Titre: Title:	On the estimation of motion and 3-D structure from monocular and stereo image sequences
Auteur: Author:	Ning Cui
Date:	1993
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Cui, N. (1993). On the estimation of motion and 3-D structure from monocular and stereo image sequences [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/57977/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/57977/
Directeurs de recherche: Advisors:	
Programme: Program:	Unspecified

UNIVERSITÉ DE MONTRÉAL

On the Estimation of Motion and 3-D Structure from Monocular and Stereo Image Sequences

Par

Ning CUI

DÉPARTEMENT DE GÉNIE ELECTRIQUE
ET DE GÉNIE INFORMATIQUE

ECOLE POLYTETHNIQUE

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU GRADE DE PHILOSOPHIAE DOCTOR (Ph.D.)

(GÉNIE ELECTRIQUE)

December 1993

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Cette thèse intitulée:

On the Estimation of Motion and 3-D Structure
from Monocular and Stereo Image Sequences

présentée par: Ning CUI

en vue de l'obtention du grade de PHILOSOPHIAE DOCTOR (Ph.D)

a été dûment acceptée par le jury d'examen constitué de:

M. BRAULT J. J., Ph.D., président

M. COHEN P., Ph.D., membre et directeur de recherche

M. FERRIE F., Ph.D., membre

M. CONAN J., Ph.D., membre

Acknowledgements

I wish to express my great appreciation to my research supervisors, Dr. Paul Cohen and Dr. Juyang Weng, for their cordial encouragement, generous support and excellent guidance throughout this research.

Special thanks are due to Hong Hai Nguyen, Michell Audette and all other members of the Perception and Robotics Laboratory of Ecole Polytechnique for their friendship and assistance.

I am grateful to my parents who always encouraged and supported me to pursue higher education; my sincere thanks also go to my husband, Jianli Sun. Their love, patience and understanding have always inspired me to overcome difficulties during my studies.

Résumé

La vision informatique traite de la construction de descriptions explicites de la géométrie des objets du monde visuel à partir d'images. Elle tente d'obtenir des résultats similaires à ceux si facilement et si rapidement obtenus par la vision humaine. En général, la vision informatique consiste en trois aspects principaux : l'acquisition, le traitement et la compréhension des images. Le traitement des images s'intéresse principalement aux transformations image à image, alors que la compréhension des images traite plutôt de l'inférence et de l'interprétation de la structure tridimensionnelle de scènes à partir d'images.

Dans le domaine de la compréhension des images, un important problème qui a été étudié de façon extensive concerne l'inférence de la structure à partir du mouvement, où l'on désire reconstruire la forme et estimer le mouvement tridimensionnel d'objets rigides à partir d'une séquence temporelle d'images.

Deux approches de base ont été proposées pour attaquer ce problème : celle basée sur le flux optique et celle basée sur les caractéristiques de l'image. Les algorithmes basés sur le flux optique tentent de retrouver le mouvement instantané à partir des dérivées spatiales et temporelles locales des valeurs d'intensité de l'image, alors que les algorithmes basés sur les caractéristiques de l'image calculent des déplacements rigides ou des paramètres cinématiques. Jusqu'à maintenant, les cas avec deux images ont été étudiés plus extensivement que les séquences d'images, en partie à cause du fait que les paramètres décrivant la réalité sont difficiles à mesurer pour des expériences avec des images réelles. En fait, pour réussir avec succès une expérience d'inférence de la structure à partir du mouvement, il faut aussi tenir compte de problèmes liés à l'acquisition et aux traitements des images.

Il y a deux étapes majeures dans les approches basées sur les caractéristiques de l'image. La première étape consiste à établir des correspondances entre certaines

caractéristiques sélectionnées de l'image, de façon à ce que des caractéristiques correspondantes dans les images proviennent d'une même entité physique dans la scène. La deuxième étape consiste à estimer les paramètres de la structure et du mouvement à partir des images. Puisque de plus en plus de techniques de correpondance pratiques et fiables sont proposées, la première étape ne représente plus maintenant un obstacle impossible à surmonter.

Alors que les approches basées sur les caractéristiques de l'image sont applicables tant à de petits qu'à de grands mouvements entre des images successives, les approches basées sur le flux optique ne s'appliquent qu'à de petits mouvements. Même si la restriction de petits intervalles inter-image simplifie à la fois l'établissement des correspondances entre les images et l'estimation des paramètres du mouvement à partir du flux optique, la fiabilité des paramètres du mouvement calculés en présence de bruit est intrinsèquement limitée puisque plusieurs techniques dans ce domaine sont basées sur les dérivées premières et secondes du flux otique, et requièrent donc les troisièmes dérivées des valeurs d'intensité de l'image. La petite quantité de mouvement est souvent noyée dans l'erreur du flux optique estimé, même lorsque qu'il peut être estimé à un niveau de précision sous-pixel.

Il est facile de voir que l'information de structure, obtenue à partir de deux images monoculaires ou d'une paire d'images stéréoscopiques, sur une grande scène est limitée en étendue et en résolution. Il existe donc un besoin au niveau de la vision dynamique, monoculaire ou stéréoscopique, dans le contexte de séquences étendues d'images. Récemment, beaucoup plus d'attention a été portée à ce sujet en raison de la maturation des algorithmes développés pour les cas avec deux images et des avantages suivants:

- Le changement progressif de point de vue amène en vue les parties cachées de la scène, permettant ainsi une description plus complète de la structure de la scène.

- La réduction de la distance de vision rapproche éventuellement les objets éloignés, permettant ainsi une description plus détaillée de ces objets.

- Puisque l'évolution de la direction et de la position du point de vue fournit habituellement plusieurs images de la même partie de la scène, une fusion de ces observations redondantes peut produire une description de la scène plus précise et plus consistante.

- En plus, le comportement cinématique et dynamique du système de senseurs peut aussi être estimé.

Même si beaucoup de travail sur l'estimation du mouvement et de la structure d'objets rigides à partir de séquences d'images bruitées a été réalisé au cours des dernières années, le problème est loin d'être résolu pour les raisons suivantes:

- Premièrement, des images réelles sont toujours contaminées par du bruit, ce qui fait que l'estimation des transformations 3-D et la fusion de vues multiples doivent prendre en considération les incertitudes variables des points 3-D estimés. En général, la composante de profondeur d'un point, déterminée par mouvement ou par triangulation, est beaucoup moins fiable que la composante latérale. Des poids scalaires affectent indistinctement l'incertitude des différentes composantes. En plus, la corrélation entre les erreurs des points 3-D ne peut pas être prise en compte adéquatement par ces poids scalaires.

- Deuxièmement, la relation entre les projections des points 3-D sur l'image et les paramètres de mouvement est non-linéaire. Il est donc critique de formuler une méthode itérative de façon à obtenir la solution la plus exacte avec les calculs les plus efficaces possibles, plutôt que de laisser l'espace de recherche croître avec le nombre de points.

- Troisièmement, dans le cas de longues séquences d'images, la quantité de données à être traitées augmente considérablement, en comparaison avec le cas à deux images. Une approche réalisant un compromis raisonnable entre la précision et l'efficacité est donc nécessaire pour des applications réelles.

- Finalement, puisque la correspondance est habituellement établie entre des paires d'images consécutives, elle sera inévitablement limitée par une quantité d'ambiguïté qui tend à s'accumuler avec le temps. Obtenir une correspondance précise pour plusieurs images consécutives, en limitant les erreurs accumulées, est donc un difficile prérequis pour l'estimation du mouvement et de la structure 3-D.

Cette thèse est donc dédiée à l'estimation du mouvement inconnu d'un système de senseurs et de la structure 3-D de la scène, en tentant de résoudre les problèmes mentionnés ci-haut. Notre approche appartient à la catégorie de celles basées sur les caractéristiques de l'image. En particulier, nous utilisons comme caractéristiques les correspondances entre les points des images. Cette étude est d'une importance grandissante en vision informatique en raison de sa nature passive et de ses applications diverses. Par exemple, il est connu que les données des senseurs de certains véhicules ne sont pas appropriés pour mesurer précisément leur mouvement en raison d'erreurs causées, par exemple, par le glissement des roues. Toutefois, la vision dynamique à partir de séquences d'images monoculaires ou stéréoscopiques peut offrir des capacités puissantes aux véhicules ou aux robots se trouvant dans cette situation. Elle peut améliorer la détection et l'évitement d'obstacles en mettant en relation la position d'un obstacle avec celle du véhicule, même lorsque l'obstacle quitte le champs de vision du système de caméra. En plus, une amélioration au niveau de la reconnaissance des objets est possible puisque plusieurs vues d'un objet peuvent être enregistrées et fusionnées lors de la navigation, ce qui rend possible la génération d'une carte globale d'un environement inconnu. En conséquence, la vision dynamique avec un système de caméra monoculaire ou stéréoscopique peut être applicable en général à la navigation robotique, l'inspection automatique et la reconnaissance et la manipulation d'objets.

Notre travail apporte de nouvelles contributions de plusieurs façons :

1. Les formulations proposées supposent une trajectoire de mouvement arbitraire afin de permettre un contrôle saccadé de la caméra et d'éviter les collisions possibles. Les paramètres de mouvement sont représentés par une matrice de rotation et un vecteur de translation.

2. Des observations répétées d'une partie de la scène sont fusionnées et l'information sur la structure de la scène, acquise à partir des images précédentes, est systématiquement intégrée aux nouvelles estimations, ce qui rend possible le traitement de longues séquences d'images monoculaires ou stéréoscopiques de façon récursive par lot et la fusion d'observations multiples de la même partie de la scène.

3. La fiabilité variable des observations et des estimés est prise en considération dans la construction des fonctions d'objectif afin d'améliorer la précision des estimés.

4. Les dimensions de l'espace de recherche lors de l'optimisation non-linéaire sont sérieusement réduites en exploitant les liens entre les paramètres de structure et de mouvement, afin que la stabilité et l'efficacité de l'optimisation soient obtenues.

5. Il est démontré que le facteur d'échelle associé à deux images consécutives de la séquence monoculaire est déterminé par le facteur d'échelle des deux premières images.

6. Des simulations et des expériences méticuleuses avec de longues séquences d'images monoculaires et stéréoscopiques de scènes réelles, incluant l'étape de calibration des caméras, ont été réalisées afin d'étudier les performances des méthodes d'optimisation développées dans cette thèse. Dans les expériences, les paramètres réels du mouvement et certains aspects de la structure de la scène, connus avec exactitude, étaient aussi disponibles pour comparaison.

Abstract

Dynamic vision provided by a monocular or stereo camera system has the capability of recovering the geometric structure of the visual environment. This ability is critical towards applications such as visually guided robot navigation, automatic surveillance, object recognition and so on. The research on this topic has been very active in the computer vision field.

This dissertation addresses the issue of optimal motion and structure estimation from monocular and stereo image sequences of rigid scenes, i.e., in which the 3-D distance between two points on any object in the scene remains constant. The proposed solutions have the following characteristics:

1. Instead of considering constrained motion, the proposed formulations allow arbitrary interframe motion, represented as a rotation matrix and a translation vector.

2. Repeated observations of a portion of a scene over successive images are fused, and the information about the structure of the scene, acquired from previous images, is systematically integrated into the new estimations, which makes it possible to deal with long monocular or stereo image sequences with a recursive-batch framework.

3. The varying reliabilities of the observations and estimates are effectively taken into account in the construction of objective functions so as to improve the accuracy of the estimates.

4. The dimension of the search space in the nonlinear optimization is drastically reduced by exploiting the relationship between structure and motion parameters, so that stability and efficiency of the optimization are achieved.

5. It is shown that the scale factor associated with any two consecutive images in a monocular sequence is determined by the scale factor of the first two images.

6. Simulations and careful experiments with long monocular and stereo image sequences of real world scenes, including camera calibration section, have been conducted to study the performance of the optimization methods developed in this dissertation. The obtained estimates have been compared to the motion and structure ground truth available.

Contents

Acknowledgements iv

Résumé v

Abstract xi

Contents xiii

List of Figures xvi

List of Tables xix

1 Introduction 1

1.1 Motivation of this research 1

1.2 New contributions of the work 5

1.3 Outline of the dissertation 7

2 Preliminaries 9

2.1 Uncertainties in the estimated 3-D positions from motion or stereo . . 9

2.2 Motion representation 11

2.2.1 Representation of relative motion 13

2.2.2 Global motion representation 15

2.3 Minimum Variance Estimation 18

2.4 Two main approaches: batch and recursive 21

3 Estimating motion and 3-D structure from monocular sequences 25

3.1 Review of the related work 25

3.2 Camera set-up 30

3.3 About the scale of the estimated structure 32

3.4 Motion and structure optimization 34

3.5 Uncertainty updating and the complete optimization procedure . . . 43

3.5.1 Updating the motion and structure uncertainties 43

3.5.2 Updating global motion and uncertainty 45

3.5.3 Steps of the complete optimization algorithm 47

3.6 Simulation and experiments 48

3.6.1 Simulation 48

3.6.2 Experiments with monocular image sequences 56

3.7 Summary 75

4 Estimating motion and 3-D structure from stereo image sequences 80

4.1 Review of related work 80

4.2 Stereo modeling 85

4.2.1 Optimal determination of a 3-D point from a pair of noisy stereo projections 86

4.3 A recursive-batch approach to process stereo image sequences 88

4.3.1 Closed-form solution from two stereo pairs 90

4.3.2 Iterative optimal solution from two stereo pairs 97

4.3.3 Estimating errors 102

4.3.4 A recursive-batch approach 103

4.3.5 Recursive-batch updating 107

4.3.6 Local and global representations 108

4.4 Simulation and experiments 111

4.4.1 Simulation for two pairs of stereo images 111

4.4.2 Simulation for stereo image sequences 112

4.4.3 Experiments 125

4.5 Further assessment of the estimated structure 133

4.6 Summary 135

5 Conclusions and discussions 138

5.1 Conclusions 138

5.2 Future work 139

Bibliography 141

Appendix 150

A Least-squares fitting of a rotation matrix 150

B The closed-form two-view algorithm 152

C Matrix-weighted centroid-coincidence Theorem 157

D The invariance of the best linear unbiased estimator under changes of scale 159

E The covariance recursive form of the structure estimator 161

F The proof of positive definiteness of the structure covariance matrix 163

G The interframe motion estimated from the monocular experiment 165

List of Figures

1.1 A monocular or stereo camera system moving in a static environment. 6

2.1 The uncertainty in the estimated coordinate of 3-D points from motion or stereo. 12

2.2 A moving monocular or stereo camera system in a static environment, with the two coordinate systems depicted. 16

3.1 Camera-centered coordinate system. 31

3.2 The framework of our recursive-batch approach for motion and structure estimation from monocular image sequences. 36

3.3 An illustration of the decomposition strategy used in the search space. 41

3.4 The navigation path of the monocular camera system in the simulation. 50

3.5 Simulation results: errors in the estimated interframe motion. (a) error in the interframe rotation matrix $R_{k,k-1}$; (b) the direction error in the interframe translation vector $\mathbf{T}_{k,k-1}$. 53

3.6 Simulation result: error in the estimated local structure. 54

3.7 Simulation result: error in the estimated global structure. 55

3.8 Simulation results: errors in the estimated global motion. (a) error in the global rotation matrix R_k; (b) the direction error in the global translation vector $\mathbf{T}_k$; (c) the absolute error in the global translation vector $\mathbf{T}_k$. 59

3.9 The calibration plate used in our experiments. The control points are the corner points of the black squares on the plate. 61

3.10 Coordinate systems for camera calibration. 63

3.11 Radial and tangential distortion 65

3.12 The flowchart of the calibration procedure. 68

3.13 The calibrated camera was mounted on the manipulator to grab image sequences. 70

3.14 The image sequence of a real scene, used in experiments. 71

3.15 Matching for the first image pair shown as vector lines. 74

3.16 The samples of the lines have been measured in the scene. 76

4.1 The estimated 3-D point in (4.7) is the midpoint of the shortest line segment that connects the two projection lines. 89

4.2 Detailed illustration of recursive-batch approach for processing long stereo sequences. 106

4.3 Simulation results from two pairs of stereo images: errors in the estimated interframe motion. (a) error in the interframe rotation matrix (the true rotation angle is 8°, the rotation axis is $(1\text{m}, 0.2\text{m}, 0.1\text{m})^{\mathsf{T}}$); (b) error in the interframe translation vector (the true translation vector is $(-0.14\text{m}, 1.35\text{m}, -0.92\text{m})^{\mathsf{T}}$). 114

4.4 Simulation results from two pairs of stereo images: errors in the estimated interframe motion. (a) error in the interframe rotation matrix (the true rotation angle is 30°, the rotation axis is $(1\text{m}, 0.2\text{m}, 0.1\text{m})^{\mathsf{T}}$); (b) error in the interframe translation vector (the true translation vector is $(-0.14\text{m}, 1.35\text{m}, -0.92\text{m})^{\mathsf{T}}$). 116

4.5 Simulation results from stereo image sequence: error in estimated interframe motion. (a) error in interframe rotation matrix $R_{k,k-1}$; (b) error in interframe translation vector $\mathbf{T}_{k,k-1}$ 120

4.6 Simulation results from stereo image sequence: error in estimated global motion. (a) error in global rotation matrix R_k; (b) error in global translation vector $\mathbf{T}_k$ 122

4.7 Simulation results from a stereo image sequence: error in the estimated local structure. 123

4.8 Simulation results from a stereo image sequence: error in the estimated global structure. 124

4.9 The stereo image sequence used in the experiments. 129

4.10 Matching for the first pair of stereo images shown as vector lines. . . 130

4.11 The depth map is shown as an intensity image from fifth stereo image pair. 131

4.12 The uncertainty in the estimated coordinate of 3- D points, due to image quantization. 136

G.1 A graphic display of the estimated values of the interframe rotation angles of the monocular experiment. The ground truth of the interframe rotation angle is 2.25°. 166

List of Tables

3.1 Calibration data for the f=8.5mm lens camera. 72

3.2 Motion estimations resulting from the linear algorithm and the nonlinear optimization. 77

3.3 Estimated motion and structure. k: time index; θ: rotation angle (degree); RMSE: root mean-squared error (mm). 79

4.1 Calibration data for the left f=8.5mm lens camera. 126

4.2 Calibration data for the right f=8.5mm lens camera. 127

4.3 Motion estimations resulting from the matrix-weighted linear algorithm and the nonlinear optimization (unit m). 134

4.4 Estimated motion and structure. k: time index; θ: rotation angle (degree); $RMSE$: root mean square error (mm); ME: mean error (mm). 134

G.1 Motion estimations resulting from the linear algorithm and the nonlinear optimization. 167

G.2 Motion estimations resulting from the linear algorithm and the nonlinear optimization. 168

G.3 Motion estimations resulting from the linear algorithm and the nonlinear optimization. 169

G.4 Motion estimations resulting from the linear algorithm and the nonlinear optimization. 170

G.5 Motion estimations resulting from the linear algorithm and the nonlinear optimization. 171

Chapter 1

Introduction

1.1 Motivation of this research

Computer vision concerns the construction of explicit descriptions of the geometry of objects in the visual world from images. It aims at achieving results similar to those vividly and effortlessly obtained by human vision. In general, computer vision consists of three main aspects: image acquisition, image processing and image understanding. While image processing is mainly concerned with image-to-image transformations, image understanding is primarily interested in the inference and the interpretation of the three-dimensional structure of scenes from images.

An important and extensively studied problem in image understanding, concerns the inference of structure from motion, where one wishes to reconstruct the shape and to estimate the 3-D motion of rigid objects from the temporal succession of their images.

Two basic approaches have been proposed to attack the problem: the optical flow based approach and the feature based approach [1]. Algorithms based on optical flow attempt to recover the instantaneous motion, relying on local spatial and temporal derivatives of the image intensity values, while feature based algorithms recover rigid displacements or kinematic parameters. Two-view cases have been studied more extensively than image sequences so far, partly due to the fact that the ground truth in experiments with real images is hard to acquire [7]. In fact, a successful structure from motion experiment has to deal with problems in image acquisition and image processing, besides the problems in structure from motion itself.

There are two major steps in feature based approaches. The first step is to establish correspondences for selected feature primitives in images, so that the matched feature primitives arise from the same physical entities in the scene. The second step is to estimate the structure and motion parameters from the concerned images. As more reliable and practical matching techniques emerged, such as [59], [60], [61], [62], the first step is no longer an intractable obstacle to overcome.

Concerning the second step, several important contributions have been made in the recent past, namely the well known eight-point algorithm for motion estimation [42], [44], [45], [46], [47], and its 3-D version, least-squares estimation of motion parameters for 3-D point correspondences [13], [23]. The degenerate configurations associated with the eight-point algorithm were investigated as well in [43], [44], [45]. It has been pointed out that three non-colinear 3-D point correspondences over two time instants determine motion uniquely [9], [7] for the approach proposed in [13], [23]. Accordingly, motion estimation algorithms based on line correspondences in the images were presented in [36], [37]. However, for the line correspondence case, a certain number of lines seen in three successive frames is required in the estimation algorithms, these algorithms are therefore less popular among researchers in computer vision community, as are the approaches posed in [38] which handle situations of 3 points in three frames, 2 points in 4 frames and 1 point in 5 frames. Recently, a matrix-weighted least-squares estimation of motion parameters for 3-D point correspondences was presented [71], which gives accurate motion parameters for noisy images. This approach requires a minimum of four points for motion uniqueness. In addition to the closed-form solutions, many different versions of iterative optimization formulas have been constructed to further improve the motion parameters from an initial solution [48], [58], [65].

While feature based approaches are applicable to both small or large interframe motions, the optical flow approaches are only applicable to small motions. Although the restriction of small interframe motion simplifies both image matching and motion parameter computation from optical flow, the reliability of the computed motion parameters in the presence of noise is intrinsically limited, since many techniques in this area are based on the first and second derivatives of optical flow and thus require the third derivatives of intensity values [1], [4], [63]. The small amount of motion is easily overridden by the error in the estimated optical flow, even if the optical flow can be estimated to subpixel accuracy.

It is easy to see that the structure information obtained from two monocular or stereo views of a large scene is limited in extent and resolution. Therefore, there is a necessity to address the issue of dynamic vision, monocular or stereo, in the context of extended image sequences. More attention has been recently devoted to this issue, due to the maturing of the algorithms developed for the two-view case and the following advantages provided by image sequences:

- The progressive change of view-point brings occluded parts of the scene into view, thus allowing a more complete description of the scene structure.

- The reduction of the viewing distance eventually draws objects that are far away to proximity, and allows a detailed analysis of these objects.

- Since the evolution of the viewing direction and position usually provides multiple images of the same part of the scene, a fusion of such redundant observations can result in more accurate and consistent descriptions of the scene.

- Moreover, the kinematic and dynamic behavior of the sensor system can also be estimated.

Although much work on the estimation of the motion and structure of rigid objects, from noisy image sequences, has been conducted in last few years [65], [53], [40], [11], the problem is far from being solved in the following respects:

- Primarily, the real images are always contaminated by noise, hence the estimation of the 3-D transformation and the fusion of multiple views ought to take into account the varying uncertainties in the estimated 3-D points. Generally speaking, the depth component of a point determined by motion or triangulation is considerably less reliable than the lateral components. Scalar weights [26], [31] indiscriminately affect the uncertainty in the different components. Furthermore, the correlation between errors in the 3-D point can not be properly accounted for by these scalar weights.

- Secondly, the relationship between the image projections of 3-D points and the motion parameters is nonlinear. Therefore, it is critical to formulate the iterative scheme in such a way to achieve the most accurate solution and most efficient computation possible, instead of letting the search space increase with the number of points as in [53], [11].

- Thirdly, in the case of long image sequences, the amount of data to be processed increases drastically as compared to the two-view analysis. An approach which has a reasonable trade-off between accuracy and efficiency is thus required for real applications.

- Finally, as matching is usually established between pairs of consecutive images in a sequence, it will inevitably be plagued by an amount of ambiguity which tends to accumulate over time. How to get an accurate matching over many frames, by limiting the accumulated error, is therefore a difficult prerequisite for the task of motion and 3-D structure estimation.

This dissertation is thus devoted to the issues of estimating the unknown motion of a sensor system and the 3-D structure of the scene, with efforts to resolve the above problems (some of our work has been published in [76], [77], [78], [79], [80], [81]). As shown in Figure 1.1, a sensor system is a monocular or stereo camera system moving in a static environment. Our approaches belong to the category of feature based approaches. In particular, we use point correspondences as features. The study is of ever-growing importance in computer vision, because of its passive nature as well as its diverse and potential applications. For example, it has been reported that the outputs of vehicle sensors themselves are not suitable for accurate motion recovery because of errors in dead reckoning resulting from problems as wheel slippage [8]; [52]. However, dynamic vision from monocular or stereo image sequences can offer a powerful visual ability for the vehicles or robots in those situations. It can improve obstacle detection and avoidance by relating the position of an obstacle to the position of the vehicle, even when the obstacle leaves the field of view of the camera system. Furthermore, object recognition will be improved since multiple views of objects can be registered and fused during the course of navigation, which makes it possible to generate a global map of an unknown environment. Consequently, dynamic vision through a monocular or stereo camera system can be applicable in general to robot navigation, automatic inspection, object recognition and manipulation.

1.2 New contributions of the work

Our work provides new contributions in several aspects: (1) The proposed formulations assume arbitrary motion trajectory to allow saccadic camera control and to avoid possible collisions. The motion parameters are represented as a rotation ma-

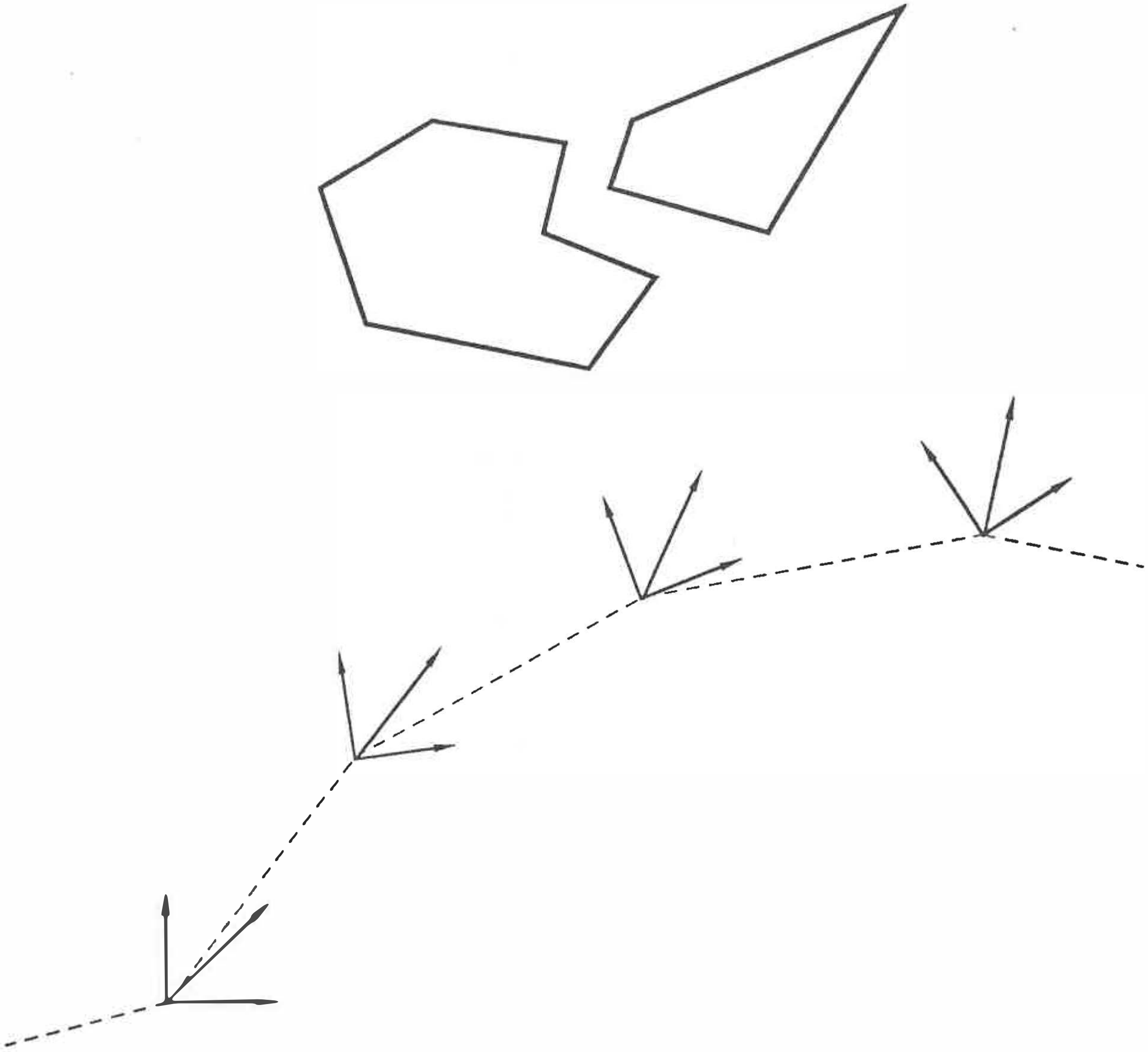


Figure 1.1: A monocular or stereo camera system moving in a static environment.

trix and a translation vector. (2) Repeated observations of a portion of a scene over successive images are fused, and the information about the structure of the scene, acquired from previous images, is systematically integrated into the new estimations, which makes it possible to deal with long monocular or stereo image sequences with a recursive-batch framework, and feasible to fuse multiple observations of some part of the scene. (3) The varying reliabilities of the observations and estimates are effectively taken into account in the construction of objective functions so as to improve the accuracy of the estimates. (4) The dimension of the search space in the nonlinear optimization is drastically reduced by exploiting the relationship between structure and motion parameters, so that stability and efficiency of the optimization are achieved. (5) It is shown that the scale factor associated with any two consecutive images in a monocular sequence is determined by the scale factor of the first two images. (6) Simulations and careful experiments with long monocular and stereo image sequences of real world scenes, including camera calibration section, have been conducted to study the performance of the optimization methods developed in this dissertation. In the experiments, ground truth concerning the motion parameter and some aspects of the scene structure were also available for comparison.

1.3 Outline of the dissertation

The dissertation is organized as follows:

Chapter 2 describes the theoretical background and some fundamental techniques in motion and structure estimation. The uncertainty problem in estimated positions of 3-D points from motion or stereo is discussed first. The motion representation, the local and global coordinate systems to be used are then introduced. The minimum variance estimation is outlined in this chapter. Finally, batch and recursive approaches, these two commonly used main approaches for estimating motion and structure, are compared according to their merits for the nonlinear problem that we want to solve.

Chapter 3 studies the subject of estimating motion and structure from monocular image sequences and includes a discussion about the relationship among the scale factors involved in any monocular image sequence. Both simulation and experiments with an image sequence of a real world scene are provided to demonstrate the performance of the proposed recursive-batch approach. The camera calibration principle which was used in the experiments is briefly described in this chapter. The strategy of combating the error accumulation in matching over many frames, uing a normalized cross-correlation, is also presented.

Chapter 4 deals with the counterpart of the monocular problem: motion and structure from stereo image sequences. Starting from a newly proposed matrix-weighted closed-form algorithm, we then process stereo image sequences with a recursive-batch approach. Simulation and experiment are presented to assess the performance of our approaches.

Chapter 5 summarizes the whole work and discusses directions for future research.

Chapter 2

Preliminaries

In this chapter, some theoretical background and fundamental techniques in motion and structure estimation are described. The uncertainty problem in estimated positions of 3-D points from motion or stereo is discussed first. Then, relative and global motion representations with respect to the local and the global coordinate systems are introduced. The minimum variance estimation principle is outlined, and finally, two commonly used techniques for motion and structure estimation, namely batch and recursive approaches, are compared in the context of the nonlinear problem we want to solve.

2.1 Uncertainties in the estimated 3-D positions from motion or stereo

We will first relate the positional uncertainty of 3-D points, reconstructed by means of triangulation, to the image positional uncertainty due to image quantization process. Properly modeling this uncertainty is essential, in order to limit its impact on motion and structure estimates.

Figure 2.1 demonstrates intuitively the relation between the image quantization noise and the estimated positions of 3-D points. One can see that identical amounts of image quantization noise induce positional uncertainties, which depend upon the spatial location. The farther the object is from the cameras, the larger the volume of the corresponding 3-D uncertainty. The same thing happens if the relative position of the object with respect to the cameras is invariant, but the distance between the two cameras decreases. The depth component of a 3-D point location is the less reliable component. These phenomena must be taken into consideration when addressing the problem of optimal motion and structure estimation.

Positional uncertainty is not caused by image quantization alone, but may also come from inproper camera calibration, feature matching errors, etc..

Several 3-D noise models were developed to represent 3-D uncertainties caused by image errors. A symmetrical 3-D noise distribution model was used in [40],

$$\begin{pmatrix} x(t_i) \\ y(t_i) \\ z(t_i) \end{pmatrix} = \begin{pmatrix} x_{tru}(t_i) \\ y_{tru}(t_i) \\ z_{tru}(t_i) \end{pmatrix} + \begin{pmatrix} n_x(t_i) \\ n_y(t_i) \\ n_z(t_i) \end{pmatrix}, \tag{2.1}$$

where noisy 3-D coordinates $(x(t_i), y(t_i), z(t_i))^\top$ of a feature point P in an arbitrary 3-D world coordinate system I are calculated from binocular images, $(x_{tru}(t_i), y_{tru}(t_i), z_{tru}(t_i))^\top$ are the noise-free 3-D coordinates of P in I and $n_x(t_i)$, $n_y(t_i)$ and $n_z(t_i)$ are the noise components, assumed to be Gaussian processes with zero mean. The measurements of 3-D coordinates are taken at discrete time t_i. This observation noise model is therefore an approximation of the observation noise model which can be derived from image noise model through triangulation. An ellipsoidal model was adopted in [52], which is similar to the symmetric 3-D distribution model.

Two other simplified models, based on scalar weights, were utilized in [26], [31]. The motivation for using scalar weights is that uncertainty grows with distance, so it can be modeled by weighting points inversely with distance.

Some 2-D image noise models, in which the measured image coordinates of the feature points are assumed to consist of the true coordinates corrupted by additive independent zero mean Gaussian noise, were also proposed in [65], [14]. Because the symmetrical 3-D noise distribution model or any other simplified model cannot appropriately capture the 3-D uncertainty shape depicted in Figure 2.1, where nearby points have a fairly compact uncertainty, whereas distant points have a more elongated uncertainty, these models will impair the quanlity of the motion and structure estimates. In our work, we use a 2-D image plane noise model as used in [65], [14]. It implicitly results in the desired distribution shape in 3-D, represented by a 3×3 error covariance matrix, through the mapping from 2-D feature points to their 3-D counterparts.

2.2 Motion representation

In this section, the motion representation for a monocular or stereo camera system will be introduced. Different situations concerning the position of the monocular or stereo camera system with respect to the scene are considered. Two coordinate systems, global and local, together with their relationship, are also introduced.

We assume that objects move rigidly in front of a monocular or stereo system. This implies that the 3-D distance between two points on any object in the scene remains constant. We can then interpret the motion in two different ways: it can correspond to the displacement of the scene, while the monocular or stereo camera system remains stationary, or vice versa. In another way, we see the motion from the rigid scene. Each of these two interpretations has its own applications; we will primarily discuss the first one.

In general, there may be several objects moving independently within the field of view of the vision system. We assume here a situation of a single moving rigid object (scene), with enough point matches in each consecutive image pair to guarantee that

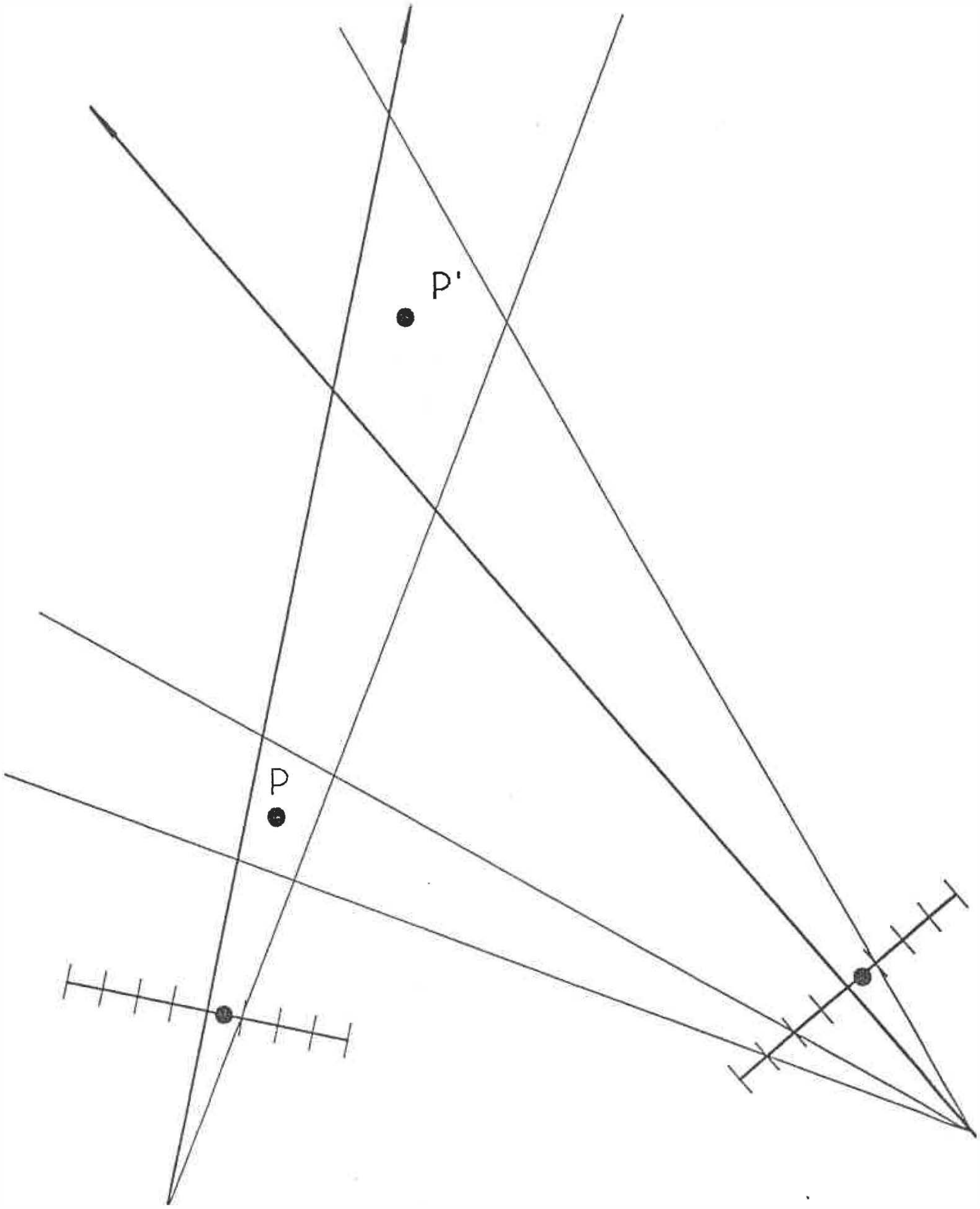


Figure 2.1: The uncertainty in the estimated coordinate of 3-D points from motion or stereo.

the linear algorithms [46], [71] are applicable, since the methodology for treating one moving object and several independently moving objects is mathematically the same.

Due to the rigidity assumption, a motion of the observed object between consecutive time instants can be represented as a rotation matrix around a chosen origin of a coordinate system followed by a translation vector. The rationale of this preference of motion representation is that it comprises fewer motion parameters than the kinematic model used in [11], [40], [53]. In addition, there are some closed-form algorithms available for the computation of an initial solution of the motion parameters.

2.2.1 Representation of relative motion

Considering the ith point on the object, its position $\mathbf{x}_{l,i}$ at time t_l and its position $\mathbf{x}_{k,i}$ at t_k are related by a rotation matrix $R_{k,l}$ (we reserve the bold face form of R for later usage) and a translation vector $\mathbf{T}_{k,l}$ in the camera-centered coordinate system (to be specified in Chapter 3 and Chapter 4 for the monocular and stereo cases, respectively) as

$$\mathbf{x}_{k,i} = R_{k,l}\mathbf{x}_{l,i} + \mathbf{T}_{k,l}, \qquad k \geq l. \tag{2.2}$$

where we impose

$$R_{k,k} \triangleq \mathbf{I}, \;\; \mathbf{T}_{k,k} \triangleq \mathbf{0}. \tag{2.3}$$

The rotation matrix corresponds to a rotation γ about the x-axis, then a rotation β about the y-axis, and then a rotation α about the z-axis:

$$R = R_z R_y R_x, \tag{2.4}$$

where

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix}, \tag{2.5}$$

$$R_y = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix}, \tag{2.6}$$

$$R_z = \begin{bmatrix} cos\alpha & -\ sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.7}$$

Alternatively, the rotation matrix corresponds to a rotation angle θ about an arbitrary axis $\mathbf{n}$:

$$R = \begin{bmatrix} n_1^2 + (1-n_1^2)cos\theta & n_1n_2(1-cos\theta) - n_3sin\theta & n_1n_3(1-cos\theta) + n_2sin\theta \\ n_1n_2(1-cos\theta) + n_3sin\theta & n_2^2 + (1-n_2^2)cos\theta & n_2n_3(1-cos\theta) - n_1sin\theta \\ n_1n_3(1-cos\theta) - n_2sin\theta & n_2n_3(1-cos\theta) + n_1sin\theta & n_3^2 + (1-n_3^2)cos\theta \end{bmatrix}. \tag{2.8}$$

Sometimes, a quaternion expression of a rotation matrix is more convenient for the rotation computation:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_2q_1 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_3q_1 - q_0q_2) & 2(q_3q_2 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}, \tag{2.9}$$

where the unit quaternion $\mathbf{q}$ can be thought as a vector with four components, $\mathbf{q} = (q_1,\ q_2,\ q_3,\ q_4)^\top$, corresponding to a rotation ϕ about a unit vector $\mathbf{w}$ [50]:

$$\mathbf{q} = cos\frac{\phi}{2} + \mathbf{w}sin\frac{\phi}{2}. \tag{2.10}$$

When $k = l + 1$ in (2.2), we have,

$$\mathbf{x}_{k,i} = R_{k,k-1}\mathbf{x}_{k-1,i} + \mathbf{T}_{k,k-1}. \tag{2.11}$$

Since $R_{k,k-1}$, $\mathbf{T}_{k,k-1}$ represent the motion between consecutive instants, it is therefore interframe motion.

Note that the expression (2.2) can be recursively transformed into

$$\begin{aligned} \mathbf{x}_{k,i} &= R_{k,k-1}\cdots R_{l+1,l}\mathbf{x}_{l,i} + R_{k,k-1}\cdots R_{l+2,l+1}\mathbf{T}_{l+1,l} \\ &\quad +R_{k,k-1}\cdots R_{l+3,l+2}\mathbf{T}_{l+2,l+1} + \cdots + \mathbf{T}_{k,k-1}. \end{aligned} \tag{2.12}$$

Comparison between (2.2) and (2.12) yields the following identities

$$\begin{aligned} R_{k,l} &= R_{k,k-1}R_{k-1,k-2}\cdots R_{l+1,l}, \qquad k > l, \\ \mathbf{T}_{k,l} &= \sum_{i=l+1}^{k} R_{k,i}\mathbf{T}_{i,i-1}, \qquad k > l. \end{aligned} \tag{2.13}$$

Letting $l = 0$, we may write the recursive equations (2.13) as:

$$\begin{aligned} R_{k,0} &= R_{k,k-1}R_{k-1,0}, \\ \mathbf{T}_{k,0} &= R_{k,k-1}\mathbf{T}_{k-1,0} + \mathbf{T}_{k,k-1}. \end{aligned} \tag{2.14}$$

It can be easily shown that reverting time leads to similar identities, namely

$$\begin{aligned} R_{l,k} &= R_{l,l+1}R_{l+1,l+2}\cdots R_{k-1,k} = R_{k,l}^{-1} \\ \mathbf{T}_{l,k} &= \sum_{i=k}^{l+1} R_{l,i-1}\mathbf{T}_{i-1,i} = -R_{k,l}^{-1}\mathbf{T}_{k,l}. \end{aligned} \tag{2.15}$$

Since there are probably more common visible points in two consecutive images than in non-consecutive ones, it seems more promising to use consecutive images for calculating the interframe motion.

