THREE DIMENSIONAL MOTION PARAMETER ESTIMATION FROM IMAGE SEQUENCES: A FEATURE BASED APPROACH.

by

Sundararaj Ragupathi, B.E., M.E.

July 1990.

A thesis submitted for the degree of
Doctor of Philosophy of the University of London
and for the Diploma of Membership of the Imperial College.

Digital Communication Section,
Electrical Engineering Department,
Imperial College of Science , Technology and Medicine,
London, SW7 2BT.

ABSTRACT

Image sequence analysis has become an important research area for its wide range of applications. Some of the applications include tracking and identification of objects , dynamic robot vision, and medical image processing. The key issue in most of the applications is motion estimation. Earlier attempts at motion estimation was made in two dimension (2-D) only. This motion estimation was used for motion compensation in the context of bandwidth compression. However, in a general scene, the object motion is a three dimensional (3-D) one.

In this thesis, an attempt is made to estimate the 3-D motion parameters from a given image sequence. Any 3-D motion can be decomposed into a translation in 3-D space followed by a rotation in that space. Our goal is to use the available 2-D images to estimate the 3-D motion parameters.

In general, there are two approaches: (a) optical flow based approach and (b) feature based approach. We , in our work use a feature based approach. Our work can be split into two parts. Part 1 dealing with point based motion estimation and Part 2 deals with a Region based motion estimation.

PART 1

Corners are used as features (or tokens) in our point correspondence based motion estimation algorithms. A corner detection algorithm is used to identify the corners present in the given images. Temporal information is also used when detecting the corners. A shuffling scheme is developed to eliminate redundant and noise corners. For matching we use a relaxation based method. This relaxation based method uses some characteristics of the tokens like cornerness, contrast, sharpness, time-varying cornerness etc., Here, the pruned corners from the shuffling algorithm are made use of. A critical analysis on an earlier motion estimation algorithm, is performed. The results showed that the noise performance of the method is poor. A modified rotation compensation is suggested and tested for different synthetic real world situations. It is seen that in most cases the modified method performs better. The methods are again tested for an Objects Motion Sequence(OMS) of images, taken from a camera attached to

a moving robot arm. Noise performance is investigated for the two methods.

PART 2

A novel region based motion estimation scheme is developed and tried. Here moment invariants are used for matching purposes. Once the regions are matched, the lower order moments are exploited for motion parameter estimation. Two theorems have been suggested and proved for the validity of the model with regard to the geometrical arrangement of the object in the 3-D scene while using the moment invariants for motion estimation. The formulation is valid for orthographic projection. We propose two iterative schemes. One , which is based on a Levinberg-Marquardt algorithm, performs better than the other iterative scheme. Our simulation results showed that the algorithm works very well. Even though the algorithm is an iterative one, it converges in most cases, that too, for a large error in the initial guess solution. Of course, like all other non-linear motion estimation algorithms , this method also suffers in noisy conditions. By noise, we mean, missing of some pixels totally, while identifying the same segments in different frames. However, we feel that this method will perform better in real image cases as the segmented patch in the three frames will be of very nearly equal size or can be made nearly equal. The main drawback with this method is that it is suitable for a small rotation. A pure translation motion estimation based on moments is also tried. The results here are better than the pure rotation case, even in the presence of large noise. A new two camera(stereo) method is proposed. This also uses lower order moments for motion estimation. The single camera , however, needs three image frames at least .

Motion estimation from images is an ill-posed inverse problem. If some more information regarding the depth (that is range) is available, then the performance of the schemes can be improved.

DEDICATION

THIS PIECE OF WORK IS DEDICATED TO

SRI MELMARUVATHUR ADI PARASAKTHI.

ACKNOWLEDGEMENTS

I wish to express my deepest sense of gratitude to Dr.Robert.A.King for his supervision of this research work. His valuable suggestions and constant encouragement was a source of inspiration during this work. The continuous interaction of ideas through discussions and cooperation, enhanced my ability in doing the work.

I am very grateful to Dr.C.England , the System Manager of the Centre for Remote Sensing, for his generous help to make use of the computing facilities extensively. The help of other members in the centre is also acknowledged.

My thanks are due to Dr.A.K.Forrest of Centre for Robotics for providing me the facilities to take the image sequences.

I owe very much to Dr.Geoff Stevenson of Mathematical department and Mrs. Pindelaksa, the department's librarian for their help.

I am indebted to all my colleagues and friends at the Digital Communication Section and also at the college, who in one way or another contributed for the successful completion of this work.

This work is done with a Post graduate fellowship from the Association of Commonwealth Universities of the Government of the United Kingdom. I am very much thankful to the Association of Commonwealth Universities and the British Council for their financial help during my course of work. I also thank the Government of India for nominating me to the Commonwealth fellowship.

I wish to thank my parents , my brothers and sisters for morally supporting me throughout my academic studies. I am indebted to my wife Uma who has supported me throughout the work and her constant encouragement helped me to complete the research successfully. Also my daughter, Loheetha's pleasant disturbances in most cases enhanced my understanding about the human pursuit of knowledge.

TABLE OF CONTENTS

Abstract 2

Dedication 4

Acknowledgements 5

Table of contents 6

List of Figures 10

List of Tables 13

CHAPTER ONE: INTRODUCTION

1.1 Introduction. 16

1.2 Motivation for the research 17

1.3 The scope of the study 20

1.3.1 The problem 20

1.4 Organisation of the thesis 21

CHAPTER TWO: MOTION ESTIMATION : PRINCIPLES AND METHODOLOGIES.

2.1 Introduction. 24

2.2 Feature Based approaches in Motion estimation. 25

2.2.1 Point correspondence methods. 25

2.2.2 Line correspondence methods. 34

2.2.3 Region correspondence methods. 41

2.2.4 Using more than two frames. 42

2.3 Optic Flow based approaches. 42

2.3.1 Optical Flow using Taylor's series. 43

2.3.2 Using points for the generation of Optic flow fields. 44

2.4 Comparison of feature based methods and Optic flow based methods. 48

Part 1: Point Based motion parameter estimation. 50

CHAPTER THREE: MOTION ESTIMATION USING LINEAR ALGORITHM AND IMPROVING THE TRANSLATION PARAMETERS.

3.1 Introduction 51

3.2 Tsai and Huangs' linear algorithm(THLAM) 54

3.2.1 Formulation 54

3.2.2 Simulation studies and results 62

3.3 An Improved method for estimating the translation parameters 66

3.3.1 Formulation 66

3.3.2 Simulation and experimental results. 70

CHAPTER FOUR: IMAGE ACQUISITION.

4.1 Introduction 99

4.1.1 Calibration of the Robot. 101

4.1.2 Intermediate process 102

4.1.3 Pre processing 102

CHAPTER FIVE: FEATURE DETECTION AND CORRESPONDENCE ANALYSIS.

5.1 Introduction 108

5.2 Feature point detection 110

5.2.1 Cubic Facet model 113

5.2.2 Time–Varying cornerness 115

5.2.3 Comparison of corner detectors....116

5.2.4 Zuniga–Haralick corner detector....119

5.3 A shuffling scheme to prune corner points....123

5.3.1 Sorting of the corner points....128

5.4 Correspondence analysis....135

5.4.1 Relaxation based token matching....139

Part 2: ***Region Based motion parameter estimation.*** 155

CHAPTER SIX: INTRODUCTION AND REVIEW OF MOMENTS.

6.1 Introduction....156

6.2 Moments and Moment Invariants....156

6.2.1 2–D moments....157

6.2.2 3–D moments....162

CHAPTER SEVEN: REGIONAL CORRESPONDENCE USING MOMENT INVARIANTS.

7.1 Introduction.... 164

7.2 Matching the moment invariants....164

CHAPTER EIGHT: USING MOMENTS FOR MOTION ESTIMATION.

8.1 Introduction....172

8.2 Formulation Using forward and backward motion: Pure rotation case....172

8.2.1 Forward motion....172

8.2.2 Backward motion....181

8.2.3 Scheme–1 186

8.2.4 Theorems for the existence of the planar coefficients 187

8.2.5. Simulation and results 194

8.3 Scheme-2: Levinberg –Marquardt Method 198

8.4 Simulation and results 199

8.5 Pure translation estimation using moments 207

CHAPTER NINE: USING STEREO IMAGE SEQUENCES FOR MOTION ESTIMATION.

9.1 Introduction 220

9.2. Stereo based formulation 220

9.3 Simulation and Results 228

CHAPTER TEN: SUMMARY AND CONCLUSIONS .

10.1 Contributions of this research 241

10.2 Suggestions for further research 244

APPENDICES.

A.1 Finding the rotational parameters. 246

A.2 Jacobian matrix elements of equation 8.36. 249

A.3 Jacobian matrix elements for LeM equation 9.21 252

REFERENCES 257

LIST OF FIGURES

Figure		page
1.1	Computational tasks involved in Computer Vision	18
2.1	Two coordinate system geometry[12].	27
2.2	Imaging geometry for line correspondence	36
2.3	Principle of angular invariance[31].	36
2.4	Geometry for optical flow based motion estimation.	45
3.1	Basic block diagram of motion estimation	53
3.2	Basic geometry for object motion	55
3.3	Effect of error in θ on translation (small translation: θ=15 $^{\circ}$)	85
3.4	Effect of error in θ on translation (small translation: θ=20 $^{\circ}$)	86
3.5	Effect of error in θ on translation (small translation: θ=25 $^{\circ}$)	87
3.6	Overall performance of motion estimation (small translation)	88
3.7	Effect of error in θ on translation (medium translation: θ=15 $^{\circ}$)	89
3.8	Effect of error in θ on translation (medium translation: θ=20 $^{\circ}$)	90
3.9	Effect of error in θ on translation (medium translation: θ=25 $^{\circ}$)	91
3.10	Overall performance of motion estimation (medium translation)	92
3.11	Effect of error in θ on translation (large translation: θ=15 $^{\circ}$)	93
3.12	Effect of error in θ on translation (large translation: θ=20 $^{\circ}$)	94
3.13	Effect of error in θ on translation (large translation: θ=25 $^{\circ}$)	95
3.14	Overall performance of motion estimation (large translation)	96
4.1	The MERLIN system	100
4.2	Points defined by MERLIN's global frame	103
4.3	Objects Motion Sequence : large motion	105

Figure		page
4.4	Objects Motion Sequence: small motion	106
5.1	Flow chart for general motion estimation algorithm	111
5.2.a	Performance of FH corner operator	117
5.2.b	Performance of KR corner operator	117
5.2.c	Performance of ZH corner operator	117
5.2.d	Relative performance of corner operators	117
5.3.	One frame in sequence " UMAS ".	118
5.4.	Result of KR operator on " UMAS ".	118
5.5	Result of FH operator on " UMAS "	118
5.6	Result of ZH operator on " UMAS "	118
5.7	Sobel operated image: Frame 1 [OMS].	122
5.8	Varying corner point threshold	122
5.9	Varying edge threshold	122
5.10.a	Pruning the corner points: stage−1	126
5.10.b	Pruning the corner points: stage−2	126
5.10.c	Pruning the corner points: stage−3	127
5.11	Performance of pruning algorithm: with and without sorting.	129
5.12	Performance of pruning algorithm: with and without sorting.	130
5.13	Results of corner detection and pruning algorithms: large motion case.	131
5.14	Results of corner detection and pruning algorithms: small motion case.	132
5.15	Flow chart for CORPROS subroutine.	134

Figure		page
5.16	An example for correspondence process.	136
7.1.	Effect of noise on ϕ_1	168
7.2.	Effect of noise on β_1	169
7.3	Effect of noise on β_4	170
7.4	Effect of noise on β_5	171
8.1.	Orthographic projection of a rigid planar surface.	173
8.2	Arrangement of image sequences.	182
8.3	Flow chart for Scheme−1	188
8.4	Indeterminate configuration of a rigid planar patch(RPP)	192
8.5	Performance of Levinberg− Marquardt method	202
8.6	Relative error performance of R_3 with depth values.	212
8.7	Relative error performance of R_{12} with depth values.	213
8.8	Relative error performance of R_6 with depth values.	214
8.9	Relative error performance of p with depth values.	215
8.10	Relative error performance of q with depth values.	216
9.1.	Geometry for stereo based motion estimation.	222
9.2	Effect of varying the number of pixels on rotation parameters.	230
9.3	Effect of varying the number of pixels on planar coefficients.	231
9.4	Effect of varying rotation angle on rotation parameters.	232
9.5	Effect of varying rotation angle on planar coefficients.	233
9.6	Effect of varying RPP distance on rotation parameters.	234
9.7	Effect of varying RPP distance on planar coefficients.	235

LIST OF TABLES

Table		Page
T 3.1	Effect of noise on motion parameters in THLAM.	63
T 3.2	Effect of increasing the number of points in THLAM.	65
T 3.3	Comparison between THLAM and the modified method.	71
T 3.4	Effect of rotation angle on THLAM and the modified method.	72
T 3.5	Effect of increasing the number of points on modified method.	73
T 3.6	Performance of THLAM and the modified method (*small translation case:* $\theta = 5^{\circ}$).	75
T 3.7	Performance of THLAM and the modified method (small translation case: $\theta = 15^{\circ}$).	76
T 3.8	Performance of THLAM and the modified method (small translation case: $\theta = 25^{\circ}$).	77
T 3.9	Performance of THLAM and the modified method (medium translation case: $\theta = 5^{\circ}$).	78
T 3.10	Performance of THLAM and the modified method (medium translation case: $\theta = 15^{\circ}$).	79
T 3.11	Performance of THLAM and the modified method (medium translation case: $\theta = 25^{\circ}$).	80
T 3.12	Performance of THLAM and the modified method (large translation case: $\theta = 5^{\circ}$).	81
T 3.13	Performance of THLAM and the modified method (large translation case: $\theta = 15^{\circ}$).	82

Table		Page
T 3.14	Performance of THLAM and the modified method (large translation case: $\theta = 25^{\circ}$).	83
T 4.1	The orientation and motion of the camera during [OMS] acquisition (large motion).	107
T 4.2	The orientation and motion of the camera during [OMS] acquisition (small motion).	107
T 5.1	Results of THLAM and the modified method for OMS: algorithm selected corner points (large motion case: N=8).	146
T 5.2	Results of THLAM and the modified method for OMS: algorithm selected corner points (large motion case: N=10).	147
T 5.3	Results of THLAM and the modified method for OMS: algorithm selected corner points (small motion case: N=8).	148
T 5.4	Results of THLAM and the modified method for OMS: algorithm selected corner points (small motion case: N=10).	149
T 5.5	Results of THLAM and the modified method for OMS: hand-picked corner points (N=8).	151
T 5.6	Results of THLAM and the modified method for OMS: hand-picked corner points (N=12).	152
T 5.7	Results of THLAM and the modified method for OMS: hand-picked corner points (N=16).	153
T 5.8	Results of THLAM and the modified method for OMS: hand-picked corner points (N=20).	154
T 8.1	Effect of rotation angles on motion parameter estimation(scheme-1).	196

Table		Page
T 8.2	Effect of depth values on motion parameter estimation(scheme-1).	197
T 8.3	Convergence of Levinberg-Marquardt algorithm (md2r1).	201
T 8.4	Motion parameter estimation(scheme-2): depth =10 units.	203
T 8.5	Motion parameter estimation(scheme-2): depth =25 units.	204
T 8.6	Motion parameter estimation(scheme-2): depth =30 units.	205
T 8.7	Motion parameter estimation(scheme-2): depth =40 units.	206
T 8.8	Effect of normalising the data: depth =30 units.	208
T 8.9	Effect of normalising the data: depth =40 units.	209
T 8.10	Effect of varying the planar orientation (no noise).	210
T 8.11	Effect of varying the planar orientation (with noise).	211
T 8.12	Motion estimation: pure translation.	219
T 9.1	Effect of varying the number of points.	237
T 9.2	Effect of varying the rotation angle.	238
T 9.3	Effect of varying the depth values.	239
T 9.4	Effect of noise on motion parameters.	240

CHAPTER ONE

INTRODUCTION.