2.2.2 Global motion representation

We now consider the global motion of the camera system as seen from a static scene. To do this we define a global coordinate system, which is fixed with the scene. Let the global coordinate system coincide with the camera-centered coordinate system called local coordinate system at time t_0. These two coordinate systems are shown in Figure 2.2.

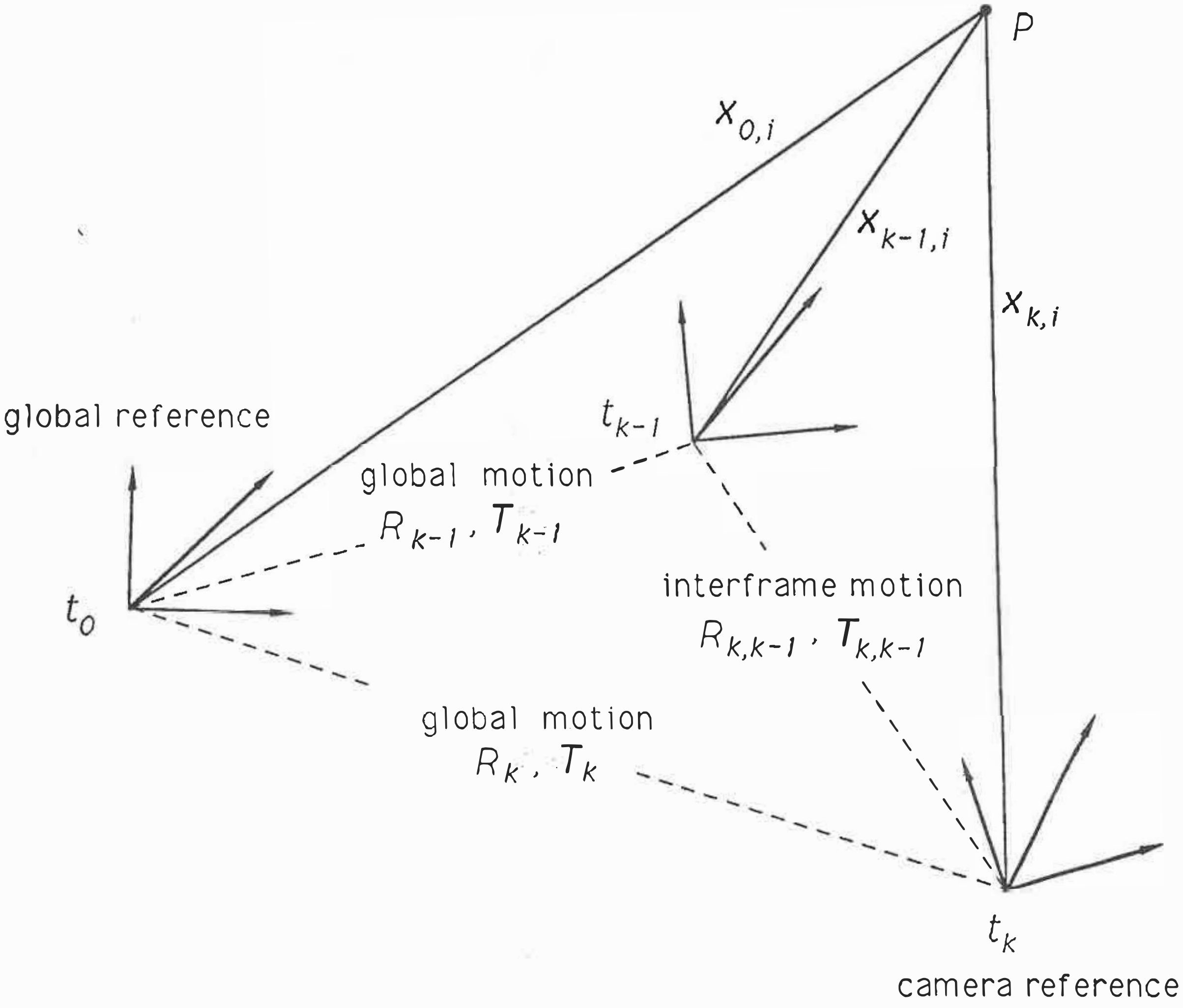


Figure 2.2: A moving monocular or stereo camera system in a static environment, with the two coordinate systems depicted.

In the global coordinate system, the current attitude, including position and orientation, of the camera system at time t_k can be achieved by moving the camera system, from its original attitude at time t_0, by a rotation about the origin, represented by a rotation matrix R_k, followed by a translation, represented by a vector $\mathbf{T}_k$. Letting $\mathbf{C}_{k,i}$ be any point on the camera system at time t_k, represented in the global coordinate system, $\mathbf{C}_{k,i}$ and $\mathbf{C}_{0,i}$ are related by

$$\mathbf{C}_{k,i} = R_k \mathbf{C}_{0,i} + \mathbf{T}_k. \tag{2.16}$$

A similar relation transforms a position vector $\mathbf{x}_{k,i}$ of an object point at t_k, represented in the local coordinate system, back to the global coordinate system:

$$\mathbf{x}_{0,i} = R_k \mathbf{x}_{k,i} + \mathbf{T}_k, \tag{2.17}$$

which is the global structure. Proceeding from t_{k-1} to t_k, we get

$$\mathbf{x}_{k,i} = R_k^{-1} R_{k-1} \mathbf{x}_{k-1,i} + R_k^{-1}(\mathbf{T}_{k-1} - \mathbf{T}_k) \tag{2.18}$$

this, combined with (2.2), leads to the global motion of the camera,

$$\begin{aligned} R_k &= R_{k-1} R_{k,k-1}^{-1}, \\ \mathbf{T}_k &= \mathbf{T}_{k-1} - R_k \mathbf{T}_{k,k-1}. \end{aligned} \tag{2.19}$$

It can be seen from (2.17), (2.18) and (2.19) that once the interframe motion $R_{k,k-1}$, $T_{k,k-1}$ and the local structure $\mathbf{x}_{k,i}$, $\mathbf{x}_{k-1,i}$ are computed, we can update the global attitude, R_k and $\mathbf{T}_k$, of the camera system from its previous attitude and current interframe motion, and further update the global structure representation $\mathbf{x}_{0,i}$ of the relative structure $\mathbf{x}_{k,i}$. Therefore, we can build up a visual map along the trajectory of the moving camera system, and specify the relative position of the camera system with respect to the scene. In other words, the task of building up the extended visual map along the navigation path as well as the global position of the camera system at different instants, can be decomposed into the subtasks of obtaining the interframe motions and local structure, and then combining these intermediate estimates. Consequently, the expressions of motion and structure in (2.17) and (2.19) are of interest for autonomous navigation.

In the more complicated situation where both the camera system and objects are moving, the only motion which can be recovered from the images is the relative motion between the camera system and each object.

Note that, with the motion representation we adopt in this research, no restriction is imposed on the type of interframe motion. However, as pointed out in the literature, for the monocular case, if the magnitude of an interframe translation vector is equal to zero, the depths of the feature points cannot be determined. Moreover, a more accurate estimation can be obtained if the interframe motion leads to the following conditions:

- feature correspondences occupy a wider field of view of the monocular or stereo camera system;
- a larger number of feature correspondences can be found in the image sequences;
- a longer average length of displacement vectors between corresponding features is achieved.

In the remainder of this dissertation, we always implicitly assume that the magnitude of any interframe translation vector is nonzero.

2.3 Minimum Variance Estimation

The minimum variance estimation is outlined here since it will be used in the following two chapters to obtain a few 3-D structure estimators.

Suppose an m-dimensional observation vector $\mathbf{y}$ is related to a n-dimensional parameter vector $\mathbf{m}$ by a linear deterministic system

$$\mathbf{y} = \mathbf{A}\mathbf{m} + \delta_{\mathbf{y}} \tag{2.20}$$

where $\mathbf{A}$ is a $m \times n$ matrix, $\delta_{\mathbf{y}}$ is a random vector with zero mean, $E\delta_{\mathbf{y}} = 0$, and an error covariance matrix

$$\mathbf{\Gamma}_{\mathbf{y}} = E[\delta_{\mathbf{y}}\delta_{\mathbf{y}}^{\mathsf{T}}]. \tag{2.21}$$

The best linear unbiased estimator of $\mathbf{m}$ is given by (*Gauss-Markov* theorem or linear minimum variance estimator) (see, e.g., [51], [69], [70])

$$\mathbf{m}^* = (\mathbf{A}^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{y}}^{-1}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{y}}^{-1}\mathbf{y} \tag{2.22}$$

with an error covariance matrix

$$\mathbf{\Gamma}_{\mathbf{m}^*} = E[(\mathbf{m}^* - \mathbf{m})(\mathbf{m}^* - \mathbf{m})^{\mathsf{T}}] = (\mathbf{A}^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{y}}^{-1}\mathbf{A})^{-1}. \tag{2.23}$$

The estimator in equation (2.22) can be seen as a weighted least squares estimator with weighting matrix $\mathbf{\Gamma}_{\mathbf{y}}^{-1}$, i.e., the objective function is to minimize $(\mathbf{y} - \mathbf{A}\mathbf{m})^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{y}}^{-1}(\mathbf{y} - \mathbf{A}\mathbf{m})$ [70], or as a linear mean-square estimator where no a priori information about the parameter vector $\mathbf{m}$ is available, i.e., ${\mathbf{\Gamma}_{\mathbf{m}}}^{-1}$ is a zero matrix [69].

If, in addition, $\delta_{\mathbf{y}}$ has a Gaussian distribution, the estimator in (2.22) is an estimator that minimizes $E\|\mathbf{m}^* - \mathbf{m}\|^2$ among all estimators (not limited to the class of linear estimators). There are two important properties of minimum variance estimates stated in the following two theorems [51].

Theorem 1. The minimum variance linear estimate of a linear function of $\mathbf{m}$, based on the ramdom vector $\mathbf{y}$, is equal to the same linear function of the minimum variance linear estimate of $\mathbf{m}$; i.e., given an arbitrary $p \times n$ matrix $\mathbf{S}$, the best estimate of $\mathbf{S}\mathbf{m}$ is $\mathbf{S}\mathbf{m}^*$.

Theorem 2. If $\mathbf{m}^* = \mathbf{K}\mathbf{y}$ is the linear minimum variance estimate of $\mathbf{m}$, then $\mathbf{m}^*$ is also the linear estimate minimizing $E[(\mathbf{m}-\mathbf{m}^*)^\top \mathbf{P}(\mathbf{m}-\mathbf{m}^*)]$ for any positive-semidefinite $n \times n$ matrix $\mathbf{P}$.

In a nonlinear problem, equation (2.20) becomes

$$\mathbf{y} = \mathbf{f}(\mathbf{m}) + \delta_{\mathbf{y}}, \tag{2.24}$$

where $\mathbf{f}$ is a nonlinear function (in our cases, $\mathbf{f}$ will represent the nonlinear relationship between the motion parameters, the 3-D structure, and the 2-D image projections). From a preliminary estimate $\mathbf{m}_0$ of the parameter vector $\mathbf{m}$, we can linearize the function $\mathbf{f}$ by the first order Taylor series expansion,.

$$\mathbf{y} \approx \mathbf{f}(\mathbf{m}_0) + \frac{\partial \mathbf{f}(\mathbf{m}_0)}{\partial \mathbf{m}}(\mathbf{m}-\mathbf{m}_0) + \delta_{\mathbf{y}}. \tag{2.25}$$

Then, the objective function to be minimized becomes:

$$\left[\mathbf{y} - \mathbf{f}(\mathbf{m}_0) - \frac{\partial \mathbf{f}(\mathbf{m}_0)}{\partial \mathbf{m}}(\mathbf{m}-\mathbf{m}_0)\right]^\top \mathbf{\Gamma}_{\mathbf{y}}^{-1} \left[\mathbf{y} - \mathbf{f}(\mathbf{m}_0) - \frac{\partial \mathbf{f}(\mathbf{m}_0)}{\partial \mathbf{m}}(\mathbf{m}-\mathbf{m}_0)\right]. \tag{2.26}$$

Comparing to (2.24), the above objective function may be replaced by

$$[\mathbf{y} - \mathbf{f}(\mathbf{m})]^\top \mathbf{\Gamma}_{\mathbf{y}}^{-1} [\mathbf{y} - \mathbf{f}(\mathbf{m})]. \tag{2.27}$$

In other words, the optimal parameter vector $\mathbf{m}^*$ is the one that minimizes the matrix-weighted discrepancy between the computed observation $\mathbf{f}(\mathbf{m})$ and the actual observation $\mathbf{y}$. If the noise vector $\delta_{\mathbf{y}}$ is uncorrelated and its components have the same variance σ^2, minimizing equation (2.27) is equivalent to minimizing

$$||\mathbf{y} - \mathbf{f}(\mathbf{m})||^2. \tag{2.28}$$

The estimate $\mathbf{m}^*$ which minimizes (2.27) has an error covariance matrix approximately estimated according to equation (2.23):

$$\mathbf{\Gamma}_{\mathbf{m}^*} = E[(\mathbf{m}^* - \mathbf{m})(\mathbf{m}^* - \mathbf{m})^\top] = \left[\frac{\partial \mathbf{f}(\mathbf{m}^*)^\top}{\partial \mathbf{m}} \mathbf{\Gamma}_{\mathbf{y}}^{-1} \frac{\partial \mathbf{f}(\mathbf{m}^*)}{\partial \mathbf{m}}\right]^{-1}. \tag{2.29}$$

Notice that the equation (2.20) is a good model only if the noise term $\delta_{\mathbf{y}}$ has a zero mean. In an estimation problem, if $\mathbf{y}$ denotes the raw data corresponding to initial measurements, it is often true that the error $\delta_{\mathbf{y}}$ approximately satisfies the above condition.

One of the advantages of using a minimum variance criterion is that one does not need to know the exact noise distribution, which is very difficult to obtain in most applications. The above objective function (2.27) does not require knowledge of more than second order statistics of the noise distribution, which can often be estimated in practice. The second advantage of the minimum variance estimator is that $\mathbf{m}^*$ is invariant under changes of scale, which is desirable for monocular motion and structure estimation (see Appendix D).

For a general nonlinear system, the estimator determined by minimizing expression (2.27) or (2.28) is not exactly a linear minimum variance estimator. However, if the noise is not very large and the correct convergence is reached, the behavior of a nonlinear system is well approximated by the Jacobian matrix of $\mathbf{f}$ in a small neighborhood around the actual parameters, and can be approximated by a linear system. So, we will use the minimum variance estimation principle to construct several objective functions for the monocular and stereo problems.

2.4 Two main approaches: batch and recursive

There are two main types of approaches commonly used in the computer vision community to estimate the interframe motion and the local structure, namely the batch and the recursive types.

In recursive or sequential approaches, like the *Extended Kalman Filtering*, the nonlinear objective function between the image observations and the motion parameters is linearized by a first order Taylor series expansion. The motion and structure update proceeds from the first observation to the last in a sequential manner.

To be more specific, if we take the monocular case as an example, the nonlinear relationship between motion parameters $\mathbf{m}_{k,k-1}$ which consists of six variables for rotation $R_{k,k-1}$ and translation $\mathbf{T}_{k,k-1}$, and the 2-D image projections $\tilde{\mathbf{u}}_{k,i}$ and $\tilde{\mathbf{u}}_{k-1,i}$ can be expressed as:

$$\tilde{\mathbf{u}}_{k,i} = \mathbf{f}(\mathbf{p}) + \delta_{\tilde{\mathbf{u}}_{k,i}}, \quad i = 1, \cdots, n. \tag{2.30}$$

where $\tilde{\mathbf{u}}_{k-1,i} = (\frac{x_{k-1,i}}{z_{k-1,i}}, \frac{y_{k-1,i}}{z_{k-1,i}})^{\mathsf{T}}$, $\tilde{\mathbf{u}}_{k,i} = (\frac{x_{k,i}}{z_{k,i}}, \frac{y_{k,i}}{z_{k,i}})^{\mathsf{T}}$, $\mathbf{f}$ represents the nonlinear relationship (motion and projection), and $\mathbf{p}$ represents the motion parameters $\mathbf{m}_{k,k-1}$. The image observation noise $\delta_{\tilde{\mathbf{u}}_{k,i}}$ is assumed to be white. The objective is to estimate the parameter vector $\mathbf{p}$ and structure $\{\mathbf{x}_{k,i}\}$, $\{\mathbf{x}_{k-1,i}\}$ through processing the observed image points $\{\tilde{\mathbf{u}}_{k,i}\}$ and $\{\tilde{\mathbf{u}}_{k-1,i}\}$.

Since *Kalman Filtering* is only applicable to linear systems, we have to linearize the nonlinear function first. From a preliminary estimate $\hat{\mathbf{p}}^{(0)}$ of the parameter vector $\mathbf{p}$ which is provided by a least-squares method, we can linearize the nonlinear function $\mathbf{f}$ in the vicinity of $\hat{\mathbf{p}}^{(0)}$ by a first order Taylor series expansion,

$$\tilde{\mathbf{u}}_{k,i} \approx \mathbf{f}(\hat{\mathbf{p}}^{(0)}) + \frac{\partial \mathbf{f}^T(\hat{\mathbf{p}}^{(0)})}{\partial \mathbf{p}}(\mathbf{p} - \hat{\mathbf{p}}^{(0)}) + \delta_{\tilde{\mathbf{u}}_{k,i}}. \tag{2.31}$$

This is approximately a linear measurement equation, in which $\tilde{\mathbf{u}}_{k,i}$ and $\tilde{\mathbf{u}}_{k-1,i}$ are the observations. Then, the objective function to be minimized becomes:

$$\sum_{i=1}^{n}\{\tilde{\mathbf{u}}_{k,i} - \mathbf{f}(\hat{\mathbf{p}}^{(0)}) - \frac{\partial \mathbf{f}^T(\hat{\mathbf{p}}^{(0)})}{\partial \mathbf{p}}(\mathbf{p} - \hat{\mathbf{p}}^{(0)})\}^{\mathsf{T}} \mathbf{\Gamma}^{-1}_{\tilde{\mathbf{u}}_{k,i}} \{\tilde{\mathbf{u}}_{k,i} - \mathbf{f}(\hat{\mathbf{p}}^{(0)}) - \frac{\partial \mathbf{f}^T(\hat{\mathbf{p}}^{(0)})}{\partial \mathbf{p}}(\mathbf{p} - \hat{\mathbf{p}}^{(0)})\}. \tag{2.32}$$

If we further assume that the components of the image noise $\delta_{\tilde{\mathbf{u}}_{k,i}}$ are mutually independent and the covariance matrix is $\delta^2_{\tilde{\mathbf{u}}}\mathbf{I}$, for $i = 1, \cdots, n$. Note that also we have a time-invariant system, i.e., parameter $\mathbf{p}$ remains unchanged with respect to point i of the matched points $\{\tilde{\mathbf{u}}_{k,i}\}$ and $\{\tilde{\mathbf{u}}_{k-1,i}\}$.

As pointed out in the literature, the criterion of *Kalman Filtering* is still the minimum-variance. For our problem and assumptions, the *Extended Kalman Filtering* minimizes finally the following objective equation in a sequential manner:

$$\sum_{i=1}^{n} ||\tilde{\mathbf{y}}_i - \mathbf{J}_i(\hat{\mathbf{p}}^{(i)})\mathbf{p}||^2$$

where $\hat{\mathbf{p}}^{(i)}$ is the estimated $\mathbf{p}$ based on the previous i points,

$$\tilde{\mathbf{y}}_i = \tilde{\mathbf{u}}_{k,i} - \mathbf{f}(\hat{\mathbf{p}}^{(i)}) + \mathbf{J}_i(\hat{\mathbf{p}}^{(i)})\hat{\mathbf{p}}^{(i)},$$

and

$$\mathbf{J}_i(\hat{\mathbf{p}}^{(i)}) = \frac{\partial \mathbf{f}^T(\hat{\mathbf{p}}^{(i)})}{\partial \mathbf{p}}.$$

The crucial difference comes from the place where the matrix $\mathbf{J}_i$ is evaluated. Remenber that *Kalman Filtering* deals with two system equations: state equation and its measurement equation, so inside the *Kalman Filtering*, the mapping matrix of the measurement equation corresponds to the matrix $\mathbf{J}_i(\hat{\mathbf{p}}^{(i)})$ for our problem. During the computation, the *Extended Kalman Filtering* updates the motion and structure estimates sequentially by the well known four-equation structure from the first observations $\tilde{\mathbf{u}}_{k,1}$ and $\tilde{\mathbf{u}}_{k-1,1}$ to the last observations $\tilde{\mathbf{u}}_{k,n}$ and $\tilde{\mathbf{u}}_{k-1,n}$. Since in the process, $\hat{\mathbf{p}}^{(i)}$ is the estimated parameter vector $\mathbf{p}$ based on the previous i points, the corresponding Jacobian matrix $\mathbf{J}_i(\hat{\mathbf{p}}^{(i)})$ might be evaluated far from the true parameters, and then results in an inaccurate system model. This can further prevent the estimates from approaching the correct solution in later processing. Therefore, as noted by many researchers [65], [54], [53], sequential methods which are derived for linear systems generally are not suitable to solve nonlinear problems.

On the contrary, if we use nonlinear batch optimization, the Jacobian matrices $\mathbf{J}_i$ are always updated, based on all observed points available to minimize the following least-square summation,

$$\sum_{i=1}^{n} ||\tilde{\mathbf{y}}_i - \mathbf{J}_i(\hat{\mathbf{p}})\mathbf{p}||^2,$$

where

$$\tilde{\mathbf{y}}_i = \tilde{\mathbf{u}}_{k,i} - \mathbf{f}(\hat{\mathbf{p}}) + \mathbf{J}_i(\hat{\mathbf{p}})\hat{\mathbf{p}},$$

and

$$\mathbf{J}_i(\hat{\mathbf{p}}) = \frac{\partial \mathbf{f}^T(\hat{\mathbf{p}})}{\partial \mathbf{p}}.$$

In other words, The matrix $\mathbf{J}_i(\hat{\mathbf{p}})$ is updated by the whole observed image points here in the batch method, while the matrix $\mathbf{J}_i(\hat{\mathbf{p}}^{(i)})$ in the previously mentioned sequential method is updated by i observed image points. Consequently, the convergence problem of sequential methods is eliminated here. Nonlinear batch optimization gives a more accurate solution for a nonlinear problem, at the cost of more computation.

From the above discussion of batch and recursive approaches, we see that sequential processing can be done simultaneously in the course of collecting data, so less memory is required and the processing is fast, while batch processing can only be accomplished after the data collection is fulfilled for a prescribed batch size, which causes high memory consumption and expensive computational cost. Since the data size in processing monocular or stereo image sequences is larger (in orders) than in the two-view case, we have to make a reasonable trade-off between accuracy and efficiency. In particular, we have to combine batch and recursive methods in a certain way in order to keep the merits of these two methods and overcome their defects for solving nonlinear problems. The smallest batch size consists two consecutive images. To obtain more accurate motion and structure estimates, the number of consecutive images inside a batch can be increased. This consideration originated our recursive-batch approaches for motion and 3-D structure estimation from monocular and stereo image sequences, which will be discussed in detail in the following two chapters.

Chapter 3

Estimating motion and 3-D structure from monocular sequences

This chapter presents, in the case of monocular sequences, an optimal strategy to compute the three-dimensional motion and scene structure from matched image points. This strategy is optimal with respect to a matrix weighted least-squares criterion, and is characterized by stability, accuracy and efficiency. We will first review the recent literature concerning this problem and then present our approach. We will also discuss the scale problem associated with the estimated structure from monocular image sequences.

3.1 Review of the related work

Estimating motion and structure of a rigid object from a sequence of monocular images involves two main steps: (1) the matching of primitives (points, lines, surfaces, etc.) between consecutive images, and (2) the estimation of the motion and structure based on the matched primitives. Since our work focuses on the second step, we will not review the extensive literature related to the matching problem. In this work, we used a general two-view matching algorithm presented in [59] to compute the displacement field of our monocular image sequences.

Although both monocular and stereo camera systems can endow machines with the ability to perceive structure from motion, there is in general a major difference between motion and structure estimations from sequences produced by these two types of systems. In the stereo case, once the baseline is calibrated, the absolute translation vector and structure can be obtained through triangulation, while in the monocular case the translation vector and structure can only be determined up to a common scale factor. Nevertheless, if absolute a priori data is available about the structure and (or) the translation vector, the global scale factor can further be determined. Stereo motion is inherently redundant since it provides several means of motion and structure estimation (left sequence, right sequence or 3-D sequence resulted from stereo pairs). Exploiting advantageously this redundancy involves a larger computation load since it requires both stereo and temporal matchings.

Several linear two-view algorithms have been proposed ([42], [44], [45], [46], [47]) to solve for motion parameters from two monocular views. The backbone of these algorithms is the introduction of "essential parameters" to make the original nonlinear problem linear in terms of the essential parameters. The essential parameters are summarized in a 3 by 3 matrix **E** defined in terms of motion parameters (see Appendix B for reference). A set of equations are established that relates the image coordinates of the matched points to the elements of the matrix **E**. Since those equations are linear and homogeneous in the element of **E**, the essential parameters can be determined up to a scale factor. Then the motion parameters are solved from the essential parameters. Finally the relative depth (depth scaled by the magnitude of translation) of each point is determined from the motion parameters and its observed projections.

The essential parameter matrix **E** has 8 degrees of freedom. Each point correspondence gives one linear equation for **E**. This is why at least 8 point correspondences are needed to solve for **E**. The relative motion between the camera and the rigid scene has 6 degrees of freedom (3 for rotation and 3 for translation). There is therefore a dependency among the 8 essential parameters. It results in a high sensitivity of the estimated motion and structure solutions to the presence of noise. The motion and structure solutions from the linear algorithms can be further optimized iteratively to improve the accuracy.

To improve the linear solution, several nonlinear optimization strategies were proposed. An earlier two-step approach of motion and structure estimation from monocular perspective images was proposed in [48]. Its first step consists of estimating the motion parameters, using a robust linear algorithm that gives a closed-form solution for motion parameters and scene structure. Its second step improves the results from the linear algorithm using a nonlinear optimization or a maximum likelihood estimation strategy. Since preliminary estimates are available, the algorithm reaches the global minimum quickly and reliably. Korsten and Houkes proposed a generic method of estimating geometry and motion of a surface from image sequences [82]. It uses the idea of linearizing a nonlinear parameter estimation problem around a previous guess. After the linearization, standard parameter estimation methods, closed form or iterative, can be applied to the problem. Spetsakis and Aloimonos [58] presented two ways to take into consideration the dependency among the intermediate variables in **E**. They formulate the problem as a quadratic minimization problem with a nonlinear constraint. Their first strategy of solution is a variation of Newton's method, and the second is a decomposition of the problem into two parts, rotation and translation, in order to reduce the dimensionality.

To directly deal with long monocular image sequences, Broida and Chellappa [65] presented recursive and batch estimation approaches to extract a type of specialized object motion parameters, 2-D constant translation and rotation, from a 1-D sequence of noisy images of known structures. It was also noted that for nonlinear optimization, batch methods are superior to sequential ones. The drawbacks of sequential methods were explained in [54] and Chapter 2 of this thesis. Recently, Broida and Chellappa extended their research to smooth 3-D motion using direct batch methods to process a whole image sequence [11]. Nonlinear optimization is used in the formulation of the batch strategy, and conjugated Gradient search is adopted for the iterative process. Kumar, Tirumalai and Jain proposed a similar batch approach [53]. The problem is formulated in terms of a nonlinear least squares minimization, and solved iteratively using the Levenberg-Marquardt method. A deficiency of these approaches is that they do not exploit the relationship between the motion parameters and the structure. This results in a high dimensional search space, and generates problems of instability. When the number of 3-D points increases the dimension of the parameter space increases quickly. Indeed, in order to combat the effect of noise, the common strategy is to use more point correspondences to form an overdetermined system. However, when the number of points increase, the dimension of the parameter space increases at the same pace, and quickly produces a prohibitively large search space. It is therefore essential, for any practical algorithm for long sequence to explore and utilize the relationship between the structure and motion, in order to limit the size of the optimization problem.

To retain the advantages of both closed form and iterative nonlinear optimization methods in case of two monocular views, and to overcome the inefficiency and instability problems of optimization algorithms previously mentioned, we propose here a recursive-batch nonlinear optimization approach for motion and structure estimation from monocular image sequences. This approach, in which the motion parameters are not constrained to remain constant, is well adapted to the intrinsic nonlinearity of the problem (induced by motion and projection), and possesses the following characteristics:

1. Using the proposed recursive-batch approach, the observation data in a monoc-

ular image sequence is subdivided into groups and estimation is done in a sequential fashion among groups of data. Within each data group, estimation is done in a batch fashion. In this way, advantages of sequential processing are kept and the performance of the algorithm is drastically improved, as compared to a direct nonlinear *Iterated Kalman Filtering* [69].

2. A matrix-weighted nonlinear objective function has been constructed in the optimization process to properly weigh the data according to the uncertainty of each information element.

3. The dimension of the search space in the optimization process is reduced by taking into account the relationship between the structure and motion parameters, so that it is equal to the number of motion parameters only (3 for translation and 3 for rotation) and does not depend upon the number of 3-D points.

4. The nonlinear optimal solution can then be achieved in two steps. The first step consists of using a linear algorithm to compute a good initial solution. There are few cases where such an initial solution can not be acquired [44], [43]. The second step uses a matrix-weighted objective function and a parameter decomposition strategy. The nonlinear optimization is accomplished using the *Gauss-Markov* theorem [51], to get approximate minimum variance estimates of motion and structure.

The scale problem of the structure is also studied. It is shown that the scale factor of any two consecutive images in a monocular image sequence is determined by the scale factor of the first two images. A simple equation is established to show this relationship. Therefore, when many image frames are involved, the number of scale factor is still one, instead of many.

A major achievement described in this chapter consists of the ability of the proposed estimation strategy to handle long sequences of natural scenes. In our experiment, an image sequence consisting of 20 images were obtained using a camera mounted on a robot manipulator, which has been calibrated to correct lens distortions. The point correspondences in the image sequence were established automatically. The computer simulations and the experiments with real images have shown that the optimization method developed not only greatly reduces the computational complexity, but also yields a substantial improvement over the results produced by linear algorithms.

3.2 Camera set-up

We assume a pin-hole camera with unit focal length. Let the origin of the camera-centered coordinate system coincide with the projection center of the camera and the z axis coincide with the optical axis. As visible rigid objects move within the field of view of the camera system, a sequence of images is acquired. This is illustrated in Figure 3.1, where $\mathbf{x}_{k-1,i}$ is the camera-centered coordinate vector of the ith point on the object at time t_{k-1} and $\mathbf{u}_{k-1,i}$ is its image coordinate vector.

At instant k, the image vector of the ith space point $\mathbf{x}_{k,i}$ is given by:

$$\mathbf{X}_{k,i} = \frac{1}{z_{k,i}}\,\mathbf{x}_{k,i} = \begin{pmatrix} u_{k,i,1} \\ u_{k,i,2} \\ 1 \end{pmatrix} \tag{3.1}$$

where $\mathbf{u}_{k,i} = (u_{k,i,1},\ \ u_{k,i,2})^T$ represent the image coordinates. The interframe motion can then be expressed in terms of the projections in consecutive images by:

$$z_{k,i}\begin{pmatrix} \mathbf{u}_{k,i} \\ 1 \end{pmatrix} = z_{k-1,i}R_{k,k-1}\begin{pmatrix} \mathbf{u}_{k-1,i} \\ 1 \end{pmatrix} + \mathbf{T}_{k,k-1}. \tag{3.2}$$

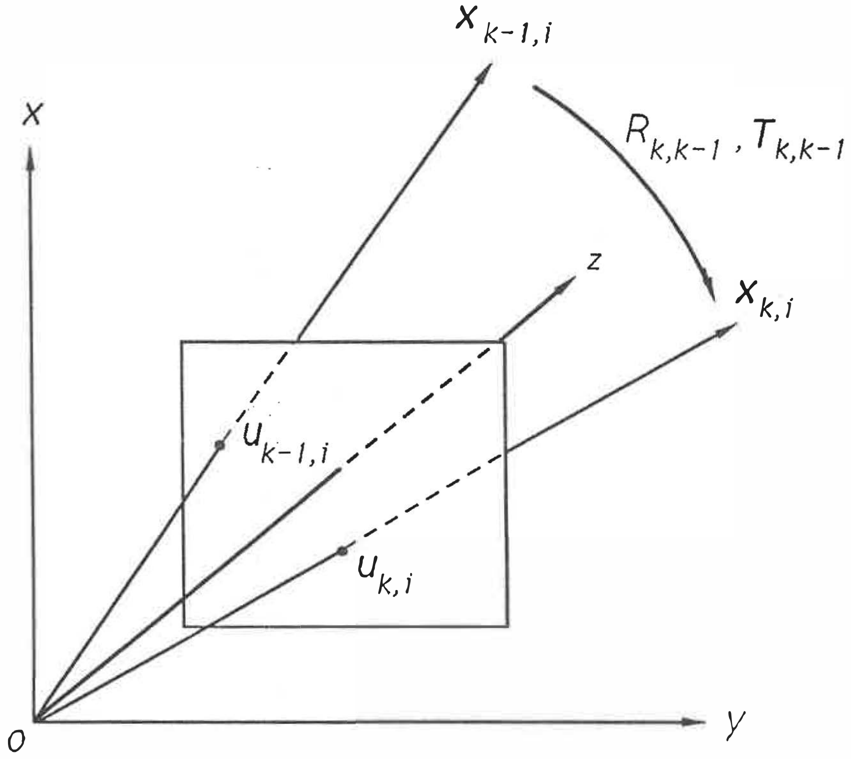


Figure 3.1: Camera-centered coordinate system.

3.3 About the scale of the estimated structure

As mentioned before, in general, there is a scale factor problem associated with any monocular image sequence. The scale factor problem arises because, in the perspective projection expression (3.1), multiples of all three coordinates $(x_{k,i},\ y_{k,i}\ z_{k,i})$ of any space point produce the same image. In this section, we discuss in detail the scale problem inherent to structure estimation from monocular sequences. It will be shown that there exists a proportional relationship between the scale factor introduced in processing any pair of consecutive images and the scale factor corresponding to the first image pair of the sequence. Consequently, only one global scale factor is involved in a monocular image sequence, instead of many.

If at time t_k, the relative displacement between the camera and any 3-D point in the scene is unknown: the image correspondences $\{\mathbf{u}_{k-1,i}\} \leftrightarrow \{\mathbf{u}_{k,i}\}$, $i = 1, \ldots, n$, can only yield the rotation $R_{k,k-1}$ and the unit translation vector $\overset{\circ}{\mathbf{T}}_{k,k-1}$. The intrinsic scale factor $\alpha_k = \| \mathbf{T}_{k,k-1} \|$ remains undetermined [45], [46], this problem will be clearer in the following derivation. However, we will show that the kth scale factor α_k is proportional to the first scale factor $\alpha_1 = \| \mathbf{T}_{1,0} \|$. First, the coordinates of any 3-D point on the observed object in the local coordinate system, as shown in Figure 3.2, at time t_1 can be related by interframe motion $R_{1,0}$, $\mathbf{T}_{1,0}$ to its coordinates at time t_0,

$$\mathbf{x}_{1,i} = R_{1,0}\mathbf{x}_{0,i} + \mathbf{T}_{1,0}, \quad i = 1,\ \ldots,\ n. \tag{3.3}$$

Dividing equation (3.3) by α_1 yields

$$\frac{\mathbf{x}_{1,i}}{\alpha_1} = R_{1,0}\frac{\mathbf{x}_{0,i}}{\alpha_1} + \overset{\circ}{\mathbf{T}}_{1,0}\ . \tag{3.4}$$

The magnitude of the translation vector α_1 and the absolute depth of the object points $z_{1,i}$ and $z_{0,i}$ cannot be determined by monocular vision. This can be seen from (3.4), which still holds when α_1, $\mathbf{x}_{1,i}$ and $\mathbf{x}_{0,i}$ are multiplied by any nonzero constant.

Note that from image vectors $\mathbf{u}_{0,i}$, $\mathbf{u}_{1,i}$ and motion parameters $R_{1,0}$, $\mathring{\mathbf{T}}_{1,0}$, we can obtain the following two vectors from one of the two-view algorithms, [50].

$$\tilde{\mathbf{x}}_{1,i} = \frac{\mathbf{x}_{1,i}}{\alpha_1}, \qquad \tilde{\tilde{\mathbf{x}}}_{0,i} = \frac{\mathbf{x}_{0,i}}{\alpha_1}, \tag{3.5}$$

which satisfy

$$\tilde{\mathbf{x}}_{1,i} = R_{1,0}\tilde{\tilde{\mathbf{x}}}_{0,i} + \mathring{\mathbf{T}}_{1,0}\,. \tag{3.6}$$

Similarly, at time t_2 we have

$$\tilde{\mathbf{x}}_{2,i} = R_{2,1}\tilde{\tilde{\mathbf{x}}}_{1,i} + \mathring{\mathbf{T}}_{2,1} \tag{3.7}$$

where,

$$\tilde{\mathbf{x}}_{2,i} = \frac{\mathbf{x}_{2,i}}{\alpha_2}, \qquad \tilde{\tilde{\mathbf{x}}}_{1,i} = \frac{\mathbf{x}_{1,i}}{\alpha_2}. \tag{3.8}$$

Combining equations (3.8) and (3.5) yields

$$\tilde{\tilde{\mathbf{x}}}_{1,i} = \frac{\mathbf{x}_{1,i}}{\alpha_2} = \frac{\alpha_1}{\alpha_2}\tilde{\mathbf{x}}_{1,i} \tag{3.9}$$

If we define

$$\beta_1 = \frac{\|\tilde{\mathbf{x}}_{1,i}\|}{\|\tilde{\tilde{\mathbf{x}}}_{1,i}\|} \tag{3.10}$$

from the expression in (3.9), we get

$$\alpha_2 = \beta_1\alpha_1. \tag{3.11}$$

Thus, from t_0 up to t_k, we have,

$$\alpha_k = \beta_{k-1}\cdots\beta_1\alpha_1, \tag{3.12}$$

where β_j is defined as

$$\beta_j = \frac{\|\tilde{\mathbf{x}}_{j,i}\|}{\|\tilde{\tilde{\mathbf{x}}}_{j,i}\|}. \tag{3.13}$$

which can be determined from some 3-D points on the object that are visible in three frames at t_{j-1}, t_j and t_{j+1}, respectively.

From equation (3.12), we observe that once α_1 is determined (given, or computed using other information) then $\alpha_k, k = 2, \ldots,$ are determined as well. However, it is important to notice that, in order to determine the value of β_k, at least one space point should remain visible at all three instants t_{k-1}, t_k and t_{k+1} and its image correspondences should be known. In fact, we can relate any α_k to any α_j, $k > j$, by

$$\alpha_k = \beta_{k-1} \cdots \beta_j \alpha_j. \tag{3.14}$$

Once the scale factor for any consecutive image pair is given, all other α_i can be determined based on expression (3.14). In the following discussion, without loss of generality, we always assume that at t_k, α_1 is unknown and $\beta_{k-1}, \ldots, \beta_1$ are all computed.

Finally, for any three nonzero 3-D vectors $\mathbf{x} = (x_1, x_2, x_3)^{\top}$, $\mathbf{y} = (y_1, y_2, y_3)^{\top}$ and $\mathbf{z} = (z_1, z_2, z_3)^{\top}$, we have

$$\frac{\|\alpha\mathbf{x} - \alpha\mathbf{y}\|}{\|\alpha\mathbf{x} - \alpha\mathbf{z}\|} = \frac{\|\mathbf{x} - \mathbf{y}\|}{\|\mathbf{x} - \mathbf{z}\|}, \tag{3.15}$$

where α is a non-zero constant. This means that the distance ratio of the scaled 3-D vectors is the same as that of the real 3-D vectors. In the presence of noise, the distance ratios of the computed 3-D vectors can be used as one of the measures to assess the accuracy of a motion and structure estimation algorithm from monocular image sequences, without requiring evaluation of the actual scale factor.

3.4 Motion and structure optimization

As mentioned earlier, batch methods generally outperform sequential methods for nonlinear problems. In addition, the estimation of early interframe motions of an image sequence can benefit from the processing of later image frames if there are multiple observations of some part of the scene. In principle, it should be desirable to use all of the image projections in a sequence as input, all the interframe motion parameters as variables, and iteratively minimize the squared sum of the discrepancies between the observed image projections and the back projections from the computed structure. However, it is only feasible under the assumption of smooth motion, and only when the number of point correspondences and the number of image frames are small as in [11]. In a general situation, the above total batch method is impractical due to possible violations of a smooth motion assumption, the enormous memory requirement and the excessive computational cost caused by the large number of point correspondences and image frames. On the other hand, a sequential technique possesses some desirable properties: old observations can be discarded once they have been used for estimation, and a relatively small amount of data and computation is required for updating estimates.

In order to achieve good performance without suffering from excessive computational cost, we use batch processing only for those data that have considerable interactions, where the improvement of the batch methods is most significant. Since there is no smooth constraint enforced upon our motion parameters, a natural data set for batch processing consists of all the points in two consecutive image frames. Sequential technique is used for those loosely related data, i.e., images which are far apart. The framework of our recursive-batch approach for motion and structure estimation from monocular image sequences is shown in Figure 3.2.