1.1 INTRODUCTION:

Vision plays a major role in understanding the environment around us, for all beings having visual capabilities. It seems effortless to understand the world present before our eyes as we open our eyes. However, this effortless action is not because of the simplicity of vision but rather from the immense amount of specialized wetware, the biological equivalent to hardware, dedicated to it [1]. Vision is , in fact , extremely complex and it has turned out to be immensely difficult to construct successful vision systems. Among the five senses of humans, vision is considered to provide about 75% of the total input to the brain about the environment. It is clear from this that the amount of data being received through two small optical sensors is enormously large [2]. Hence vision is considered to be an important sensory modality. Scientists and engineers have been trying to incorporate such human-like capabilities into machines for a long time [3]. The purpose of computer vision is to enable a computer to understand its environment from the visual information supplied to it by one or more cameras or other similar sensors. The computational tasks involved in computer vision is given in Figure 1.1

Low level processing starts with the image acquisition either from a camera or from other similar sensors. The images thus acquired are processed to remove noise. Features like edges, lines and regions are detected from these images. The detected features from a given image frame are then analysed so as to get

corresponding features in the given frames. These front end operations are more important as the higher level operations depend on these low level vision processes.

In the intermediate level processing , the features detected in the low level processing are used to derive various 3-D information such as motion, structure etc.,

In the high level processing a global understanding of the scene is performed. In addition, objects present in a scene are recognized. This interpretation can help a vision system to model the scene. In some cases, a domain specific knowledge base is created and stored in the computer and this helps the vision system to understand the environment. This approach is known as the *bottom up approach* in computer vision. All the different levels of a computer vision system are highly interdependent.

From Figure 1.1 it can be seen that motion analysis forms an intermediate level processing. It provides important cues for high level scene interpretation.

1.2. MOTIVATION FOR THE RESEARCH

From a single image frame, it is seldom possible to say whether objects present in the frame are in motion. Thus motion analysis needs a sequence of image frames (at least two) to know about the moving objects in a scene. One of the most important aspects of image sequence analysis is motion estimation, which is of particular significance in machine or computer vision and artificial intelligence. To motivate motion analysis some of the applications [4,5,6] are listed below:

(1) Military applications: Camera can be used as a surveillance device in military.

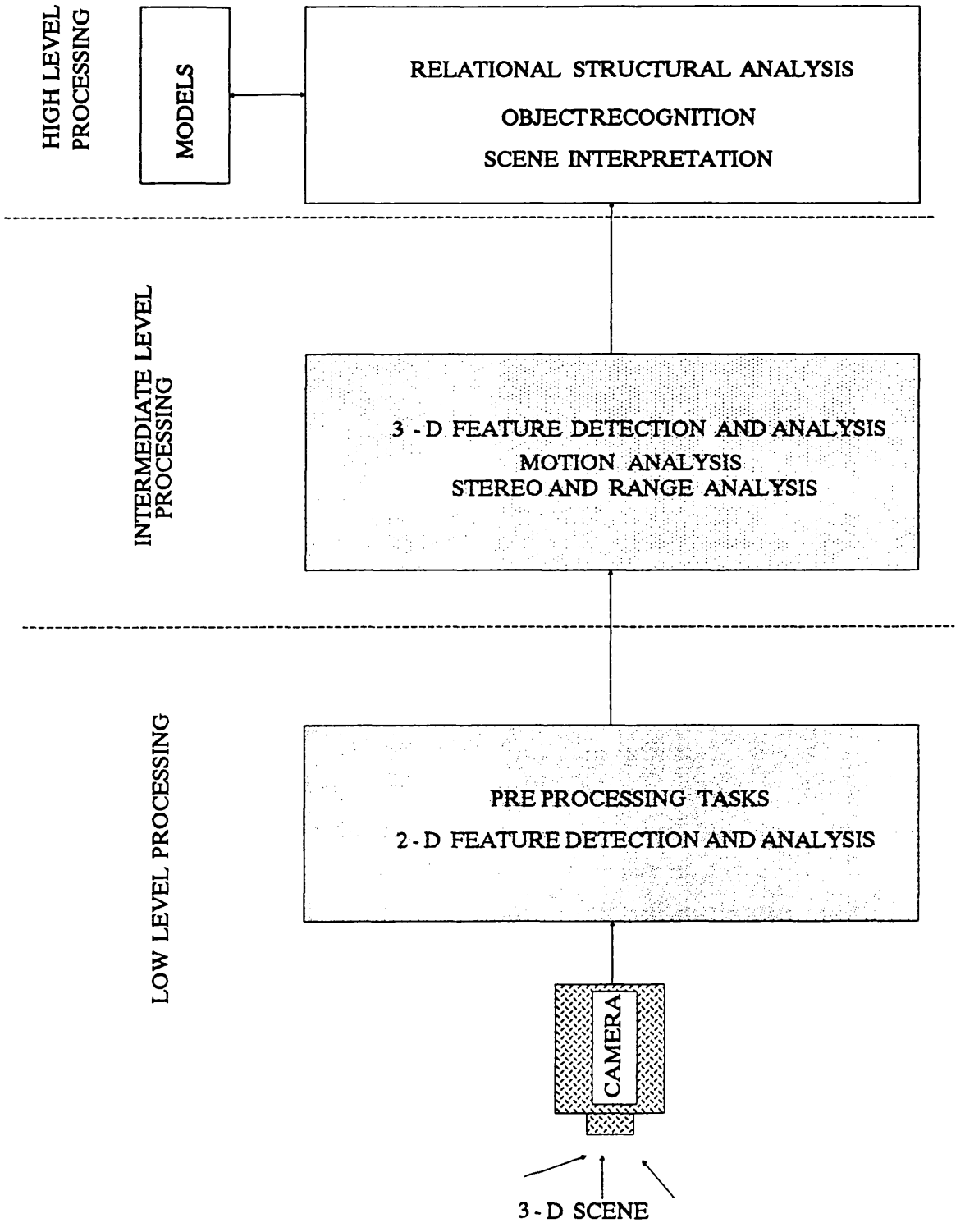


Figure 1.1
Computational tasks involved in Computer Vision

The camera is used to detect the motion of objects and to track these objects. Besides, the image sequences obtained from a camera can be used for object recognition. Another important application for image sequences is in autonomous moving vehicles, which is of special interest.

(2) Medical applications: Study of cell motion by microcinematography and study of heart motion from images will be useful for doctors. The latter application will be useful in detecting abnormalities of hearts and also their normal functions. CAT images are used to obtain qualitative information about blood circulation in human organs.

(3) Meteorological applications: Satellite images are analysed for forecasting the weather. Furthermore, the images from the satellites are used in agriculture for estimating crop yield and for controlling pests efficiently.

(4) Traffic monitoring: Image sequences are used by transport engineers and police officers for the identification of a vehicle, its speed etc., and there by, for ticketing the speeding vehicle.

(5) Bandwidth compression: From the image sequences the velocity of moving objects are found. By sending these velocity components as information bits , the transmission rate can be reduced and hence the bandwidth requirement.

(6) Industrial automation and robotics: Motion analysis is an important requirement for the hand-eye co-ordination of a robot working on conveyor belts of assembly lines. Motion analysis is used to determine the moving robots trajectory and to construct a model of its environment in order to avoid obstacles.

Current research in motion analysis tends to concentrate on finding solutions for general three-dimensional (3-D) motion and modelling a general phenomenon such as the relative motion between the observer, scene components and light sources.

The importance of motion estimation will be obvious from a large number of conferences , workshops and NATO advanced study programmes held during the past few years. Besides, a number of special issues on the topic is a common feature in international journals. For some of these citings , one can see the recent review by Aggarwal and Nandhakumar [7].

1.3 THE SCOPE OF THE STUDY:

The current research in motion analysis may be divided into two main categories:

(a) research done by biologists and psychophysiologists to understand the process of motion perception by biological systems.

(b) research done by computer scientists and engineers to develop computational algorithms for motion estimation.

Our scope is limited to the latter one. The computational analysis of time-varying images may be divided into two broad classes.

(a) Feature based motion analysis

(b) Optical flow based analysis

We concentrate on the first class of motion analysis.

1.3.1 THE PROBLEM

We are interested in deriving the motion parameters of a moving object/objects from the image sequences that are available.

We assume that the 'correspondence problem' is solved in the case of synthetic images and for real images we establish the correspondence by a method which uses both inter and intra frame information. The objects are assumed rigid

and undergoing a constant velocity.

The study is organised into two parts. In the first part, we used points as features in three consecutive image frames. We critically analysed a linear method of motion estimation as proposed by Tsai and Huang [8]. An extension to the method is proposed and it yielded an improvement for the translation parameter estimation. In the real images we used time-varying corners as features. The correspondence is solved using a relaxation based method.

In the second part we used regions as features for motion estimation. For matching regions in a sequence of image frames we used moment invariants of regions and for the actual motion estimation we used lower order moments.

1.4 ORGANISATION OF THE THESIS:

The organisation of the thesis is as follows.

In the next chapter(Chapter 2) we discuss the different methodologies of motion estimation. We analysed both the feature based and the optic flow based methods though the latter is not extensively covered. It is to be mentioned here, that we deviate from using some of the standard notations used in motion analysis literature, especially in the case of optical flow based methods. This is to have a consistent and simple presentation.

Chapter 3 deals with the motion estimation using a linear algorithm. A modification to the method is suggested and is found to improve the translation parameter estimation. The results are given for synthetically generated image points.

In Chapter 4 we briefly explain the method of image acquisition of Objects Motion Sequence [OMS] using a CCD camera attached to a Robot's moving arm.

Detection of corner points as features from a few real images and the correspondence of these corner points in a pair of two images are given in Chapter 5. We also present some results for manually picked corner points as well as results for the corner points detected by a corner detection and correspondence algorithm.

Chapters 3,4 and 5 collectively form Part 1 of the thesis. We can see these chapters deal with the point correspondence method of motion estimation.

The second part of the thesis deals with the motion parameter estimation using regions as features. Region based methods do not need any correspondence problem to be solved prior to estimating the motion. Chapter 6 gives an introduction to using regions as features for motion estimation. We also review in this chapter, the moment invariants of regions which are well known for their use in object recognition.

Matching the selected regions in a pair of images using moment invariants is given in Chapter 7.

In Chapter 8 we show a method of using the lower order moments for motion estimation purpose. The formulation is first derived for a pure rotation case. Then we prove mathematically that for some configurations the method will give degenerate results. Two iterative schemes are presented and extensive simulation tests are performed on this method. In addition a method is given for translation parameter estimation in the orthographic projection case. A planar patch motion is considered here.

We give a novel formulation for deriving the motion of a planar patch from stereo images in Chapter 9. This method also uses moments as features for motion estimation. Again, a planar patch motion is considered here.

Finally, in Chapter 10, we give a general conclusion for motion estimation. We also suggest some further research areas and extensions of the present research in this motion parameter estimation field.

CHAPTER TWO

MOTION ESTIMATION : PRINCIPLES AND METHODOLOGIES

2.1 INTRODUCTION:

It is well known that when an object in 3-D space moves , the object will experience a rotational and/or translational motion. In our work, by motion estimation, we mean the problem of estimating the motion parameters like the rotational and the translational velocities of the object/objects in the 3-D coordinate axes from a sequence of image frames. The general problem is of computing the 3-D information from a given 2-D frame. As can be expected , due to the dimensionality reduction, some of the desired parameters like the translation velocity in the z-direction may not be recoverable. Once the motion parameters are found, the structure of the object points can easily be obtained. This problem of finding the Structure of the objects from Motion is known as ' Structure from Motion (SfM) ' problem.

In this chapter we discuss the various formulations reported in the 3-D motion estimation field. Basically , there are two distinct approaches in motion estimation. They are (a) Feature (Token) Based Approach (FBA) and (b) Optical flow Based Approach (OBA). These different approaches will be discussed at length in the following sections. In section 2.2 we concentrate on the methods in FBA and in section 2.3 we consider the Optic flow based methods. A comparison and the problems of these two approaches are made in section 2.4.

We use the terms ' tokens' and ' features' equivalently throughout the thesis.

2.2 FEATURE BASED APPROACHES IN MOTION ESTIMATION:

The Feature based method relies on extracting a relatively sparse set of highly discriminative features in the image corresponding to three dimensional object features in the scene. These features may be 'corners', occluding boundaries of surfaces, lines, curves and regions. Inter-frame correspondence is then made between these features. Constraints are formulated based on the physical laws of 3-D objects in motion. Such constraints in most cases result in non-linear equations. These equations are then solved to estimate the motion parameters.

FBA is further divided into three types. These divisions are a result of the type of tokens used. They are (a) Point Correspondence methods (PC), (b) Line Correspondence methods (LC) and (c) Region Correspondence methods (RC).

2.2.1 Point Correspondence methods:

Most of the motion parameter estimation techniques use the point correspondence method and it is the earliest of all the tokens used for such tasks. Points such as corners, centers of gravity of a surface etc., are some very distinguishable features in the 3-D objects. Some of the earliest methods were reported in [9,10,11]. Aggarwal et al [9,10] analysed the feasibility of estimating the motion from point features in given images. In [11], Roach and Aggarwal showed a method of extracting the motion(translational and rotational) of a camera and the 3-D structure from five points in two views.

The object is assumed stationary and the camera is moved from one

position to the other. Perspective projection was also assumed. The camera positions are given by the three position coordinates (x_0,y_0,z_0) and three angle coordinates(θ,ϕ,ψ): θ, and ϕ are the pan and the tilt of the camera respectively and ψ is the angle of rotation of the image plane coordinate axes relative to the global coordinate axes. The equations that relate the 3-D coordinates of a point p(x,y,z) and its image point P(X,Y) are as follows.

$$X = F \frac{a_{11}(x-x_0)+a_{12}(y-y_0)+a_{13}(z-z_0)}{a_{31}(x-x_0)+a_{32}(y-y_0)+a_{33}(z-z_0)} \tag{2.1}$$

$$Y = F \frac{a_{21}(x-x_0)+a_{22}(y-y_0)+a_{23}(z-z_0)}{a_{31}(x-x_0)+a_{32}(y-y_0)+a_{33}(z-z_0)} \tag{2.2}$$

where,

F is the focal length, (x_0, y_0, z_0) is the projection center,

and a_{ij} (i,j ϵ {1,2,3}) are functions of (θ,ϕ,ψ). Roach et al [11] related the number of unknowns and the number of equations available for the solution of 3-D points and the motion parameters. The 3-D coordinates of the points are unknowns: five points give rise to 15 unknowns. The camera position and orientation (x_0, y_0, z_0, θ,ϕ,and ψ) for each camera add another 12 unknowns. Hence , the total number of unknowns are 27. Each 3-D point provided two projection equations (as in 2.1 and 2.2) per camera position. Hence five points will give 20 equations. In order to solve the above system it is required that seven of the unknowns should be known *a priori*. The location and the orientation of the first camera is taken as 0 (origin) and one of the z coordinates of the five points is fixed to a known +ve scaling constant. Thus they have to solve 20 non-linear equation in 20 unknowns. They solved this large system of non-linear equations by

iterative methods. However, this method seems to be very sensitive to noise and with a sufficiently over determined system they could get a reasonable estimation of the parameters. In order to get fairly reasonable results they used two views of 12 or even 15 points or three views of at least 7 points.