At time t_k, there are two sets of data available, $\{\tilde{\mathbf{u}}_{k,i}\}$ and $\{\tilde{\mathbf{u}}_{k-1,i}\}$, $i = 1,\cdots,n$. These are point projections at the previous and current instants. Since the information accumulated through all preceding frames can be well conveyed by the estimated structure $\mathbf{x}^*_{k-1,i}$, represented in the local coordinate system at the previous instant, $\{\tilde{\mathbf{u}}_{k-1,i}\}$ $i = 1,\cdots,n$ can be implicitly represented by $\mathbf{x}^*_{k-1,i}$, $i = 1,\cdots,n$. Each $\mathbf{x}^*_{k-1,i}$ has an error covariance matrix $\Gamma_{\mathbf{x}^*_{k-1,i}}$, which is obtained by the structure estimator and indicates the expected accuracy of the estimate. The interframe motion

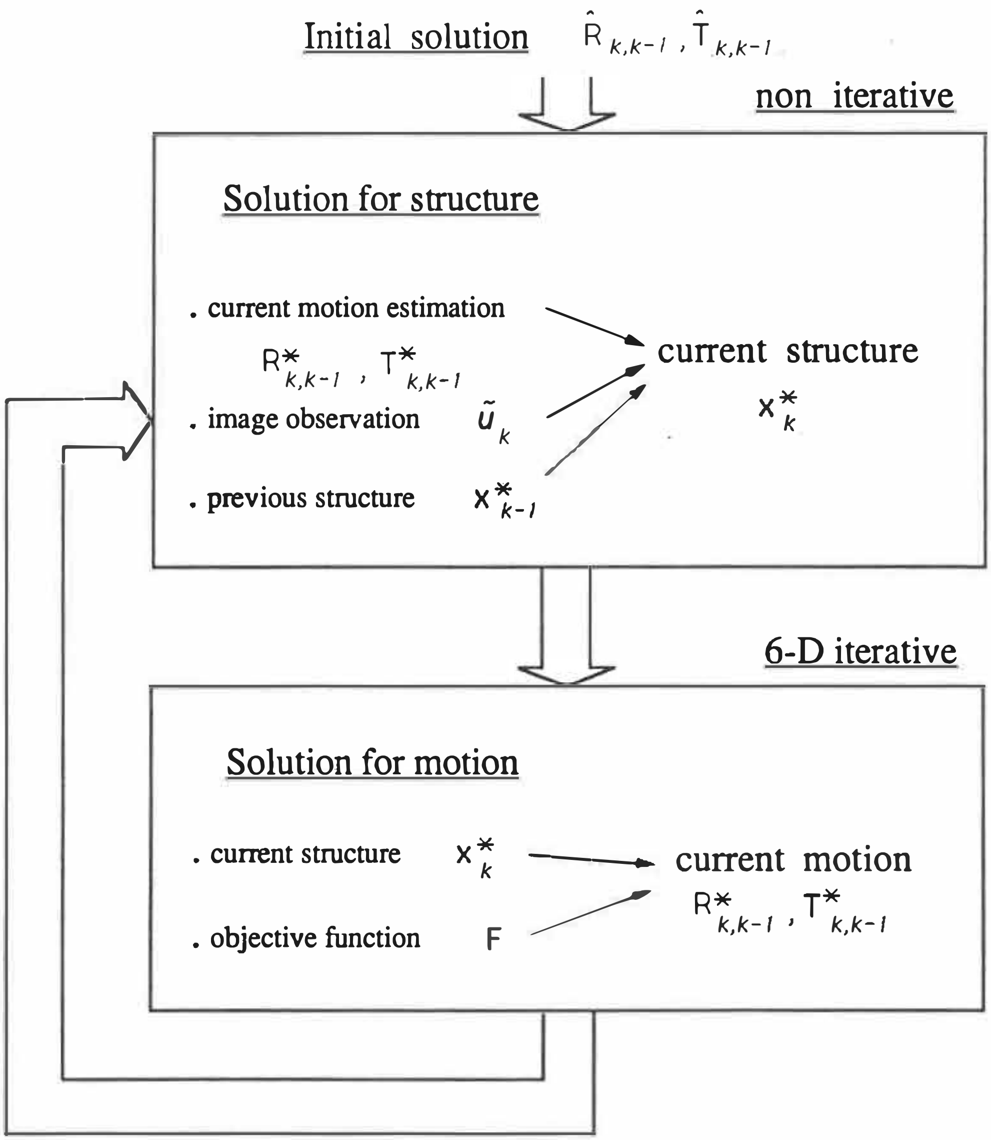

Initial solution $\hat{R}_{k,k-1}, \hat{T}_{k,k-1}$
non iterative
Solution for structure
. current motion estimation
$R^*_{k,k-1}, T^*_{k,k-1}$
. image observation $\tilde{u}_k$
. previous structure x^*_{k-1}
current structure
x^*_k
6-D iterative
Solution for motion
. current structure x^*_k
. objective function F
current motion
$R^*_{k,k-1}, T^*_{k,k-1}$

Figure 3.2: The framework of our recursive-batch approach for motion and structure estimation from monocular image sequences.

from the $(k-1)$th frame to the kth frame, represented by $\mathbf{m}_{k,k-1}$ designating the pair ($R_{k,k-1}$ and $\mathbf{T}_{k,k-1}$), transforms the noise free structure $\mathbf{x}_{k-1,i}$ at t_{k-1} into $\mathbf{x}_{k,i}$ at t_k:

$$\mathbf{x}_{k,i} = \mathbf{m}_{k,k-1}(\mathbf{x}_{k-1,i}), \quad i = 1, \ldots, n. \tag{3.16}$$

In reality, only the estimated structure $\mathbf{x}^*_{k-1,i}$ is available at time t_k, which can be expressed as

$$\mathbf{x}^*_{k-1,i} = \mathbf{x}_{k-1,i} + \delta_{\mathbf{x}^*_{k-1,i}}. \tag{3.17}$$

Given any set of motion parameters $\mathbf{m}_{k,k-1}$, we can establish a prediction $\hat{\mathbf{x}}_{k,i}$ of the structure $\mathbf{x}_{k,i}$ at the current time t_k:

$$\hat{\mathbf{x}}_{k,i} = R_{k,k-1}\mathbf{x}^*_{k-1,i} + \mathbf{T}_{k,k-1} = \mathbf{x}_{k,i} + \delta_{\hat{\mathbf{x}}_{k,i}}, \tag{3.18}$$

and estimate the prediction error: $\delta_{\hat{\mathbf{x}}_{k,i}} = R_{k,k-1}\delta_{\mathbf{x}^*_{k-1,i}}$, so the error covariance of the predicted structure $\hat{\mathbf{x}}_{k,i}$ is given by

$$\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} = E[\delta_{\hat{\mathbf{x}}_{k,i}}\delta^{\mathrm{T}}_{\hat{\mathbf{x}}_{k,i}}] = R_{k,k-1}\mathbf{\Gamma}_{\mathbf{x}^*_{k-1,i}}R^{\mathrm{T}}_{k,k-1}. \tag{3.19}$$

The image observation $\tilde{u}_{k,i}$ is related to the structure $\mathbf{x}_{k,i}$ by

$$\tilde{\mathbf{u}}_{k,i} = \mathbf{u}_{k,i}(\mathbf{x}_{k,i}) + \delta_{\tilde{\mathbf{u}}_{k,i}}, \quad i = 1, \ldots, n \tag{3.20}$$

where $\mathbf{u}_{k,i}(\mathbf{x}_{k,i})$ is the noise-free image point and $\delta_{\tilde{\mathbf{u}}_{k,i}}$ is the observation noise in $\mathbf{u}_{k,i}$. Using (3.16), equation (3.20) can also be written as

$$\tilde{\mathbf{u}}_{k,i} = \mathbf{u}_{k,i}(\mathbf{m}_{k,k-1}(\mathbf{x}_{k-1,i})) + \delta_{\tilde{\mathbf{u}}_{k,i}}, \quad i = 1, \ldots, n. \tag{3.21}$$

This is a nonlinear equation that relates the previous structure $\mathbf{x}_{k-1,i}$, the new observations $\tilde{u}_{k,i}$ and the interframe motion $\mathbf{m}_{k,k-1}$. A direct sequential approach [65] solves for $\mathbf{m}_{k,k-1}$ through updating sequentially on i, which results in a long convergence period and poor performance.

We assume that the components of the image noise $\delta_{\tilde{\mathbf{u}}_{k,i}}$ caused by quantization error, calibration error, feature matching error, etc., are zero mean, mutually independent, and independent of $\mathbf{x}_{k,i}$, with a error covariance matrix $\sigma_{\mathbf{u}}^2\mathbf{I}$.

From the two sets of observations $\{\hat{\mathbf{x}}_{k,i}\}$ and $\{\tilde{\mathbf{u}}_{k,i}\}$, $i = 1, \cdots, n$, we form our objective function as a nonlinear weighted least squares sum:

$$\begin{aligned} & f(\mathbf{m}_{k,k-1}, \mathbf{x}_{k,\bullet}) = \\ & \quad \sum_{i=1}^{n} \Big\{ (\hat{\mathbf{x}}_{k,i} - \mathbf{x}_{k,i})^{\mathsf{T}} \mathbf{\Gamma}^{-1}_{\hat{\mathbf{x}}_{k,i}} (\hat{\mathbf{x}}_{k,i} - \mathbf{x}_{k,i}) \\ & \quad + [\tilde{\mathbf{u}}_{k,i} - \mathbf{u}_{k,i}(\mathbf{x}_{k,i})]^{\mathsf{T}} \mathbf{\Gamma}^{-1}_{\tilde{\mathbf{u}}_{k,i}} [\tilde{\mathbf{u}}_{k,i} - \mathbf{u}_{k,i}(\mathbf{x}_{k,i})] \Big\} . \end{aligned} \tag{3.22}$$

Its parameter set consists of the interframe motion $\mathbf{m}_{k,k-1}$ and the unknow structure $\mathbf{x}_{k,\bullet} = \{\mathbf{x}_{k,i} \;\; i = 1, \cdots, n\}$. This objective function contains two terms for each point i. The first term measures the discrepancy between the improved structure estimate of $\mathbf{x}_{k,i}$ and the predicted structure $\hat{\mathbf{x}}_{k,i}$ . .The second term corresponds to moving the previous estimated structure $\mathbf{x}^*_{k-1,i}$ according to the current motion estimate $\mathbf{m}_{k,k-1}$, then projecting the result onto the image plane to obtain $\mathbf{u}_{k,i}$ and comparing it to the actual observation $\tilde{\mathbf{u}}_{k,i}$. The reason why we take weighted least squares other than unweighted least squares is because the reliability of each information element is not the same. The inverses of the error covariance matrices in the objective function are symmetric and positive definite. They account for the different reliabilities of the different information elements. That is, a more reliable information element will have a larger weight in the objective function, and a less reliable information element will have a smaller weight.

It is shown in (2.11) that the structure $\mathbf{x}_{k,i}$ depends upon the interframe motion $\mathbf{m}_{k,k-1}$, once the previous structure $\mathbf{x}_{k-1,i}$ is fixed. Then, once the interframe motion $\mathbf{m}_{k,k-1}$ is known, the optimal estimation of the structure $\mathbf{x}^*_{k,i}$ at time t_k can be directly determined from $\mathbf{m}_{k,k-1}$, the image projections $\tilde{\mathbf{u}}_{k,i}$ and the previous structure estimate $\mathbf{x}^*_{k-1,i}$, without resorting to an iteration process. We can thus reduce the number of search parameters of the objective function in the iterative process. Namely, we rewrite the minimization of the objective function, owing to its continuity, as:

$$\min_{\mathbf{m}_{k,k-1},\mathbf{x}_{k,\bullet}} f(\mathbf{m}_{k,k-1},\mathbf{x}_{k,\bullet}) = \min_{\mathbf{m}_{k,k-1}} g(\mathbf{m}_{k,k-1}) \tag{3.23}$$

where

$$g(\mathbf{m}_{k,k-1}) = \min_{\mathbf{x}_{k,\bullet}} f(\mathbf{m}_{k,k-1},\mathbf{x}_{k,\bullet}). \tag{3.24}$$

This indicates that the whole minimization can be decomposed into two phases, the iterative search phase in motion space and the noniterative estimation phase (minimum variance estimation) for fixed motion parameters in structure space.

We now describe the minimum variance structure estimator for any set of giving motion parameters. Linearizing $\mathbf{u}_{k,i}(\mathbf{x}_{k,i})$ within a small neighborhood of the predicted structure $\hat{\mathbf{x}}_{k,i}$ with a given motion $\mathbf{m}_{k,k-1}$ yields

$$\mathbf{u}_{k,i}(\mathbf{x}_{k,i}) \approx \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i}) + \frac{\partial \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i})}{\partial \mathbf{x}_{k,i}}(\mathbf{x}_{k,i} - \hat{\mathbf{x}}_{k,i}). \tag{3.25}$$

By substituting this expression of $\mathbf{u}_{k,i}(\mathbf{x}_{k,i})$ into (3.24), the objective function becomes quadratic in $\mathbf{x}_{k,i}$. Or, more procisely, we have two measurements about the structure $\mathbf{x}_{k,i}$ at time t_k:

$$\hat{\mathbf{x}}_{k,i} = R_{k,k-1}\mathbf{x}^*_{k-1,i} + \mathbf{T}_{k,k-1} = \mathbf{x}_{k,i} + \delta_{k,i}, \tag{3.26}$$

and

$$\tilde{\mathbf{u}}_{k,i} - \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i}) + \frac{\partial \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i})}{\partial \mathbf{x}_{k,i}}\hat{\mathbf{x}}_{k,i} = \frac{\partial \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i})}{\partial \mathbf{x}_{k,i}}\mathbf{x}_{k,i} + \delta_{\tilde{\mathbf{x}}_{k,i}}. \tag{3.27}$$

These two equations have the same parameter vector $\mathbf{x}_{k,i}$, and $\hat{\mathbf{x}}_{k,i}$ and $\tilde{\mathbf{u}}_{k,i}$ are known. The optimal structure estimate $\mathbf{x}^*_{k,i}$ can be computed based on the linear minimum variance estimation presented in Chapter 2:

$$\mathbf{x}^*_{k,i} = (\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{A})^{-1}\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{B} \tag{3.28}$$

where

$$\mathbf{A} = (\mathbf{I} \quad \mathbf{A}_{k,i})^{\top} = \begin{pmatrix} \mathbf{I} \\ \frac{\partial \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i})}{\partial \mathbf{x}_{k,i}} \end{pmatrix},$$

$$\mathbf{B} = (\mathbf{b}_1 \quad \mathbf{b}_2)^{\top} = \begin{pmatrix} \hat{\mathbf{x}}_{k,i} \\ \tilde{\mathbf{u}}_{k,i} - \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i}) + \frac{\partial \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i})}{\partial \mathbf{x}_{k,i}} \hat{\mathbf{x}}_{k,i} \end{pmatrix},$$

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} & 0 \\ 0 & \mathbf{\Gamma}_{\tilde{\mathbf{u}}_{k,i}} \end{pmatrix},$$

$$\mathbf{\Gamma}_{\tilde{\mathbf{u}}_{k,i}} = \begin{pmatrix} \sigma_{\mathbf{u}}^2 & 0 \\ 0 & \sigma_{\mathbf{u}}^2 \end{pmatrix}.$$

and the error covariance matrix is

$$\mathbf{\Gamma}_{\mathbf{x}^*_{k,i}} = E[(\mathbf{x}^*_{k,i} - \mathbf{x}_{k,i})(\mathbf{x}^*_{k,i} - \mathbf{x}_{k,i})^{\top}] = (\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{A})^{-1}, \tag{3.29}$$

where $\mathbf{A}$ is a 5×3 matrix and $\mathbf{\Gamma}$ is a 5×5 matrix. One of the important aspects of this procedure is that the minimum variance estimation provides at each instant an assessment of the structure uncertainty in the form of the error covariance matrices $\mathbf{\Gamma}_{\mathbf{x}^*_{k,i}}$. This uncertainty is then transfered to the next estimation step, and thus, in case of multiple observations of the same part of the scene, the redundancy of the observations is implicitly taken into account by the structure estimator (3.28).

At time t_k, the nonlinear weighted least squares minimization in (3.22) is thus performed in such a way that for each $\mathbf{m}_{k,k-1}$, the matrix-weighted square discrepancies in equation (3.22) are computed from the predicted structure $\hat{\mathbf{x}}_{k,i}$, the noniterative structure estimate $\mathbf{x}^*_{k,i}$ and the observed image projection $\tilde{\mathbf{u}}_{k,i}$. The motion parameters $\mathbf{m}^*_{k,k-1}$ which minimize the objective function (3.22) is determined using an iterative search in a 6-D motion space, and the corresponding optimal structure $\mathbf{x}^*_{k,\bullet}$

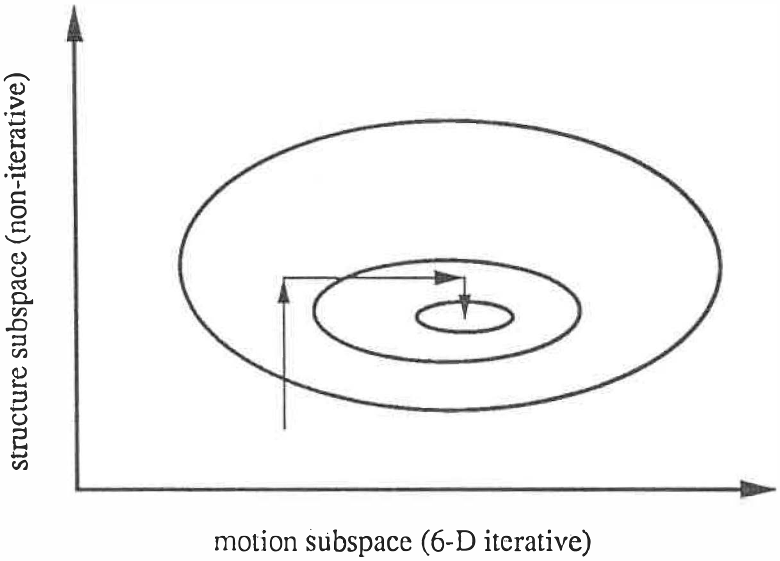

structure subspace (non-iterative)
motion subspace (6-D iterative)

Figure 3.3: An illustration of the decomposition strategy used in the search space.

is obtained using equation (3.28). This optimization process is illustrated in Figure 3.3.

In summary, two measures are taken which make our approach significantly more efficient than those described in [53] and [65]: (1) the recursive-batch method with a batch corresponding to an image frame; (2) the exclusion of structure from the iterative search space by exploring the relationship between the motion parameters and the structure.

Given a monocular image sequence, a linear algorithm [50] is used to directly compute an initial guess of $\mathbf{m}_{k,k-1}$, thus avoiding divergence and speeding up the computation. The image frames are processed sequentially, and the accumulated information is kept in the form of the 3-D structure (i.e. no past image frames need to be kept). The points in each frame are processed in a batch fashion to obtain the current interframe motion and the updated structure. Our approach is thus of a recursive-batch type.

Any unbiased linear minimum variance estimator in batch form can be converted into one of the two categories of recursive forms: information form or covariance form. The information form is often more useful than the covariance form in analytical studies. In on-line applications, where speed of computation is often the most important consideration, the covariance form is preferable to the information form. This is because a smaller matrix needs to be inverted in the covariance form [70]. In order to facilitate the implementation of the motion and structure estimation, we use the well known inverse theorem [55], [69], [70] to change equation (3.28) into the following covariance recursive form (the derivation is shown in Appendix E):

$$\mathbf{x}^*_{k,i} = (\mathbf{A}^\top \mathbf{\Gamma}^{-1} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{\Gamma}^{-1} \mathbf{B} = \hat{\mathbf{x}}_{k,i} + \mathbf{G}_{k,i}(\mathbf{b}_2 - \mathbf{A}_{k,i}\hat{\mathbf{x}}_{k,i}), \tag{3.30}$$

where the gain matrix is

$$\mathbf{G}_{k,i} = \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} \mathbf{A}^\top_{k,i} (\mathbf{A}_{k,i} \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} \mathbf{A}^\top_{k,i} + \mathbf{\Gamma}_{\tilde{\mathbf{u}}_{k,i}})^{-1},$$

and the error covariance matrix is

$$\mathbf{\Gamma}_{\mathbf{x}^*_{k,i}} = (\mathbf{A}^\top \mathbf{\Gamma}^{-1} \mathbf{A})^{-1} = \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} - \mathbf{G}_{k,i}\mathbf{A}_{k,i}\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}, \tag{3.31}$$

because that only 2×2 matrix inverse is required in (3.30) compared to 3×3 matrix inverse required in (3.28).

3.5 Uncertainty updating and the complete optimization procedure

In this section, we will first discuss how to update the relative motion and the structure uncertainties. These uncertainties estimates, in the form of error covariance matrices, are not only crucial for obtaining the expected accuracy of the estimates, but also important for processing the next image frame based on the current computed estimates. The uncertainties in the estimates of the corresponding global motion and structure can be obtained in a similar way. The complete optimization procedure will then be presented.

3.5.1 Updating the motion and structure uncertainties

Let a rotation matrix $\mathbf{R}$ be expressed as: $R = R(\lambda)$ with $\lambda = (\alpha \;\; \beta \;\; \gamma)$, where α, β are two variables representing the rotation axis and γ represents the rotation angle around this axis (alternatively α, β, γ can represent the three rotation angles around z, y, x axes, respectively). The interframe motion vector $\mathbf{m}_{k,k-1}$ is thus defined as:

$$\mathbf{m}_{k,k-1} = (\alpha_{k,k-1},\; \beta_{k,k-1},\; \gamma_{k,k-1},\; t_{x,k,k-1},\; t_{y,k,k-1},\; t_{z,k,k-1})^\top. \tag{3.32}$$

At time t_k, each visible point provides 2 "observations": the predicted structure, $\hat{\mathbf{x}}_{k,i}$, and the current image observations, $\tilde{\mathbf{u}}_{k,i}$,

$$\tilde{\mathbf{y}}_i = \begin{pmatrix} \hat{\mathbf{x}}_{k,i} \\ \tilde{\mathbf{u}}_{k,i} \end{pmatrix}$$

with the error covariance matrix

$$\mathbf{\Gamma}_{\tilde{\mathbf{y}}_i} = \begin{pmatrix} \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} & 0 \\ 0 & \mathbf{\Gamma}_{\tilde{\mathbf{u}}_{k,i}} \end{pmatrix}.$$

When we obtain the interframe motion estimate $\mathbf{m}^*_{k,k-1}$ which minimizes the objective function (3.22) using the space decomposition method described in the last section, its error covariance matrix is, according to equation (2.23),

$$\begin{aligned} \mathbf{\Gamma}_{\mathbf{m}^*_{k,k-1}} &= E[(\mathbf{m}^*_{k,k-1} - \mathbf{m}_{k,k-1})(\mathbf{m}^*_{k,k-1} - \mathbf{m}_{k,k-1})^\top] \\ &= \left\{ \sum_{i=1}^{n} \left[\left(\frac{\partial \mathbf{x}_{k,i}(\mathbf{m}^*_{k,k-1})}{\partial \mathbf{m}_{k,k-1}} \right)^\top \mathbf{\Gamma}^{-1}_{\hat{\mathbf{x}}_{k,i}} \left(\frac{\partial \mathbf{x}_{k,i}(\mathbf{m}^*_{k,k-1})}{\partial \mathbf{m}_{k,k-1}} \right) \right.\right. \\ &\quad \left.\left. + \frac{1}{\sigma^2_{\tilde{\mathbf{u}}}} \left(\frac{\partial \mathbf{u}_{k,i}(\mathbf{m}^*_{k,k-1})}{\partial \mathbf{m}_{k,k-1}} \right)^\top \left(\frac{\partial \mathbf{u}_{k,i}(\mathbf{m}^*_{k,k-1})}{\partial \mathbf{m}_{k,k-1}} \right) \right] \right\}^{-1}. \end{aligned} \tag{3.33}$$

The uncertainty of the estimated optimal structure $\mathbf{x}^*_{k,i}$ can then be derived as follows. Let the previous relative structure $\mathbf{x}_{k-1,i}$ and the current relative structure $\mathbf{x}_{k,i}$ be related by the interframe motion $R_{k,k-1}$ and $\mathbf{T}_{k,k-1}$,

$$\mathbf{x}_{k,i} = R_{k,k-1}\mathbf{x}_{k-1,i} + \mathbf{T}_{k,k-1} = \mathbf{f}_1(\mathbf{m}_{k,k-1}, \mathbf{x}_{k-1,i}). \tag{3.34}$$

We expand the above equation in the vicinity of the estimated optimal motion parameters and the optimal structure at the previous instant, $(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})$, by Taylor Polynomial, while neglecting the terms of order higher than two,

$$\begin{aligned} \mathbf{x}_{k,i} &= \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i}) + \frac{\partial \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})}{\partial \mathbf{m}_{k,k-1}}(\mathbf{m}_{k,k-1} - \mathbf{m}^*_{k,k-1}) \\ &\quad + \frac{\partial \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})}{\partial \mathbf{x}_{k-1,i}}(\mathbf{x}_{k-1,i} - \mathbf{x}^*_{k-1,i}) \end{aligned} \tag{3.35}$$

After rearranging the above equation, we have

$$\begin{aligned} \hat{\mathbf{x}}_{k,i} - \mathbf{x}_{k,i} &= \frac{\partial \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})}{\partial \mathbf{m}_{k,k-1}}(\mathbf{m}^*_{k,k-1} - \mathbf{m}_{k,k-1}) + R^*_{k,k-1}(\mathbf{x}^*_{k-1,i} - \mathbf{x}_{k-1,i}) \\ &= \frac{\partial \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})}{\partial \mathbf{m}_{k,k-1}}\delta_{\mathbf{m}^*_{k,k-1}} + R^*_{k,k-1}\delta_{\mathbf{x}^*_{k-1,i}}. \end{aligned} \tag{3.36}$$

Where predicted structure $\hat{\mathbf{x}}_{k,i} = \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})$. The error covariance matrix of $\hat{\mathbf{x}}_{k,i}$ is

$$\begin{aligned}
\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} &= E[(\hat{\mathbf{x}}_{k,i} - \mathbf{x}_{k,i})(\hat{\mathbf{x}}_{k,i} - \mathbf{x}_{k,i})^{\top}] \\
&= \left(\frac{\partial \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})}{\partial \mathbf{m}_{k,k-1}}\right) \mathbf{\Gamma}_{\mathbf{m}^*_{k,k-1}} \left(\frac{\partial \mathbf{f}_1(\mathbf{m}^*_{k,k-1}, \mathbf{x}^*_{k-1,i})}{\partial \mathbf{m}_{k,k-1}}\right)^{\top} \\
&+ R^*_{k,k-1} \mathbf{\Gamma}_{\mathbf{x}^*_{k-1,i}} R^{*T}_{k,k-1}.
\end{aligned} \tag{3.37}$$

Notice that equation (3.37) is different from equation (3.19), where $\mathbf{m}_{k,k-1}$ is a given variable vector only. So, after the optimal motion and structure estimates, the uncertainty of the structure $\mathbf{x}^*_{k,i}$ is

$$\mathbf{\Gamma}_{\mathbf{x}^*_{k,i}} = (\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{A})^{-1} \tag{3.38}$$

where

$$\mathbf{A} = (\mathbf{I} \quad \mathbf{A}_{k,i})^{\top} = \begin{pmatrix} \mathbf{I} \\ \frac{\partial \mathbf{u}_{k,i}(\hat{\mathbf{x}}_{k,i})}{\partial \mathbf{x}_{k,i}} \end{pmatrix},$$

and

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}_{\bar{\mathbf{u}}_{k,i}} \end{pmatrix}.$$

In the above block matrix, $\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}$ is determined as in equation (3.37). The traces of error covariance matrices $\mathbf{\Gamma}_{\mathbf{m}^*_{k,k-1}}$ and $\mathbf{\Gamma}_{\mathbf{x}^*_{k,i}}$ give us the expected square norm of the error in the corresponding estimated vectors.

3.5.2 Updating global motion and uncertainty

Between instants t_l and t_k, the trajectory of the moving section of the scene (when the camera is stationary) is defined by the expressions:

$$\begin{aligned}
R_{k,l} &= R_{k,k-1}R_{k-1,k-2}\cdots R_{l+1,l}, \qquad k > l, \\
\mathbf{T}_{k,l} &= \sum_{i=l+1}^{k} R_{k,i}\mathbf{T}_{i,i-1}, \qquad k > l.
\end{aligned} \tag{3.39}$$

When the scene is static and the camera moving the following expression can be used to update the global position of the camera:

$$\begin{aligned} R_k &= R_{k-1}R_{k,k-1}^{-1}, \\ \mathbf{T}_k &= \mathbf{T}_{k-1} - R_k\mathbf{T}_{k,k-1}. \end{aligned} \tag{3.40}$$

For example, let us assume that we wish to represent the current motion with respect to t_0, and at each time instant, we update it from the previous position. Let

$$\mathbf{m}_{k,0} = \mathbf{F}(\mathbf{m}_{k,k-1}, \mathbf{m}_{k-1,0}). \tag{3.41}$$

Where $\mathbf{m}_{k,k-1}$ represents interframe motion $(R_{k,k-1}, \mathbf{T}_{k,k-1})$. And $\mathbf{m}_{k,0}$ and $\mathbf{m}_{k-1,0}$ represent global motions at instants t_k and t_{k-1} respectively. This equation can be linearized in the vicinity of the estimated motion $\mathbf{m}^*_{k,k-1}$ and $\mathbf{m}^*_{k-1,0}$. Then, the difference between the estimated motion and the true motion parameters, neglecting the terms of order higher than two, is

$$\begin{aligned} \delta_{\mathbf{m}^*_{k,o}} &= \mathbf{m}^*_{k,0} - \mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0}) \\ &= \frac{\partial\mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0})}{\partial\mathbf{m}_{k,k-1}}\delta_{\mathbf{m}^*_{k,k-1}} \\ &\quad + \frac{\partial\mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0})}{\partial\mathbf{m}_{k-1,0}}\delta_{\mathbf{m}^*_{k-1,0}}. \end{aligned} \tag{3.42}$$

So the error covariance matrix of the global motion can be updated at each time t_k by

$$\begin{aligned} \Gamma_{\mathbf{m}^*_{k,0}} &= E[(\mathbf{m}^*_{k,0} - \mathbf{m}_{k,0})(\mathbf{m}^*_{k,0} - \mathbf{m}_{k,0})^{\top}] \\ &= \left(\frac{\partial\mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0})}{\partial\mathbf{m}_{k,k-1}}\right)\Gamma_{\mathbf{m}^*_{k,k-1}}\left(\frac{\partial\mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0})}{\partial\mathbf{m}_{k,k-1}}\right)^{\top} \\ &+ \left(\frac{\partial\mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0})}{\partial\mathbf{m}_{k-1,0}}\right)\Gamma_{\mathbf{m}^*_{k-1,0}}\left(\frac{\partial\mathbf{F}(\mathbf{m}^*_{k,k-1}, \mathbf{m}^*_{k-1,0})}{\partial\mathbf{m}_{k-1,0}}\right)^{\top} \end{aligned} \tag{3.43}$$

where $k > 1$.

3.5.3 Steps of the complete optimization algorithm

In summary, the recursive-batch optimization algorithm for motion and structure estimation from monocular image sequences can be described as follows:

Step 1 Compute $\mathbf{x}^*_{0,i}$ and $\mathbf{\Gamma}_{\mathbf{x}^*_{0,i}}$, for all point i.

Compute the scaled 3-D scene points $\mathbf{x}^*_{0,1}, \cdots, \mathbf{x}^*_{0,n}$ at t_0 using the five-step two-view linear algorithm shown in Appendix B (see also [54]). Their error covariance matrices at time t_0 can also be obtained. Let

$$\mathbf{x}^*_{0,i} = z^*_{0,i}[u^*_{0,i,1} \quad u^*_{0,i,2} \quad 1]^{\mathsf{T}}, \tag{3.44}$$

$$\delta_{\mathbf{x}^*_{0,i}} = [u^*_{0,i,1} \quad u^*_{0,i,2} \quad 1]^{\mathsf{T}} \delta_{z^*_{0,i}} + [\delta_{u^*_{0,i,1}} \quad \delta_{u^*_{0,i,2}} \quad 0]^{\mathsf{T}} z^*_{0,i}. \tag{3.45}$$

Where $u^*_{0,i,1}$ and $u^*_{0,i,2}$ are the image projections of 3-D coordinates $\mathbf{x}^*_{0,i}$. If we denote $\mathbf{u}^*_i = [u^*_{0,i,1} \quad u^*_{0,i,2} \quad 1]^{\mathsf{T}}$ and $\delta_{\mathbf{u}_i} = [\delta_{u^*_{0,i,1}} \quad \delta_{u^*_{0,i,2}} \quad 0]^{\mathsf{T}}$, then the error covariance matrix of $\mathbf{x}^*_{0,i}$ is

$$\mathbf{\Gamma}_{\mathbf{x}^*_{0,i}} = \mathbf{u}^*_i \sigma^2_{z^*_{0,i}} (\mathbf{u}^*_i)^{\mathsf{T}} + z^{*2}_{0,i} \delta_{\mathbf{u}_i} \delta^{\mathsf{T}}_{\mathbf{u}_i}. \tag{3.46}$$

This error covariance matrix is always positive definite, see Appendix F for proof.

Step 2 At t_k, $k \geq 1$:

(2.1) **Linear solution**: from $\tilde{\mathbf{u}}_{k,i}$, $\tilde{\mathbf{u}}_{k-1,i}$, compute the initial solution $\mathring{\mathbf{m}}_{k,k-1}$ (i.e., $R_{k,k-1}, \mathring{\mathbf{T}}_{k,k-1}$), using the five-step linear two-view algorithm presented in Appendix B (see also [50]).

(2.2) **Optimization**: compute $\mathbf{x}^*_{k,i}$, and $\mathbf{m}^*_{k,k-1}$ (i.e., $R^*_{k,k-1}, \mathbf{T}^*_{k,k-1}$) by improving $\mathring{\mathbf{m}}_{k,k-1}$ according to (3.23) and (3.28).

(2.3) Compute $\mathbf{\Gamma}_{\mathbf{m}^*_{k,k-1}}$, and $\mathbf{\Gamma}_{\mathbf{x}^*_{k,i}}$ using equations (3.33) and (3.38).

(2.4) In the case of a moving scene and a stationary camera, compute $R^*_{k,0}$, $\mathbf{T}^*_{k,0}$ and $\mathbf{\Gamma}_{\mathbf{m}^*_{k,0}}$ using equations (2.14) and (3.43). In the case of a moving camera and a stationary scene, use equations (2.19) and (3.43).

Step 3 Terminate the procedure if at the last monocular image, otherwise $k := k+1$, and go back to step 2).

3.6 Simulation and experiments

To assess the performance of the recursive-batch optimization algorithm for motion and 3-D structure estimation from monocular image sequences, simulations as well as experiments with a real monocular image sequence of a natural scene have been conducted.

3.6.1 Simulation

The objective of the simulation is to investigate the performance of our approach, when the ground truth is known and the noise level is controlled, by precisely measuring errors in the estimates. In addition, a statistical assessment can be achieved through numerous trials.

In the simulation, 3-D points are generated randomly for each trial, between depth 0 and 200 meters, with a uniform distribution. The simulated camera has a square image frame whose field of view (side to side) is about 44°. The error in the image projections of the space points is simulated by additive zero-mean independent Gaussian noise, whose variance is equal to that of a uniform digitization with a 256×256- pixel image. To establish the scale factor, we supposed that the average distance $\sum_1^n\{z_{0,i}\}$ of the visible points is known, noting that

$$\alpha_1 = \frac{\sum_1^n\{z_{0,i}\}}{\sum_1^n\{\tilde{z}_{0,i}\}}, \tag{3.47}$$

where $\{\tilde{z}_{0,i}\}$ can be computed from the two-view linear algorithm shown in Appendix B.

We let the simulated monocular camera system move along a zigzag path, while undergoing slight rotations through the static environment similar to the movement of a human head, as illustrated in Figure 3.4.

In order to simulate reasonable conditions of visibility, only those points lying within a range of 20 meters in the local coordinate system are used for motion analysis. Fifty images were generated in this way, and the visible 3-D points of the last view were totally different from those of the first. For the first pair of images about 300 points were used to compute the initial structure at t_0, while 70 points were used for the remaining images.

The true rotation matrix R_k of the monocular camera system with respect to the global coordinate system was 5° around a rotation axis $(0.3m, 0.3m, 1m)$ for odd k, and $-5°$ around the same rotation axis for even k. The given translation vector $\mathbf{T}_k$ of the monocular camera system with respect to the global coordinate system was $(0.5, -0.5, 1.2k)$ for odd k, and $(-0.5, 0.5, 1.2k)$ for even k, with meter as the unit. After 50 images, the forward distance travelled by the monocular camera system was equal to 60 meters.

The error in rotation was measured as the relative error in the rotation matrix, defined as the Euclidean norm of the matrix difference between the estimated and true rotation matrices, divided by the norm of the true rotation matrix, $\|\hat{R}-R\|/\|R\|$. Since $\|\hat{R}I-RI\| = \|\hat{R}-R\|$, the geometrical meaning of the relative error in a rotation matrix is that it measures the root mean-squared error (RMSE) in the 3 unit vectors of a rotated orthonormal frame. This error measure gives a relatively stable measure over a wide range of rotation angles, so it can be used as a normalized measure to compare the accuracy between different rotations. The error in the interframe translation vector $\mathbf{T}_{k,k-1}$ is defined as the direction error, i.e., the Euclidean norm of the vector difference between the estimated unit translation vector and the true unit translation vector. The error in the global translation vector $\mathbf{T}_k$ consists of a direction

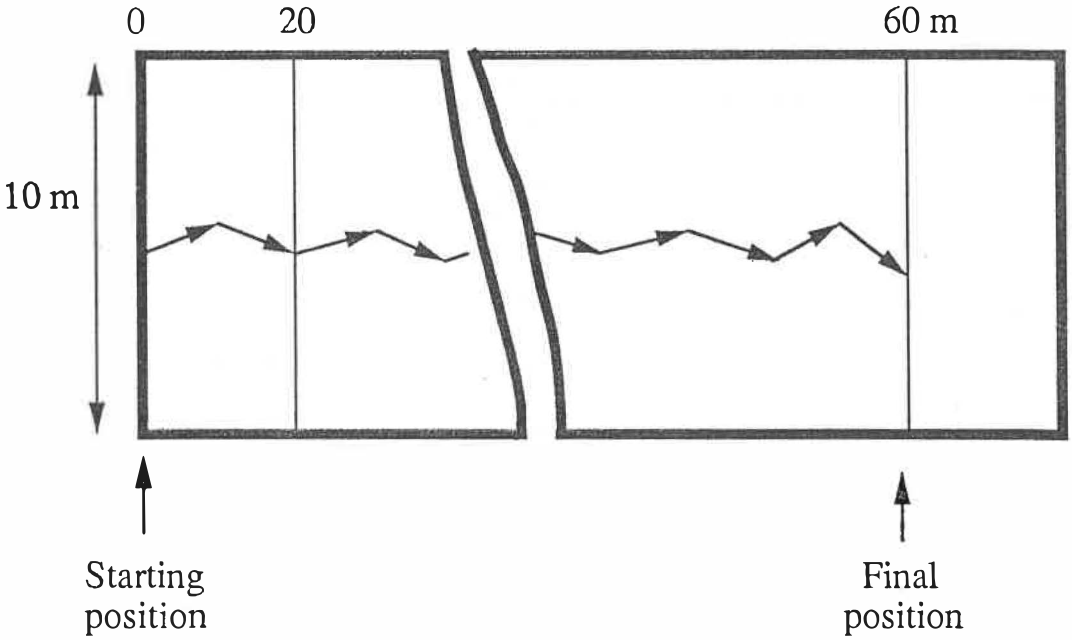


Figure 3.4: The navigation path of the monocular camera system in the simulation.

error as well as an absolute error, i.e., the Euclidean norm of the vector difference between the estimated and true translation vectors. For the iterative optimization, the "dunlsf" subroutine in IMSL library was used.

The simulation consists of 100 trials to obtain average errors. For each trial, a different set of 3-D points was used.

Figure 3.5 shows the estimation error for the interframe motion. We can see that the average direction error in the interframe translation and the relative error in the interframe rotation have been reduced by more than 65% as compared to the errors produced by the linear algorithm.

To measure the error in the estimated structure, we use the root mean-squared error (RMSE) of the positional errors of the 3-D points. A 3-D point $\mathbf{x}_i$ is estimated up to a scale factor by $\alpha\tilde{\mathbf{x}}_i$. We determine the scale factor α such that

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \alpha\tilde{\mathbf{x}}_i\|^2} \tag{3.48}$$

is minimized. This measure indicates the best scale fit to the 3-D positions.

We have computed RMSE at each frame for the local structure, with respect to the current camera reference, and the global structure, with respect to the global coordinate system. The results are depicted in Figure 3.6 and 3.7 respectively.

A substantial decrease in the local structure error in Figure 3.6 appears in the first few frames only. This is due to the overdetermination available through more views. But the local structure error does not continue to decrease further with time since new points come in and old points go out continuously. The global structure error shown in Figure 3.7 increases with time because of the accumulated error in β_i's in (3.12). It shows that a single scale factor α_1 can not fit well every frame due to the presence of noise. This is a very important phenomenon in long monocular sequence analysis.

Figure 3.8 illustrates the estimation error for the motion with respect to the global

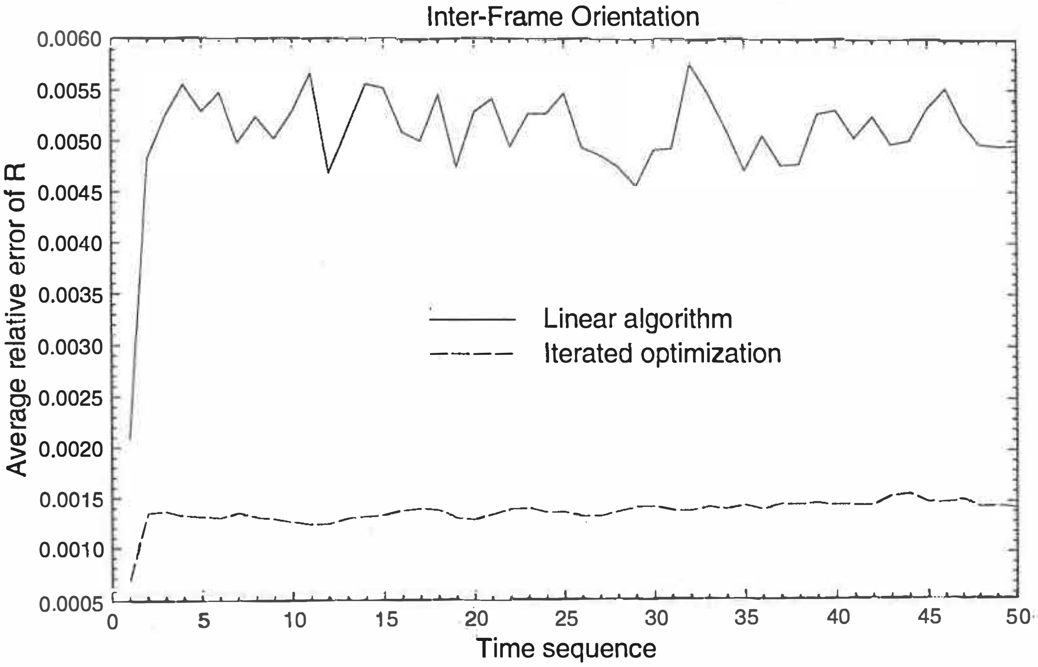

Inter-Frame Orientation
Average relative error of R
0.0060
0.0055
0.0050
0.0045
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0
5
10
15
20
25
30
35
40
45
50
Time sequence
Linear algorithm
Iterated optimization

Figure 3.5 (a)

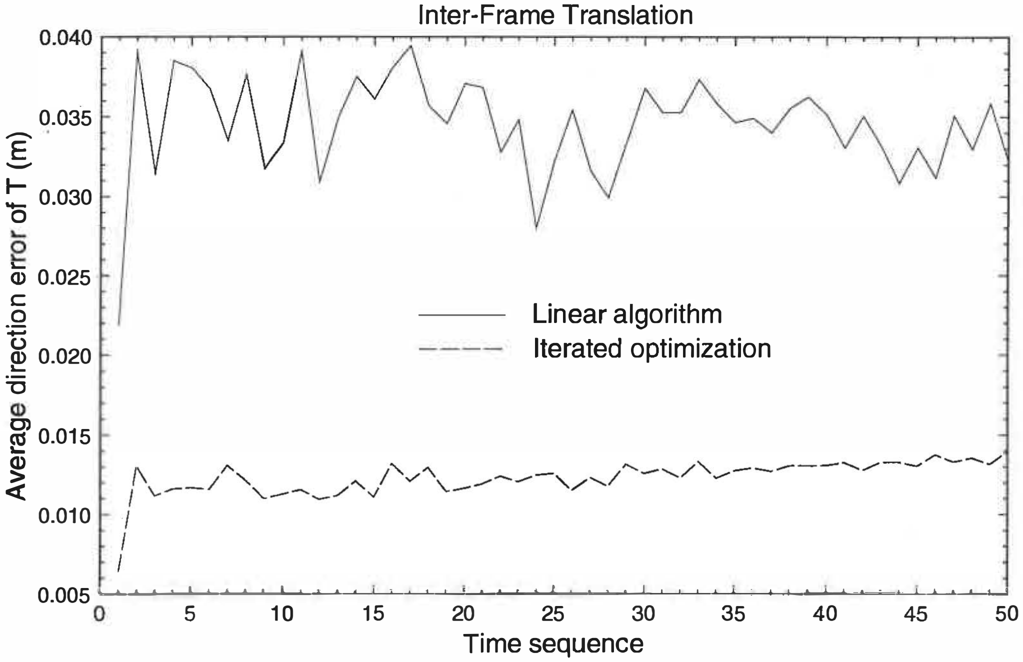


Figure 3.5 (b)

Figure 3.5: Simulation results: errors in the estimated interframe motion. (a) error in the interframe rotation matrix $R_{k,k-1}$; (b) the direction error in the interframe translation vector $\mathbf{T}_{k,k-1}$.