Nagel [12] employed a compact vector notation for solving the SfM problem. The camera is assumed stationary and the object is moving. There are two coordinate systems; one attached to the camera and the other attached to the object. In the camera coordinate system , a 3-D point 'm' in the ' i th view is expressed as C_{mi} c_{mi} where c_{mi} is a unit vector(See figure 2.1). The same point

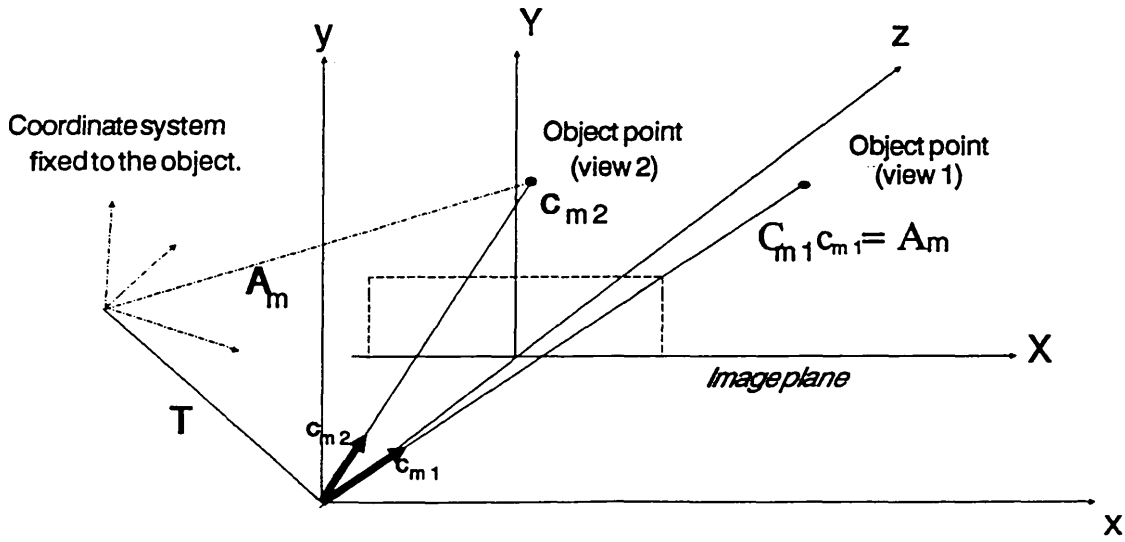


Figure 2.1

Two coordinate system geometry [12]

in the object coordinate system is represented as A_m . The object undergoes a translation T followed by a rotation **R**. The relationships for the two views are given by,

$$C_{m1}\, c_{m1} = A_m \tag{2.3}$$

$$C_{m2}\, c_{m2} = (A_m + T)\, \mathbf{R} \tag{2.4}$$

Here c_{m1} and c_{m2} are known from the image plane coordinates and lens center location whereas, C_{mi}, A_m, T and **R** are unknowns. The lens is assumed to be at the origin. Nagel outlined a procedure by which the various parameters are computed in terms of a scale factor C_{11}. He extensively used vector algebra in deriving the above results and used five points in two views to derive the motion and structure parameters.

Tsai and Huang [13] in their work approached the motion estimation problem in a different manner. They assumed that the 3-D scene contains a rigid planar patch characterized by a set of eight or more points. They solved the problem in two steps: first they defined some intermediate parameters called 'essential parameters' and solved for these parameters from two image frames. Then the actual motion parameters were calculated from the essential parameters. Their formulation is as follows:

The motion of a point p(x,y,z) to p′(x′,y′,z′) is expressed as

$$x' = \mathbf{R}\, x + T \tag{2.5}$$

where,

$$x = (x,y,z)^\top \qquad ; x' = (x',y',z')^\top$$

$$\mathbf{R} = \begin{bmatrix} n_1^2+(1-n_1^2)\cos\theta & n_1n_2(1-\cos\theta)-n_3\sin\theta & n_1n_3(1-\cos\theta)+n_2\sin\theta \\ n_1n_2(1-\cos\theta)+n_3\sin\theta & n_2^2+(1-n_2^2)\cos\theta & n_2n_3(1-\cos\theta)-n_1\sin\theta \\ n_1n_3(1-\cos\theta)-n_2\sin\theta & n_2n_3(1-\cos\theta)+n_1\sin\theta & n_3^2+(1-n_3^2)\cos\theta \end{bmatrix} \tag{2.6}$$

$n_{j(\ j=1,2,3)}$s' are the directional cosines of the rotation axis, θ being the angle of rotation.

The perspective constraints are written as,

$$X = F\,\frac{x}{z}\ ; \quad Y = F\,\frac{y}{z}$$

$$X' = F\,\frac{x'}{z'}\ ; \quad Y' = F\,\frac{y'}{z'} \tag{2.7}$$

where

' F ', the focal length of the camera can be assumed unity.

The planar patch equation is given by

$$ax + by + cz = 1 \tag{2.8}$$

where {a,b,c} are known as the planar coefficients.

When the angle of rotation θ, is small , we use the following approximation for the rotation matrix,

$$\mathbf{R}_a = \begin{bmatrix} 1 & -\phi_3 & +\phi_2 \\ +\phi_3 & 1 & -\phi_1 \\ -\phi_2 & +\phi_1 & 1 \end{bmatrix} \tag{2.9}$$

where

$$\phi_j = n_j.\theta \quad (j=1,2,3)$$

From (2.7),(2.5) and (2.8) it is possible to get,

$$X' = \frac{a_1.X + a_2.Y + a_3}{a_7.X + a_8.Y + a_9} \qquad Y' = \frac{a_4.X + a_5.Y + a_6}{a_7.X + a_8.Y + a_9} \tag{2.10}$$

where a_i's are functions of the unknowns (ϕ_1,ϕ_2,ϕ_3, a, b, c, t_x , t_y and t_z).

Here a_9 is taken as the scaling factor and the a_i's are called 'essential parameters'. After tedious manipulations they got a sixth order polynomial in terms of only $\frac{t_x}{t_z}$. Though potentially 6 roots were possible it seems the authors have got only two real roots, however no theoretical proof is given. One of the solutions corresponding to the two roots gave the required true solution whereas the other was a reflection of the true solution. The uniqueness of the result was proved using Lie group theory.

In [14,15], Fang et al derived a non-linear equation involving the three rotational parameters (ϕ_1,ϕ_2 and ϕ_3) and the two scaled translational parameters (t'_x, t'_y). Using five point correspondences, it was shown that the motion parameters can be estimated by iterative methods. The non-linear equation is of the form,

$$\{[-(F^2+Y\,Y')\,t'_x + X'Y\,t'_y]\phi_1 + [X\,Y'\,t'_x - (F^2 + X\,X')\,t'_y]\phi_2 + (X\,t'_x + Y\,t'_y)\,F\phi_3 - F\,\Delta Y\,t'_x + F\Delta X\,t'_y + F^2\,X'\phi_1 + F^2\,Y'\phi_2 - F(X\,X' + Y\,Y')\phi_3 + X'\Delta Y\,F - Y'\Delta X\,F\} = 0 \tag{2.11}$$

$$\Delta X = X' - X; \quad \Delta Y = Y' - Y \tag{2.12}$$

where,

$$t'_x = F.\frac{t_x}{t_z} \text{ and } t'_y = F.\frac{t_y}{t_z} .$$

Equation (2.12) represents the displacements in the X , Y directions in the image frames. These two quantities will be frequently used and are known as ' Flow vectors ' in Optic flow based methods.

First the equations for ΔX and ΔY are derived from equations (2.5) and (2.7). Equating the common term $\frac{Z}{t_z}$ in ΔX and ΔY relations, the above non-linear equation can easily be obtained. The method was tested using several non-linear algorithms. It is shown that the convergence depends on the number of points , their geometrical configurations and the initial guess solution.

In [16], a direct method of motion estimation is given. The brightness of pixels are used in estimating the motion parameters. Let,

f_1 (X,Y) = brightness of a pixel at time t_1

$f_2(X',Y')$ = brightness of a pixel at time t_2 where X' and Y' are given by equation (2.7). Then the method is to find ΔX and ΔY so as to minimize D [$f_1(X,Y)$, $f_2(X+\Delta X,Y+\Delta Y)$] where D is a distance measure. That is,

$$D = [\, f_1(X,Y) - f_2(\, X+\Delta X,\, Y+\Delta Y)\,]^2 \tag{2.13}$$

Using Taylor's series expansion for $f_2(X',Y')$ around (X,Y) and taking only linear terms we get,

$$D = \sum_{X,Y} (\, \Delta f - \Delta X.\frac{\partial f_2}{\partial X} - \Delta Y.\, \frac{\partial f_2}{\partial Y}\,)^2 \tag{2.14}$$

where

$$\Delta f(\, X,Y) = f_1(X,Y) - f_2(X',Y')$$

Minimizing D in the above equations is equivalent to finding a least-square

solution of the set of linear equations

$$\Delta f_i = \Delta X \left(\frac{\partial f_2}{\partial X} \right)_i + \Delta Y \left(\frac{\partial f_2}{\partial Y} \right)_i \quad , i = 1,2,\ldots N \tag{2.15}$$

where N is the number of points.

Now from (2.10) it is easy to show that,

$$\Delta X = \frac{a_1X+a_2Y+a_3-a_7X^2-a_8XY-a_9X}{a_7X+a_8Y+a_9} \tag{2.16a}$$

$$\Delta Y = \frac{a_4X+a_5Y+a_6-a_7XY-a_8Y^2-a_9Y}{a_7X+a_8Y+a_9} \tag{2.16b}$$

Substituting equations(2.16a and 2.16b) in equation (2.15) we get an equation which is a function of the essential parameters, a_i's(i=1,2,...9). Hence, with 8 or more points it is possible to get the estimates for the essential parameters up to a scale factor. From these estimates the actual motion parameters can easily be calculated.

Independently Longuet-Higgins [17] and Tsai and Huang [8] gave a closed form solution for deriving the motion parameters from 8 or more points in two views. The later work is based on a Singular Value Decomposition(SVD) approach and they provided some results for their algorithm. More about this and a modification is dealt with, in chapter-3. Longuet-Higgins [18] showed that some of the configurations will cause this 8 point linear algorithm to fail.

In [19,20], Tsai et al showed that aside from a scale factor for translation parameters , the number of solutions is either one or two depending on the

multiplicity of the singular values of the A matrix which is formed by the essential parameters a_i' s as below:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \tag{2.17}$$

The problem of estimating the motion parameters of an RPP in three views was considered in [21] and it was proved that given 4 point correspondences in three perspective views, the number of solutions for the motion parameters is always one.

In [22], Tsai presented an efficient method for estimating 3-D motion parameters and object surface orientation from the motion parallax of 3-D moving planar contour or coplanar edges in the form of conic arc segments in two perspective views. Fang and Huang [23] gave an elaborative account of the 8 point algorithm and some real world applications. They used corners as tokens and derived the motion parameters of moving rigid objects.

Several modifications to the basic linear algorithm have been attempted [6,24,25,26,27]. The linear algorithms require an *a priori* knowledge with regard to a non-zero translation. The main improvement being, the applicability of the linear algorithm to the pure rotation case. It is also shown [24,25] that if the observed points lie on a specific quadratic surface passing through the origin, the motion parameters are not unique. Faugeras et al [26] and Nagel [6] discuss a more robust estimation of motion parameters using a least mean square error minimization approach.

In [27], Tosconi et al presented an iterative method of extracting structure

from motion from noisy feature points. The basic idea is to look, among all the possible motions, for the one that minimizes the difference between the measured actual motion and the one obtained,synthesized using the motion. They used a reconstruction and projection technique for deriving the motion. It was shown that the method is robust even in the presence of incorrect matches of the feature points.

2.2.2 Line Correspondence Methods:

The use of lines as token points for Structure from Motion has been relatively neglected for a long time. One notable exception is the work of Ben and Huang [28]. They proposed an iterative method based on the observation of seven line correspondences in three views for the case of general motion between the views. They have also considered a linear method based on eight line correspondences for the special case where the rotational axes of the displacement between the three views are all parallel to each other that is the orientation of the these axes is constant.

A.Mitiche et al, [29,30] exploited the principle of conservation of distances with respect to rigid object motion. This was similar to using the rigidity constraint. The formulation is as follows:

Assume a static view and two cameras(S_1 and S_2) having focal length F_1 and F_2. Let P_i be a point in the 3-D space whose coordinates in S_1 be (x_i, y_i, z_i) and in S_2 be (u_i, v_i, w_i). Let the corresponding image point be $p_i(X_i, Y_i)$ on the image plane I_1 and $q_i(U_i, V_i)$ on the image plane I_2. The geometry[29] is as given in figure 2.2. In figure 2.2 two specific points P_1 and P_2 are taken for illustration. Since P_i is a point on the line S_1P_i there exists a real number $\lambda_i > 0$ such that,

$$x_i = \lambda_i X_i \quad ; y_i = \lambda_i Y_i \qquad \text{and } z_i = (1-\lambda_i)\, F_1 \tag{2.18}$$

If we have a line segment with end points P_i and P_j in space then we can write its length as

$$d_{ij}^2(S_1) = (x_i-x_j)^2 + (y_i-y_j)^2+(z_i-z_j)^2 \tag{2.19}$$

Substituting equation (2.18) in the above we get,

$$d_{ij}^2(S_1) = (\lambda_i X_i-\lambda_j X_j)^2+(\lambda_i Y_i-\lambda_j Y_j)^2+F_1^2(\lambda_i-\lambda_j)^2 \tag{2.20}$$

Similarly for the same points in the second camera system we have,

$$u_i = \gamma_i U_i \quad ; v_i = \gamma_i V_i \qquad \text{and } w_i = (1-\gamma_i)\, F_2 \tag{2.21}$$

Writing the length of the line segment defined by P_i and P_j in the second camera coordinate system we get,

$$d_{ij}^2(S_2) = (u_i-u_j)^2 + (v_i-v_j)^2+(w_i-w_j)^2 \tag{2.22}$$

Substituting equation(2.21) we get,

$$d_{ij}^2(S_2) = (\gamma_i U_i-\gamma_j U_j)^2+(\gamma_i V_i-\gamma_j V_j)^2+F_2^2(\gamma_i-\gamma_j)^2 \tag{2.23}$$

However , the two distances $d_{ij}(S_1)$ and $d_{ij}(S_2)$ represent the same length in the 3-D scene.This is known as the *constancy of distance* principle. Hence equating them we get a non-linear equation.That is,

$$(\lambda_i X_i-\lambda_j X_j)^2+(\lambda_i Y_i-\lambda_j Y_j)^2+F_1^2(\lambda_i-\lambda_j)^2=$$
$$(\gamma_i U_i-\gamma_j U_j)+(\gamma_i V_i-\gamma_j V_j)^2+F_2^2(\gamma_i-\gamma_j)^2 \tag{2.24}$$

It is easy to see that each end point gives rise to two unknowns (λ_i,γ_i) and two points defining a line segment give one non-linear equation(2.24). Hence if we have 5 points in the two camera systems we will have 10 line segments and 10 unknowns and hence the system is soluble. However, Mitiche et al [29], fixed one of the unknowns, as the distances can be found only up to a scale factor. Hence