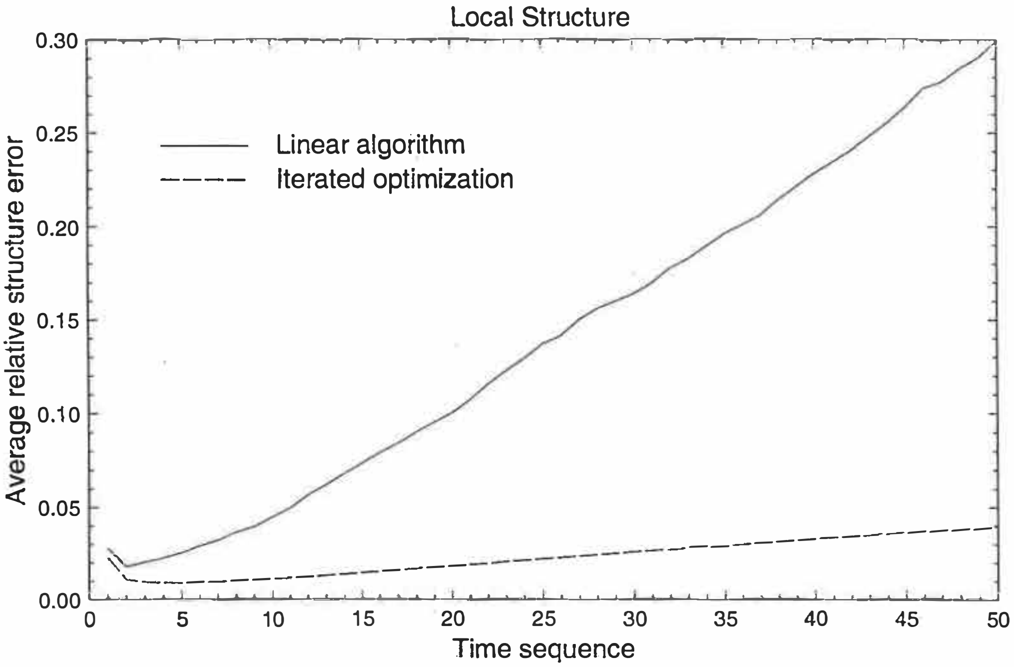


Figure 3.6: Simulation result: error in the estimated local structure.

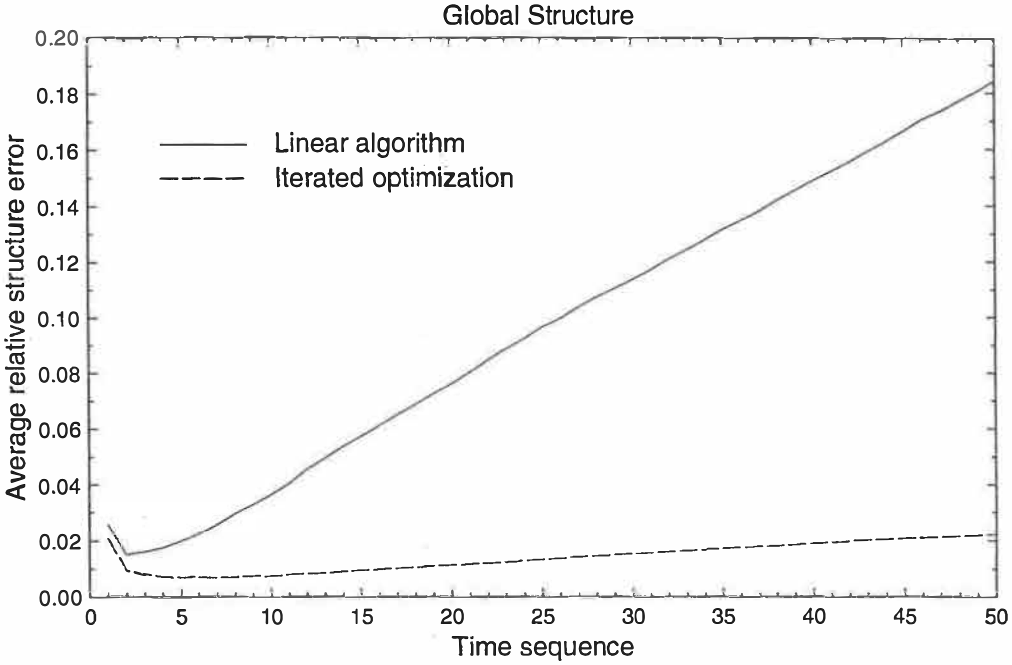


Figure 3.7: Simulation result: error in the estimated global structure.

coordinate system, which at t_0 coincides with the camera coordinate system. We can see that the global motion error increases gradually. This is because the global motion was computed from the interframe motions as in (2.19), so its error was accumulated through every related interframe motion. In Figure 3.8, we notice that the direction of translation is improved in the early stages, but due to the increasing magnitude of the translation vector, it degrades later when the accumulated error predominates. In the simulation, only certain type of interframe motion ground truth was used. More simulations with different kinds of interframe motions are needed to be conducted for investigating the relationship between the accuracy of the estimates and the motion type.

3.6.2 Experiments with monocular image sequences

Principles of camera calibration

Before describing our experiments with real monocular image sequences, we will first describe the camera calibration procedure which was adopted in the experiments.

The calibration of cameras is considered to be an essential part of an artificial vision system. An accurate calibration of cameras is especially crucial for applications that involve quantitative measurements such as dimensional measurements, depth from stereoscopy or motion from images.

The objective of camera calibration is to estimate the internal and external parameters of each camera. To acquire a large field of view, the lenses for a vision system must be wide-angle ones. Therefore, nonlinear distortion is significant. The camera model used in our experiments is a distortion model which accounts for major sources of camera distortion: radial, decentering and thin prism distortions, characterized by a set of distortion parameters (see [66] for details). The calibration consists of two steps. In the first step, calibration parameters are estimated using a closed-form so-

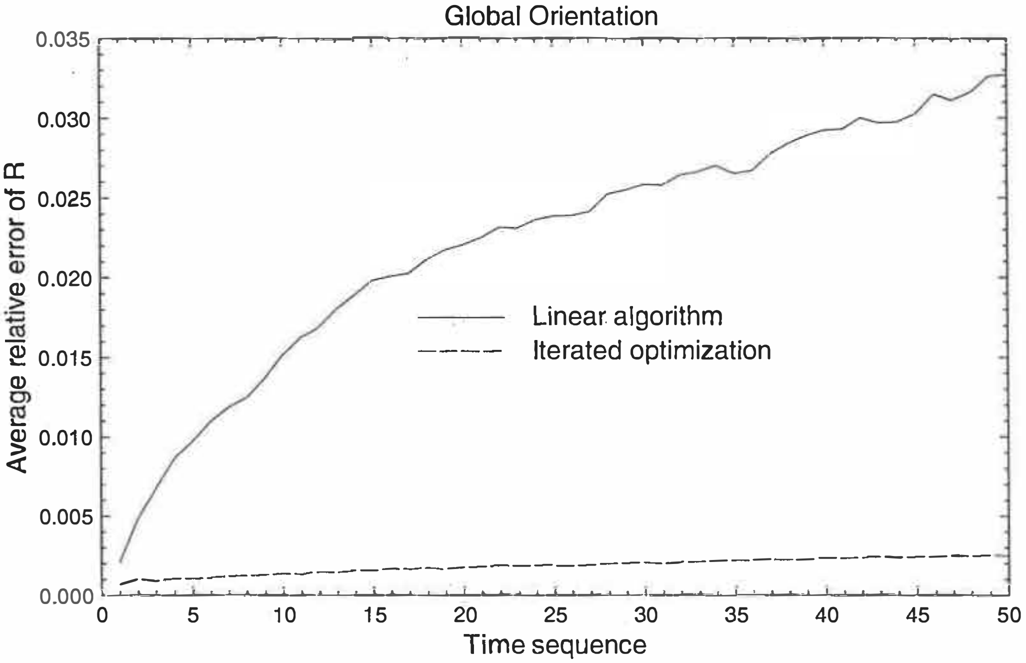

Global Orientation
Average relative error of R
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25
30
35
40
45
50
Time sequence
Linear algorithm
Iterated optimization

Figure 3.8 (a)

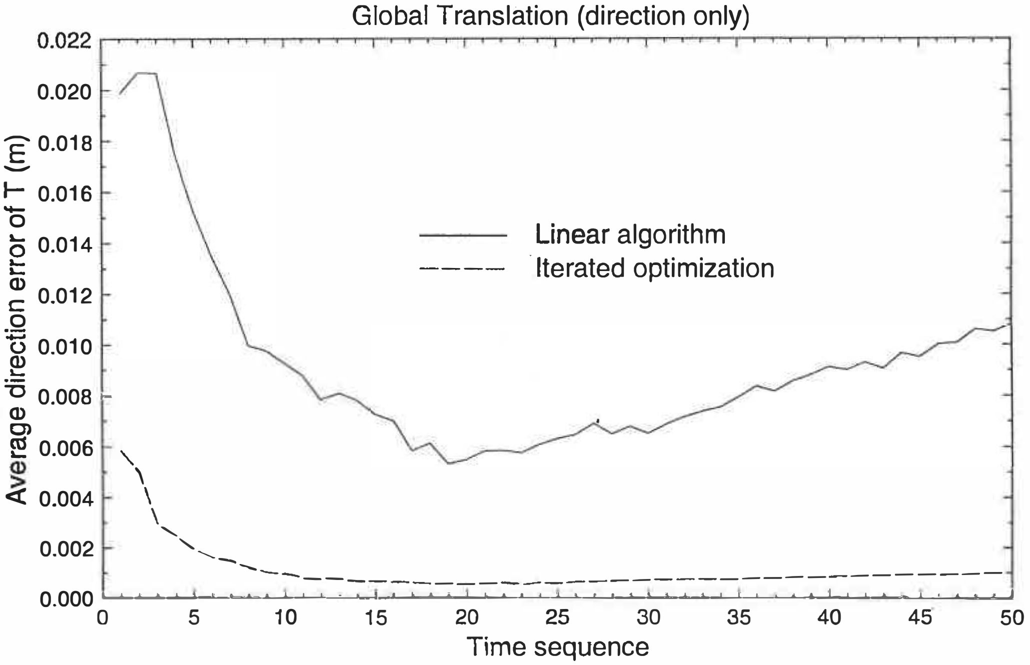

Global Translation (direction only)
Average direction error of T (m)
0.022
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
5
10
15
20
25
30
35
40
45
50
Time sequence
Linear algorithm
Iterated optimization

Figure 3.8 (b)

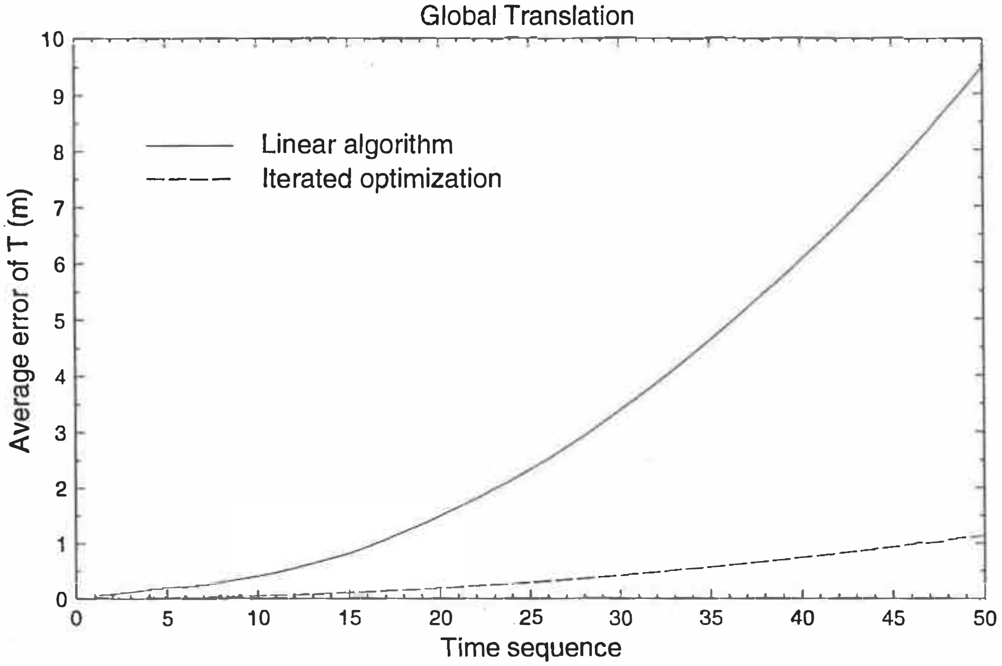


Figure 3.8 (c)

Figure 3.8: Simulation results: errors in the estimated global motion. (a) error in the global rotation matrix R_k; (b) the direction error in the global translation vector $\mathbf{T}_k$; (c) the absolute error in the global translation vector $\mathbf{T}_k$.

lution based on a distortion-free camera model. In the second step, the parameters estimated in the first step are improved iteratively through nonlinear optimization, taking into account camera distortions. According to the minimum variance estimation principle, the objective function to be minimized is the mean-squared discrepancy between the observed image points of a set of known points in space, called control points, and their inferred image projections that are computed based on the estimated calibration parameters. In our calibration process, the control points are the corner points of the black squares on the calibration plate, shown in Figure 3.9.

The camera model used in the calibration process will be briefly introduced in the following.

Let (x, y, z) represent the coordinates of any visible point P in a fixed reference system (world coordinate system) and let (x_c, y_c, z_c) represent the coordinates of the same point in a camera-centered coordinate system (note that these coordinate systems are independent of the coordinate systems used for estimating motion and structure). As illustrated in Figure 3.10, the origin of the camera-centered coordinate system coincides with the optical center of the camera, and the z_c axis coincides with its optical axis. The image plane, which corresponds to the image sensing array, is assumed to be parallel to the (x_c, y_c) plane and at a distance f to the origin, where f represents the (effective) focal length of the camera. The relationship between the world and camera-centered coordinate systems is given by

$$(x_c, y_c, z_c)^\top = R(x, y, z)^\top + \mathbf{T} \tag{3.49}$$

where $R = (r_{i,j})$ is a 3×3 rotation matrix defining the camera orientation and $\mathbf{T} = (t_1, t_2, t_3)^\top$ is a translation vector defining the camera position.

We now define in the image plane the image coordinate system (O', u, v) where O' represents the principal point of the image plane (i.e., the intersection of the image plane with the optical axis) and where the u and v axes are chosen parallel to the

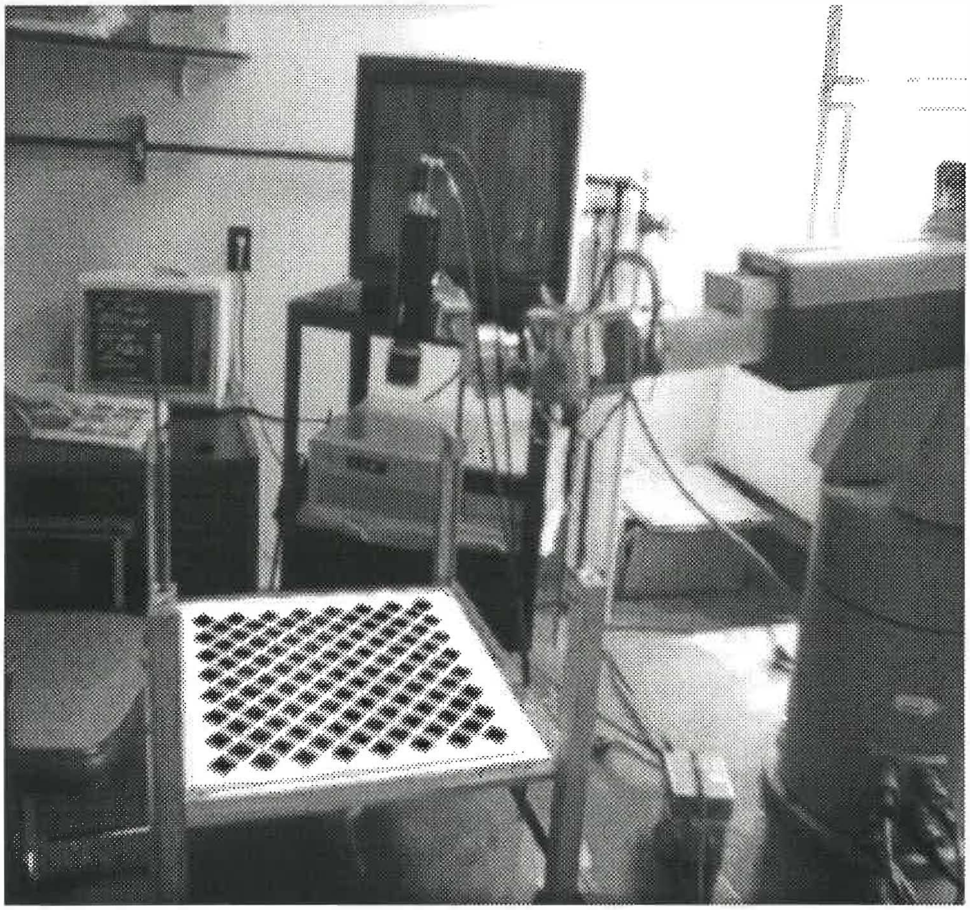


Figure 3.9: The calibration plate used in our experiments. The control points are the corner points of the black squares on the plate.

x_c and y_c axes. It should be stressed that, owing to possible misalignments of the CCD array, O' does not necessarily coincide with the geometrical center of the image plane. The image plane coordinates of the point P are given by the equations

$$\begin{aligned} u &= f x_c / z_c \\ v &= f y_c / z_c \end{aligned} \tag{3.50}$$

Finally, if we denote by (r, c) the position of the corresponding pixel in the digitized image, this position is related to the image plane coordinates by the expressions

$$\begin{aligned} r - r_0 &= s_u u \\ c - c_0 &= s_v v \end{aligned} \tag{3.51}$$

where (r_0, c_0) denotes the pixel position of the principal point O'. The coordinates (r, c) can be considered as the row and column numbers in a CCD array. In other words, the x_c and y_c axes are chosen to be parallel to row and column directions, respectively. As can be noticed in Figure 3.10, the adopted conventions impose s_u to be negative and s_v to be positive. Combining (3.49), (3.50) and (3.51) leads to the following expressions that relate the pixel position, the world coordinates and the various parameters to be calibrated

$$\begin{aligned} \frac{u}{f} = \frac{r - r_0}{f_u} &= \frac{r_{1,1}x + r_{1,2}y + r_{1,3}z + t_1}{r_{3,1}x + r_{3,2}y + r_{3,3}z + t_3} \overset{\text{def}}{=} \dot{u} \\ \frac{v}{f} = \frac{c - c_0}{f_v} &= \frac{r_{2,1}x + r_{2,2}y + r_{2,3}z + t_2}{r_{3,1}x + r_{3,2}y + r_{3,3}z + t_3} \overset{\text{def}}{=} \dot{v} \end{aligned} \tag{3.52}$$

where $(\dot{u}, \dot{v})$ defines the coordinates in the normalized image plane that is located at $z = 1$, and $f_u = s_u f$ and $f_v = s_v f$ are called the row focal length and the column focal length, respectively. The internal parameters r_0, c_0, f_u, f_v determine the image coordinates of a point, given the spatial position of the corresponding 3-D point with respect to the camera. $\mathbf{T}$ and R are called the external parameters of the camera.

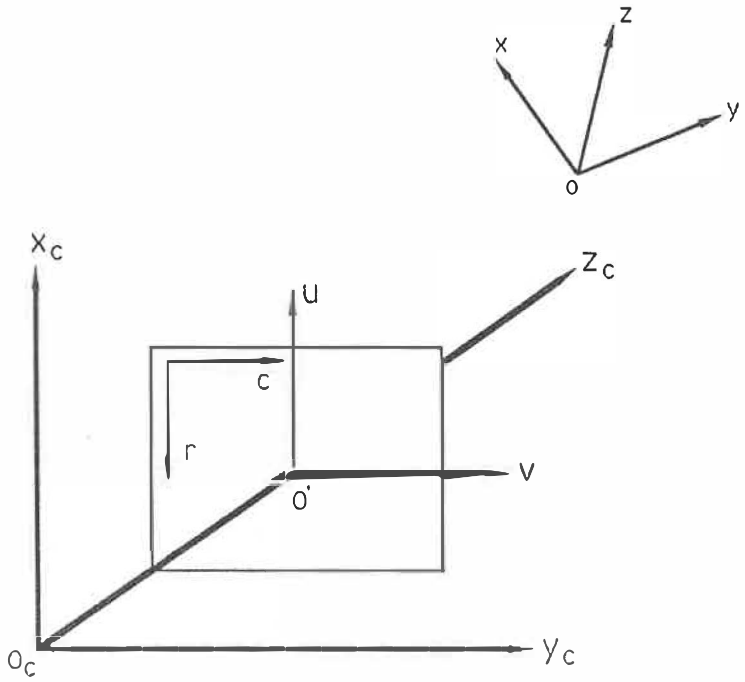


Figure 3.10: Coordinate systems for camera calibration.

These external parameters characterize the geometrical relation between a camera and a scene, or between different cameras.

Because of several types of imperfections in the design and assembly of lenses composing the optical system, expressions (3.50) do not hold true and must be replaced by expressions which explicitly take into account the positional error thus introduced:

$$\begin{aligned} u' &= u + \delta_u(u,v) \qquad (3.53)\\ v' &= v + \delta_v(u,v) \end{aligned}$$

where u and v are the non-observable, distortion-free image coordinates and u' and v' are the corresponding coordinates with distortion. As indicated by (3.53), the amount of positional error along each coordinate usually depends upon the point position. In order to compensate for the distortion, we need to analyze the various sources of distortion and model their effects in the image plane. Three types of distortion are considered in the calibration process. The first one is caused by imperfections of the lens shape, and manifests itself by radial positional errors only. The second and the third types of distortion are generally caused by improper lens and camera assembly and generate both radial and tangential errors in image point positions, as shown in Figure 3.11.

Taking into account the radial distortion, the decentering distortion and the thin prism distortion along the u and v axes, we obtain the following total distortion model when assuming that terms of order higher than 3 are negligible:

$$\begin{aligned} \delta_u(u,v) &= s_1(u^2+v^2) + 3p_1u^2 + p_1v^2 + 2p_2uv + k_1u(u^2+v^2), \qquad (3.54)\\ \delta_v(u,v) &= s_2(u^2+v^2) + 2p_1uv + p_2u^2 + 3p_2v^2 + k_1v(u^2+v^2). \end{aligned}$$

Letting $g_1 = s_1 + p_1$, $g_2 = s_2 + p_2$, $g_3 = 2p_1$, $g_4 = 2p_2$, expressions (3.54) become:

$$\begin{aligned} \delta_u(u,v) &= (g_1+g_3)u^2 + g_4uv + g_1v^2 + k_1u(u^2+v^2), \qquad (3.55)\\ \delta_v(u,v) &= g_2u^2 + g_3uv + (g_2+g_4)v^2 + k_1v(u^2+v^2). \end{aligned}$$

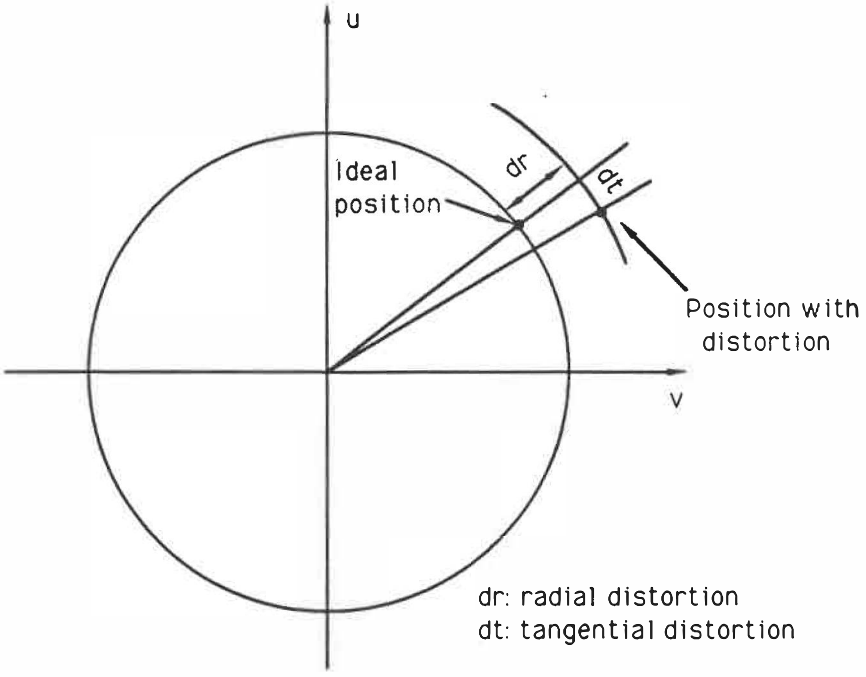


Figure 3.11: Radial and tangential distortion

Then, the relationship between the distortion-free image point (u, v) and its corresponding pixel location is given by:

$$\begin{aligned} u + \delta_u(u, v) &= (r - r_0)/s_u \\ v + \delta_v(u, v) &= (c - c_0)/s_v. \end{aligned} \tag{3.56}$$

Introducing the new variables:

$$\begin{aligned} \hat{u} &= (r - r_0)/f_u \\ \hat{v} &= (c - c_0)/f_v \end{aligned} \tag{3.57}$$

equation (3.56) becomes

$$\begin{aligned} \frac{u}{f} &= \hat{u} - \frac{\delta_u(u, v)}{f} \\ \frac{v}{f} &= \hat{v} - \frac{\delta_v(u, v)}{f}. \end{aligned} \tag{3.58}$$

Because the exact u, v cannot be obtained from actual noise-contaminated observations, the arguments of the modeled distortion are replaced by $\hat{u}$, $\hat{v}$, which leads to

$$\begin{aligned} \frac{u}{f} &= \hat{u} - \delta'_u(\hat{u}, \hat{v}) \\ \frac{v}{f} &= \hat{v} - \delta'_v(\hat{u}, \hat{v}). \end{aligned} \tag{3.59}$$

This replacement is reasonable because (a) the distortion at the exact image plane projection is approximately equal to that in the actual projection, and (b) the actual distortion coefficients in δ'_u and δ'_v will be estimated based on $\hat{u}$ and $\hat{v}$. So, the actual model fitting will be better than what is stated in (a).

Finally, the complete camera model with the considered three types of geometrical distortion is:

$$\frac{r_{1,1}x + r_{1,2}y + r_{1,3}z + t_1}{r_{3,1}x + r_{3,2}y + r_{3,3}z + t_3} = \hat{u} + (g_1 + g_3)\hat{u}^2 + g_4\hat{u}\hat{v} + g_1\hat{v}^2 + k_1\hat{u}(\hat{u}^2 + \hat{v}^2), \tag{3.60}$$

$$\frac{r_{2,1}x + r_{2,2}y + r_{2,3}z + t_2}{r_{3,1}x + r_{3,2}y + r_{3,3}z + t_3} = \hat{v} + g_2\hat{u}^2 + g_3\hat{u}\hat{v} + (g_2 + g_4)\hat{v}^2 + k_1\hat{v}(\hat{u}^2 + \hat{v}^2). \qquad (3.61)$$

It is clear that these expressions are linear with respect to the distortion coefficients k_1, g_1, g_2, g_3, g_4. The calibration problem can now be stated in the following terms:

Given a sufficient number of visible control points (x_i, y_i, z_i) and their corresponding pixel locations (r_i', c_i'), estimate in some optimal sense the set of external and internal non-distortion parameters:

$$\mathbf{m} = (r_0, c_0, f_u, f_v, \mathbf{T}, \alpha, \beta, \gamma)^\top$$

(where α, β and γ are three independent parameters of the rotation matrix R) and the set of distortion parameters:

$$\mathbf{d} = (k_1, g_1, g_2, g_3, g_4)^\top.$$

In the parameter estimation process, we first use the projections of the control points around the center of the images, where $\mathbf{d}$ is approximately zero, to compute $\mathbf{m}$ in closed-form. The projections of all the control points in the images are then used to iteratively improve $\mathbf{m}$ and $\mathbf{d}$, where $\mathbf{d}$ can be found analytically when $\mathbf{m}$ is fixed. The flowchart of the calibration procedure is illustrated in Figure 3.12.

Once the calibration is done for the camera, the estimated parameters $\mathbf{m}$ and $\mathbf{d}$ can be used to compensate for the distortion and determine the 3-D back-projection line of each sensed point. First, the measured r and c values of the sensed point give $\hat{u}$ and $\hat{v}$ according to (3.57). Then, the values of $\hat{u}, \hat{v}$ are used to evaluate the right-hand sides of the two equations in (3.60) and (3.61), whose values correspond to the distortion-corrected projection of the point in the normalized image plane $(\dot{u}, \dot{v})$. Finally, the two equations in(3.60) and (3.61) determine the back-projection line of the sensed point in the world coordinate system.

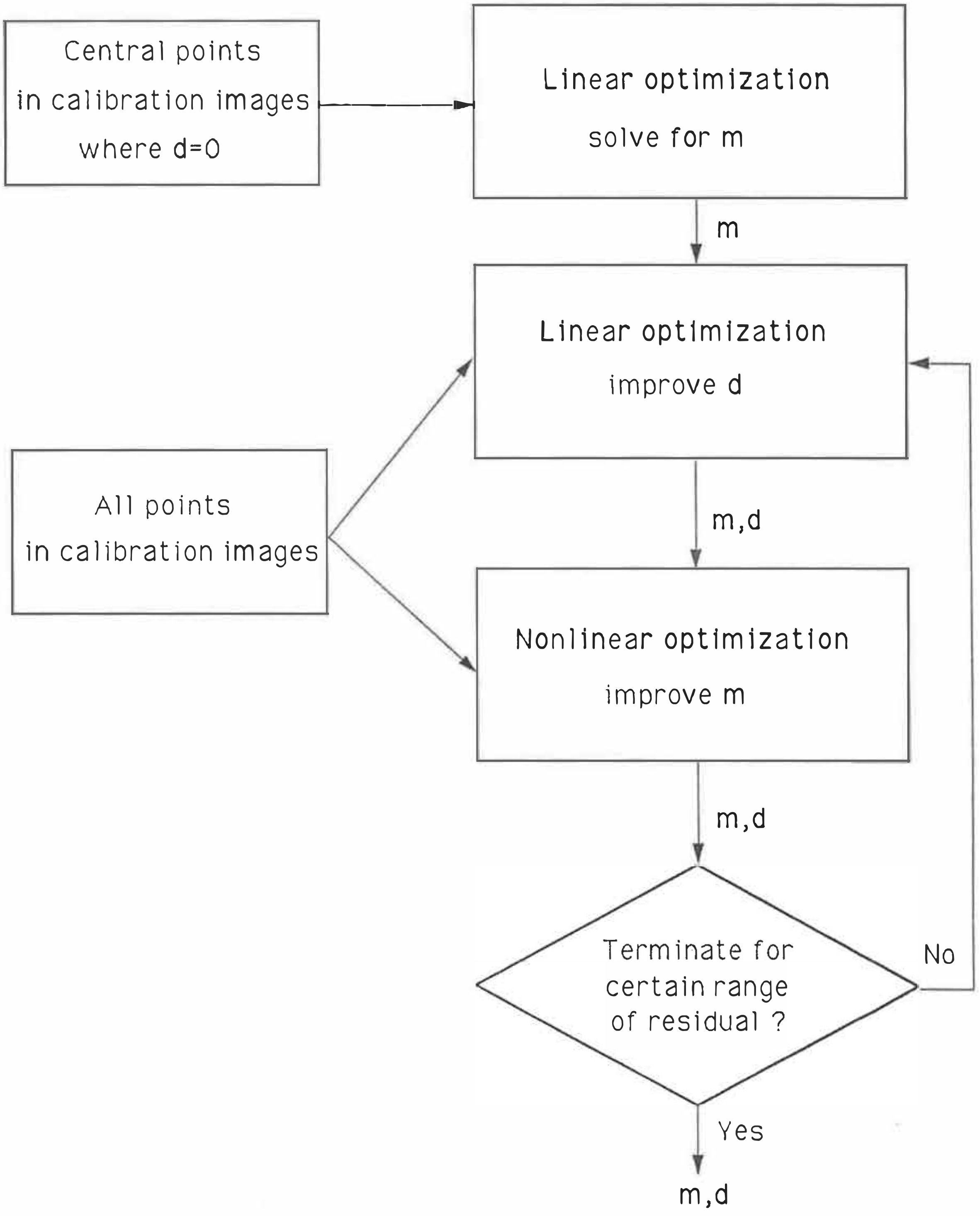


Figure 3.12: The flowchart of the calibration procedure.

Experiments with a monocular image sequence

A TM-840 PULNiX camera of $f = 8.5mm$ wide-angle lens was used in the experiments. Unlike some other high resolution cameras, the TM-840 imager cells are nearly square ($11.5\mu m$ (h) × $13.5\mu m$ (v)) instead of the more common long vertical cells. Therefore the vertical resolution and physical pixel positioning is best in the EIA (NTSC) TV format. In order to compute the internal parameters (focal length and principal point position) and to compensate for lens distortion, this camera was calibrated together with another camera where the two cameras form a stereo setup, using the method briefly described before (see [66] for detail). The calibrated data is listed in Table 3.1, in which NSCE (normalized stereo calibration error) parameter measures the mean of the ratio of the lateral triangulation error to the lateral standard deviation of the pixel digitization noise at any estimated depth. Therefore, NSCE$\approx$1 implies a good calibration in which the residual distortion is negligible compared with image digitization noise at that depth.

The camera was mounted on the tip of a robotic manipulator as shown in Figure 3.13. Since the laboratory where our six-joint robot manipulator was installed is quite small (about $3m \times 3m$ only), it is hard to stretch the manipulator forward sufficiently to create long image sequences. Instead, we let the manipulator rotate laterally around a vertical revolute joint. After each image was grabbed, the manipulator was controlled to rotate by 2.25° around the vertical revolute joint. Viewed from the camera coordinate system, the translation is almost horizontal, but the exact translation is unknown. The interframe rotation angle 2.25° is the only ground truth available in our experiments.

The image sequence obtained by the monocular camera, as shown in Figure 3.14, consists of 20 images. The scene in the first image is totally different from that in the last. The depth of the scene is about 0.6 to 2 meters.

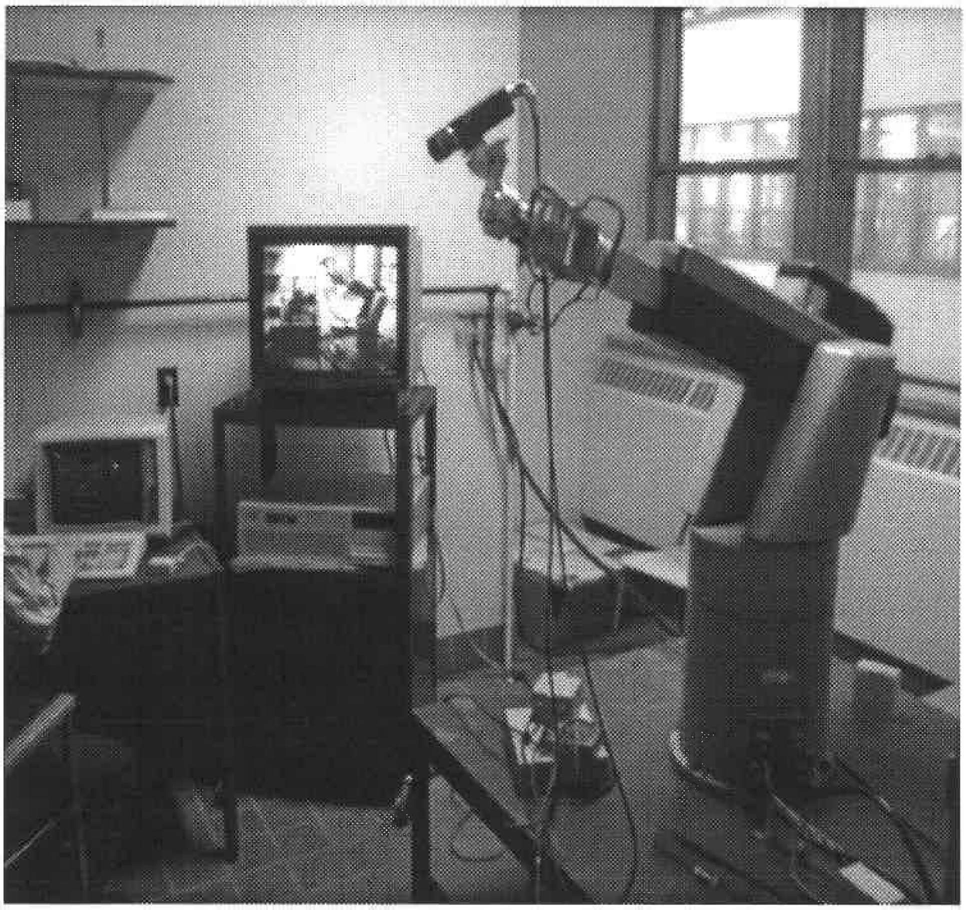
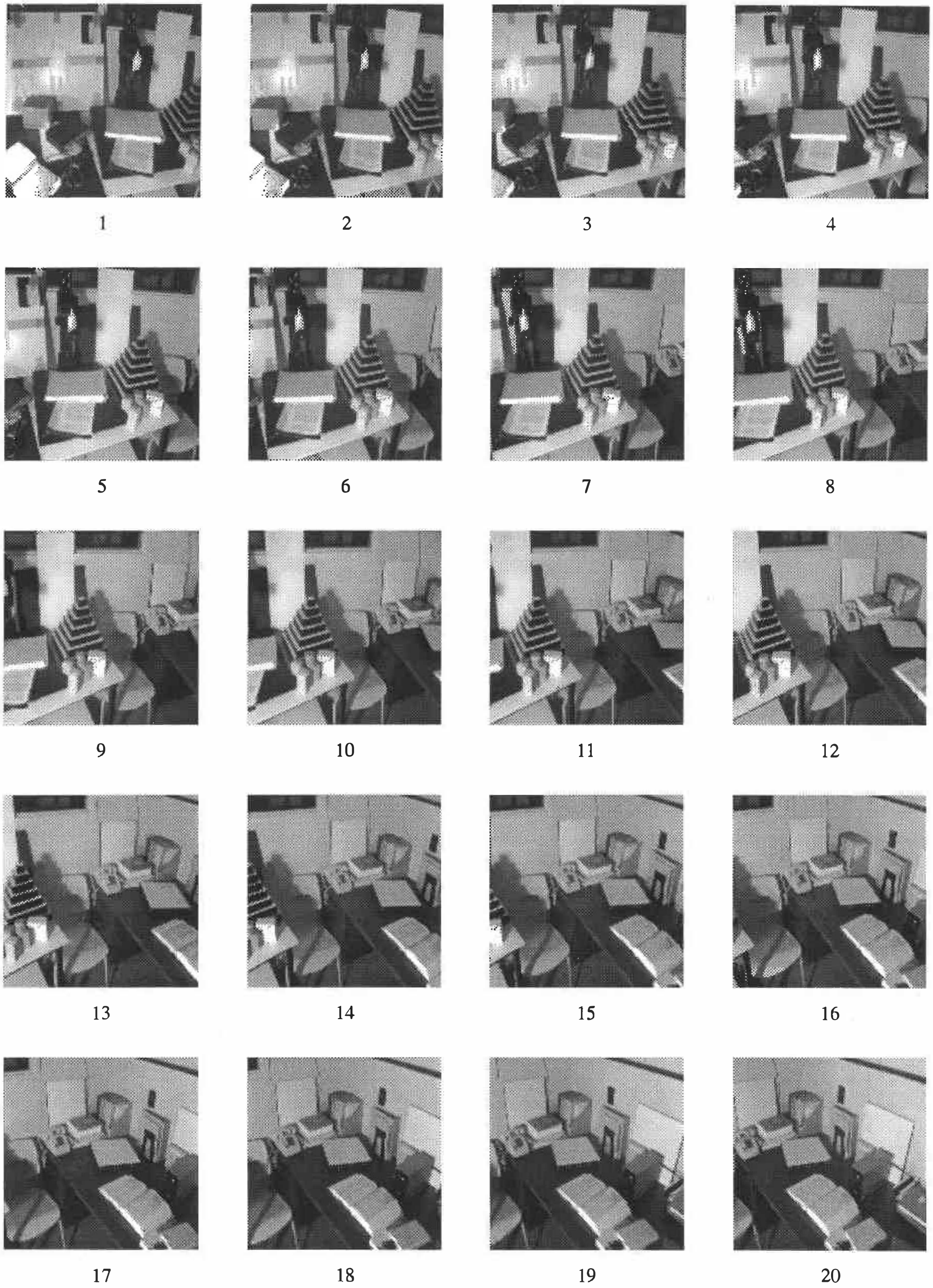


Figure 3.13: The calibrated camera was mounted on the manipulator to grab image sequences.



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Figure 3.14: The image sequence of a real scene, used in experiments.

Table 3.1: Calibration data for the f=8.5mm lens camera.

Focal length:	f_u	-639.10026
	f_v	527.09577
Center coordinate:	r_0	251.07143
	c_0	260.01047
distortion parameter:	k_1	0.17645
	g_1	-0.00390
	g_2	0.00093
	g_3	0.01522
	g_4	0.00373
test parameter	NSCE	1.30032

For any pair of consecutive images of the monocular image sequence, matching was automatically done using the method presented in [59]. This matching method uses multiple attributes associated with a pixel to yield a generally overdetermined system of constraints, taking into account possible structural discontinuities and occlusions. In the algorithm, intensity, edgeness, and cornerness attributes are used in conjunction with the constraints arising from intraregional smoothness, field continuity and discontinuity, and occlusions to compute dense displacement fields. A multiresolution structure is employed to deal with large disparities. However, if we track a feature point successively among many consecutive images with the disparity values provided by the algorithm [59], we observe that after 4 to 5 frames the accumulated disparity error is as much as 4 to 7 pixels. This makes the motion and structure estimation erroneous. Since the disparity values obtained between any two consecutive images by the matching algorithm [59] were within a range of 2 pixels around the correct values, we used a post-refining process of intensity-based normalized cross-correlation to reduce the amount of mismatching.

This normalized cross-correlation was done between a 9×9 intensity window centered at the feature location in the earlier image, and a 9×9 window in the latter image, scanning a 5×5 neighborhood centered at the preliminary matching location. This refining process related a feature point to its earliest past so that the error in interframe matching would not quickly accumulate through multiple views. The feature points consisted of manually selected corners in the image. This selection could be automatically done by a simple corner detector such as in [59]. The matching for the first image pair is shown as vector lines in Figure 3.15.

The points are classified into two categories: old and new. The old points are visible both in the current image and the previous one. The new points are visible in the current image but not in the previous one. A point is no longer considered as old if its neighborhood changed drastically due to motion. Only the old points (about 70 for each image pair) were included in the iterative optimization. The structure of the new points was estimated after motion.

The first two interframe motions and the last two interframe motions are listed in Table 3.2, the complete interframe motion estimates are shown in Appendix G. The estimated angle of interframe rotation is about 2.39° to 2.61° as listed in Table 3.3. As can be seen, they are quite accurate compared to the ground truth of 2.25°. During the interframe computation, the nonlinear optimization converges and the magnitude of the average difference between image points projected from the structure estimates and the observed image points is less than half a pixel.

The accuracy of the estimated structure was tested using the lengths of several lines of the 3-D structure. Any line with end points x_i and x_i' has a true length $l_i = \|x_i - x_i'\|$. From the 3-D points determined up to a scale factor α, $x_i = \alpha \tilde{x}_i$, $x_i' = \alpha \tilde{x}_i'$, its scaled length is $\tilde{l}_i = \|\tilde{x}_i - \tilde{x}_i'\|$. In a noise free situation $l_i = \alpha \tilde{l}_i$, while in

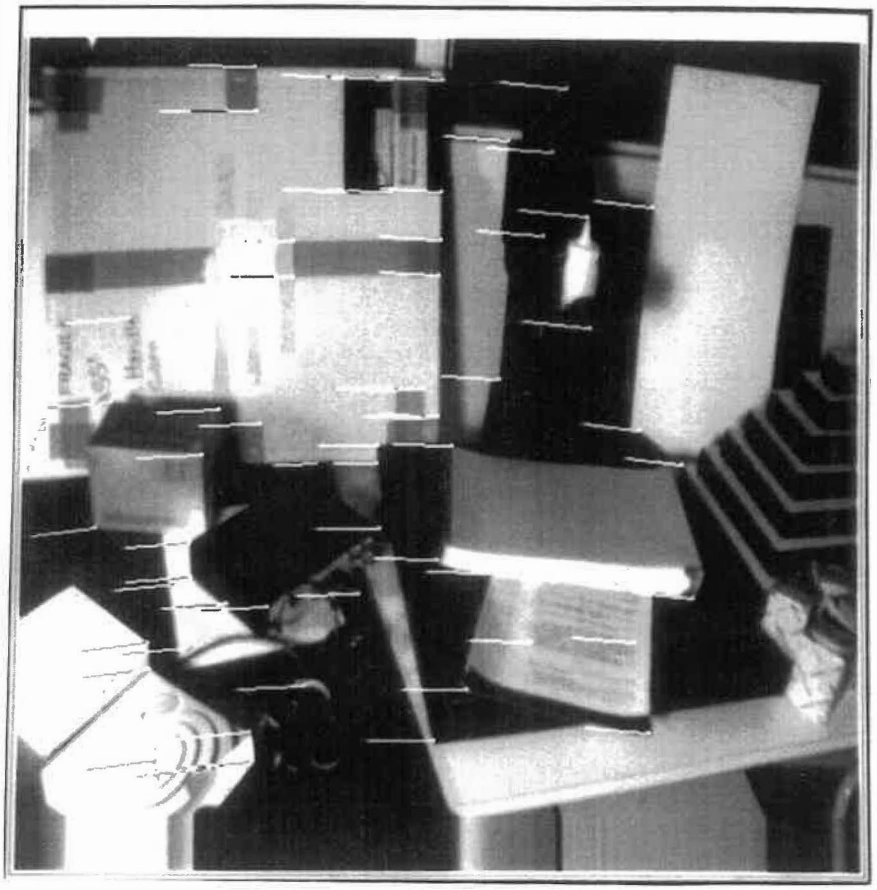


Figure 3.15: Matching for the first image pair shown as vector lines.

the presence of noise, $\tilde{l}_i$ has errors. We can determine the scale factor, α, such that the following root mean-squared error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(l_i - \alpha\tilde{l}_i)^2} \qquad (3.62)$$

is minimized. So, RMSE indicates the best scale fit between the estimated structure and the true structure.

However, it should be noted here that this type of error measure is not as good as the positional error measurement we used in simulations. Identical amounts of error in the two end points may be cancelled out in the length measurement. We used length measurement since the global positions of points are not available here.

We measured 65 lines in the entire scene (see Figure 3.16), and recorded the visible lines in each view. The value of RMSE was computed at each frame using these visible lines. These RMSE values indicate that the structure estimation has an accuracy of about 6 to 32 mm, with respect to the world reference. We have observed that the most part of the error concerns the depth components and is due to the relatively small interframe motion. The integration of multiple views may reduce the error, as shown in the table. But the error in motion parameters is accumulated, which makes the improvement on the estimated structure saturated soon.

Note that a wide angle lens was used in this experiment. A tele lens will result in unreliable estimates as discussed in [50]. A forward (or backward) camera motion may lead to more accurate motion parameter estimations [50], but structure estimates will be very bad for points near the focus of expansion of the images.

3.7 Summary

We have thus far investigated the problem of optimal motion and structure estimation for long monocular image sequences. It was shown that any scale factor of two

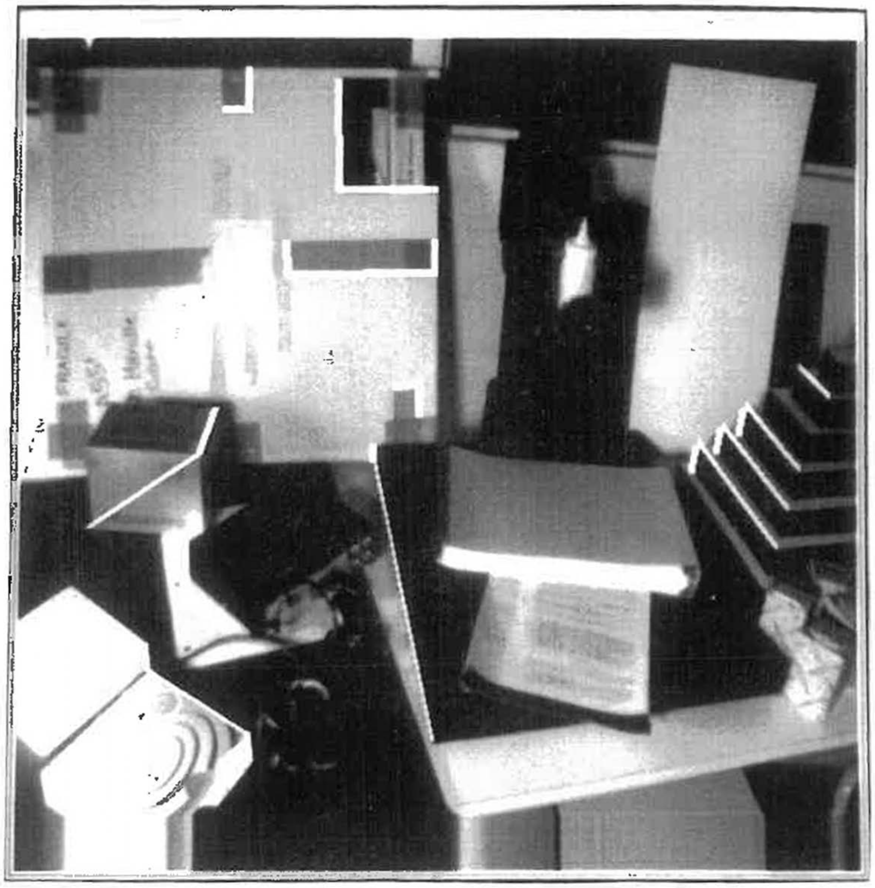


Figure 3.16: The samples of the lines have been measured in the scene.

Table 3.2: Motion estimations resulting from the linear algorithm and the nonlinear optimization.

motion parameters		linear algorithm	nonlinear optimization
motion	$\mathbf{M}_{1,0}$		
translation	t_x	-0.004081	-0.000615
(scaled)	t_y	-0.140111	-0.192057
	t_z	-0.116833	-0.044210
	length	0.182476	0.197081
rotation axis	N_x	0.927024	0.896551
	N_y	0.127492	0.049395
	N_z	-0.352664	-0.440179
rotation angle	$\theta(^\circ)$	2.955706	2.514551
motion	$\mathbf{M}_{2,1}$		
translation	t_x	-0.012594	-0.005608
(scaled)	t_y	-0.040259	-0.190828
	t_z	-0.116480	-0.042419
	length	0.123883	0.195567
rotation axis	N_x	0.987225	0.894357
	N_y	0.088764	0.057969
	N_z	-0.132316	-0.443582
rotation angle	$\theta(^\circ)$	3.777868	2.510903
motion	$\mathbf{M}_{18,17}$		
translation	t_x	0.019225	-0.010592
(scaled)	t_y	-0.073187	-0.200072
	t_z	-0.136523	-0.042506
	length	0.156091	0.204812
rotation axis	N_x	0.959765	0.890507
	N_y	0.069962	0.060714
	N_z	-0.271949	-0.450900
rotation angle	$\theta(^\circ)$	3.484531	2.400253
motion	$\mathbf{M}_{19,18}$		
translation	t_x	-0.070523	0.000453
(scaled)	t_y	-0.202018	-0.206084
	t_z	0.438972	-0.041304
	length	0.488345	0.210183
rotation axis	N_x	0.950502	0.890109
	N_y	-0.062418	0.058372
	N_z	-0.304383	-0.451993
rotation angle	$\theta(^\circ)$	3.422166	2.438417

corresponding consecutive images can be expressed as a proportional function of the scale factor of the first image pair. A recursive-batch nonlinear optimization approach has been presented to estimate the motion and three-dimensional structure of the scene from long monocular image sequences, allowing arbitrary interframe motion between any two consecutive images in the sequences.

The fundamental difference between our approach and some others in dealing with long image sequences is that we have fully used the relationship between motion and structure. By using this relationship, the extremely large parameter space was reduced to a 6-dimensional (the 6 independent motion variables) space. In addition, we adopted the recursive-batch method by preserving and updating the structure through time. In order to improve the numerical stability and reduce the computational cost, an initial motion solution of the nonlinear optimization was provided to the nonlinear optimization stages by means of a linear algorithm. Finally, the different uncertainties of the different components of the space points were taken into account in the optimization process. These strategies formulate the intractable optimization problem into a practical one, while still obtaining a good performance. Our method gives accurate results from the first two frames, on the contrary, the conventional *Kalman filtering* method needs a long period of up to 30-40 frames to converge.

Our objective was to validate our motion and structure analysis with thorough experiments, in order to establish the credibility of our approach. Both simulations and experiments with real image sequences were conducted for assessing the performance of the proposed approach. Our experiments with long image sequences, automatic matching and a calibrated camera provide a detailed assessment of the accuracy of the estimated motion and structure in real world situations. Despite the presence of quantization, calibration and matching errors induced through the long image sequences, the motion and structure estimated from the nonlinear optimization appear satisfactory, based on the available partial ground truth.

Table 3.3: Estimated motion and structure. k: time index; θ: rotation angle (degree); RMSE: root mean-squared error (mm).

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
θ	2.51	2.51	2.43	2.58	2.51	2.45	2.50	2.46	2.48	2.48	2.54	2.41	2.57	2.61	2.39	2.47	2.57	2.40	2.44
RMSE	32	13	21	13	17	12	10	20	17	16	23	14	31	14	11	13	6	9	27

Chapter 4

Estimating motion and 3-D structure from stereo image sequences

This chapter deals with the counterpart of the monocular problem: optimal motion and structure estimation from stereo image sequences. Starting from a newly proposed matrix-weighted closed-form algorithm, we process stereo image sequences with a recursive-batch approach. Simulation and experiment are presented to assess the performance of our approaches.

4.1 Review of related work

We consider in this chapter a stereo camera system, with a fixed but general stereo configuration, moving in a static environment. The evolution of the viewing direction and position of the stereo camera system usually induces multiple images of the same part of a scene. Our objective is to estimate the unknown motion of the stereo camera system and the 3-D structure of the scene, by fusing partially overlapping views so that a more accurate and consistent description of the visual world can be derived.

Different components of a 3-D point determined through stereo triangulation have different uncertainties: typically, the depth component of an estimated 3-D point is much less reliable than the other two lateral components. The accuracy of structure and motion estimation depends on how the varying uncertainties in the estimated 3-D points are treated, and how the nonlinear optimization is performed. Least-squares closed-form solutions for motion estimation based on estimated 3-D points from triangulation have been proposed in [13], [18], [23], [24], [34]. One of the most important advantages of the closed-form solutions is that the corresponding algorithms are fast and the solutions are guaranteed. However, since the various 3-D points and the different components of a 3-D point are treated equally in the objective function, a least-squares solution is not optimal.

Estimation based on a proper error modeling can essentially improve the accuracy of the estimates. Hallam [22] used Gaussian noise to model the errors in range data and employed *Kalman filtering* to estimate the motion of a mobile robot. Broida and Chellappa [65] and Young and Chellappa [40] modeled errors in images by independent Gaussian noise and used *Kalman filtering* to estimate motion parameters based on correspondences through a monocular image sequence. Ayache and Faugeras [14] applied similar techniques to estimate motions and fuse stereo images. Matthies and Shafer [52] studied some related issues of error modeling for stereo navigation. They model the error of a 3-D point, which is constructed through stereo triangulation, by a 3-D random vector with Gaussian distribution (called ellipsoidal model). Given a set of corresponding 3-D points $\{\mathbf{x}_i\}$ before motion and $\{\mathbf{x}'_i\}$ after motion, the interframe motion, represented by a rotation matrix R and a translation matrix $\mathbf{T}$, is determined to minimize

$$\sum_{i=1}^{n}\{R\mathbf{x}_i + \mathbf{T} - \mathbf{x}'_i\}^{\mathsf{T}} V_i \{R\mathbf{x}_i + \mathbf{T} - \mathbf{x}'_i\}, \tag{4.1}$$

where the weighting matrix V_i is the inverse of $R\Gamma_{\mathbf{x}_i}R^{-1} + \Gamma_{\mathbf{x}'_i}$, $\Gamma_{\mathbf{x}_i}$ and $\Gamma_{\mathbf{x}'_i}$ being the error covariance matrices of $\mathbf{x}_i$ and $\mathbf{x}'_i$, respectively. A closed-form solution to this problem was not found, neither in the general case, nor for a simplified case where the weighting matrix V_i is not a function of R. Their method consists of solving the problem iteratively using a least-squares solution as an initial guess. Kiang et al [26] replaced the matrix V_i in (4.1) by a scalar w_i^2. The distribution of error in the 3-D position of a point is then simplified to an uncertainty line segment along the projection line. The scalar weight w_i^2 is estimated based on some relative geometrical configurations between the two corresponding uncertainty lines. A few iterations lead to improved accuracy, compared to a least-squares solution. A simpler scalar weight, which is inversely proportional to the depth of a point, has also been used by Moravec in [31].

Although scalar-weighted closed-form solutions may yield better results than unweighted closed-form solutions, scalar weights still indiscriminately treat the uncertainties in the different components of a 3-D point. This implies that either the reliable components are under-trusted or the unreliable components are over-trusted. Furthermore, the correlation between errors in the 3-D point cannot be properly accounted for by scalar weights. A matrix-weighted method is a correct solution to these problems.

Although most of the above mentioned models were developed to represent 3-D triangulation errors induced by errors in the image planes, these models are not directly related to the image plane noise, and assume a symmetrical 3-D noise distribution. However, the distribution of errors in a 3-D point is not simple and cannot be assumed to be symmetrical. In fact, the shape of the probability distribution has a shrunken end towards the camera and a swollen end away from the camera, as shown in Chapter 2. A symmetrical ellipsoidal model or any other simplified model not only will limit the performance of the estimation but also will cause bias in the estimates. While the 3-D error distribution of a point is complex, the distribution of 2-D errors in image plane is simpler. For an iterative algorithm, modeling 3-D errors will not gain any substantial efficiency over modeling 2-D image errors. In this chapter we consider a 2-D image plane noise model which will implicitly result in a desired distribution shape in 3-D.

With a long stereo image sequence, the amount of data to be processed is drastically larger than that for two stereo image pairs. *Kalman filtering* [25], [20], [29], [55] [35], [30], [21], [39] is a method used to compute a weighted least-squares solution for a dynamic system. It is a sequential technique in the sense that the observation data are sequentially fed into the algorithm and new estimates are recursively computed from previous estimates and current observations. In contrast, a batch technique is such that all observations are processed together in a batch fashion and estimates are determined directly from all observations. A sequential technique has some desirable properties. First, old observations can be discarded once they have been used for estimation. This is a must if the whole data set is so large that it exceeds the capacity of memory (e.g., in an extended navigation). Second, the technique is efficient since a relatively small amount of computation is required for updating estimates with each observation. Third, since a new estimate is computed at every time of observation, one does not need to wait for a complete set of observations to be collected before receiving estimates. However, sequential techniques have a relatively poor performance for nonlinear problems. As discussed in [54] and in Chapter 2, the estimated system matrices (i.e., Jacobian matrices) used for earlier observations are not updated after new observations are collected. Because the system matrices of a nonlinear system depend on the estimated system parameters, the system matrices for earlier observations are evaluated with parameters values close to the initial solution, and therefore, are very inaccurate. Those inaccurate system matrices almost always cause an early divergence, which requires many image frames to graduately "pull" the state trajectory back. The linear-model based error covariance matrices, which determine the amount of parameter updating, are unable to reflect such a divergence and underestimate the errors in the current estimates [33], [19] [54]. Consequently, these error covariance matrices impede the correction of divergence and parameter updating. This explains why *Kalman filtering* requires many image frames (typically 20 to 30 frames) to converge to an acceptable solution. Such a slow convergence also implies a slow response to system changes, such as those caused by change of motion direction.

Starting from a newly proposed closed-form matrix-weighted least-squares solution to motion parameters from 3-D point correspondences [71], we employ a recursive-batch approach to deal with stereo image sequences in this chapter. This closed-form solution computes the motion parameters that minimize (4.1), in the case where the weighting matrix V_i is simplified so that it does not depend upon the unknown rotation matrix R. This method leads to a remarkable improvement over unweighted or scalar-weighted closed-form solutions. Its solution can be directly used in situations where the speed of the algorithm is critical to the intended applications. In the recursive-batch approach, the observed projection data are divided into groups of images, each data group consisting of two pairs of stereo images. Estimation is done in a sequential fashion among these groups. Within each group the estimation of the motion and the structure is completed in a batch fashion for all the corresponding projected points. From the view-point of *Kalman filtering*, the states (the motion parameter solutions) are directly observed (provided by a batch solution, either non-iterative or iterative). The recursive *Kalman filtering* equations are used to properly weigh the uncertainties in observations (the 3-D structure computed from triangulation) as well as states (the motion parameter solutions) and update the uncertainty in the new states. In other words, by properly defining states and observations, we convert the nonlinear *Kalman filtering* problem to a linear *Kalman filtering* problem, which is, with regard to the formulation to which the *Kalman filter* is applied, equivalent to a batch solution! In this way, advantages of sequential processing are kept and the performance of the algorithm is drastically improved from direct nonlinear iterated *Kalman filtering*.

Since the dimension of the parameter space for each data group consisting of two pairs of stereo images, as described before, is still very large (including the 3-D coordinates of all points as well as the motion parameters), a direct search in such a large space is computationally prohibitive. Instead of using direct batch processing, we explore the nature of the problem so that the search space is reduced to motion parameters only. In order to reduce the number of iterations and prevent divergence, the closed-form matrix-weighted least-squares solution is used as an initial solution for batch optimization within each data group. It should be mentioned that our formulation is independent of the way the batch solution are computed. While iterative optimal solutions lead to more accurate results, the non-iterative matrix-weighted solution often suffices, especially in situations where the speed of the algorithm is critical.

We also investigate the representation of motion of the stereo camera system and 3-D structure of the scene in different coordinate systems: the local and global coordinate systems. In self-guided navigation, the use of the local reference system is preferable. However, if the navigation has to refer to a map or to construct a global 3-D map of the sensed world, a representation in the global coordinate system is more appropriate.

To study the performance of proposed approaches, simulation and careful experiments with a real stereo image sequence have been carried out.

4.2 Stereo modeling

Most of the previous published works use a simplified stereo camera system in which the two image planes are coplanar and the corresponding image coordinate vectors (corresponding to row or column directions) in the two image planes are parallel. In reality, however, it is difficult to align two cameras physically so that the internal optical geometry of the cameras satisfies the above requirements. As a matter of fact, such an alignment is often not desirable. For example, the common field of view of the two cameras covers a larger scene if the two cameras gaze at the scene of interest. In this research, we use a general stereo setup, since it agrees more closely with situations of real applications.

4.2.1 Optimal determination of a 3-D point from a pair of noisy stereo projections

For convenience, we choose the coordinate system centered at the left camera as the local (camera-centered) reference for the stereo system. The orientation and position of the right camera, with respect to the left camera, is specified by a rotation matrix M, and a translation vector $\mathbf{B}$. A vector $\mathrm{x}_r = (x_r\,, y_r\,, z_r)^\mathsf{T}$ represented in the right-camera-centered system is related to $\mathrm{x}_l = (x_l\,, y_l\,, z_l)$ in the local system by

$$\mathrm{x}_l = M\mathrm{x}_r + \mathbf{B}, \tag{4.2}$$

where M and $\mathbf{B}$ are usually determined through camera calibration. Since we can always derive a normalized pin-hole camera model from a real model in which the focal length is equal to 1 and the image plane is at $z = 1$, for a 3-D point $\mathrm{x} = (x\,, y\,, z)^\mathsf{T}$, its corresponding image vector $\mathbf{X}$ satisfies:

$$\mathbf{X} = \mathrm{x}/z.$$

The first two components $\mathbf{u} = (x/z\,, y/z)^\mathsf{T}$ in $\mathbf{X}$ are the image coordinates of the point.

Thus, the depths z_l and z_r of the point, in the local and the right-camera-centered systems, respectively, can be determined from (4.2):

$$\mathbf{X}_l z_l = M\mathbf{X}_r z_r + \mathbf{B}. \tag{4.3}$$

This is a vector equation with three scalar equations and two unknowns. In the absence of noise, the three scalar equations are always consistent since the true depths are the solutions. Equation (4.3) expresses the epipolar constraint: $\mathbf{X}_l$, $M\mathbf{X}_r$ and $\mathbf{B}$ are coplanar (linearly dependent). Geometrically, the constraint means that two projection lines intersect in space. In the presence of noise, the epipolar constraint may be violated. We need to determine the optimal 3-D position of a point from the noise contaminated observations so as to further estimate motion parameters. Let the noise-contaminated observations in the left and right images be $\tilde{\mathbf{u}}_l$ and $\tilde{\mathbf{u}}_r$, respectively. We have a 2-D image plane noise model as:

$$\tilde{\mathbf{u}}_r = \mathbf{u}_r + \delta_{\mathbf{u}_r}, \qquad \tilde{\mathbf{u}}_l = \mathbf{u}_l + \delta_{\mathbf{u}_l}, \tag{4.4}$$

where $\delta_{\mathbf{u}_r}$ and $\delta_{\mathbf{u}_l}$ are additive noise vectors. They account for image quantization noise, edge detecting error, feature matching error etc..

We assume that the correlation between image errors is negligible. We also assume the same error variance in its different components (This is not true for CCD arrays with rectangular sensing cells, but an extension for this case is straight-forward, according to the above discussion). Let $\mathbf{u}_l(\hat{\mathbf{x}})$ and $\mathbf{u}_r(\hat{\mathbf{x}})$ represent the projections of the estimated 3-D point $\hat{\mathbf{x}}$ in the left and right images, respectively. According to the principle of minimum variance estimation, the optimal 3-D point $\hat{\mathbf{x}}$ should minimize

$$\|\tilde{\mathbf{u}}_l - \mathbf{u}_l(\hat{\mathbf{x}})\|^2 + \|\tilde{\mathbf{u}}_r - \mathbf{u}_r(\hat{\mathbf{x}})\|^2. \tag{4.5}$$

This is a nonlinear minimization problem. The best linear estimation, in the least-squares sense, is obtained by solving for z_l and z_r which minimize

$$\|M\tilde{\mathbf{X}}_r z_r + \mathbf{B} - \tilde{\mathbf{X}}_l z_l\|. \tag{4.6}$$

The estimated $\hat{\mathbf{x}}$ is then determined by

$$\hat{\mathbf{x}} = \{M\tilde{\mathbf{X}}_r z_r + \mathbf{B} + \tilde{\mathbf{X}}_l z_l\}/2. \tag{4.7}$$

The geometrical interpretation of this approximate solution is the following: Owing to noise, the two projection lines of the point from left and right cameras, respectively, do not intersect in space. The solution in (4.7) is the midpoint of the shortest line segment that connects these two projection lines, as shown in Figure 4.1. From this approximate solution, a few iterations can be performed to minimize (4.5). To conclude, from a pair of stereo projections $\tilde{\mathbf{u}}_l$ and $\tilde{\mathbf{u}}_r$, we have constructed a function $\mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)$ that gives the estimated 3-D position of a point:

$$\hat{\mathbf{x}} = \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r). \tag{4.8}$$

The errors in the estimated position $\hat{\mathbf{x}}$ need also to be determined. From (4.8), we have

$$\hat{\mathbf{x}} - \mathbf{x} \cong \frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_l}(\tilde{\mathbf{u}}_l - \mathbf{u}_l) + \frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_r}(\tilde{\mathbf{u}}_r - \mathbf{u}_r), \tag{4.9}$$

or,

$$\delta_{\mathbf{x}} \cong \frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_l}\delta_{\mathbf{u}_l} + \frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_r}\delta_{\mathbf{u}_r}. \tag{4.10}$$

The error covariance matrix of the estimated 3-D point $\hat{\mathbf{x}}$ is thus,

$$\Gamma_{\mathbf{x}} \cong \frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_l}\Gamma_{\mathbf{u}_l}\frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_l}^{\mathsf{T}} + \frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_r}\Gamma_{\mathbf{u}_r}\frac{\partial \mathbf{c}(\tilde{\mathbf{u}}_l\,, \tilde{\mathbf{u}}_r)}{\partial \mathbf{u}_r}^{\mathsf{T}} \tag{4.11}$$

assuming $\delta_{\mathbf{u}_l}$ and $\delta_{\mathbf{u}_r}$ are uncorrelated.

4.3 A recursive-batch approach to process stereo image sequences

Suppose that the stereo camera system is moving in a static surrounding and a sequence of stereo images is taken. We first discuss the issue in the estimation of interframe motion, which serves as a backbone in the analysis of long sequences. Two methods are discussed, a closed-form matrix-weighted least-squares solution and an iterative optimal solution. Using each of these two methods for two-view analysis, we

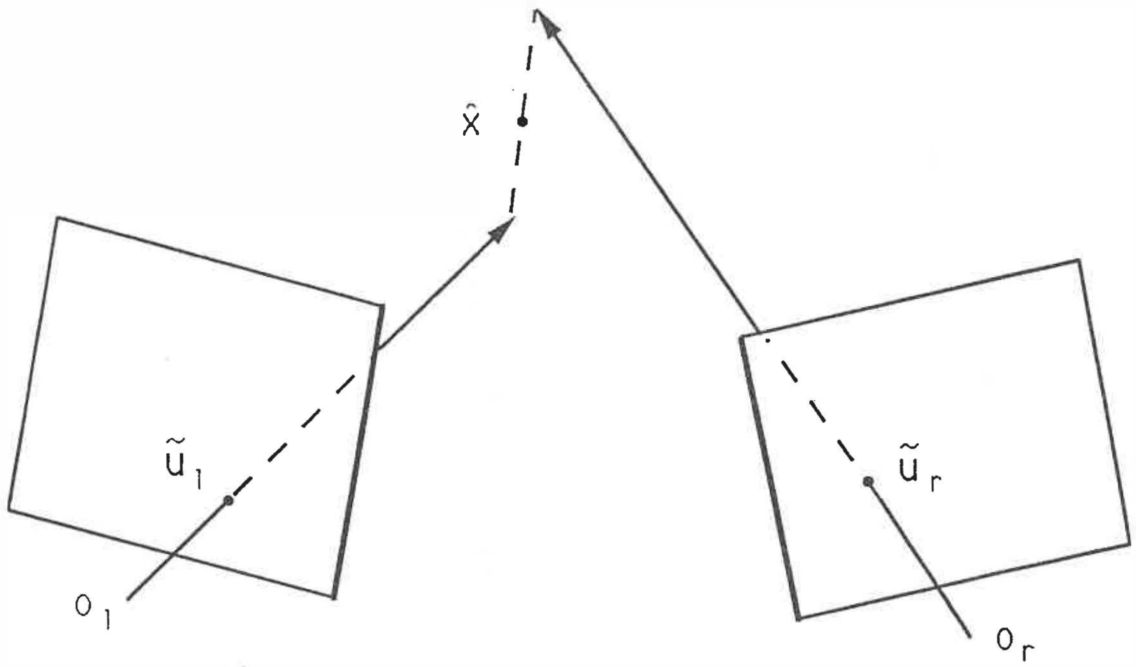


Figure 4.1: The estimated 3-D point in (4.7) is the midpoint of the shortest line segment that connects the two projection lines.

proceed to deal with long image sequences. A recursive-batch approach is adopted to fuse multiple stereo views in order to achieve higher performance without suffering from excessive computational cost.

4.3.1 Closed-form solution from two stereo pairs

We consider now how to determine the motion parameters of the scene in the local system from two consecutive pairs of stereo images without iteration. In this system, let a point $\mathbf{x}_{0,i}$ at time t_0 be moved to $\mathbf{x}_{1,i}$ at time t_1. They are related by

$$\mathbf{x}_{1,i} = R_{1,0}\mathbf{x}_{0,i} + \mathbf{T}_{1,0} \tag{4.12}$$

where $R_{1,0}$ is a rotation matrix and $\mathbf{T}_{1,0}$ is a translation vector describing the inter-frame motion. To simplify the following derivation, R and $\mathbf{T}$ are used instead of $R_{1,0}$ and $\mathbf{T}_{1,0}$. The objective here is to determine R and $\mathbf{T}$ from a sequence of estimated 3-D point correspondences: $\{(\hat{\mathbf{x}}_{0,i}\ , \hat{\mathbf{x}}_{1,i})\}$.

1) Unweighted and scalar-weighted closed-form solutions

In the presence of noise, it is necessary to take into account the different uncertainties in the points that are constructed by stereoscopic triangulation. Using the estimated 3-D positions $\hat{\mathbf{x}}_{1,i} = \mathbf{x}_{1,i} + \delta_{\mathbf{x}_{1,i}}$ and $\hat{\mathbf{x}}_{0,i} = \mathbf{x}_{0,i} + \delta_{\mathbf{x}_{0,i}}$, equation (4.12) gives

$$\hat{\mathbf{x}}_{1,i} = R\hat{\mathbf{x}}_{0,i} + \mathbf{T} + \delta_i, \tag{4.13}$$

where

$$\delta_i = \delta_{\mathbf{x}_{1,i}} - R\delta_{\mathbf{x}_{0,i}}. \tag{4.14}$$

Suppose that the errors in the observed points are uncorrelated between instants t_0 and t_1. It follows from (4.14) that the residual vector δ_i has a error covariance matrix

$$\Gamma_i = E\delta_i\delta_i^{\mathrm{T}} = \Gamma_{\mathbf{x}_{1,i}} + R\Gamma_{\mathbf{x}_{0,i}}R^{-1}. \tag{4.15}$$

We now suppose that the observation vector consists of a sequence of 3-D points at two time instants and that the errors in these observations are uncorrelated between the different points and the instants. Based on the principle of weighted least squares, the motion parameters should thus be determined by minimizing

$$\sum_{i=1}^{n}\{R\hat{\mathbf{x}}_{0,i}+\mathbf{T}-\hat{\mathbf{x}}_{1,i}\}^{\top}\Gamma_i^{-1}\{R\hat{\mathbf{x}}_{0,i}+\mathbf{T}-\hat{\mathbf{x}}_{1,i}\}, \tag{4.16}$$

where n is the number of 3-D points. Letting $\mathbf{a}$ denote a three dimensional vector consisting of the three independent parameters of the rotation matrix R, the expression (4.16) is a nonlinear function of a six-dimensional parameter vector

$$\mathbf{m}=(\mathbf{a}^{\top},\ \mathbf{T}^{\top})^{\top}. \tag{4.17}$$

The objective is thus to determine $\mathbf{m}$ which minimizes (4.16).

A close-form solution to $\mathbf{m}$ that minimizes a special case of (4.16), namely:

$$\sum_{i=1}^{n}||R\hat{\mathbf{x}}_{0,i}+\mathbf{T}-\hat{\mathbf{x}}_{1,i}||^2, \tag{4.18}$$

has already been described in the literature [34], [18], [23] [13], [24]. This objective function leads to an unweighted least-squares solution in that the weighting matrix Γ_i^{-1} is replaced by an identity matrix. The Γ_i^{-1} has been replaced by a scalar w_i^2 in [26] to minimize

$$\sum_{i=1}^{n}w_i^2||R\hat{\mathbf{x}}_{0,i}+\mathbf{T}-\hat{\mathbf{x}}_{1,i}||^2, \tag{4.19}$$

which becomes to a scalar-weighted least-squares solution. However, since the depth component of a point is significantly less reliable than its lateral components, and the errors in these three components have considerable correlations (the uncertainty volume is elongated and tilted), an unweighed or even a scalar-weighted objective function cannot properly treat these uncertainties.

2) A Closed-Form Solution with a Matrix-Weighted Objective Function

We now introduce the closed-form solution for the matrix-weighted objective function (4.16) presented in [71]. First, we need to simplify the weighting matrix in (4.15) so that it does not depend on the rotation matrix being computed. With a small rotation, the rotation matrix is roughly equal to an identity matrix, $R \approx I$, and the weighting matrix in (4.15) does not depend very much on R. So, the weighting matrix can be approximated by

$$\Gamma_i = \Gamma_{\mathbf{x}_{1,i}} + \Gamma_{\mathbf{x}_{0,i}}. \tag{4.20}$$

If the rotation is so large that the simplification in (4.20) is not allowed, we can use the weighting matrix in (4.15) but R in (4.15) is replaced by a fixed rotation matrix which is estimated by a closed-form scalar-weighted least-squares solution to be discussed soon.

We first state the matrix-weighted centroid-coincidence theorem (MWCC theorem for short), whose proof is included in Appendix C. Its unweighted version was originally proved in [23].

MWCC Theorem. If R^* and T^* minimize (4.16) with the weighting matrix Γ_i^{-1} not depending on either R or $\mathbf{T}$, then the matrix-weighted centroids of $\{\hat{\mathbf{x}}_{1,i}\}$ and $\{R^*\hat{\mathbf{x}}_{0,i} + \mathbf{T}^*\}$ must coincide:

$$\sum_{i=1}^{n} \Gamma_i^{-1}\{R^*\hat{\mathbf{x}}_{0,i} + \mathbf{T}^*\} = \sum_{i=1}^{n} \Gamma_i^{-1}\hat{\mathbf{x}}_{1,i}. \tag{4.21}$$

Before looking for a closed-form solution to the matrix-weighted problem, we now consider the scalar-weighted least-squares solution that minimizes (4.19). Replacing Γ_i^{-1} by matrix $(1/n)I$, the MWCC theorem gives the unweighted centroids coincidence theorem originally presented in [23]:

$$\frac{1}{n}\sum_{i=1}^{n}\{R^*\hat{\mathbf{x}}_{0,i} + \mathbf{T}^*\} = \frac{1}{n}\sum_{i=1}^{n}\hat{\mathbf{x}}_{1,i}. \tag{4.22}$$

Replacing Γ_i^{-1} by a scalar matrix $w_i^2 I$, the MWCC theorem takes the form

$$\sum_{i=1}^{n} w_i^2 \{R^* \hat{\mathbf{x}}_{0,i} + \mathbf{T}^*\} = \sum_{i=1}^{n} w_i^2 \hat{\mathbf{x}}_{1,i} \tag{4.23}$$

which can be rewritten as

$$R^* \sum_{i=1}^{n} w_i^2 \hat{\mathbf{x}}_{\bullet,i} + \sum_{i=1}^{n} w_i^2 \mathbf{T}^* = \sum_{i=1}^{n} w_i^2 \hat{\mathbf{x}}_{1,i}. \tag{4.24}$$

Letting

$$\bar{\mathbf{x}}_0 = \sum_{i=1}^{n} w_i^2 \hat{\mathbf{x}}_{0,i} / \sum_{i=1}^{n} w_i^2, \qquad \bar{\mathbf{x}}_1 = \sum_{i=1}^{n} w_i^2 \hat{\mathbf{x}}_{1,i} / \sum_{i=1}^{n} w_i^2,$$

it follows from (4.24) that

$$\mathbf{T}^* = \bar{\mathbf{x}}_1 - R^* \bar{\mathbf{x}}_0. \tag{4.25}$$

Then,

$$w_i\{R^*\hat{\mathbf{x}}_{0,i} + \mathbf{T}^* - \hat{\mathbf{x}}_{1,i}\} = w_i\{R^*\hat{\mathbf{x}}_{0,i} + \bar{\mathbf{x}}_1 - R^*\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_{1,i}\} = R^* w_i\{\hat{\mathbf{x}}_{0,i} - \bar{\mathbf{x}}_0\} - w_i\{\hat{\mathbf{x}}_{1,i} - \bar{\mathbf{x}}_1\}. \tag{4.26}$$

If R^* and $\mathbf{T}^*$ minimize the scalar-weighted objective function (4.19), we conclude from (4.26) that R^* must minimize

$$\sum_{i=1}^{n} \|R w_i\{\hat{\mathbf{x}}_{0,i} - \bar{\mathbf{x}}_0\} - w_i\{\hat{\mathbf{x}}_{1,i} - \bar{\mathbf{x}}_1\}\|^2. \tag{4.27}$$

Noticing that the term under the summation has a form $\|R\mathbf{x} - \mathbf{y}\|^2$, we have

$$\|R\mathbf{x} - \mathbf{y}\|^2 = \{R\mathbf{x} - \mathbf{y}\}^\top \{R\mathbf{x} - \mathbf{y}\} = \mathbf{x}^\top R^\top R \mathbf{x} - 2\mathbf{x}^\top R^\top \mathbf{y} + \|\mathbf{y}\|^2. \tag{4.28}$$

Because R is orthonormal ($R^\top R = I$), (4.28) is a linear function in the elements of R. The rotation matrix R^* that minimizes (4.27) can be solved for in closed-form (with a noniterative algorithm) by the method presented in Appendix A. An alternative way to solve for R to minimize (4.27) consists of using singular value decomposition, as presented in [13]. Once R^* is determined, $\mathbf{T}^*$ is determined based on (4.25).

We further consider the matrix-weighted solution. Since Γ_i^{-1} is a positive definite matrix, there is a matrix $\mathbf{W}_i$, obtained by *Cholesky decomposition* [75], such that

$$\Gamma_i^{-1} = \mathbf{W}_i^\top \mathbf{W}_i. \tag{4.29}$$

Because Γ_i^{-1} is just a 3 by 3 matrix, $\mathbf{W}_i$ can be computed by a non-iterative algorithm. The objective function (4.16) can then be rewritten as

$$\sum_{i=1}^{n} \|\mathbf{W}_i\{R\hat{\mathbf{x}}_{0,i} + \mathbf{T} - \hat{\mathbf{x}}_{1,i}\}\|^2. \tag{4.30}$$

From (4.21) of the MWCC theorem, we have

$$\sum_{i=1}^{n} \Gamma_i^{-1} R^* \hat{\mathbf{x}}_{0,i} + \sum_{i=1}^{n} \Gamma_i^{-1} \mathbf{T}^* = \sum_{i=1}^{n} \Gamma_i^{-1} \hat{\mathbf{x}}_{1,i}. \tag{4.31}$$

It follows that

$$\mathbf{T}^* = \{\sum_{i=1}^{n} \Gamma_i^{-1}\}^{-1} \sum_{i=1}^{n} \Gamma_i^{-1} \hat{\mathbf{x}}_{1,i} - \{\sum_{i=1}^{n} \Gamma_i^{-1}\}^{-1} \sum_{i=1}^{n} \Gamma_i^{-1} R^* \hat{\mathbf{x}}_{0,i}. \tag{4.32}$$

Substituting $\mathbf{T}$ in (4.30) by the right-hand side of (4.32), we get an expression which is quadratic in the elements of R. This implies that when using general matrix weights, we cannot simplify the objective function to a linear expression in the elements of R as we did for the unweighted or the scalar-weighted cases in (4.28). This is due to the fact that matrix multiplication is generally not commutative except for some special cases. To give a concise form of the quadratic expression, we represent a rotation matrix by the corresponding vector (denoted by bold font) which consists of its rows. Letting $R = [\mathbf{R}_1,\ \mathbf{R}_2,\ \mathbf{R}_3]^\top$, we have

$$R\mathbf{x} = C(\mathbf{x})\mathbf{R} \tag{4.33}$$

where the mapping from a three-dimensional vector $\mathbf{x}$ to a 3 by 9 matrix $C(\mathbf{x})$ is

$$C(\mathbf{x}) = \begin{bmatrix} \mathbf{x} & 0 & 0 \\ 0 & \mathbf{x} & 0 \\ 0 & 0 & \mathbf{x} \end{bmatrix}^\top \tag{4.34}$$

From (4.32), it follows that

$$\mathbf{T}^* = \{\sum_{i=1}^{n} \Gamma_i^{-1}\}^{-1} \sum_{i=1}^{n} \Gamma_i^{-1} \hat{\mathbf{x}}_{1,i} - \{\sum_{i=1}^{n} \Gamma_i^{-1}\}^{-1} \sum_{i=1}^{n} \Gamma_i^{-1} C(\hat{\mathbf{x}}_{0,i}) R^* = \mathbf{d} - C\mathbf{R}^*. \tag{4.35}$$

Then

$$\begin{aligned} \mathbf{W}_i\{R^* \hat{\mathbf{x}}_{0,i} + \mathbf{T}^* - \hat{\mathbf{x}}_{1,i}\} &= \mathbf{W}_i\{C(\hat{\mathbf{x}}_{0,i})\mathbf{R}^* + \mathbf{d} - C\mathbf{R}^* - \hat{\mathbf{x}}_{1,i}\} \qquad (4.36) \\ &= \mathbf{W}_i\{C(\hat{\mathbf{x}}_{0,i}) - C\}\mathbf{R}^* - \mathbf{W}_i\{\hat{\mathbf{x}}_{1,i} - \mathbf{d}\} \\ &= A_i\mathbf{R}^* - \mathbf{b}_i. \end{aligned}$$

With n point correspondences, define a new matrix A and a new vector $\mathbf{b}$ by

$$A = \begin{bmatrix} A_1 \\ . \\ . \\ . \\ A_n \end{bmatrix} \tag{4.37}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ . \\ . \\ . \\ \mathbf{b}_n \end{bmatrix} \tag{4.38}$$

According to (4.36), if we substitute R^* and $\mathbf{T}^*$ into the objective function in (4.30), we get

$$||A\mathbf{R}^* - \mathbf{b}||. \tag{4.39}$$

Therefore, the solution to the rotation matrix R^* must be such that (4.39) is minimized. The nine-dimensional vector $\mathbf{R}^*$ in (4.39) is subject to the constraint that it represents a rotation matrix. An iterative algorithm is required to search for a $\mathbf{R}^*$ that satisfies the constraint and minimizes (4.39). To acquire a closed-form solution, we begin by solving for an intermediate $\hat{R}$ that minimizes (4.39) without the rotation matrix constraint:

$$\hat{\mathbf{R}} = (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}\mathbf{b}. \tag{4.40}$$

The rotation matrix R^* is then resolved with the rotation matrix constraint through minimization of

$$\|R^* - \hat{R}\|^2 \tag{4.41}$$

using the method presented in Appendix A. Finally, the translation is determined according to (4.35):

$$\mathbf{T}^* = \mathbf{d} - C\mathbf{R}^*. \tag{4.42}$$

Since the constraint in the corresponding vector of the rotation matrix is not considered in minimizing (4.39) but, instead, it is compensated later in minimizing (4.41), the performance is penalized. However, such a penalty is expected to be much less significant than the penalty caused by improper weighting with typical stereo setups. The simulation results have showed that this closed-form solution is significantly more reliable than both unweighted and scalar-weighted least-squares solutions.

Because the noise may cause a degenerate matrix to become nondegenerate, the uniqueness question should be studied in the absence of noise. A rigid motion can be uniquely determined from three nonlinear points. However, the above algorithm for the matrix-weighted solution requires at least 4 point correspondences, since the constraint in the intermediate rotation matrix $\hat{R}$ is not considered. In (4.39), $\mathbf{R}^*$ is a 9-dimensional vector. Each point correspondence gives 3 scalar equations in the corresponding noise-free equation:

$$A\mathbf{R}^* = \mathbf{b}. \tag{4.43}$$

It seems that 3 point correspondences might be enough to uniquely determine $\mathbf{R}^*$. However, this is not the case, since the matrix A may not have a full rank. We consider a noise-free case, for which we can replace Γ_i^{-1} by $(1/n)I$ for all i. Based on the derivation of the closed-form solution of the matrix-weighted objective function, it can be seen that (4.43) is equivalent to the noise-free equation that corresponds to (4.27):

$$R\{\mathbf{x}_{0,i} - \bar{\mathbf{x}}_0\} = \{\mathbf{x}_{1,i} - \bar{\mathbf{x}}_1\}. \tag{4.44}$$

Since three points are always coplanar, the three vectors $\mathbf{x}_{0,i} - \bar{\mathbf{x}}_0$, $i = 1,\ 2,\ 3$, are coplanar. Consequently, R cannot be uniquely determined by three point correspondences based on (4.44) without imposing a constraint on R. With four non-coplnar point, R is uniquely determined by (4.44). So, four point correspondences are enough in general to uniquely determine the intermediate matrix $\hat{R}$. The fitting of a rotation matrix to the intermediate matrix $\hat{R}$ will improve the rotation matrix, but will not affect uniqueness. Once the rotation matrix is determined, the translation vector can be determined by (4.25) (or (4.42) for the matrix-weighted solution).