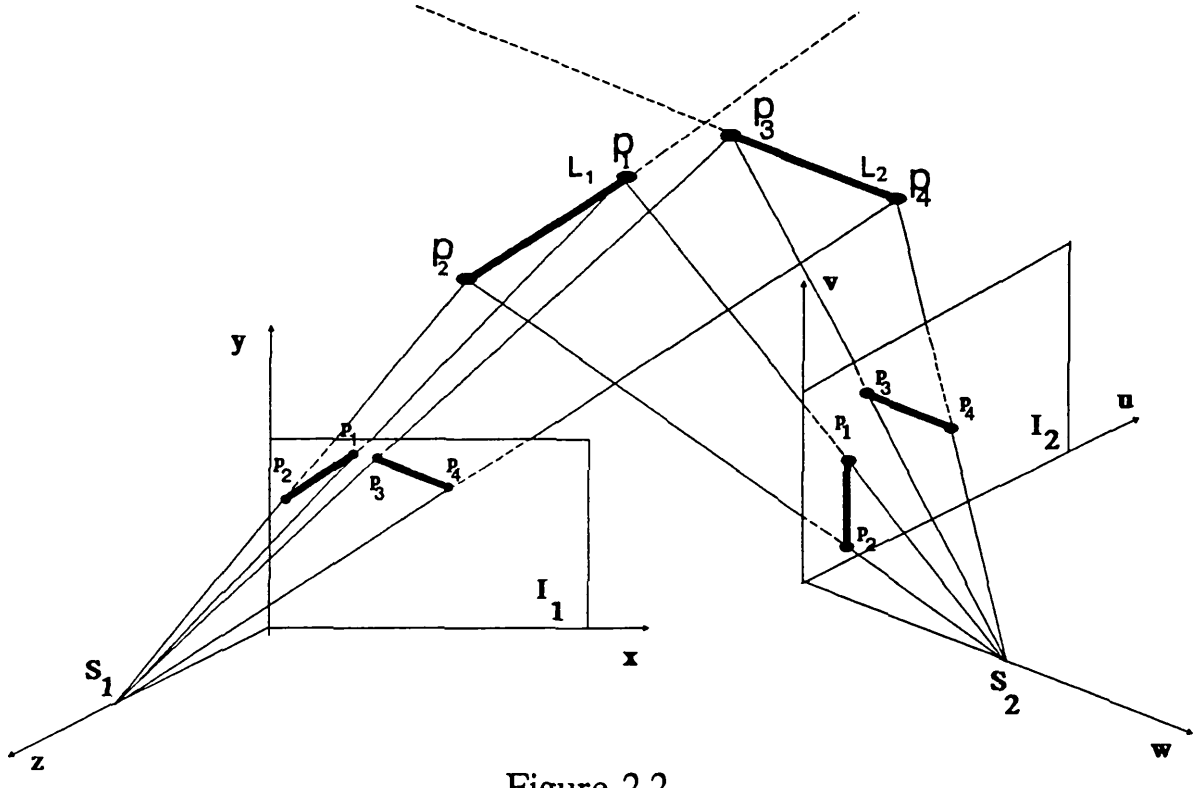


Figure 2.2
Imaging geometry for line correspondence.

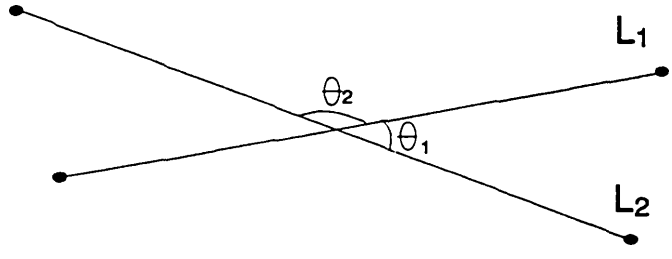


Figure 2.3
Principle of angular invariance [31]

with 10 equations they tried to find a solution using an iterative method. Having found the position, using any of the four points (which are non-coplanar) they got the transformation matrix. With a fairly good initial guess, they got good results when there is no noise. When noise is present the method gives poor results. It was suggested if more than five points were used the performance will be better. The number of non-linear equations that can be formed from N points will be $\binom{N}{2}$.

Mitiche, Seida and Aggarwal [31] also considered the case of four lines in three views. They used the concept of *angular invariance* of rigid motion. This principle states that " angles between lines of a set of lines in space do not change as a result of a rigid motion of this set of lines ". In figure 2.3 we show two lines L_1 and L_2 which are intersecting in the 3-D space. As we have two angles θ_1 and θ_2 we may not be in a position to say which of the two angles we are actually observing. Hence, Mitiche et al used the $\cos^2\theta$ value. In figure 2.2 we have given the same two lines L_1 and L_2. The lines can be represented in vector notation as P_{21} and P_{43}. The components of these vectors are given by,

$$P_{21} = \{ (x_2 - x_1); (y_2 - y_1) ; (z_2 - z_1) \} \tag{2.25a}$$

$$P_{43} = \{ (x_4 - x_3) ; (y_4 - y_3); (z_4 - z_3)\} \tag{2.25b}$$

The components of the above vectors can be written in terms of the image coordinates using equation 2.18 as,

$$P_{21} = \{ (\lambda_2 X_2 - \lambda_1 X_1); (\lambda_2 Y_2 - \lambda_1 Y_1) ; (\lambda_1 - \lambda_2) F_1\} \tag{2.26a}$$

$$P_{43} = \{ (\lambda_4 X_4 - \lambda_3 X_3); (\lambda_4 Y_4 - \lambda_3 Y_3) ; (\lambda_3 - \lambda_4) F_1\} \tag{2.26b}$$

Then

$$\cos^2\theta = \left[\frac{P_{21} \cdot P_{43}}{|P_{21}| \, |P_{43}|} \right]^2 \tag{2.27}$$

Dividing the components of P_{21} by λ_1 and writing the components of the unit vector as,

$$p_{21} = \left\{ \frac{\mu X_2 - X_1}{a}; \frac{\mu Y_2 - Y_1}{a} ; \frac{(1-\mu) F_1}{a} \right\} \tag{2.28a}$$

$$p_{43} = \left\{ \frac{\Omega X_4 - X_3}{b}; \frac{\Omega Y_4 - Y_3}{b} ; \frac{(1-\Omega) F_1}{b} \right\} \tag{2.28b}$$

where $\mu = \frac{\lambda_2}{\lambda_1}$, $\Omega = \frac{\lambda_4}{\lambda_3}$, 'a' and 'b' are the lengths of the vectors p_{21} and p_{43} and are given by,

$$a = \sqrt{(\mu X_2 - X_1)^2 + (\mu Y_2 - Y_1)^2 + (1-\mu)^2 F_1^2} \tag{2.29a}$$

$$b = \sqrt{(\Omega X_4 - X_3)^2 + (\Omega Y_4 - Y_3)^2 + (1-\Omega)^2 F_1^2} \tag{2.29b}$$

If we observe the same lines in another frame of reference as in figure 2.2 then one can write another equation for θ as,

$$\cos^2\theta = \left[\frac{P'_{21} \cdot P'_{43}}{|P'_{21}| \, |P'_{43}|} \right]^2 \tag{2.30}$$

where P'_{21} and P'_{43} are the vector equation for the same lines in the new frame of reference. Substituting the image coordinates as in equation 2.21 we get the vector components for the new set of lines as,

$$P'_{21} = \{ (\gamma_2 U_2 - \gamma_1 U_1); (\gamma_2 V_2 - \gamma_1 V_1) ; (\gamma_2 - \gamma_1) F_2 \} \tag{2.31a}$$

$$P'_{43} = \{ (\gamma_4 U_4 - \gamma_3 U_3); (\gamma_4 V_4 - \gamma_3 V_3) ; (\gamma_4 - \gamma_3) F_2 \} \tag{2.31b}$$

Writing in a similar manner we get the components of the unit vectors as,

$$p'_{21} = \left\{ \frac{\mu' U_2 - U_1}{a'}; \frac{\mu' V_2 - V_1}{a'} ; \frac{(1-\mu') F_2}{a'} \right\} \tag{2.32a}$$

$$p'_{43} = \left\{ \frac{\Omega' U_4 - U_3}{b'}; \frac{\Omega' V_4 - V_3}{b'} ; \frac{(1-\Omega') F_2}{b'} \right\} \tag{2.32b}$$

where $\mu' = \frac{\gamma_2}{\gamma_1}$, $\Omega' = \frac{\gamma_4}{\gamma_3}$, ' a' ' and ' b' ' are the lengths of the vectors p'_{21} and p'_{43}. The lengths ' a' ' and ' b' ' can be calculated using expressions similar to equations (2.29 a) and (2.29 b).