In summary, while 3 point correspondences is the minimum number needed for the unweighted or scalar-weighted solutions discussed in Subsection 4.3.1, the closed-form matrix-weighted least-squares solution requires at least 4 point correspondences. This is due to the fact that the constraint on the rotation matrix R is not considered in solving (4.39) for the intermediate rotation matrix R.

4.3.2 Iterative optimal solution from two stereo pairs

From two pairs of stereo images with 3-D point correspondences, the parameters to be estimated are the structure of the points

$$\mathbf{x} = (\mathbf{x}_1^{\mathsf{T}},\ \mathbf{x}_2^{\mathsf{T}},\ \cdots,\mathbf{x}_n^{\mathsf{T}})^{\mathsf{T}} \tag{4.45}$$

and the motion parameter vector $\mathbf{m}$ as defined in (4.17). Let $\mathbf{t}$ denote all the parameters to be estimated from two pairs of stereo images:

$$\mathbf{t} = (\mathbf{x}^{\top},\ \mathbf{m}^{\top})^{\top}, \tag{4.46}$$

and let the two-dimensional image coordinate vector of the i-th point in image j ($j = 1$ for left image and $j = 2$ for right image) at time t_k be $\mathbf{u}_{k,j,i}$. Suppose observation vector $\tilde{\mathbf{u}}$ consists of all image vectors at time t_0 and t_1. Given $\mathbf{t}$, the noise-free projection vector $\mathbf{u}$ can be directly determined through projection:

$$\mathbf{u} = \mathbf{f}(\mathbf{t}). \tag{4.47}$$

In the presence of noise, the contaminated observation vector $\tilde{\mathbf{u}}$ is given by

$$\tilde{\mathbf{u}} = \mathbf{f}(\mathbf{t}) + \delta_{\mathbf{u}}. \tag{4.48}$$

The noise term $\delta_{\mathbf{u}}$ accounts for measurement noise in the image plane. Supposing $\delta_{\mathbf{u}}$ has an approximately zero mean, and a error covariance matrix $\sigma^2 I$, as discussed in Section 4.3.1 the optimal $\mathbf{t}$ minimizes

$$\|\tilde{\mathbf{u}} - \mathbf{f}(\mathbf{t})\|. \tag{4.49}$$

This objective function is based on the 2-D image plane noise model (4.48), while the complicated 3-D uncertainties in the measured 3-D points will be implicitly taken into account. The matrix-weighted or the scalar-weighted least-squares solution is used as an initial solution for an iterative algorithm (e.g., Levenberg-Marquardt method or conjugate gradient method) that improves the initial solution to minimize (4.49). The expected error in the parameter $\mathbf{t}$ is provided by a error covariance matrix similar to (2.29).

However, several points should be considered.

(1) The above method is computationally expensive. The main reason for this is the large dimension of the parameter space. For two pairs of stereo images with n

point correspondences, the parameter spacei s $(3n+6)$-dimensional. For example, with 20 point correspondences, the iterative algorithm has to search in a 66-dimensional space!

(2) The direct extension to deal with long image sequences is computationally prohibitive and complicated. If many images are used, $\mathbf{u}$ include all image points that have ever appeared in some images. The number of such points may be extremely large in the case of extended navigation. Due to occlusions and other reasons, a point can disappear and reappear many times in an image sequence. In other words, there is only a moderately large number of points that are currently visible in each time instant.

(3) The model is not suited for recursive computation. In extended navigation, it is impossible to store all the data. The old information should be stored in a concise manner and be efficiently used.

We present in the following a modified method, in which the structure of 3-D points is not included in the search space, and the corresponding model can be directly extended to recursive computation from long image sequences.

Let the true but unknown 3-D position of the i-th point at time t_k, in the local coordinate system, be $\mathbf{x}_{k,i}$ and the collection of all such points at time t_k be $\mathbf{x}_{k,\bullet}$. Given two pairs of stereo images, (corresponding to instants t_0 and t_1), the parameter vector $\mathbf{t}$ to be estimated consists of the interframe motion parameter vector $\mathbf{m}$, and the structure of the 3-D points $\mathbf{x}_{0,\bullet}$ at time t_0: $\mathbf{t} = (\mathbf{m}^{\mathsf{T}},\ \mathbf{x}_{0,\bullet}^{\mathsf{T}})^{\mathsf{T}}$ (equivalently, we can consider the structure at time t_1). The set of image observation vectors consists of all noise corrupted versions $\tilde{\mathbf{u}}_{k,j,i}$ of $\mathbf{u}_{k,j,i}$. The objective function in (4.49) can now be rewritten i n detail as

$$f(\mathbf{m},\ \mathbf{x}_{0,\bullet}) = \sum_{i=1}^{n} \sum_{j=1}^{2} \sum_{k=0}^{1} \sigma^{-2} \|\mathbf{u}_{k,j,i}(\mathbf{m},\ \mathbf{x}_{0,i}) - \tilde{\mathbf{u}}_{k,j,i}\|^2 \tag{4.50}$$

where $\mathbf{u}_{k,j,i}(\mathbf{m},\ \mathbf{x}_{0,i})$ is the noise-free projection computed from $\mathbf{m}$ and $\mathbf{x}_{0,i}$, and σ^2 is the image noise variance.

The above model is a natural model with two pairs of stereo images. Another alternative model, although less natural, is useful for our recursive estimation from long image sequences. We consider as an "observation" at time t_0, the estimate $\mathbf{x}_{0,i}$ computed by the method presented in Section 4.2. The objective function to be minimized then becomes

$$f(\mathbf{m},\ \mathbf{x}_{0,\bullet}) = \sum_{i=1}^{n}\{(\mathbf{x}_{0,i} - \hat{\mathbf{x}}_{0,i})^{\mathsf{T}}\Gamma_{\mathbf{x}_{0,i}}^{-1}(\mathbf{x}_{0,i} - \hat{\mathbf{x}}_{0,i}) + \sum_{j=1}^{2}\sigma^{-2}\|\mathbf{u}_{1,j,i}(\mathbf{m},\ \mathbf{x}_{0,i}) - \tilde{\mathbf{u}}_{1,j,i}\|^2\} \tag{4.51}$$

where we assume that (a) errors in $\mathbf{x}_{0,\bullet}$ estimated from images before t_1 are uncorrelated, (b) errors in the measured image coordinates of points $\{\tilde{\mathbf{u}}_{1,j,i}\}$ are all uncorrelated and have the same variance σ^2 and (c) errors in previously estimated $\mathbf{x}_{0,\bullet}$ are uncorrelated with the errors in the currently measured image coordinates of points. Due to the non-symmetrical nature of the distribution of errors in the 3-D coordinates of a point constructed by triangulation, the objective function in (4.51) is not as good as that in (4.50), it is developed for recursive estimation from long image sequences.

The objective functions in (4.51) are neither linear nor quadratic, and an iterative algorithm is required to get a solution. Instead of performing a computationally expensive direct optimization, we reduce the dimension of parameter space first. Since the objective functions are continuous, we have

$$f(\hat{\mathbf{m}},\ \hat{\mathbf{x}}_{0,\bullet}) = \min_{\mathbf{m},\,\mathbf{x}_{0,\bullet}} f(\mathbf{m},\ \mathbf{x}_{0,\bullet}) = \min_{\mathbf{m}}\{\min_{\mathbf{x}_{0,\bullet}} f(\mathbf{m},\ \mathbf{x}_{0,\bullet})\} = \min_{\mathbf{m}} g(\mathbf{m},\ \hat{\mathbf{x}}_{0,\bullet}) \tag{4.52}$$

where

$$g(\mathbf{m},\ \hat{\mathbf{x}}_{0,\bullet}) = \min_{\mathbf{x}_{0,\bullet}} f(\mathbf{m},\ \mathbf{x}_{0,\bullet}) \tag{4.53}$$

is the smallest "cost", computed by choosing the "best" structure $\mathbf{x}_{0,\bullet}$, with a given motion parameter vector $\mathbf{m}$. This means that the space $(\mathbf{m},\ \mathbf{x}_{0,\bullet})$ is decomposed into two subspaces, corresponding to $\mathbf{m}$ and $\mathbf{x}_{0,\bullet}$, respectively. In the subspace of $\mathbf{m}$, an iterative algorithm (e.g, Levenberg-Marquardt method or conjugate gradient method) is used. In the subspace of $\mathbf{x}_{0,\bullet}$, an non-iterative method is used that gives the best $\mathbf{x}_{0,\bullet}$ for any given $\mathbf{m}$. According to the decomposition shown in (4.52), the search space in

$$\min_{\mathbf{m}} g(\mathbf{m},\ \hat{\mathbf{x}}_{0,\bullet})$$

is just the 6-dimensional motion parameter space. With a good initial solution of $\mathbf{m}$ provided by the matrix-weighted solution, few iterations are needed to reach the optimal solution. Since the dimension of $\mathbf{x}$ is very large ($3n$-dimensional with n point correspondences), this decomposition significantly reduces the computational cost.

We now consider how to compute the best $\mathbf{x}_{0,\bullet}$ in (4.53), without resorting to iterations. In (4.51), there are two terms for each point, one is a matrix-weighted discrepancy of $\mathbf{x}_{0,i} - \hat{\mathbf{x}}_{0,i}$

$$\{\mathbf{x}_{0,i} - \hat{\mathbf{x}}_{0,i}\}^{\mathsf{T}} \Gamma^{-1}_{\mathbf{x}_{0,i}} \{\mathbf{x}_{0,i} - \hat{\mathbf{x}}_{0,i}\} \tag{4.54}$$

the other is

$$\sum_{j=1}^{2} \sigma^{-2} \|\mathbf{u}_{1,j,i}(\mathbf{m},\ \mathbf{x}_{0,\bullet}) - \tilde{\mathbf{u}}_{1,j,i}\|^2. \tag{4.55}$$

The latter term can be approximated by the estimated 3-D position at time t_1 through triangulation. We have described in Section 4.2 a method to get an estimate $\hat{\mathbf{x}}_{1,i}$ that minimizes (4.55). The corresponding error covariance matrix $\Gamma_{\mathbf{x}_{1,i}}$ of the point $\hat{\mathbf{x}}_{1,i}$ can also be estimated. In other words, we have two sample data here for the same unknown parameter vector $\mathbf{x}_{0,i}$. One is $\mathbf{p} = \hat{\mathbf{x}}_{0,i}$ with the error covariance matrix $\Gamma_{\mathbf{p}} = \Gamma_{\mathbf{x}_{0,i}}$ and the other is the point moved back from $\hat{\mathbf{x}}_{1,i}$: $\mathbf{q} = R^{\mathsf{T}}\{\hat{\mathbf{x}}_{1,i} - \mathbf{T}\}$ with the error covariance matrix $\Gamma_{\mathbf{q}} = R^{\mathsf{T}}\Gamma_{\mathbf{x}_{1,i}}R$. According to the principle of weighted least squares, the optimal $\mathbf{x}_{0,i}$ should minimize

$$\{\mathbf{x}_{0,i} - \mathbf{p}\}^{\mathsf{T}} \Gamma^{-1}_{\mathbf{p}} \{\mathbf{x}_{0,i} - \mathbf{p}\} + \{\mathbf{x}_{0,i} - \mathbf{q}\}^{\mathsf{T}} \Gamma^{-1}_{\mathbf{q}} \{\mathbf{x}_{0,i} - \mathbf{q}\}. \tag{4.56}$$

For a given motion, the optimal $\mathbf{x}^*_{0,i}$ that minimizes (4.56) is thus directly computed (without iteration) by

$$\mathbf{x}^*_{0,i} = \Gamma_{\mathbf{q}}\{\Gamma_{\mathbf{p}} + \Gamma_{\mathbf{q}}\}^{-1}\mathbf{p} + \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}} + \Gamma_{\mathbf{q}}\}^{-1}\mathbf{q} = \mathbf{p} + \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}} + \Gamma_{\mathbf{q}}\}^{-1}\{\mathbf{q} - \mathbf{p}\}. \quad (4.57)$$

Based on (4.57), a sequence of best points can be calculated for any given motion parameter vector $\mathbf{m}$, and the corresponding residual in (4.51) is then computed. After the best $\mathbf{m}$ is determined by an iterative algorithm to minimize the residual in (4.51), the corresponding set of points is the best solution for the structure.

Through investigation of the nature of the objective functions (4.51), we have explored the relationships between $\mathbf{m}$ and $\mathbf{x}_{0,\bullet}$ so that the constraint related them is fully utilized. The computationally almost intractable optimization problem in the space of $\mathbf{t} = (\mathbf{m}^\top,\ \mathbf{x}^\top_{0,\bullet})$ is decomposed into two levels: (a) At the higher level is an iterative algorithm in the 6-dimensional subspace of $\mathbf{m}$ starting with a good initial solution; (b) At the lower level is a non-iterative optimization algorithm that directly computes the optimal solution in the large subspace of $\mathbf{x}_{0,\bullet}$.

4.3.3 Estimating errors

Since the actual errors in the solutions depend on random noise, it is reasonable to estimate the expected errors. More specifically, we estimate the error covariance matrix of the estimated parameters. These error covariance matrices not only enable us to assess the expected accuracy of the estimates, but also are important for further estimation using the obtained estimates.

In the objective function in (4.51), each point has three "observations": $\hat{\mathbf{x}}_{0,i}$, $\tilde{\mathbf{u}}_{1,1,i}$ and $\tilde{\mathbf{u}}_{1,2,i}$. We define a 7-dimensional observation vector $\mathbf{v}_i$ which consists of these three observations. Namely $\tilde{\mathbf{u}}$ in (4.48) consists of $\mathbf{v}_i$, $i = 1,\ 2,\ \cdots,\ n$. Suppressing $\mathbf{x}_{0,\bullet}$ in (4.48) we have

$$\tilde{\mathbf{u}} = \mathbf{f}(\mathbf{m}) + \delta_{\mathbf{u}} \quad (4.58)$$

where $\mathbf{f}(\mathbf{m})$ is the computed observation from $\mathbf{m}$ ($\hat{\mathbf{x}}_{0,\bullet}$ is computed from $\mathbf{m}$ and image plane observations as discussed in Subsection 4.3.2).

The estimated error covariance matrix $\Gamma_{\mathbf{m}}$ of motion parameter vector $\mathbf{m}$ is given in (2.29) where A is replaced by

$$\frac{\partial \mathbf{f}(\hat{\mathbf{m}})}{\partial \mathbf{m}}, \tag{4.59}$$

which is evaluated with the optimal estimates of $\mathbf{m}$ and $\mathbf{x}_{0,\bullet}$. According to the assumption that the image noise components in $\delta_{\mathbf{u}}$ are uncorrelated between different points, the error covariance matrix of $\delta_{\mathbf{u}}$ is a block diagonal matrix.

For efficiency, the error matrix of a point $\mathbf{x}_{0,i}$ is estimated based on the space-decomposition method. In Subsection 4.3.2, the error covariance matrix of $\mathbf{q}$ was given by $\Gamma_{\mathbf{q}} = R^{-1}\Gamma_{\mathbf{x}_{i,1}}R$, which is a conditional error covariance matrix conditioned on the given motion parameters. For error estimation here, the error covariance matrix of $\mathbf{q}$ is unconditional and should take into account the errors in the estimated motion parameters. From the definition of vector $\mathbf{q}(\mathbf{m}) = R^{-1}(\hat{\mathbf{x}}_{i,1} - \mathbf{T})$, the error covariance matrix of $\mathbf{q}$ should be

$$\Gamma_{\mathbf{q}} = R^{-1}\Gamma_{\mathbf{x}_{i,1}}R + \frac{\partial \mathbf{q}(\mathbf{m})}{\partial \mathbf{m}}\Gamma_{\mathbf{m}}\frac{\partial \mathbf{q}(\mathbf{m})^{\top}}{\partial \mathbf{m}}. \tag{4.60}$$

According to (2.29) the error covariance matrix of the estimated 3-D position of the point $\mathbf{x}^*_{i,0}$ in (4.57) is estimated by

$$\Gamma_{\mathbf{x}^*_{i,0}} = \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}} + \Gamma_{\mathbf{q}}\}^{\top}\Gamma_{\mathbf{q}} = \Gamma_{\mathbf{p}} - \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}} + \Gamma_{\mathbf{q}}\}^{\top}\Gamma_{\mathbf{p}}. \tag{4.61}$$

The diagonal elements of this error covariance give the expected error variances of the corresponding components of the estimated vector.

4.3.4 A recursive-batch approach

With a stereo image sequence, the parameters to be estimated include the 3-D positions of feature points of a scene, represented in some coordinate system and the interframe motion parameters between every pair of consecutive instants. The global attitude of the camera system can be determined from interframe motions based on (2.19).

Based on the foregoing discussion, we know that the estimation of structure and interframe motions are closely related. The accuracy of the estimated structure influences the accuracy of the estimated motion parameters, and vice versa. Due to this type of interaction between the structure and interframe motion, a higher estimation accuracy can be obtained if all the image frames to be considered are processed in a batch fashion. In reality, however, this is a computationally prohibitive task if the image sequence is long.

In fact, a 3-D point may not be visible for a long time. It will likely go out of the field of view and disappear from the stereo image sequence after a while. This implies that in a long stereo image sequence, the relation between two pairs of stereo images is weak if the two pairs are far apart. Consequently, the accuracy of the estimated structure in a section of a scene will not affect very much the accuracy of the estimated structure of a different section of the same scene.

As discussed in Chapter 2, *Kalman filtering* techniques have some desirable advantages. For a linear problem, theoretically, the result of *Kalman filtering* is the same as that of a batch method. However, for a nonlinear problem, the result of *Kalman filtering* is not as good. The key problem with *Kalman filtering* for a nonlinear problem is that the system Jacobian matrix for each old observation is not updated when new observations are processed. This is a fundamental structure of sequential processing. If all observations are processed in a batch fashion, the modification of parameters is very reliable and the system matrix of every observation is updated at each iteration from all observations. In other words, with a sequential algorithm, the contribution or influence of the later observations to the evaluation of the system matrices for the early observations are neglected (this is not a problem for a linear problem since the system Jacobian matrix is constant).

In order to achieve good performance without excessive computational cost, we need batch processing only for those data that have considerable interactions. The above observations motivate our recursive-batch approach which is illustrated in Figure 4.2. In this approach, the observed stereo sequence is processed in relatively small groups. For each group of data, estimates are determined in a batch fashion from old estimates and the current group of data. The approach is recursive because the processing step is repeated for each batch of data and the newly estimated result depends on previous result. From a global point of view, the data is processed sequentially by feeding through a processing algorithm that cover a certain length (batch size) of the sequence. This approach has the following advantages:

(1) It can process virtually very long image sequences with a limited memory.

(2) Since old estimates can be used for new estimates, and a limited amount of computation is required to update estimates, the algorithm is relatively efficient.

(3) The algorithm outperforms a straight sequential technique (e.g, *Kalman filtering* technique) because the data is processed in a batch fashion in which each overlapping batch sufficiently covers the interaction among data.

If the batch size of the recursive-batch approach is so large that the batch covers the whole image sequence, the recursive-batch approach degenerates into a pure batch approach. On the other hand, if the batch size is equal to one 3-D point, the recursive-batch approach becomes an iterated extended *Kalman filter*. The choice of an appropriate batch size is important. For the problem studied in this research, a natural batch size corresponds to all observations in a single stereo image pair. With each new pair of stereo images, a new interframe motion needs to be estimated and the structure of points is updated. The performance may be further improved if several stereo pairs are processed as a batch set (the batch will slide through the sequence so that batches are overlapping). However, this will significantly increase

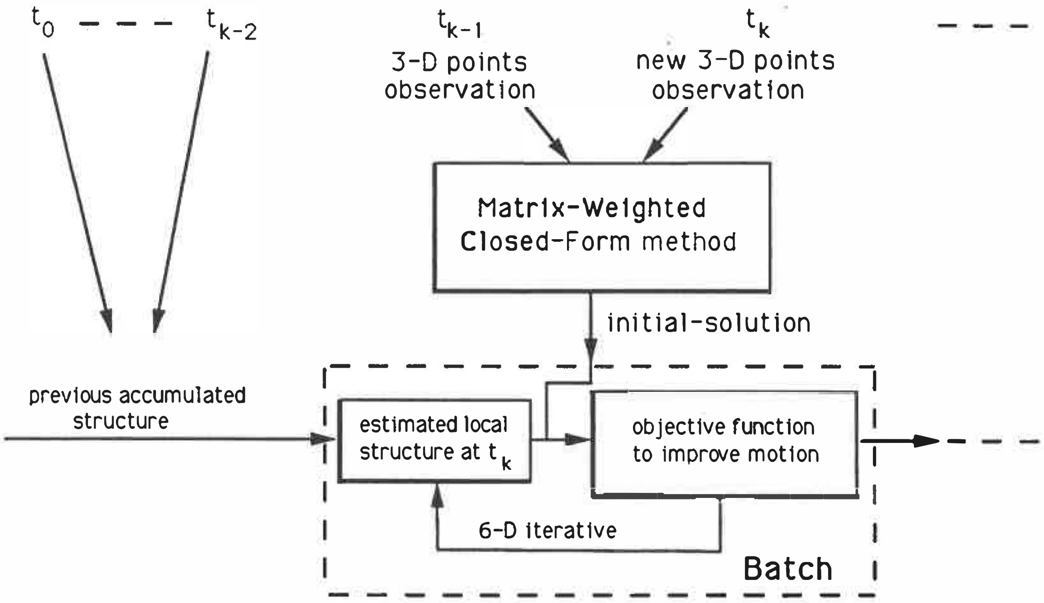


Figure 4.2: Detailed illustration of recursive-batch approach for processing long stereo sequences.

the computational cost.

4.3.5 Recursive-batch updating

The model represented by the objective function (4.51) is very suitable for recursive-batch updating. With each new pair of stereo images, the interframe motion is computed in closed-form as explained in Subsection 4.3.1. The motion parameters are further optimized by the method presented in Subsection 4.3.2. The fusion of the estimated structure accumulated up to previous time instant with the new pair of stereo images updates the structure, which will be used further for the next pair of stereo images. In other words, the structure of points in the local coordinate system is the parameter vector that is updated through the stereo image sequence, while the interframe motion determines how the previous structure evolves into the current time instant. In order to provide the following recursive step with the updated structure and the associated uncertainty, we need to compute the local structure at the current instant together with its error covariance matrix.

After the interframe motion parameter vector $\mathbf{m}_{k+1,k}$ is estimated based on points $\{\hat{\mathbf{x}}_{k,i}\}$ before motion and $\{\hat{\mathbf{x}}_{k+1,i}\}$ after motion, we compute the error covariance matrix of the motion parameter vector $\Gamma_{\mathbf{m}_{k+1,k}}$ as discussed in Subsection 4.3.3. The structure at time t_{k+1} can be updated in a way similar to that for the structure at time t_k: Let $\hat{\mathbf{x}}_{k,i}$ be the estimated structure of $\mathbf{x}_{k,i}$ provided by the processing of the previous interframe motion $\mathbf{m}_{k,k-1}$ (for the first interframe motion, it is estimated from stereo triangulation), and $\Gamma_{\hat{\mathbf{x}}_{k,i}}$ be the associated error covariance matrix. Moving the point $\hat{\mathbf{x}}_{k,i}$ to time t_{k+1} gives

$$\mathbf{p} = R_{k+1,k}\hat{\mathbf{x}}_{k,i} + \mathbf{T}_{k+1,k} \tag{4.62}$$

with the associated error covariance matrix

$$\Gamma_{\mathbf{p}} = R_{k+1,k}\Gamma_{\hat{\mathbf{x}}_{k,i}}R^{-1}_{k+1,k} + \frac{\partial \mathbf{p}(\mathbf{m}_{k+1,k})}{\partial \mathbf{m}_{k+1,k}}\Gamma_{\mathbf{m}_{k+1,k}}\frac{\partial \mathbf{p}(\mathbf{m}_{k+1,k})^T}{\partial \mathbf{m}_{k+1,k}}. \tag{4.63}$$

From triangulation at time t_{k+1} we get the estimated position $\mathbf{q}$ of $\mathbf{x}_{k+1,i}$ and the associated error covariance matrix $\Gamma_{\mathbf{q}}$. According to (2.22), the updated estimates of $\mathbf{x}_{k+1,i}$ is given by

$$\Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\mathbf{q}+\Gamma_{\mathbf{q}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\mathbf{p} = \mathbf{q}+\Gamma_{\mathbf{q}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\{\mathbf{p}-\mathbf{q}\} \tag{4.64}$$

with a error covariance matrix from (2.23)

$$\Gamma_{\mathbf{q}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\Gamma_{\mathbf{p}} = \Gamma_{\mathbf{q}} - \Gamma_{\mathbf{q}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\Gamma_{\mathbf{q}} \tag{4.65}$$

The algorithm starts at time t_0 with the structure estimated through triangulation and the associated error covariance matrix as in Section 4.2. Then the recursive estimation proceeds by incrementing t_k to get the following stereo pair. The interframe motion is first estimated using the batch method together with the associated error covariance matrix of the motion parameters as in Subsections 4.3.3. The structure after the interframe motion and its error covariance matrix are estimated using the method presented in this subsection. These estimates are used for the next recursive step as the estimates for structure before the next interframe motion.

4.3.6 Local and global representations

As discussed in Chapter 2, when a stereo camera system moves in a static environment, the two coordinate systems are used to represent the local and global motion of the camera system as well as the structure of the scene. The structure representation in the local coordinate system is useful for path planning with respect to the motion of the navigation system on which the camera system is mounted. For example, it can be used directly to plan for the direction and distance of the next motion. Its representation in the global coordinate system is useful for extended visual map generation, since what is perceived in the moving local coordinate system needs to be registed on a map which is represented in the fixed global coordinate system. The global attitude of the current camera system is also a critical information for map guided navigation, where the navigation system constantly needs to keep track its current position on a map.

Equations (2.19) give the relation that can be used to update the current global attitude of the camera system (represented by a vector $\mathbf{m}_{k+1}$ (i.e., R_{k+1}, $\mathbf{T}_{k+1}$) that consists of 6 independent motion parameters) from the previous attitude $\mathbf{m}_k$ (i.e., R_k, $\mathbf{T}_k$) and the current interframe motion $\mathbf{m}_{k+1,k}$ (i.e., $R_{k+1,k}$, $\mathbf{T}_{k+1,k}$). They define a vector updating function $\mathbf{h}$:

$$\mathbf{m}_{k+1} = \mathbf{h}(\mathbf{m}_k,\ \mathbf{m}_{k+1,k}). \tag{4.66}$$

Letting the difference between the estimated and true vector be denoted by δ with the vector as a subscript, from (4.66) we have

$$\delta_{\mathbf{m}_{k+1}} = \frac{\partial \mathbf{h}(\hat{\mathbf{m}}_k,\ \hat{\mathbf{m}}_{k+1,k})}{\partial \mathbf{m}_k}\delta_{\mathbf{m}_k} + \frac{\partial \mathbf{h}(\hat{\mathbf{m}}_k,\ \hat{\mathbf{m}}_{k+1,k})}{\partial \mathbf{m}_{k+1,k}}\delta_{\mathbf{m}_{k+1,k}}. \tag{4.67}$$

So the error covariance matrix of the global attitude of the stereo camera system is updated by

$$\begin{aligned}\Gamma_{\mathbf{m}_{k+1}} &= \frac{\partial \mathbf{h}(\hat{\mathbf{m}}_k,\ \hat{\mathbf{m}}_{k+1,k})}{\partial \mathbf{m}_k}\Gamma_{\mathbf{m}_k}\frac{\partial \mathbf{h}(\hat{\mathbf{m}}_k,\ \hat{\mathbf{m}}_{k+1,k})^\top}{\partial \mathbf{m}_k} \\ &\quad + \frac{\partial \mathbf{h}(\hat{\mathbf{m}}_k,\ \hat{\mathbf{m}}_{k+1,k})}{\partial \mathbf{m}_{k+1,k}}\Gamma_{\mathbf{m}_{k+1,k}}\frac{\partial \mathbf{h}(\hat{\mathbf{m}}_k,\ \hat{\mathbf{m}}_{k+1,k})}{\partial \mathbf{m}_{k+1,k}}\end{aligned} \tag{4.68}$$

where $\Gamma_{\mathbf{m}_{k+1,k}}$ is the error covariance matrix of the interframe motion parameter vector $\mathbf{m}_{k+1,k}$ estimated by the method presented in Subsection 4.3.3.

Similarly the global position of the perceived structure can also be updated to take into account the new observations in the current stereo pair. Let $\mathbf{p} = \hat{\mathbf{x}}_{0,i}$ be the current estimate of a point. The global position of the new observed point $\mathbf{x}_{k+1,i}$, represented in the local coordinate system, is given from (2.17):

$$\mathbf{q} = R_{k+1}\hat{\mathbf{x}}_{k+1,i} + \mathbf{T}_{k+1} \tag{4.69}$$

with an associated error covariance matrix

$$\Gamma_{\mathbf{q}} = R_{k+1}\Gamma_{\hat{\mathbf{x}}_{k+1,i}}R_{k+1}^{-1} + \frac{\partial\mathbf{q}(\hat{\mathbf{m}}_{k+1})}{\partial\mathbf{m}_{k+1}}\Gamma_{\mathbf{m}_{k+1}}\frac{\partial\mathbf{q}(\hat{\mathbf{m}}_{k+1})}{\partial\mathbf{m}_{k+1}}^{\mathsf{T}}. \tag{4.70}$$

According to (2.22), the updated global structure of the point is given by

$$\Gamma_{\mathbf{q}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\mathbf{p} + \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\mathbf{q} = \mathbf{p} + \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}(\mathbf{q}-\mathbf{p}). \tag{4.71}$$

Based on (2.23) the error covariance matrix of the global point is estimated by

$$\Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\Gamma_{\mathbf{q}} = \Gamma_{\mathbf{p}} - \Gamma_{\mathbf{p}}\{\Gamma_{\mathbf{p}}+\Gamma_{\mathbf{q}}\}^{-1}\Gamma_{\mathbf{p}} \tag{4.72}$$

The following is an outline of the recursive-batch algorithm used to estimate motion and 3-D structure from stereo image sequences:

Step 1): Let $k = 0$. Get stereo pair at time t_0 with stereo point correspondences. For all points compute the estimated 3-D position $\hat{\mathbf{x}}_{0,i}$, at time t_0, and the associated error covariance matrix as in Section 4.2. At time t_0, the local and the global coordinate systems coincide: $R_0 = I$ and $\mathbf{T}_0 = \mathbf{0}$.

Step 2): Get stereo pair at time t_{k+1} with stereo point correspondences. For all points compute the estimated local 3-D position $\hat{\mathbf{x}}_{k+1,i}$ at time t_{k+1}, and the associated error covariance matrix (Section 4.2).

Step 3): Compute the closed-form matrix-weighted solution for the interframe motion parameter vector $\mathbf{m}_{k+1,k}$ from 3-D point correspondences from t_k to t_{k+1} (Subsection 4.3.1).

Step 4): Further optimize the above solution for $\mathbf{m}_{k+1,k}$, through a few iterations, and compute the associated error covariance matrix (Subsection 4.3.2).

Step 5): Update the local coordinates of the points at time t_{k+1} and the associated error covariance matrices (Subsection 4.3.5).

Step 6): Update the global attitude of the stereo camera system $\mathbf{m}_k$ (i.e., R_k and $\mathbf{T}_k$), and the associated error covariance matrices (Subsection 4.3.6). Update global position of points and the associated error covariance matrices (Subsection 4.3.6).

Step 7): If not at the last pair of stereo images, let $k \leftarrow k+1$ and go to Step 2). Otherwise, stop.

4.4 Simulation and experiments

4.4.1 Simulation for two pairs of stereo images

The purpose of the simulation for two-view situations is to compare the performances of unweighted, scalar-weighted (a scalar weight is inversely proportional to the depth) and the matrix-weighted closed-form solutions.

For the first simulation, the 3-D points were generated randomly for each trial, between depth 2m and 15m, with a uniform distribution. The field of view of the two stereo cameras with a squared image plane was about 53°. The two stereo cameras were arranged in such a way that the optical axes of the two cameras intersect at the center of the block where the random points were generated. They were separated along the y-axis by a baseline of 0.5m. Only those points that fall into the field of view of both stereo cameras at instants t_0 and t_1 were used for motion analysis. The variance of Gaussian noise added to the image points was equal to that of a uniform digitization noise in a 256×256-pixel image. The true motion was a rotation about a rotation axis $(1\text{m}, 0.2\text{m}, 0.1\text{m})^{\mathsf{T}}$ by an angle of 8°, followed by a translation of (-0.14m, 1.35m, -0.92m) $^{\mathsf{T}}$. Figure 4.3 shows the simulation results, where the rotation error is measured as the relative error in the rotation matrix, the translation error is measured as the norm of the error vector as in the simulation with monocular image sequences. The average error was obtained through 500 random trials each with a different set of 3-D points. For the matrix-weighted least-squares solution, the weighting matrices were simplified to be independent of motion (that is, the rotation matrix R in the weighting matrices was replaced by an identity matrix I). We can see in Figure 4.3 that the matrix-weighted solution outperforms other non-iterative methods, given a sufficient number of points. But the iterative optimization gives the most accurate results, at the cost of more computation.

For the second simulation, everything was similar to the first simulation, except that the rotation angle here was 30° instead of 8°. In this simulation, the output of the rotation matrix from the scalar-weighted closed-form method was used as the rotation matrix in the weighting matrices of the matrix-weighted closed-form approach. Comparing the three closed-form solutions as shown in Figure 4.4, the matrix-weighted approach still yields better performance than the other two methods, when enough points are provided.

From these two simulations, we conclude that the proposed matrix-weighted closed-form algorithm for motion parameters works in a wide range of situations, with satisfactory speed and accuracy properties.

4.4.2 Simulation for stereo image sequences

The performance of the proposed recursive-batch algorithm for the processing of stereo image sequences have also been explored through simulation, in which the ground truth and the amount of noise can be well controlled and the errors in the estimates can be accurately measured. In particular, this performance enables us to perceive the relationship between errors in motion and in structure in the local and the global coordinate systems.

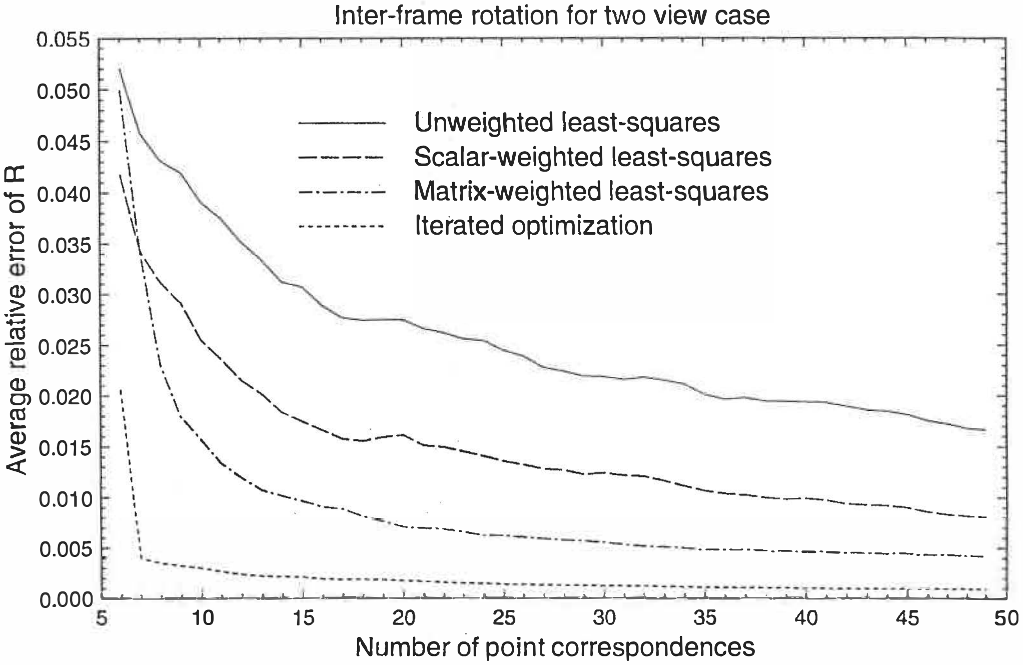

Inter-frame rotation for two view case
Average relative error of R
Number of point correspondences
0.055
0.050
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
5
10
15
20
25
30
35
40
45
50
Unweighted least-squares
Scalar-weighted least-squares
Matrix-weighted least-squares
Iterated optimization

Figure 4.3 (a)

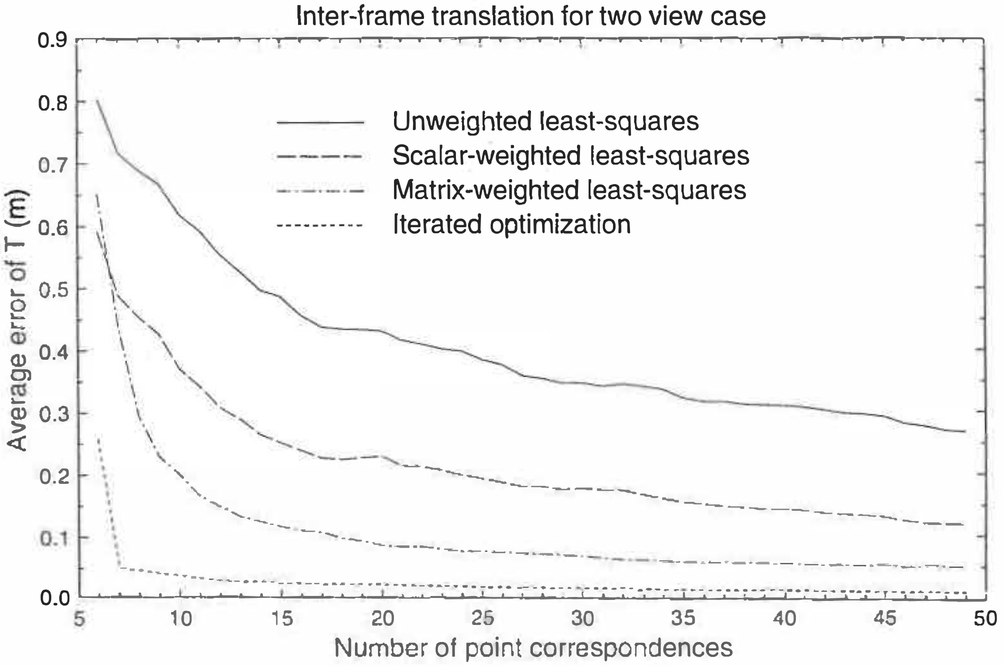


Figure 4.3 (b)

Figure 4.3: Simulation results from two pairs of stereo images: errors in the estimated interframe motion. (a) error in the interframe rotation matrix (the true rotation angle is 8°, the rotation axis is (1m, 0.2m, 0.1m)^T); (b) error in the interframe translation vector (the true translation vector is (-0.14m, 1.35m, -0.92m) ^{T}).

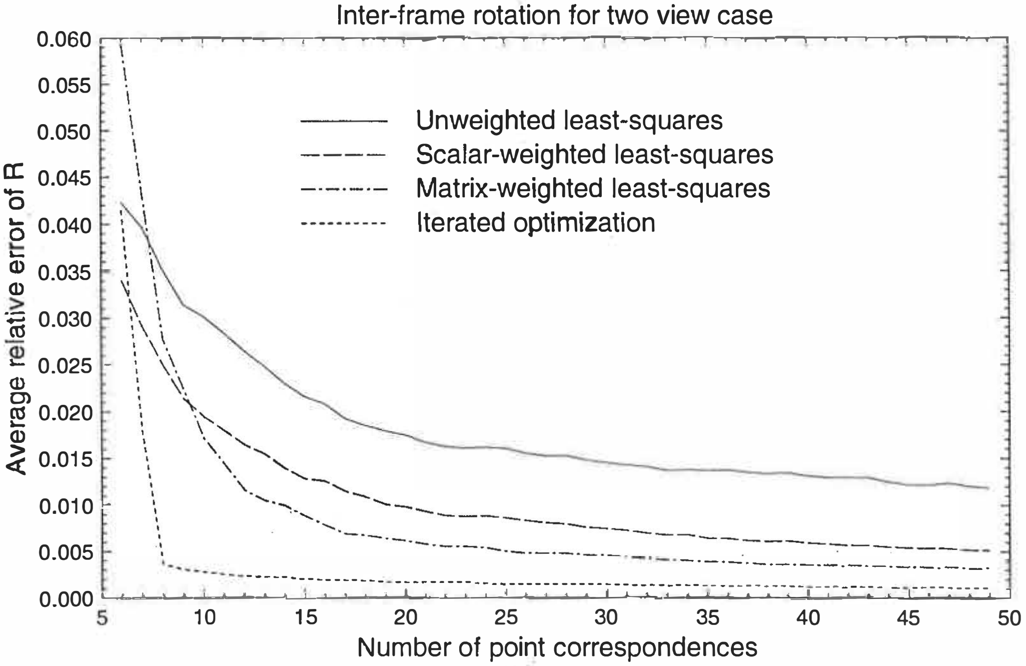

Inter-frame rotation for two view case
Unweighted least-squares
Scalar-weighted least-squares
Matrix-weighted least-squares
Iterated optimization
Average relative error of R
0.060
0.055
0.050
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
5
10
15
20
25
30
35
40
45
50
Number of point correspondences

Figure 4.4 (a)

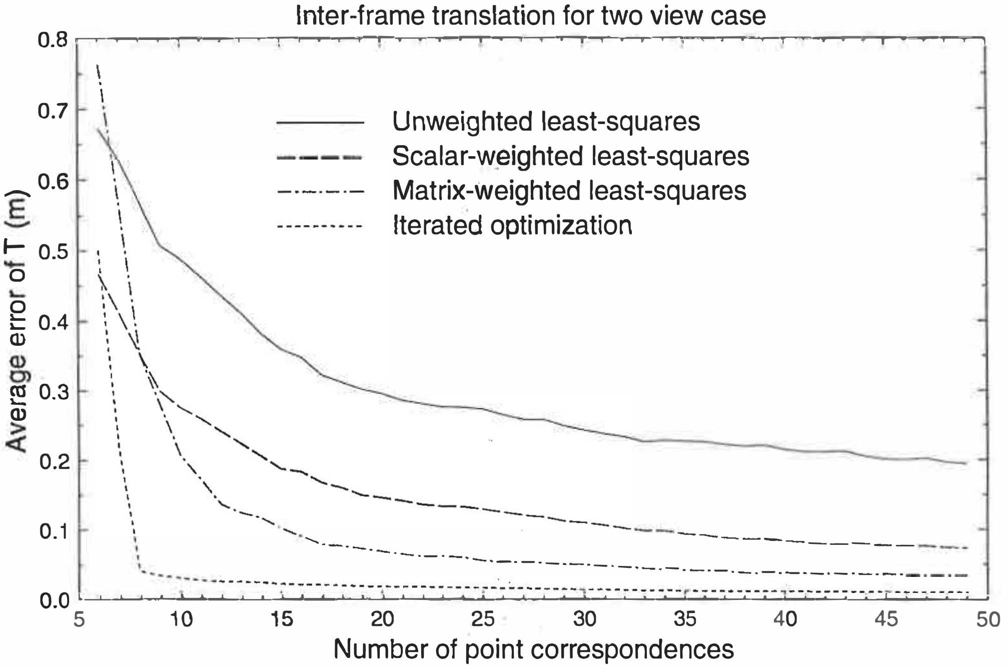


Figure 4.4 (b)

Figure 4.4: Simulation results from two pairs of stereo images: errors in the estimated interframe motion. (a) error in the interframe rotation matrix (the true rotation angle is 30°, the rotation axis is $(1\text{m}, 0.2\text{m}, 0.1\text{m})^{\mathsf{T}}$); (b) error in the interframe translation vector (the true translation vector is $(-0.14\text{m}, 1.35\text{m}, -0.92\text{m})^{\mathsf{T}}$).

As in the simulation of monocular image sequences, the error in rotation in the simulation of stereo image sequences was also measured as the relative error in the rotation matrix (i.e. the norm of the difference matrix between the estimated and true rotation matrices divided by the norm of the true rotation matrix, where the norm of a matrix $R = [r_{ij}]$ is defined as the square root of the sum over all the squared elements r_{ij}^2. The error in the translation vector $\mathbf{T}$ is defined as the norm of the difference vector between the estimated and true vectors. The relative error in the translation vector is the error in translation divided by the norm of the true vector. Notice that relative error may become large if the true vector itself is small. The error in the image projections of points was simulated by additive zero-mean independent Gaussian noise. For the iterative optimization, the "dunlsf" subroutine in IMSL library was used.

In the simulation, the 3-D points were generated between $z = 1$m to $z = 80$m with a uniform distribution. The simulated stereo cameras have a field of view of about 38° and a resolution of 512 × 512 pixels. The stereo setup has a baseline length of 0.2m, and the optical axes of the two cameras form a vergence angle of 10°. The stereo camera system navigates forward through the scene with slight rotations and lateral translations. To keep a relatively constant number of visible point in each image, only those points that lie in the depth range [2 , 10m], in the local coordinate system, are used for motion analysis. This arrangement was designed to simulate situations where the stereo system navigates through a zigzag path and the points in the last view are totally different from those in the first. The sequence we obtained consisted of 40 stereo frames. In the sequence, the stereo camera system has traveled a distance of 60 meters in depth in total. About 60 point correspondences were available between two consecutive stereo frames. The exact number of point correspondences available may vary slightly from frame to frame. Average errors of the algorithms were accumulated through 50 random trials, each with a completely new set of scene points. The global orientation and position of the stereo camera system at any instant were specified by a rotation matrix R_k and a translation vector $\mathbf{T}_k$ as defined in Chapter 2. The true rotation matrix R_k of the stereo camera system with respect to the global coordinate system was 8° around a rotation axis (0.3m, 0.3m, 1m) for odd k, and $-8°$ around the same rotation axis for even k. The given translation vector $\mathbf{T}_k$ of the stereo camera system with respect to the global coordinate system was $(0.5, -0.5, 1.5k)$ for odd k, and $(-0.5, 0.5, 1.5k)$ for even k, with meter as the unit. Figure 4.5 and 4.6 show the average error in the local and global attitude of the stereo camera system, respectively. Three algorithms were employed for estimating interframe motions: unweighted least-squares, matrix-weighted least-squares and iterative optimization. The solution of the matrix-weighted least-squares was used as initial solutions for the iterative optimization.

We can see that the error of the estimated global attitude of the stereo camera system has accumulated through navigation, which reflects the nature of the problem here, because errors are accumulated through every related interframe motion by using the formula in (2.19). We can also see that the error after iterative optimization is much smaller than that of the two closed-form solutions. It does not exhibit a long divergence period as observed in iterated nonlinear *Kalman filtering*. The result is accurate starting from the beginning (time t_1 when the second stereo frame is available). By comparison, the local and global motion solution of the matrix-weighted least-squares is superior to that of the unweighted least-squares.

Figure 4.7 and 4.8 demonstrate relative error in the estimated local and global structure from the matrix-weighted least-squares and iterative optimization. Since the structure estimator of linear approaches is based on triangulation using least squares, the structure estimator of the unweighted least-squares is the same as that of the matrix-weighted least-squares. But at the cost of more computation, the local and global structure from iterative optimization are more accurate.

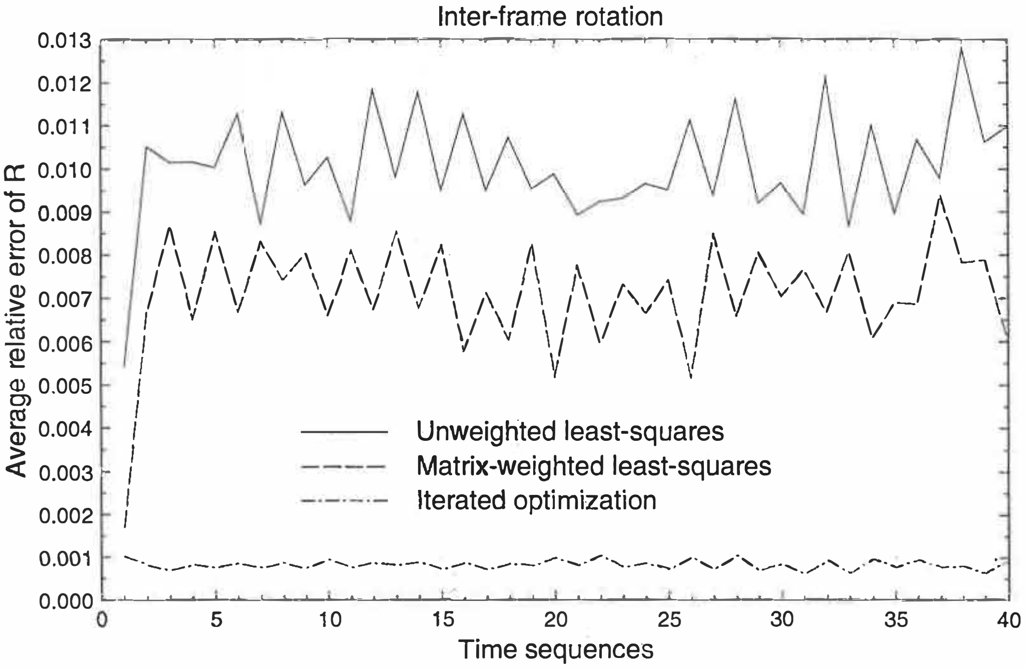

Inter-frame rotation
Average relative error of R
0.013
0.012
0.011
0.010
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
5
10
15
20
25
30
35
40
Time sequences
Unweighted least-squares
Matrix-weighted least-squares
Iterated optimization

Figure 4.5 (a)

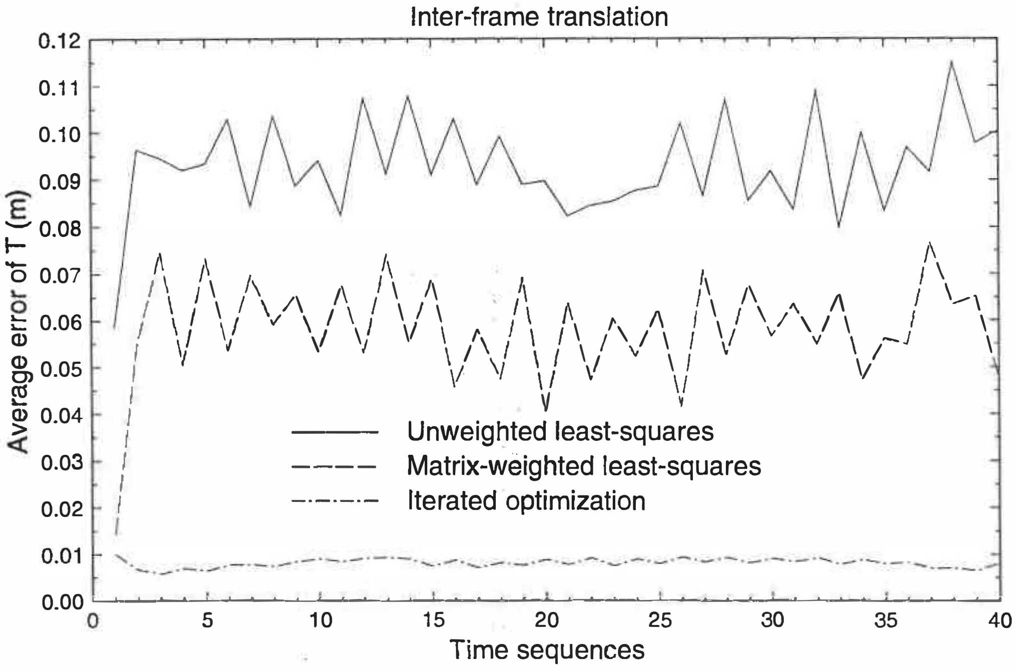


Figure 4.5 (b)

Figure 4.5: Simulation results from stereo image sequence: error in estimated interframe motion. (a) error in interframe rotation matrix $R_{k,k-1}$; (b) error in interframe translation vector $\mathbf{T}_{k,k-1}$

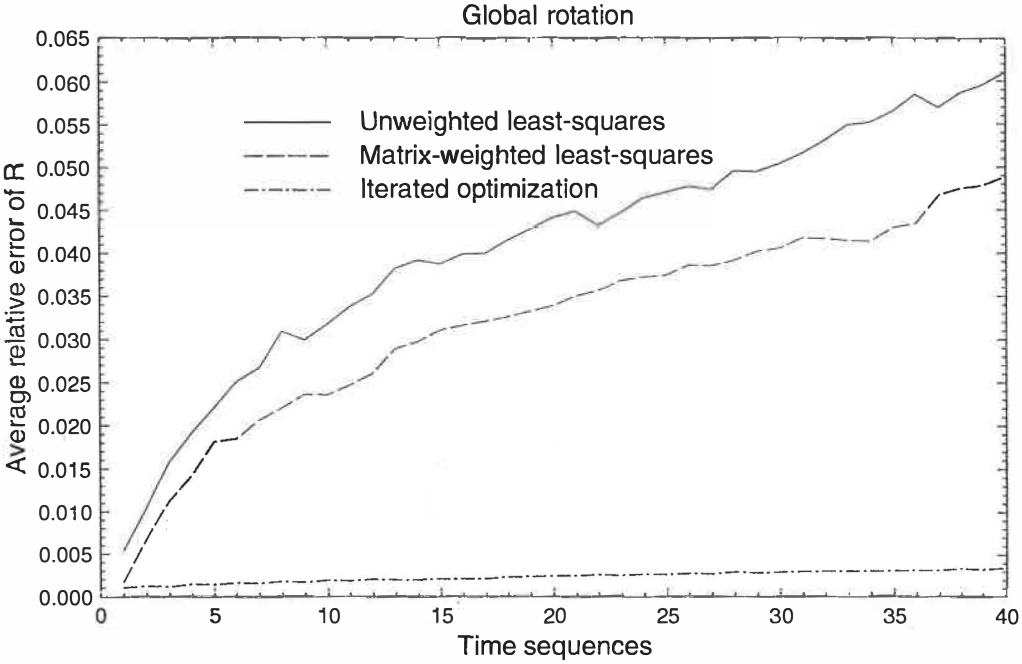

Global rotation
Unweighted least-squares
Matrix-weighted least-squares
Iterated optimization
Average relative error of R
0.065
0.060
0.055
0.050
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
15
20
25
30
35
40
Time sequences

Figure 4.6 (a)

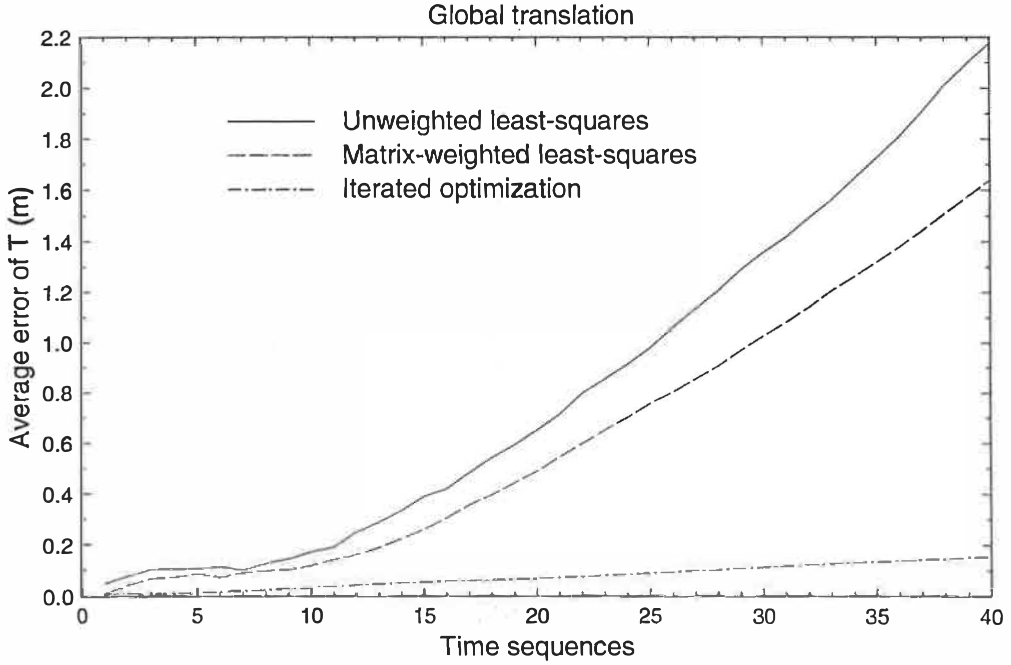


Figure 4.6 (b)

Figure 4.6: Simulation results from stereo image sequence: error in estimated global motion. (a) error in global rotation matrix R_k; (b) error in global translation vector $\mathbf{T}_k$

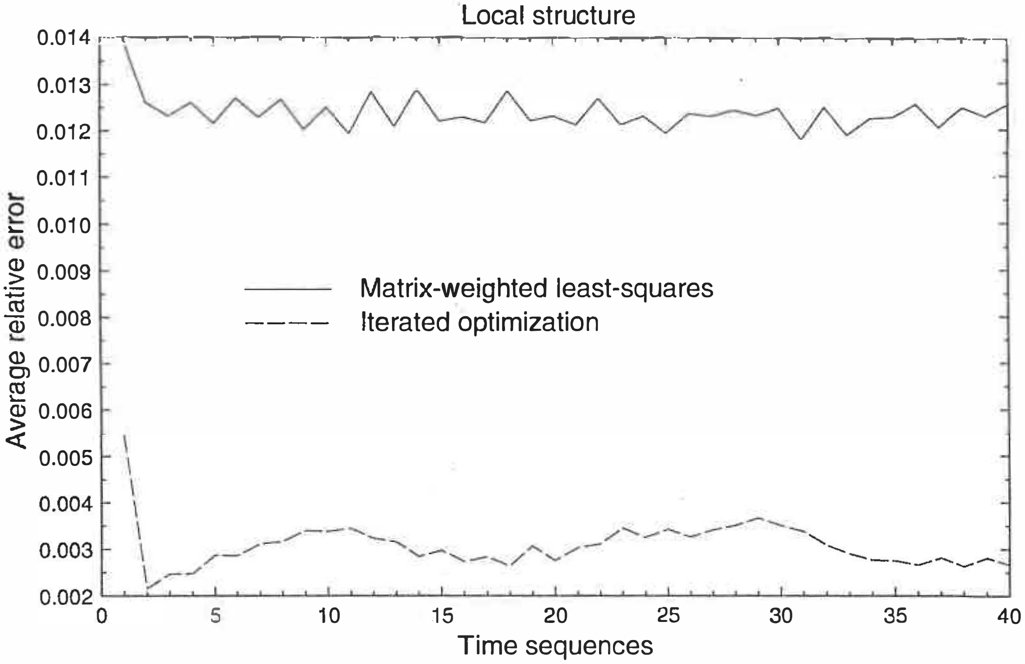


Figure 4.7: Simulation results from a stereo image sequence: error in the estimated local structure.

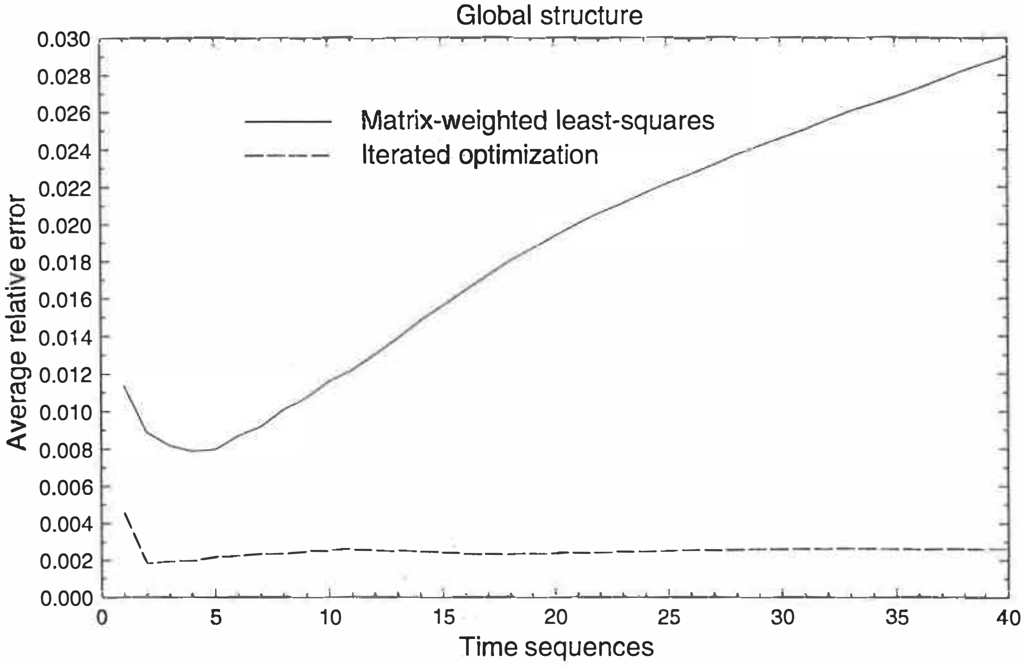


Figure 4.8: Simulation results from a stereo image sequence: error in the estimated global structure.

4.4.3 Experiments

In our experiments with long image sequence, the stereo system consisted of two TM-840 high resolution PULNiX CCD cameras with f=8.5mm wide-angle lenses, mounted on the tip of a high-precision six-joint robot arm. Every digital image grabbed from each camera has 480×512 pixels. After each stereo image pair was grabbed the manipulator was imposed a rotation of 2.25° (this is the only motion ground truth available) about a vertical revolute joint. The stereo cameras were calibrated in order to compensate for lens distortion, and compute internal and external parameters of each camera [66]. The calibration principle used in the stereo image experiments was briefly described in the experimental part of Chapter 3. The calibrated internal and external parameters of the two stereo cameras are listed in Table 4.1 and Table 4.2 for the left and the right cameras, respectively.

The relative orientation between the stereo cameras (M and $\mathbf{B}$) was directly computed from the external parameters of the stereo cameras, as illustrated in Figure 3.10. Since the relative position of the two cameras were calibrated with respect to the world coordinate system through

$$\begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} = R_l \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \mathbf{T}_l \tag{4.73}$$

and

$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = R_r \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \mathbf{T}_r, \tag{4.74}$$

then we have

$$\begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} = R_l R_r^{\mathsf{T}} \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} - R_l R_r^{\mathsf{T}} \mathbf{T}_r + \mathbf{T}_l. \tag{4.75}$$

Table 4.1: Calibration data for the left f=8.5mm lens camera.

Focal length:	f_u	-639.10
	f_v	527.09
Center coordinate:	r_0	251.07
	c_0	260.01
Distortion parameter:	k_1	0.17645
	g_1	-0.00390
	g_2	0.00093
	g_3	0.01522
	g_4	0.00373
External parameters:		
Rotation angle (°)	θ	6.2704
Rotation axis	n_x	0.8791
	n_y	0.4119
	n_z	0.2396
Translation (mm)	t_1	-147.74
	t_2	-191.45
	t_3	390.83

Table 4.2: Calibration data for the right f=8.5mm lens camera.

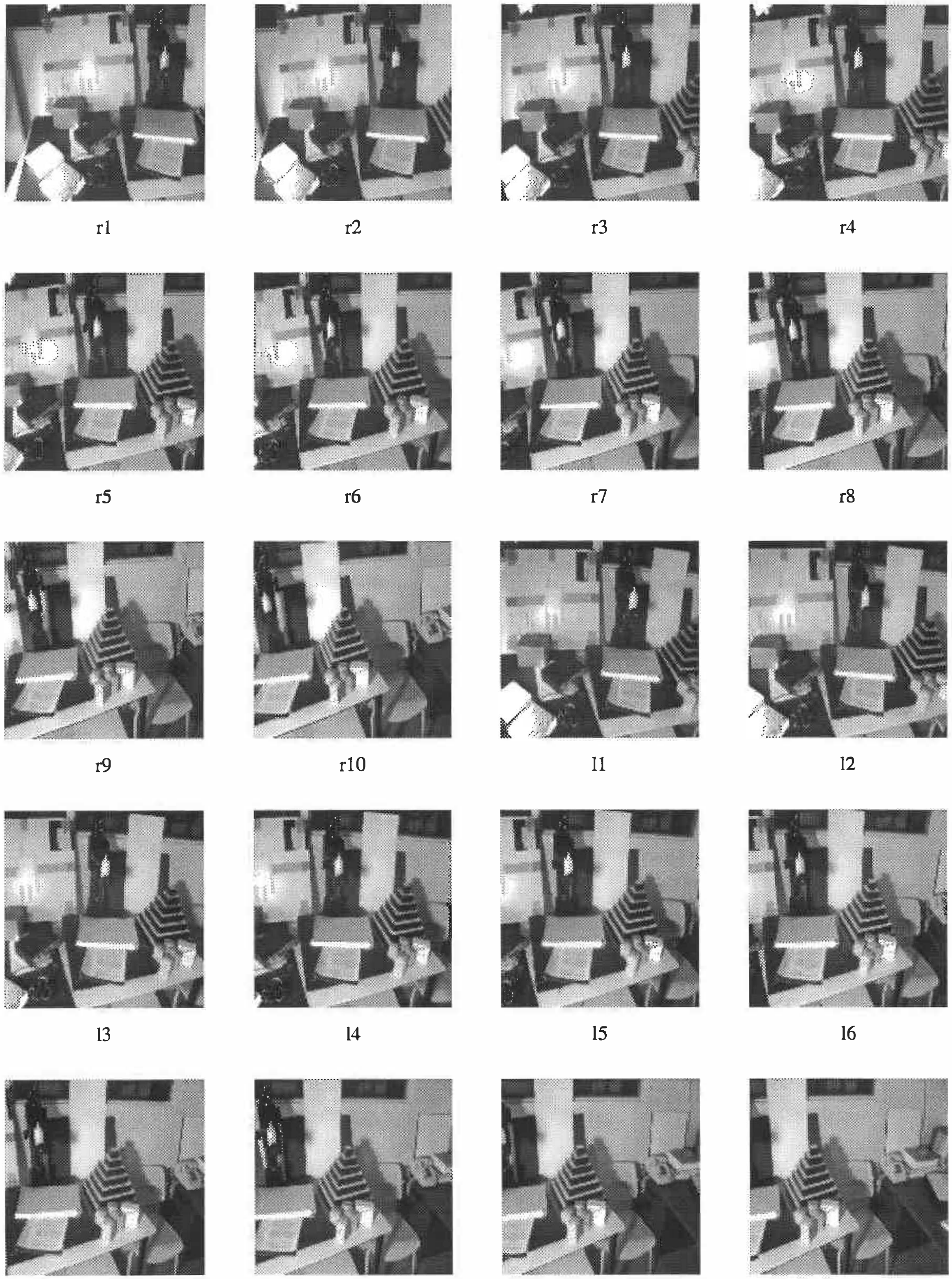
Focal length:	f_u	-639.11
	f_v	527.87
Center coordinate:	r_0	243.85
	c_0	261.79
distortion parameter:	k_1	0.17727
	g_1	-0.00440
	g_2	0.00081
	g_3	0.01768
	g_4	0.01098
External parameters:		
Rotation angle (°)	θ	6.8387
Rotation axis	n_x	-0.9249
	n_y	0.3654
	n_z	-0.1041
Translation (mm)	t_1	-150.28
	t_2	-200.69
	t_3	434.35

which means that $M = R_l R_r^\mathsf{T}$ and $\mathbf{B} = -R_l R_r^\mathsf{T} \mathbf{T}_r + \mathbf{T}_l$ in our experiments. In particular, the rotation angle was 11.86°, the rotation axis was $(0.99,\ 0.01,\ 0.04)^\mathsf{T}$ and the translation vector $\mathbf{B}$ was $(0.00,\ 0.09,\ 0.01)^\mathsf{T}$ (unit: meters). Due to the relative configuration of the stereo cameras and the depth range of the scene, the common field of view of the stereo cameras was about 362-pixel wide. The stereo image sequence consists of 10 consecutive stereo image pairs shown in Figure 4.9. The observed scene was approximately 1.2m away from the stereo system.

The algorithm described in [59] was used to compute stereo and temporal matchings. It is a general matching algorithm without any epipolar line constraints. One example of stereo matching in our experiments is shown as crosses in Figure 4.10. The image matching algorithm [59] establishes correspondence for each pixel. Since matching is more accurate where the image texture is abundant, the feature points used for motion computation consisted of a set of manually selected corner points in the first left image. These corners were then tracked automatically in consecutive left images by temporal matching, and the matching was refined by an intensity-based normalized cross-correlation process to eliminate the possible accumulated error over multiple frames. The depth map from the last stereo image pair is shown in Figure 4.11.

The feature points are classified into two categories: old and new. The old points are visible both in the current image and the previous one. The new points are visible in the current image but not in the previous one. A point is no longer considered as old if its neighborhood changed drastically due to motion. Only the old points (about 56 for each image pair) were included in the iterative optimization. The structure of the new points was estimated after motion.

In our experiment, two solutions were computed, the matrix-weighted close-form solution and the iterative optimal solution. These two solutions are very close but the iterative optimal solution is slightly better in terms of the root mean-squared error in



r1 r2 r3 r4

r5 r6 r7 r8

r9 r10 l1 l2

l3 l4 l5 l6

l7 l8 l9 l10

Figure 4.9: The stereo image sequence used in the experiments.

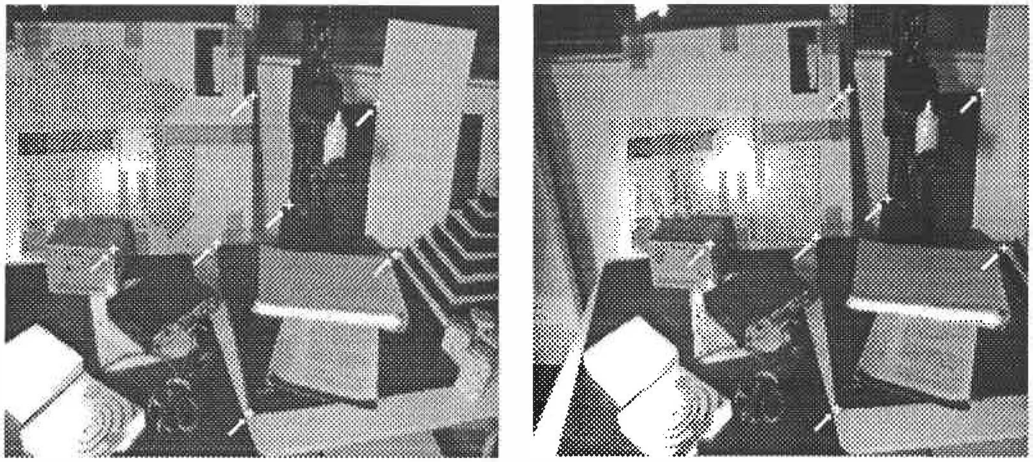


Figure 4.10: Matching for the first pair of stereo images shown as crosses.

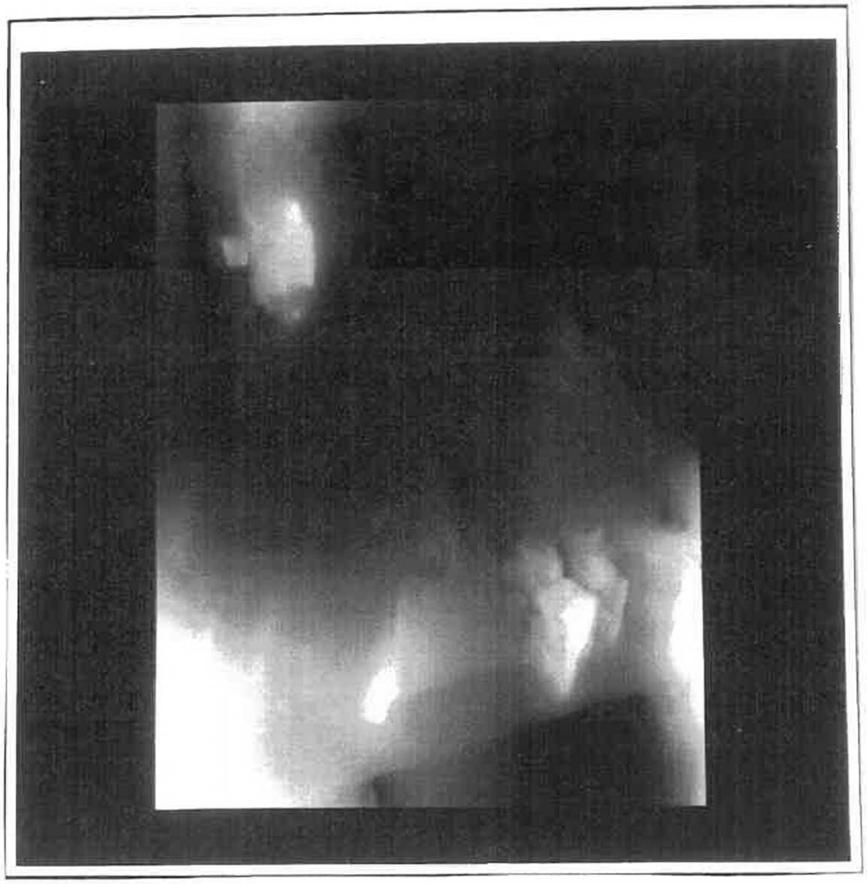


Figure 4.11: The depth map is shown as an intensity image from fifth stereo image pair.

the estimated structure. This indicates that the matching is roughly correct on the corner points. It also indicates that the noise level is much higher in the experiments than in the simulation, so that, in the real experiments, we cannot expect that the improvement brought about by the iterative optimization will be as remarkable as in the simulation. On the contrary, it shows that the performance of the matrix-weighted closed-form solution is rather stable in the processing of real stereo image sequences. The first and last interframe motions, are listed in Table 4.3. The estimated angle of interframe rotation from the optimization is listed in Table 4.4. As can be seen they are quite accurate. We have noted in the experiments that if point correspondences are well spread in some stereo image pairs, covering as large a common field of views as possible, then the corresponding motion and 3-D structure estimates are more accurate and stable.

In order to test the accuracy in the structure estimation, we manually measured the length l_i of approximately 40 lines in the scene as ground truth, like the lines shown in Figure 3.16. From the estimated length $\tilde{l}_i$, we can compute the root mean squared error of the estimated structure as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(l_i - \tilde{l}_i)^2},$$

and its mean error as

$$ME = \frac{1}{n}\sum_{i=1}^{n}\|(l_i - \tilde{l}_i)\|.$$

Table 4.4 illustrates for each view the RMSE and ME values for visible lines. It can be seen that the 3-D structure error increases with the frame number. One reason for this is that the corner points in the latter frames were not as well spread as in the first few frames, which may have resulted in a less accurate motion estimation (as the interframe translation ground truth was not available in the experiments, it is hard to assess the accuracy of the estimated motion for each frame). The second reason is that the number, the lengths and the directions of the visible lines varied from frame to frame, which also made the values of the structure error different. Finally, since corner points are usually located where the depth of the scene changes drastically, small mismatches may cause large errors in the estimated structure.

What we should mention here is that we used wide angle lenses. This implies that a pixel corresponds to a larger area in the scene than with a normal lens or a tele-lens. Also, a short baseline was used (about the distance between human eyes). If a tele-lens or wider baseline were employed, the structure error would decrease significantly.

It should be mentioned that if outliers exist in the data to be processed, method of robust statistics [72], [73] can then be invoked to detect the outliers and suppress their harmful effects.

4.5 Further assessment of the estimated structure

We now roughly estimate the structure uncertainty caused by each image pixel, in order to get a better assessment of the structure accuracy obtained in the experiments on monocular and stereo image sequences. Let us look at Figure 4.5, where b represents the length of the base-line between the two stereo cameras, and d represents the average depth of the scene concerned.

Let w and h represent respectively the width and the half height of the volume of the structure uncertainty induced by one image pixel. We can calculate the values of w and h from

$$\frac{h}{d} = \frac{w}{b} \tag{4.76}$$

and

$$\frac{w}{1} = \frac{d}{f}. \tag{4.77}$$

Table 4.3: Motion estimations resulting from the matrix-weighted linear algorithm and the nonlinear optimization (unit m).

motion parameters		linear	nonlinear
motion	$\mathbf{M}_{1,0}$		
translation	t_x	-0.003	-0.002
	t_y	-0.044	-0.047
	t_z	-0.007	-0.008
	length	0.045	0.048
rotation axis	N_x	0.872	0.858
	N_y	0.112	0.099
	N_z	-0.474	-0.502
rotation angle	$\theta(°)$	2.299	2.169
motion	$\mathbf{M}_{9,8}$		
translation	t_x	-0.001	0.001
	t_y	-0.045	-0.046
	t_z	-0.009	-0.010
	length	0.046	0.047
rotation axis	N_x	0.853	0.855
	N_y	0.072	0.035
	N_z	-0.516	-0.516
rotation angle	$\theta(°)$	2.237	2.192

Table 4.4: Estimated motion and structure. k: time index; θ: rotation angle (degree); $RMSE$: root mean square error (mm); ME: mean error (mm).

k	1	2	3	4	5	6	7	8	9
θ	2.16	2.18	2.08	2.33	2.20	2.30	2.19	2.22	2.19
$RMSE$	9.79	9.64	19.05	22.92	26.49	29.38	26.56	27.49	30.30
ME	6.64	5.74	9.67	13.25	17.45	20.40	19.39	20.33	23.43

In our experiments, d equals 1200mm, b equals 100mm and f equals 550. Then w and h are 2.2mm and 26mm, respectively, while $2h$ equals to 52mm. For one end of any horizontal line parallel to the baseline b, the uncertainty induced by image quantization is w, while for one end of any vertical line perpendicular to the baseline, the uncertainty is $2h$. Note that the vertical uncertainty $2h$ is much larger than the horizontal uncertainty w.

Since each line has two ends, and we have measured both horizontal and vertical lines in the scene, the estimated structure error from both monocular and stereo experiments shown in Table 3.3 and 4.4 fall into the allowable structure error range of the image pixel, which is roughly three times of the average uncertainty $\sqrt{w^2 + (2h)^2}$ (52.05mm) according to [5].

4.6 Summary

Our approach to motion and structure analysis through long image sequences is characterized by the following aspects:

(1) A closed-form matrix-weighted solution is used to obtain a reliable solution to interframe motion.

(2) To further improve the closed-form solution, an iterative optimization is formulated, using a space decomposition strategy to reduce the cost of computation and improve the numerical stability of the algorithm. The parameter space is decomposed in such a way that the structure of points is not included in the search space. This framework can be directly extended to a recursive estimation from long stereo image sequences.

(3) A recursive-batch approach is employed to process long stereo image sequences. Simulation and careful experiments have been carried out to investigate the performance of our methods. The experimental results have been compared to the available

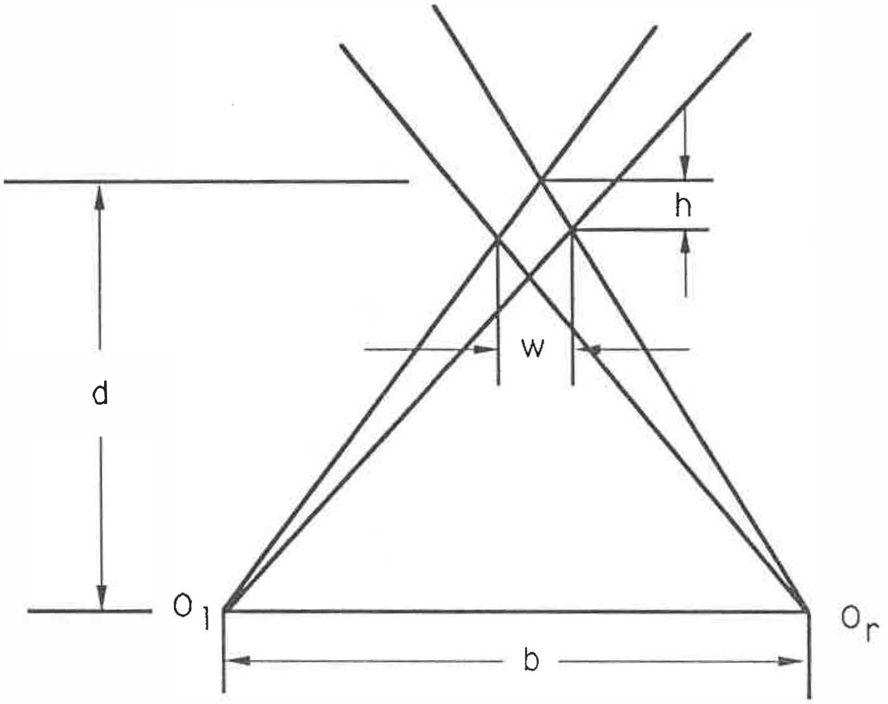


Figure 4.12: The uncertainty in the estimated coordinate of 3-D points, due to image quantization.

ground truth.

Chapter 5

Conclusions and discussions

5.1 Conclusions

The problem of dynamic motion and 3-D structure estimation from monocular and stereo image sequences has been studied in this dissertation. Our formulation considered the case of general motion, represented as a rotation matrix and a translation vector. Our approach falls into the category of the feature based approaches (using point correspondences for motion and structure computations).

It was seen that nonlinearity and the large dimensionality of the parameter space are two obstacles to the solution of the motion problem. In order to substantially improve the accuracy of the estimated motion and 3-D structure, and the stability as well as the efficiency of our optimization methods, two main measures have been undertaken. The first is applied in the data part, i.e., the varying reliabilities of the observations and estimates have been taken into account in the construction of the objective functions. The second is applied in the search procedure, i.e., the dimension of the search space in the nonlinear optimization is drastically reduced by exploiting the relationship between structure and motion, so that the search space only includes the 6 independent motion parameters. Since good initial motion solutions are provided by closed-form formulations for both the monocular and stereo cases, few iterations are required to converge to an optimal solution.

To process monocular and stereo image sequences a recursive-batch framework has been adopted, which combines the advantages of recursive and batch methods by preserving and updating the previous structure through time. In the case of multiple observation of some part of a scene, this method can provide more accurate estimates by exploiting the redundancy in the observation of the structure.

Experiment with real image sequences is commonly regarded as difficult in computer vision community. Great efforts have been made in this research to complete careful experiments with long stereo and monocular image sequences of a real world scene, including the camera calibration section, to investigate the performance of the proposed optimization methods. The estimates obtained have been compared to the motion and structure ground truth available.

The scale factor problem which is intrinsic in monocular image sequences was analyzed. It was shown that the scale factor associated with any two consecutive images in a monocular sequence is determined by the scale factor of the first two images.

5.2 Future work

Due to the time limitation, I have not made the selection of feature points from the image sequences fully automatic. As described in our experiments of the monocular and stereo image sequences, the initial selection of feature points was done manually. However, once the initial selection is performed, the tracking of the feature points in the successive images was completed automatically. Algorithms for fully automatic feature point selection which can select the most reliable feature points from the starting image and track them in the following images of the sequence is obviously an important and interesting topic for the work of future. Unlike other features, corner point correspondences do not suffer from the aperture problem in the matching process. Because of this advantage, corner points play an important role in the analysis of long image sequences (In fact, the manually selected feature points used in our experiments of the monocular and stereo image sequences consisted mostly of corner points.).

Although our recursive-batch framework set up one model for fusing redundant observations, new strategies for fusing multiple views in image sequences should be explored further. For example, the new strategy can be based on a new noise model other than that of Gaussian noise.

In reality, it is possible that there are several objects moving independently in different ways in the observed scene, segmentation algorithms which are able to handle general scenes consisting of both static and moving objects should be intensively studied. As many other motion and 3-D structure estimation approaches, we assume also that the necessary segmentation is done. In order to apply our approach to non-static environment, a robust segmentation is critical to the final results of the motion and 3-D structure estimates.

As more kinds of 3-D cameras using laser scanners developed, a reliable and high precision vision system should comprise both passive and active vision systems. To build such a vision system is an attractive project which can find many application fields.

Bibliography

[1] AGGARWAL, J. K. and NANDHAKUMAR, N., "On the computation of motion from sequences of images - a review", TR-88-2-47 April 1988, Computer and Vision Research Center, The University of Texas at Austin.

[2] ALOIMONOS, J., "Purposive and qualitative active vision", in *The proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, June 16-21, 1990, pp. 346-360.

[3] HUANG, T. S., "Modeling, analysis, and visualization of nonrigid object motion", in *The proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, June 16-21, 1990, pp. 361-364.

[4] JAIN, R. C. and BINFORD, T. O., "Dialogue, ignorance, myopia, and naivete in computer vision systems", *CVGIP: Image Understanding*, Vol. 53, No. 1, Jan, pp. 112-117, 1991.

[5] SNYDER, M. A., "Reply, a commentary on the paper by Jain and Binford", *CVGIP: Image Understanding*, Vol. 53, No. 1, Jan, pp. 118-119, 1991.

[6] ALOIMONOS, Y. and ROSENFELD, A., "Reply, a response to 'Ignorance, myopia, and naivete in computer vision systems' by R. C. Jain and T. O. Binford", *CVGIP: Image Understanding*, Vol. 53, No. 1, Jan, pp. 120-124, 1991.

[7] HUANG, T. S., "Reply, computer vision needs more experiments and applications", *CVGIP: Image Understanding*, Vol. 53, No. 1, Jan, pp. 125-126, 1991.

[8] HORN, B. K. P., "Rigid body motion from range image sequences", *CVGIP: Image Understanding*, Vol. 53, No. 1, Jan, pp. 120-124, 1991.

[9] HORN, B. K. P., *Robot Vision*, The MIT Press, Cambridge, Massachusetts, 1986.

[10] ROGERS, D. F. and ADAMS, J. A., *Mathematical elements for computer graphics*, McGraw-Hill, 1976.

[11] BROIDA, T. J. and CHELLAPPA, R., "Estimating the kinematics and structure of a rigid object from a sequence of monocular images", *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 13, No. 6, pp. 497-513, June. 1991.

[12] ALOIMONOS, J., WEISS, I. and BANDYOPADHYAY, A., "Active vision", in *Proc. 1st International Conference on Computer Vision*, London, England, June 8-11, 1987, pp. 35-54.

[13] ARUN, K. S., HUANG, T. S. and BLOSTEIN, S. D., "Least-squares fitting of two 3-D point sets", *IEEE Trans. Pattern Anal. Machine Intell.* vol. PAMI-9, No. 5, pp. 698-700, 1987.

[14] AYACHE, N. and FAUGERAS, O., "Building, registration, and fusing noisy visual maps", in *Proc. First International Conference on Computer Vision*, London, England, June 8-11, 1987, pp. 73-82.

[15] BANDYOPADHAY, A., CHANDRA, B. and BALLARD, D., "Egomotion using active vision", in *Proc. IEEE Conference Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, pp. 498-503.

[16] BROWN, K. M . and DENNIS, J. E., "Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation", *Numeriche Mathematik.* vol 18, 1972, pp. 289-297.

[17] CRAMÉR, H., *Mathematical Methods of Statistics.* Princeton Univ. Princeton, New Jersey, 1946.

[18] FAUGERAS, O. D. and HEBERT, M., "A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces", *Proc. 8th Int. Joint Conf. Artificial Intell.*, Karlsruhe, W. Germany, Aug. 1983 pp. 996-1002

[19] FITZGERALD, R. J., "Divergence of the Kalman Filter", *IEEE Trans. Automat. Contr.*, vol. AC-16, pp. 736-747, Dec. 1971.

[20] GELB, A. (ed), *Applied Optimal Estimation.* M.I.T. Press, Cambridge, MA, 1974.

[21] GIORDANO, A. A. and HSU, F. M., *Least Squares Estimation with Applications to Digital Signal Processing*, Wiley, New York, 1985.

[22] HALLAM, J., "Resolving observer motion by object tracking", in Proc. *Inter. Joint Conf. on Artificial Intelligence*, 1983.

[23] HUANG, T. S., BLOSTEIN, S. D. and MARGERUM, E. A., "Least-squares estimation of motion parameters from 3-D point correspondences", in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, pp. 198-201.

[24] HORN, B. K., "Closed-form solution of absolute orientation using unit quaternions", *Journal of the Optical Society of America*, A, Vol. 4, pp. 629-642, April 1987.

[25] KALMAN, R. E., *A new approach to linear filtering and prediction problems*, *J. Basic Eng.*, Series 82D, 1960, pp. 35-45.

[26] KIANG, S. M., CHOU, R. J. and AGGARWAL, J. K., "Triangulation errors in stereo algorithms", in *Proceedings IEEE Workshop on Computer Vision*, Miami Beach, FL, Nov. 1987, pp. 72-78.

[27] LEVENBERG, K., "A method for the solution of certain nonlinear problems in least squares", *Quart. Appl. Math.*, 2, 1944, pp. 164-168.

[28] MARQUARDT, D. W., "An algorithm for least squares estimation of nonlinear parameters", *SIAM J. Appl. Math.*, 11, 1963, pp. 431-441.

[29] MAYBECK, P. S., *Stochastic Models, Estimation, and Control*, Vol 1, Academic Press, New York, 1979.

[30] MAYBECK, P. S., *Stochastic Models, Estimation, and Control*, Vol 2, Academic Press, New York, 1982.

[31] MORAVEC, H. P., Obstacle avoidance and navigation in the real world by a seeing robot rover, Ph.D. dissertation, Stanford Univ. Stanford, CA, Sept. 1980.

[32] RAO, C. R., *Linear Statistical Inference and Its Applications*, 2nd Ed., Wiley, New York, 1973.

[33] SCHLEE, F. H., STANDISH, C. J. and TOTA, N. F., "Divergence in the Kalman Filter", *AIAA J*, vol. 5, pp. 1114-1120, June 1967.

[34] SHUSTER, M. D., "Approximate algorithms for fast optimal attitude computation", *Proc. AIAA Guidance and Control Specialist Conf.*, Palo Alto, Aug. 1978, pp. 88-95.

[35] SORENSON, H. W., *Parameter Estimation: Principles and Problems*, Marcel Dekker, New York, 1980.

[36] WENG, J., LIU, Y., HUANG, T. S. and AHUJA, N., "Estimating motion/structure from line correspondences: a robust linear algorithm and unique-

ness theorems", in Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June, 1988, pp. 387-392.

[37] SPETSAKIS, M. and ALOIMONOS, J., "Closed form solution to the structure from motion problem from line correspondences", in *Proc. Sixth AAAI National Conference on Artificial Intelligence*, Seattle, Washington, July 1987, pp. 738-743.

[38] SHARIAT, H. and PRICE, K. E., "Motion estimation from more than two frames", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, No. 5, pp. 417-434, May, 1990.

[39] SORENSON, H. W. (ed), *Kalman Filtering: Theory and Application*, IEEE Press, New York, 1985.

[40] YONG, G. and CHELLAPPA, R., "3-D motion estimation using noisy stereo images", in Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June, 1988, pp. 710-716.

[41] ZACKS, S., *The Theory of Statistical Inference*, Wiley, New York, 1971.

[42] LONGUET-HIGGINS, H. C., "A computer program for reconstructing a scene from two projections", *Nature*, Vol. 293, pp. 133-135, Sept. 1981.

[43] LONGUET-HIGGINS, H. C., "The reconstruction of a scene from two projections - configurations that defeat the 8-point algorithm", *Proceedings of First Conference on Artificial Intelligence Applications*, Denver, Colorado, USA, 1984, pp. 395-397.

[44] TSAI, R. Y. and HUANG, T. S., "Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, No. 1, pp. 13-27, 1984.

[45] ZHUANG, X., HUANG, T. S. and HARALICK, R. M., "Two-view motion analysis: a unified algorithm", *Journal of Optimal Society of America*, A, Vol. 3, No. 9, pp. 1492-1500, sept. 1986.

[46] WENG, J., AHUJA, N. and HUANG, T. S., "Error analysis of motion parameters estimation from image sequences", in *Proc. Inter. Conf. Computer Vision*, London, England, June, 1987.

[47] FAUGERAS, O. D., LUSTMAN, F., TOSCANI, G., "Motion and structure from motion from point and line matches", in *IEEE Proceedings of First International Conference on Computer Vision*, June 8-11, 1987, Londen England, pp. 25-34.

[48] WENG, J., HUANG, T. S. and AHUJA, N., "A two-step approach to optimal motion and structure estimation", in *Proceedings of IEEE Computer Society Workshop on Computer Vision*, Nov. 30-Dec. 2, 1987, Miami Beach, Florida, pp. 355-357.

[49] WENG, J., AHUJA, N. and HUANG, T. S., "Close-form + maximum likelihood: a robust approach to motion and structure estimation", in *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June 5-9, 1988, pp. 381-386.

[50] WENG, J., HUANG, T. S. and AHUJA, N., "Motion and Structure from two perspective views: algorithms, error analysis, and error estimation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 2, No. 5, pp. 451-476, May 1989.

[51] LUENBERGER, D. G., *Optimization by vector space methods*, John Wiley & Sons, 1969.

[52] MATTHIES, L. and SHAFER, S. A., "Error modeling in stereo Navigation", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, pp. 239-248, June

1987.

[53] KUMAR, R. V. R., TIRUMALAI, A. and JAIN, R. C., "A nonlinear optimization algorithm for the estimation of structure and motion parameters", *The Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 1989, pp. 136-143.

[54] WENG, J., AHUJA, N. and HUANG, T. S., "Optimal motion and structure estimation", in *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 1989, pp. 144-152.

[55] ANDERSON, B. D. O., MOORE, J. B., *Optimal filtering*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1979.

[56] SORENSON, H. W. (ed), *Least-squares estimation: from Gauss to Kalman, Kalman filtering: theory and application*, IEEE Press, 1985.

[57] TAYLOR, J. R., *An introduction to error analysis, the study of uncertainties in physical measurements*, University Science Books, Mill Valley, California, 1982.

[58] SPETSAKIS, M. E. and ALOIMONOS, J., "Optimal computing of structure from motion using point correspondence in two frames", *The Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 1989, pp. 449-453.

[59] WENG, J., AHUJA, N. and HUANG, T. S., "Two view matching", *The proceedings of Second International Conference on Computer Vision*, Innisbrook Resort, Tampa, Florida, USA, December 5-8, 1989, pp. 64-73.

[60] FLEET, D. J., JEPSON, A. D. and JENKIN, M. R. M., "Phase-based disparity measurement", *CVGIP: Image Understanding*, Vol. 53, No. 2, March, pp. 198-210, 1991.

[61] JENKIN, M. R. M., JEPSON, A. D. and TSOTSOS, J. K., "Techniques for disparity measurement", *CVGIP: Image Understanding*, Vol. 53, No. 1, January, pp. 14-30, 1991.

[62] WENG, J., "A theory of image matching", *Proceedings of third International Conference on Computer Vision*, Osaka, Japan, December 4-7, pp. 200-209, 1990.

[63] WENG, J., HUANG, T. S. and AHUJA, N., "Motion from images: image matching, parameter estimation and intrinsic stability", *The proceedings of CVPR*, 1989, pp. 359-366.

[64] FAUGERAS, O. D. and MAYBANK, S., "Motion from point matches: multiplicity of solutions", *The proceedings of CVPR*, 1989, pp.248-255.

[65] BROIDA, T. J. and CHELLAPPA, R., "Estimation of object motion parameters from noisy images", *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. PAMI-8, pp. 90-99, Jan. 1986.

[66] WENG, J., COHEN, P. and HERNIOU, M., Calibration of stereo cameras using a non-linear distortion model, in *The proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, June 16-21, 1990, pp. 246-253.

[67] ROSENFELD, A. and KAK, A. C., *Digital picture processing*, Academic Press, Vol. 2, 1982.

[68] NAHI, N. E., *Estimation theory and applications*, John Wiley & Sons, 1969.

[69] SORENSON, H. W., *Parameter estimation principles and problems*, Marcel Dekker, 1980.

[70] MENDEL, J. M., *Lessons in digital estimation theory*, Prentice-Hall, 1987.

[71] WENG, J. and COHEN, P., Fusion of stereo views: estimating structure and motion using a robust method, in *Proceedings of SPIE Symposium on Advanced in Intelligent Systems*, Boston, MA, Nov. 1990.

[72] WENG, J. and COHEN, P., Robust motion estimation using stereo vision, in *Proc. IEEE International Workshop on Robust Computer Vision*, Seattle, WA, Oct. 1990, pp. 367-388.

[73] TIRUMALAI, A. P., SCHUNCK, B. G. and JAIN, R. C., Dynamic stereo with self-calibration, *IEEE Proceedings of Third International Conference on Computer Vision*, Dec. 4-7, 1990, Osaka, Japan, pp. 466-470.

[74] CRAIG, J., *Introduction to robotics mechanics and control*, Addison-Wesley, 1989.

[75] STRANG, G., *Linear algebra and its applications*, Academic Press, 1980.

[76] CUI, N., WENG, J. and COHEN, P., "Extended structure and motion analysis from monocular image sequences", *Proc. of IEEE Third International Conference on Computer Vision*, Osaka, Japan, pp. 222-229, Dec. 1990.

[77] CUI, N., "Motion and 3-D structure estimation from long stereo image sequences", in *Proc. of IRIS PRECARN FIRST ANNUAL CONFERENCE*, Vancouver, pp. S06, June 12-14, 1991.

[78] AUDETTE, M., CUI, N., COHEN, P. and WENG, J., "An approach to the estimation of ground structure and aircraft motion from aerial image sequences", in *Proc. of Canadian Conference on Electrical and Computer Engineering*, Quebec City, Quebec, Canada, Sept. 25-27, 1991.

[79] CUI, N., WENG, J. and COHEN, P., "Dynamic stereo with visual integration for sensory motion and environmental reconstruction", in *Proc. International*

Advanced Robotics Program: Second Workshop on Sensor Fusion and Environmental Modeling, Oxford, UK, Sept. 1991.

[80] CUI, N., WENG, J. and COHEN, P.,"Motion and structure from long stereo image sequences", in *Proc. IEEE Workshop on Visual Motion*, Princeton, New Jersey, Oct. 7-10, 1991, pp. 75-80.

[81] CUI, N., WENG, J. and COHEN, P.,"Recursive-batch estimation of motion and structure from monocular image sequences", accepted for publication by *CVGIP: Image Understanding*, September 24, 1993.

[82] KORSTEN, M. J. and HOUKES, Z., "The estimation of geometry and motion of a surface from image sequences by means of linearization of a parametric model", *CVGIP*, 50, 1-28, 1990.

Appendix A

Least-squares fitting of a rotation matrix

Given two matrices $C = [\mathbf{C}_1,\ \mathbf{C}_2,\ \cdots,\ \mathbf{C}_n]$, and $D = [\mathbf{D}_1,\ \mathbf{D}_2,\ \cdots,\ \mathbf{D}_n]$, where $\mathbf{C}_i$ and $\mathbf{D}_i$ are 3-component vectors, we look for 3×3 matrix R which minimizes the norm:

$$\|RC - D\| = min \qquad subject\ to:\ R\ is\ a\ rotation\ matrix. \tag{A.1}$$

Define a 4 by 4 matrix B by

$$B = \sum_{i=1}^{3} B_i^\top B_i \tag{A.2}$$

where

$$B_i = \begin{bmatrix} 0 & (\mathbf{C}_i - \mathbf{D}_i)^\top \\ \mathbf{D}_i - \mathbf{C}_i & [\mathbf{D}_i + \mathbf{C}_i]_\times \end{bmatrix} \tag{A.3}$$

where we define a mapping $[.]_\times$ from a 3-dimensional vector to a 3 by 3 matrix by:

$$[(x_1,\ x_2,\ x_3)^\top]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{A.4}$$

Let $\mathbf{q} = (q_0,\ q_1,\ q_2,\ q_3)^\top$ be a unit eigenvector of B associated with the smallest eigenvalue. The solution of the rotation matrix R in (A.1) is

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_2 q_1 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_3 q_1 - q_0 q_2) & 2(q_3 q_2 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{A.5}$$

For proofs see [34], [18] or [50].

Appendix B

The closed-form two-view algorithm

The closed-form two-view algorithm outlined here is used to compute the initial motion and structure solutions in the case of monocular image sequences. We cite here the main computation steps for obtaining the initial solution. Further details about the algorithm can be found in [50].

Let the coordinate system be fixed on the camera with the origin coinciding with the optical axis and pointing forwards. Without loss of generality, we assume that the focal length is unity. Thus the image plane is located at $z = 1$. Visible objects are always located in front of the image plane, i.e., $z > 1$.

We define a mapping $[\cdot]_\times$ from a 3-D vector to a 3 by 3 matrix by:

$$[(x_1, x_2, x_3)^\top]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{B.1}$$

Using this mapping, we can express cross operation of two vectors by the matrix multiplication of a 3 by 3 matrix and a comumn matrix:

$$\mathbf{X} \times \mathbf{Y} = [\mathbf{X}]_\times \mathbf{Y}. \tag{B.2}$$

Consider a point P on the object which is visible at two time instants, with the following notations:

$$\mathbf{x} = (x, y, z)^{\mathsf{T}} \quad \textit{spatial vector of } P \textit{ at time } t_1$$

$$\mathbf{x}' = (x', y', z')^{\mathsf{T}} \quad \textit{spatial vector of } P \textit{ at time } t_2$$

$$\mathbf{X} = (u, v, 1)^{\mathsf{T}} = (x/z, y/z, 1)^{\mathsf{T}} \quad \textit{image vector of } P \textit{ at time } t_1$$

$$\mathbf{X}' = (u', v', 1)^{\mathsf{T}} = (x'/z', y'/z', 1)^{\mathsf{T}} \quad \textit{image vector of } P \textit{ at time } t_2$$

where (u,v) and (u',v') are the image coordinates of the point. Let R and $\mathbf{T}$ be the rotation matrix and the translational vector, respectively. The spatial points at the two time instants are related by

$$\mathbf{x}' = R\mathbf{x} + \mathbf{T}, \tag{B.3}$$

or for image vectors:

$$z'\mathbf{X}' = zR\mathbf{X} + \mathbf{T}. \tag{B.4}$$

If $\|\mathbf{T}\| \neq 0$, from the above equation we get

$$\frac{z'}{\|\mathbf{T}\|}\mathbf{X}' = \frac{z}{\|\mathbf{T}\|}R\mathbf{X} + \hat{\mathbf{T}} \tag{B.5}$$

where

$$\hat{\mathbf{T}} = \frac{\mathbf{T}}{\|\mathbf{T}\|}. \tag{B.6}$$

The algorithm is as follows:

(i) *Solve for the intermediate unknown vector* h:

Let $\mathbf{X}_i = (u_i, v_i, 1)^{\mathsf{T}}$, $\mathbf{X'}_i = (u'_i, v'_i, 1)^{\mathsf{T}}$, $i = 1, 2, \cdots, n$, be the corresponding image vectors of n ($n \geq 8$) points. Let

$$A = \begin{bmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_nu_n' & u_nv_n' & u_n & v_nu_n' & v_nv_n' & v_n & u_n' & v_n' & 1 \end{bmatrix} \tag{B.7}$$

$$\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)^{\mathsf{T}}. \tag{B.8}$$

We solve for unit vector $\mathbf{h}$ such that

$$||A\mathbf{h}|| = min. \tag{B.9}$$

The solution of $\mathbf{h}$ is the unit eigenvector of $A^{\mathsf{T}}A$ associated with the smallest eigenvalue. We then define matrix $\mathbf{E}$ as :

$$\mathbf{E} = [\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3] = \sqrt{2} \begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix}. \tag{B.10}$$

(ii) *Determine a unit vector* $\mathbf{T}_s$ *with* $\hat{\mathbf{T}} = \pm\mathbf{T}_s$

Using standard least-squares method we solve for the unit vector $\mathbf{T}_s$ such that

$$||\mathbf{E}^{\mathsf{T}}\mathbf{T}_s|| = min \tag{B.11}$$

If

$$\sum_i (\mathbf{T}_s \times \mathbf{X}_i') \cdot (\mathbf{E}\mathbf{X}_i) < 0, \tag{B.12}$$

then $\mathbf{T}_s$ is replaced by $-\mathbf{T}_s$. The summation in (B.12) is over several values of i's to reduce the instability due to noise (usually three or four values of i will suffice).

(iii) *Determine rotation matrix R*

Without noise we have

$$\mathbf{E} = [\mathbf{T}_s]_\times R \tag{B.13}$$

or

$$R^\top [-\mathbf{T}_s]_\times = \mathbf{E}^\top. \tag{B.14}$$

In the presence of noise, we find the rotation matrix R such that

$$\|R^\top [-\mathbf{T}_s]_\times - \mathbf{E}^\top\| = min, \quad subject\ to: \ R\ is\ a\ rotation\ matrix. \tag{B.15}$$

Alternatively, we can find R directly. Let

$$\mathbf{W} = [\mathbf{E}_1 \times \mathbf{T}_s + \mathbf{E}_2 \times \mathbf{E}_3 \quad \mathbf{E}_2 \times \mathbf{T}_s + \mathbf{E}_3 \times \mathbf{E}_1 \quad \mathbf{E}_3 \times \mathbf{T}_s + \mathbf{E}_1 \times \mathbf{E}_2], \tag{B.16}$$

without noise we have $R = \mathbf{W}$. In the presence of noise, we find rotation matrices R such that

$$\|R - \mathbf{W}\| = min, \quad subject\ to: \ R\ is\ a\ rotation\ matrix. \tag{B.17}$$

We can use either (B.16) or (B.17) to find R. They both have the form

$$\|RC - D\| = min, \quad subject\ to: \ R\ is\ a\ rotation\ matrix \tag{B.18}$$

where $C = [C_1 \quad C_2 \quad C_3]$, $D = [D_1 \quad D_2 \quad D_3]$. Then we can use the method of "least-squares fitting of a rotation matrix" to compute the rotation matrix R.

(iv) *Check* $\mathbf{T} = 0$. *If* $\mathbf{T} \neq 0$, *determine* $\hat{\mathbf{T}} = \mathbf{T}_s$ *or* $\hat{\mathbf{T}} = -\mathbf{T}_s$

Let α be a small threshold. If $\frac{\|\mathbf{X}'_i \times R\mathbf{X}_i\|}{\|\mathbf{X}'_i\| \, \|\mathbf{X}_i\|} \leq \alpha$ for all $1 \leq i \leq n$, then report $\mathbf{T} \simeq 0$. Otherwise determine the sign for $\hat{\mathbf{T}}$: if

$$\sum_i (\mathbf{T}_s \times \mathbf{X}'_s) \cdot (\mathbf{X}'_i \times R\mathbf{X}_i) > 0, \tag{B.19}$$

then $\hat{\mathbf{T}} = \mathbf{T}_s$. Otherwise $\hat{\mathbf{T}} = -\mathbf{T}_s$. Similar to (B.12), summation (B.19) is over several value of i.

(v) *If* $\mathbf{T}$ *does not vanish, estimate the relative depths*

For i, $1 \le i \le n$, find the relative depth

$$\mathbf{Z}_i = (\frac{z_i'}{\|\mathbf{T}\|}, \frac{z_i}{\|\mathbf{T}\|})^{\mathsf{T}} \tag{B.20}$$

such that

$$\|[\mathbf{X}_i' \quad - R\mathbf{X}_i]\mathbf{Z}_i - \hat{\mathbf{T}}]\| = min \tag{B.21}$$

using standard least-squares method for linear equations.

Appendix C

Matrix-weighted centroid-coincidence Theorem

MWCC Theorem [71]. If R^* and $\mathbf{T}^*$ minimize (4.16) with the weighting matrix Γ_i^{-1} not depending on either R or $\mathbf{T}$, then the matrix-weighted centroids of $\{\hat{\mathbf{x}}_i'\}$ and $\{R^*\hat{\mathbf{x}}_i + \mathbf{T}^*\}$ must coincide:

$$\sum_{i=1}^{n} \Gamma_i^{-1}\{R\hat{\mathbf{x}}_i + \mathbf{T}^*\} = \sum_{i=1}^{n} \Gamma_i^{-1}\hat{\mathbf{x}}_i'. \tag{C.1}$$

Proof. Let

$$\hat{\mathbf{x}}_i'' = (\hat{x}_{i1}'',\ \hat{x}_{i2}'',\ \hat{x}_{i3}'')^{\mathsf{T}} = R\hat{\mathbf{x}}_i + \mathbf{T} \quad i = 1,\ 2,\ \cdots,\ n. \tag{C.2}$$

Minimizing (4.16) is equivalent to the following: Given $\{\hat{\mathbf{x}}_i\}$ and $\{\hat{\mathbf{x}}_i'\}$, $i = 1,\ 2,\ \cdots,\ n$, determine $\{\hat{\mathbf{x}}_i''\}$ to minimize

$$\sum_{i=1}^{n} \{\hat{\mathbf{x}}_i'' - \hat{\mathbf{x}}_i'\}^{\mathsf{T}} \Gamma_i^{-1} \{\hat{\mathbf{x}}_i'' - \hat{\mathbf{x}}_i'\} \tag{C.3}$$

subject to the rigidity constraints

$$\|\hat{\mathbf{x}}_i'' - \hat{\mathbf{x}}_j''\|^2 = \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2 \quad 1 \le i \le n,\ \ 1 \le j \le n. \tag{C.4}$$

As a necessary condition of this minimization problem with equality constraints, the partial derivatives of the corresponding Lagrangian

$$\mathbf{L} = \sum_{i=1}^{n} \{\hat{\mathbf{x}}_i'' - \hat{\mathbf{x}}_i'\}^{\mathsf{T}} \Gamma_i^{-1} \{\hat{\mathbf{x}}_i'' - \hat{\mathbf{x}}_i'\} + \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_{ij}\{\|\hat{\mathbf{x}}_i'' - \hat{\mathbf{x}}_j''\|^2 - \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2\} \tag{C.5}$$

where $\lambda_{ij} = \lambda_{ji}$, must vanish. Differentiating $\mathbf{L}$ with respect to $\hat{x}''_{kl}$, $1 \leq k \leq n$ and $1 \leq l \leq 3$, yields,

$$0 = \frac{\partial \mathbf{L}}{\partial \hat{x}''_{kl}} = 2\Gamma^{-1}_{kl}\{\hat{\mathbf{x}}''_k - \hat{\mathbf{x}}'_k\} + 2\sum_{j=1}^{n} \lambda_{kj} 2\{\hat{x}''_{kl} - \hat{x}''_{jl}\}, \tag{C.6}$$

where Γ^{-1}_{kl} denotes the l-th row of the matrix Γ^{-1}_{k}. Since Γ^{-1}_{k} in (4.20) does not depend on either R or $\mathbf{T}$, it does not depend on $\hat{\mathbf{x}}''_k$. Summing up (C.6) for $k = 1$ to n gives

$$\sum_{i=1}^{n} \Gamma^{-1}_{kl}\{\hat{\mathbf{x}}''_i - \hat{\mathbf{x}}'_i\} = 0 \tag{C.7}$$

since $\sum_{k=1}^{n} \sum_{j=1}^{n} \lambda_{kj}\{\hat{x}''_{kl} - \hat{x}''_{jl}\} = 0$ for any integer l with $1 \leq l \leq 3$. According to the definition of $\hat{x}''_k$, (C.1) directly follows from (C.7). This completes the proof.

Appendix D

The invariance of the best linear unbiased estimator under changes of scale

Theorem. $\hat{\mathbf{m}}_{BLU}$ is invariant under changes of scale [70].

Proof. Assume that observers $O1$ and $O2$ are observing a process; but, observer $O1$ reads the measurements in one set of units and $O2$ in another. Let S be a symmetric matrix of scale factors relating $O1$ to $O2$, and $\mathbf{y}_{O1}(k)$ and $\mathbf{y}_{O2}(k)$ denote the total measurement vectors of $O1$ and $O2$, respectively. Then

$$\mathbf{y}_{O2}(k) = \mathbf{A}_{O2}(k)\mathbf{m} + \delta_{\mathbf{y}_{O2}} = \mathbf{S}\mathbf{y}_{O1}(k) = \mathbf{S}\mathbf{A}_{O1}(k)\mathbf{m} + \mathbf{S}\delta_{\mathbf{y}_{O1}} \tag{D.1}$$

which means that

$$\mathbf{A}_{O2}(k) = \mathbf{S}\mathbf{A}_{O1}(k) \tag{D.2}$$

$$\delta_{\mathbf{y}_{O2}} = \mathbf{S}\delta_{\mathbf{y}_{O1}} \tag{D.3}$$

and

$$\Gamma_{\mathbf{y}_{O2}} = \mathbf{S}\Gamma_{\mathbf{y}_{O1}}\mathbf{S}^{\top} = \mathbf{S}\Gamma_{\mathbf{y}_{O1}}\mathbf{S} \tag{D.4}$$

Let $\hat{\mathbf{m}}_{O1,BLU}(k)$ and $\hat{\mathbf{m}}_{O2,BLU}(k)$ denote the best linear unbiased estimators associated with observers $O1$ and $O2$, respectively; then,

$$\hat{\mathbf{m}}_{O2,BLU}(k) \quad = \quad \left[\mathbf{A}_{O2}^{\top}(k)\Gamma_{\mathbf{y}_{O2}}^{-1}\mathbf{A}_{O2}^{\top}(k)\right]^{-1}\mathbf{A}_{O2}^{\top}(k)\Gamma_{\mathbf{y}_{O2}}^{-1}\mathbf{y}_{O2}(k) \tag{D.5}$$

$$\begin{aligned}
&= \left[\mathbf{A}_{O1}^{\mathsf{T}}(k)\mathbf{S}(\mathbf{S}\Gamma_{\mathbf{y}_{O1}}\mathbf{S})^{-1}\mathbf{S}\mathbf{A}_{O1}(k)\right]^{-1}\mathbf{A}_{O1}^{\mathsf{T}}(k)\mathbf{S}(\mathbf{S}\Gamma_{\mathbf{y}_{O1}}\mathbf{S})^{-1}\mathbf{S}\mathbf{y}_{O1}(k) \\
&= \left[\mathbf{A}_{O1}^{\mathsf{T}}(k)\Gamma_{\mathbf{y}_{O1}}^{-1}\mathbf{A}_{O1}^{\mathsf{T}}(k)\right]^{-1}\mathbf{A}_{O1}^{\mathsf{T}}(k)\Gamma_{\mathbf{y}_{O1}}^{-1}\mathbf{y}_{O1}(k) \\
&= \hat{\mathbf{m}}_{O1,BLU}(k).
\end{aligned}$$

Appendix E

The covariance recursive form of the structure estimator

Matrix Inversion Lemma ([55], [69], [70]),

$$(\mathbf{\Sigma}^{-1} + \mathbf{H}\mathbf{R}^{-1}\mathbf{H}^{\top})^{-1} = \mathbf{\Sigma} - \mathbf{\Sigma}\mathbf{H}(\mathbf{H}^{\top}\mathbf{\Sigma}\mathbf{H} + \mathbf{R})^{-1}\mathbf{H}^{\top}\mathbf{\Sigma} \tag{E.1}$$

where $\mathbf{\Sigma}$ is a $n \times n$ matrix, $\mathbf{R}$ is a $p \times p$ matrix, and $\mathbf{H}$ is a $n \times p$ matrix. All matrix inverses are assumed to exist.

By using the Matrix Inversion Lemma, we can derive the covariance recursive form of the structure estimator in (3.30) from (3.28). Since

$$\begin{aligned}(\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{A})^{-1} &= (\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}^{-1} + \mathbf{A}_{k,i}^{\top}\mathbf{\Gamma}_{\hat{\mathbf{u}}_{k,i}}^{-1}\mathbf{A}_{k,i})^{-1} \\ &= \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}\mathbf{A}_{k,i}^{\top}(\mathbf{A}_{k,i}\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}\mathbf{A}_{k,i}^{\top} + \mathbf{\Gamma}_{\hat{\mathbf{u}}_{k,i}})^{-1}\mathbf{\Gamma}_{\hat{\mathbf{u}}_{k,i}}(\mathbf{A}_{k,i}^{\top})^{-1},\end{aligned} \tag{E.2}$$

and

$$\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{B} = \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}^{-1}\mathbf{b}_1 + \mathbf{A}_{k,i}^{\top}\mathbf{\Gamma}_{\hat{\mathbf{u}}_{k,i}}^{-1}\mathbf{b}_2, \tag{E.3}$$

we can thus rewrite (3.28) as the following:

$$\begin{aligned}\mathbf{x}_{k,i}^{*} &= (\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{A})^{-1}\mathbf{A}^{\top}\mathbf{\Gamma}^{-1}\mathbf{B} \\ &= [\mathbf{I} - \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}\mathbf{A}_{k,i}^{\top}(\mathbf{A}_{k,i}\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}\mathbf{A}_{k,i}^{\top} + \mathbf{\Gamma}_{\hat{\mathbf{u}}_{k,i}})^{-1}\mathbf{A}_{k,i}]\mathbf{b}_1 \\ &\quad + \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}\mathbf{A}_{k,i}^{\top}(\mathbf{A}_{k,i}\mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}}\mathbf{A}_{k,i}^{\top} + \mathbf{\Gamma}_{\hat{\mathbf{u}}_{k,i}})^{-1}\mathbf{b}_2.\end{aligned} \tag{E.4}$$

Substituting $\mathbf{b}_1$ into the above equation, we have the desired covariance recursive form of the structure estimator for (3.28):

$$\mathbf{x}^*_{k,i} = (\mathbf{A}^\top \mathbf{\Gamma}^{-1} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{\Gamma}^{-1} \mathbf{B} = \hat{\mathbf{x}}_{k,i} + \mathbf{G}_{k,i}(\mathbf{b}_2 - \mathbf{A}_{k,i}\hat{\mathbf{x}}_{k,i}) \tag{E.5}$$

where the gain matrix is

$$\mathbf{G}_{k,i} = \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} \mathbf{A}^\top_{k,i} (\mathbf{A}_{k,i} \mathbf{\Gamma}_{\hat{\mathbf{x}}_{k,i}} \mathbf{A}^\top_{k,i} + \mathbf{\Gamma}_{\tilde{\mathbf{u}}_{k,i}})^{-1}.$$

Appendix F

The proof of positive definiteness of the structure covariance matrix

$\mathbf{\Gamma}_{\mathbf{x}^*_{0,i}}$ in (3.46) can be expressed in the following form:

$$\mathbf{\Gamma}_{\mathbf{x}^*_{0,i}} = c_1 \mathbf{u}^*_i (\mathbf{u}^*_i)^\top + c_2 \mathbf{D} \tag{F.1}$$

where $c_1 = \sigma^2_{z^*_i}$, $c_2 = z^{*2}_i$ and

$$\mathbf{D} = \begin{bmatrix} \sigma^2_u & 0 & 0 \\ 0 & \sigma^2_u & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{F.2}$$

We prove now that

$$\begin{aligned} \mathbf{x}^\top \mathbf{\Gamma}_{\mathbf{x}^*_{0,i}} \mathbf{x} &= c_1 \mathbf{x}^\top \mathbf{u}^*_i (\mathbf{u}^*_i)^\top \mathbf{x} + c_2 \mathbf{x}^\top \mathbf{D} \mathbf{x} \\ &= c_1 (\mathbf{u}^*_i \cdot \mathbf{x})^2 + c_2 \mathbf{x}^\top \mathbf{D} \mathbf{x} \\ &> 0, \end{aligned} \tag{F.3}$$

for all nonzero vectors $\mathbf{x} = (x_1\, x_2\, x_3)^\top$.

Note that $\mathbf{u}^*_i \neq \mathbf{0}$, since its third component is a constant 1. The second term in (F.3) can be simplified as

$$c_2 \mathbf{x}^\top \mathbf{D} \mathbf{x} = c_2 (x_1^2 \sigma^2_u + x_2^2 \sigma^2_u). \tag{F.4}$$

For any nonzero vector $\mathbf{x} = (x_1\, x_2\, x_3)^\top$, if $x_3 = 0$, then $c_2\mathbf{x}^\top\mathbf{D}\mathbf{x} \neq 0$; if $x_1 = 0$ and $x_2 = 0$, then $c_1(\mathbf{u}_i^* \cdot \mathbf{x})^2 \neq 0$. So we conclude that $\mathbf{\Gamma}_{\mathbf{x}_{0,i}^*}$ is always positive definite.

Appendix G

The interframe motion estimated from the monocular experiment

All the interframe motion estimates obtained from the monocular experiment are shown in the following figure and tables.

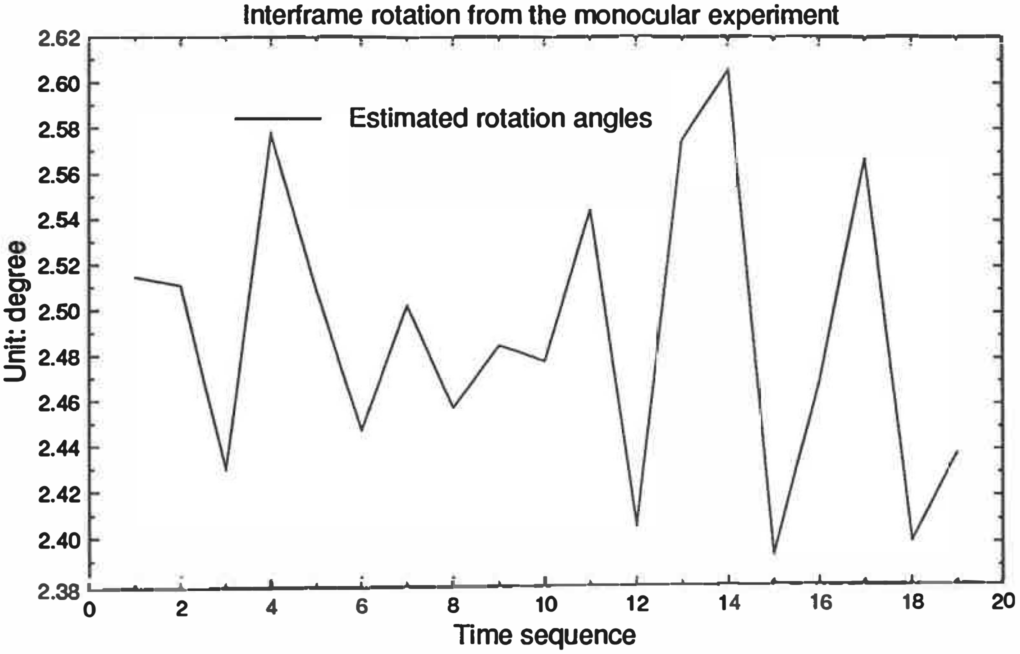


Figure G.1: A graphic display of the estimated values of the interframe rotation angles of the monocular experiment. The ground truth of the interframe rotation angle is 2.25°.

Table G.1: Motion estimations resulting from the linear algorithm and the nonlinear optimization.

motion parameters		linear algorithm	nonlinear optimization
motion	$\mathbf{M}_{1,0}$		
translation	t_x	-0.004081	-0.000615
(scaled)	t_y	-0.140111	-0.192057
	t_z	-0.116833	-0.044210
	length	0.182476	0.197081
rotation axis	N_x	0.927024	0.896551
	N_y	0.127492	0.049395
	N_z	-0.352664	-0.440179
rotation angle	$\theta(^\circ)$	2.955706	2.514551
motion	$\mathbf{M}_{2,1}$		
translation	t_x	-0.012594	-0.005608
(scaled)	t_y	-0.040259	-0.190828
	t_z	-0.116480	-0.042419
	length	0.123883	0.195567
rotation axis	N_x	0.987225	0.894357
	N_y	0.088764	0.057969
	N_z	-0.132316	-0.443582
rotation angle	$\theta(^\circ)$	3.777868	2.510903
motion	$\mathbf{M}_{3,2}$		
translation	t_x	-0.010199	-0.008147
(scaled)	t_y	-0.031638	-0.197683
	t_z	-0.126912	-0.045876
	length	0.131193	0.203100
rotation axis	N_x	0.987154	0.885395
	N_y	0.064054	0.068052
	N_z	-0.146370	-0.459830
rotation angle	$\theta(^\circ)$	3.870644	2.430236
motion	$\mathbf{M}_{4,3}$		
translation	t_x	-0.010558	-0.011871
(scaled)	t_y	-0.060952	-0.194237
	t_z	-0.123074	-0.030931
	length	0.137746	0.197042
rotation axis	N_x	0.976017	0.894446
	N_y	0.099821	0.105399
	N_z	-0.193462	-0.434578
rotation angle	$\theta(^\circ)$	3.507770	2.577579

Table G.2: Motion estimations resulting from the linear algorithm and the nonlinear optimization.

motion parameters		linear algorithm	nonlinear optimization
motion	$\mathbf{M}_{5,4}$		
translation	t_x	-0.004424	-0.000455
(scaled)	t_y	-0.090996	-0.196497
	t_z	-0.179526	-0.036368
	length	0.201319	0.199835
rotation axis	N_x	0.953410	0.888732
	N_y	0.122819	0.047206
	N_z	-0.275545	-0.455990
rotation angle	$\theta(°)$	3.214753	2.509122
motion	$\mathbf{M}_{6,5}$		
translation	t_x	-0.002598	-0.013570
(scaled)	t_y	0.034913	-0.192642
	t_z	-0.125652	-0.043876
	length	0.130438	0.198041
rotation axis	N_x	0.985159	0.905226
	N_y	0.002010	0.094617
	N_z	-0.171632	-0.414262
rotation angle	$\theta(°)$	4.341894	2.447395
motion	$\mathbf{M}_{7,6}$		
translation	t_x	-0.000212	-0.005927
(scaled)	t_y	-0.128437	-0.197732
	t_z	-0.258612	-0.038415
	length	0.288749	0.201516
rotation axis	N_x	0.936065	0.892702
	N_y	0.115364	0.068716
	N_z	-0.332375	-0.445378
rotation angle	$\theta(°)$	3.161157	2.502223
motion	$\mathbf{M}_{8,7}$		
translation	t_x	0.006445	0.000301
(scaled)	t_y	-0.124258	-0.197155
	t_z	-0.205979	-0.039485
	length	0.240642	0.201070
rotation axis	N_x	0.924629	0.888798
	N_y	0.095265	0.046924
	N_z	-0.368763	-0.455891
rotation angle	$\theta(°)$	2.936972	2.457414

Table G.3: Motion estimations resulting from the linear algorithm and the nonlinear optimization.

motion parameters		linear algorithm	nonlinear optimization
motion	$\mathbf{M}_{9,8}$		
translation	t_x	0.005957	-0.006870
(scaled)	t_y	-0.072216	-0.191468
	t_z	-0.214220	-0.045614
	length	0.226143	0.196946
rotation axis	N_x	0.958225	0.900382
	N_y	0.083626	0.066008
	N_z	-0.273516	-0.430064
rotation angle	$\theta(°)$	3.238676	2.484936
motion	$\mathbf{M}_{10,9}$		
translation	t_x	0.009495	-0.007293
(scaled)	t_y	-0.017878	-0.195638
	t_z	-0.149888	-0.033243
	length	0.151249	0.198576
rotation axis	N_x	0.959894	0.902316
	N_y	0.057237	0.061092
	N_z	-0.274460	-0.426725
rotation angle	$\theta(°)$	3.590450	2.477988
motion	$\mathbf{M}_{11,10}$		
translation	t_x	0.010524	-0.002607
(scaled)	t_y	-0.006316	-0.189613
	t_z	-0.128659	-0.039179
	length	0.129243	0.193636
rotation axis	N_x	0.965102	0.892462
	N_y	0.065311	0.069414
	N_z	-0.253598	-0.445750
rotation angle	$\theta(°)$	3.647439	2.544469
motion	$\mathbf{M}_{12,11}$		
translation	t_x	0.019834	-0.008744
(scaled)	t_y	-0.018138	-0.194748
	t_z	-0.135338	-0.035821
	length	0.137981	0.198208
rotation axis	N_x	0.967331	0.902118
	N_y	0.045610	0.061712
	N_z	-0.249379	-0.427054
rotation angle	$\theta(°)$	3.514187	2.406422

Table G.4: Motion estimations resulting from the linear algorithm and the nonlinear optimization.

motion parameters		linear algorithm	nonlinear optimization
motion	$\mathbf{M}_{13,12}$		
translation	t_x	0.022363	-0.001111
(scaled)	t_y	-0.087605	-0.192015
	t_z	-0.164568	-0.042067
	length	0.187769	0.196572
rotation axis	N_x	0.949348	0.888060
	N_y	0.046863	0.046597
	N_z	-0.310714	-0.457359
rotation angle	$\theta(°)$	3.244757	2.574573
motion	$\mathbf{M}_{14,13}$		
translation	t_x	0.022448	-0.009339
(scaled)	t_y	0.014461	-0.190505
	t_z	-0.146642	-0.047326
	length	0.149053	0.196517
rotation axis	N_x	0.973143	0.905118
	N_y	0.022836	0.093418
	N_z	-0.229065	-0.414769
rotation angle	$\theta(°)$	3.793730	2.605713
motion	$\mathbf{M}_{15,14}$		
translation	t_x	0.011349	-0.007113
(scaled)	t_y	-0.045599	-0.196871
	t_z	-0.103114	-0.045662
	length	0.113316	0.202223
rotation axis	N_x	0.965373	0.888405
	N_y	0.027323	0.057948
	N_z	-0.259438	-0.455389
rotation angle	$\theta(°)$	3.279208	2.394105
motion	$\mathbf{M}_{16,15}$		
translation	t_x	0.012836	-0.005137
(scaled)	t_y	-0.175316	-0.208194
	t_z	-0.131395	-0.035274
	length	0.219466	0.211224
rotation axis	N_x	0.891003	0.886930
	N_y	0.064878	0.062890
	N_z	-0.449338	-0.457603
rotation angle	$\theta(°)$	2.859024	2.469579

Table G.5: Motion estimations resulting from the linear algorithm and the nonlinear optimization.

motion parameters		linear algorithm	nonlinear optimization
motion	$\mathbf{M}_{17,16}$		
translation	t_x	0.022908	-0.010166
(scaled)	t_y	-0.152211	-0.203621
	t_z	-0.149965	-0.041988
	length	0.214901	0.208154
rotation axis	N_x	0.934230	0.882817
	N_y	0.075973	0.096769
	N_z	-0.348486	-0.459640
rotation angle	$\theta(^\circ)$	2.902700	2.566996
motion	$\mathbf{M}_{18,17}$		
translation	t_x	0.019225	-0.010592
(scaled)	t_y	-0.073187	-0.200072
	t_z	-0.136523	-0.042506
	length	0.156091	0.204812
rotation axis	N_x	0.959765	0.890507
	N_y	0.069962	0.060714
	N_z	-0.271949	-0.450900
rotation angle	$\theta(^\circ)$	3.484531	2.400253
motion	$\mathbf{M}_{19,18}$		
translation	t_x	-0.070523	0.000453
(scaled)	t_y	-0.202018	-0.206084
	t_z	0.438972	-0.041304
	length	0.488345	0.210183
rotation axis	N_x	0.950502	0.890109
	N_y	-0.062418	0.058372
	N_z	-0.304383	-0.451993
rotation angle	$\theta(^\circ)$	3.422166	2.438417