Due to the angular invariance principle equations (2.27 and 2.30) are equal in value. Using equations 2.28a, 2.28b, 2.32a, and 2.32b we get,

$$\frac{[(\mu X_2 - X_1)\ (\Omega X_4 - X_3) + (\mu Y_2 - Y_1)\ (\Omega Y_4 - Y_3) + (1-\mu)(1-\Omega)\ F_1^2]^2}{a^2\ b^2}$$

$$= \frac{[(\mu'\ U_2 - U_1)(\Omega'\ U_4 - U_3) + (\mu' V_2 - V_1)(\Omega'\ V_4 - V_3) + (1-\mu')(1-\Omega')\ F_2^2]^2}{a'^2\ b'^2} \qquad (2.33)$$

where μ, Ω, μ' and Ω' are the unknowns. a, b , a' and b' are the lengths of the lines L_1 , L_2 , L'_1 and L'_2 , in the 2-D image plane, and hence can be calculable. Each line gives one unknown per view (like μ) and two intersecting lines give rise to one equation as in 2.33 With four lines in three views Mitiche et al got 12 equations in 12 unknowns. Each of the equations are second order equations involving only 4 of the 12 unknowns appearing in each of the equations.However, they had to use four lines in three views so as to get significant information. After finding μ, μ', Ω and Ω' a method is given for finding the motion parameters. The equations were complicated and non-linear. As the error in the initial guess increases the error in finding the rotation angle also increases. The motion parameters were not involved in these equations. The unknowns were found by an iterative method from which the rotation parameters are derived. They separated the rotation and translation component of the motion. Once the rotation component is determined they used this to cancel this component from the general motion resulting in a pure translation motion. They used the normals of the lines and derived two linear equations in 6 unknowns for the two translation

motions. Hence with 3 lines it was possible to solve for translation. The experimental results shown were for synthetic motion and they reported results for the angle estimation alone. No error performance for the translation was given. The results showed that the Average angle error was less than 6% if the initial guess was within 10% of the actual values. The authors have added noise to the image pixels by moving the image points within a 5 x 5 pixel area thus creating the noisy line correspondences.

Liu and Huang [32,33] presented a method of estimating the rigid body motion using straight line correspondences. They showed for a pure rotation case, 2 line correspondences over 2 frames were sufficient to determine the rotation matrix. In the case of pure translation they required at least 6 line correspondences over 3 frames. In the case of a general motion they separated the rotation and translation motions. The rotational part was found by an iterative method. Having found the rotational part they cancelled the rotational component from the general motion and the resulting motion was a pure translation. Hence the translational part was found by a set of linear equations. The general principle in their work was based on the normals of the lines in the 3-D space. However, there were a few drawbacks in the above work. Firstly, the iterative method needs an initial guess solution and if the guess is poor the method converges to a local minimum and not to the global minimum. The few results reported in the papers are for 0% error in the initial guess. Secondly, as is to be expected, the translation parameters are very sensitive to the errors made in solving the rotation parameter. They also studied the effect of noise in the input parameters namely the normals of the line in three frames. In general, they showed that if the noise is within 1.0% reasonable results can be obtained. As the

noise increases the number of iterations needed to find the rotation parameters is more and hence the computing time.

Independently Liu and Huang [34] and Spetsakis and Aloimonos [35] gave a linear algorithm for finding the motion parameters using line correspondences. Both the methods were similar, in the sense that they determined an intermediate parameter (like the essential parameters in point correspondence methods) which were functions of the motion parameters. First they solved for the intermediate parameters and then the actual motion parameters were found. Liu and Huang gave an elaborate procedure and gave an algorithm for finding the motion parameters in their work. The results shown were only for the noise free case. They report that the algorithm is very sensitive to noise. Spetsakis et al explain the theory behind their algorithm. Both the methods require 13 line correspondences over three frames. It is very difficult to find 13 LCs over three frames and the real question is, is it possible to find a linear algorithm which requires less than 13 LCs?. The answer to the question is not available at the time of writing this thesis.

2.2.3 Region correspondence methods:

Matching the point and line features between frames is often encountered with great difficulty. Point features are very much susceptible to noise. On the other hand, identifying a region correspondence is relatively easier and is more accurate. As we use a large number of points while forming the regions , the noise effect may be less compared to point correspondence methods. In Part 2 of the thesis we discuss a method of using moment invariants for finding the region correspondence. Nevertheless, the performance of a region based method depends

on the accuracy of the segmentation process. Also at least one visible planar surface is needed in a given set of image frames.

In [36], a method is given to detect the change from multisensor images. They used a region based method to detect the changes from these images. Young and Wang [37] proposed a region correspondence method for a rigid planar patch motion estimation. Legters and Young [38] discussed an iterative method for computing the shape-change and 2-D motion parameters from a given set of images of an RPP. They used an operator formulation for motion estimation. In [39], Gunesekaran and Young considered the problem of recovering 3-D motion and structure from dynamic scenes using regions. They extended the operator formulation proposed in [38] for the 3-D motion case. They assumed that the initial orientation and the range of the planes were available.

2.2.4 Using more than two frames:

Several attempts have been made for improving the accuracy of the FBA methods by using more than two frames at a time. Broida and Chellappa [40,41] used a long sequence of about 20 – 30 frames for motion parameter estimation. They have applied an Iterated Extended Kalman Filtering for increasing the accuracy of the estimation. Shariat [42,43] used more than three frames for motion estimation. Tseng et al [44], using optimal control theory and long sequences, obtained the motion estimation of objects.

2.3. OPTIC FLOW BASED APPROACHES.

Optic flow relates the spatio-temporal variations in the image plane to the velocity fields. There are two basic steps involved. They are (1) generation of

velocity fields from the given image frames and (2) interpretation of the optic flow fields. Velocity at each pixel is computed by some local operators and thus a velocity field is formed. Usually only partial computation of the velocity is possible and hence a global aggregation process is essential. However, the interpretation of the optical flow field is often difficult. In OBA , it is assumed that the image frames are very narrowly separated and hence the rotation is very small in the 3-D space. In the following sections we will see the basics of OBA.

2.3.1 Optical flow using Taylor's series:

Let f(X,Y,t) be the image gray level at point (X,Y) at time t. If 'f' is an analytic function and it remains constant while (X,Y) moves to (X+ΔX, Y+ΔY) at time (t+Δt) we have

$$\begin{aligned} f(X,Y,t) &= f(X+\Delta X,\ Y+\Delta Y,\ t+\Delta t) \\ &= f(X,Y,t) + f_x(X,Y,t)\Delta X + f_y(X,Y,t)\Delta Y + f_t(X,Y,t)\Delta t + O^2(.) + \\ &\quad \text{Higher order partial derivatives of f and powers of } \Delta X,\Delta Y,\Delta t. \end{aligned} \tag{2.34}$$

If the higher order terms are sufficiently small , we have,

$$f_x\Delta X + f_y\Delta Y + f_t\Delta t = 0.$$

Dividing by Δt and letting $\Delta t \longrightarrow 0$ we have $\frac{\Delta X}{\Delta t} \longrightarrow u$ and $\frac{\Delta Y}{\Delta t} \longrightarrow v$ so that,

$$f_x\, u + f_y\, v = -f_t \tag{2.35}$$

where u , v are the velocity components of X,Y in the image plane. This gives us a constraint on (X,Y) at each point of the image. Note that it gives us only the component of (u,v) in the gray level gradient direction. Using additional assumptions it is possible to compute the optical flow. Some of the assumptions used are (i) Optical flow is smooth and neighbouring points have similar

velocities, (ii) Optical flow is constant over a small segment and (iii) optical flow is the result of a restricted motion eg. planar motion.

Horn and Schunck [45] imposed the smoothness constraint by minimizing the error in optical flow as

$$E^2(X,Y) = \text{error in } (f_x u + f_y v + f_t)^2 + \lambda^2[(u_x^2+u_y^2)+(v_x^2+v_y^2)] \quad (2.36)$$

where,

λ is a constant

u_x and u_y represent the components of u in the x and y direction.

similarly v_x and v_y represent the components of v in the x and y direction.

The aim here is to find u and v so as to minimize R in the following equation.

$$R = \int\int \left\{(f_x u + f_y v + f_t)^2 + \lambda^2[(u_x^2+u_y^2)+(v_x^2+v_y^2)]\right\} dx\, dy \quad (2.37)$$

Using calculus of variations the above equation is solved. Several iterative schemes were developed to obtain the optical flow.

2.3.2. Using points for the generation of Optic flow fields:

Consider the imaging geometry as in figure 2.4. Let p(x,y,z) and p′(x′,y′,z′) be an object point before and after motion. The basic equation relating the motion of a point is given in equation (2.5). The rotation must be small in the OBA case, and hence the rotation matrix given in equation(2.9) is used. For simplicity we write (2.9) as,

$$\mathbf{R}_a = \begin{bmatrix} 1 & -\Omega_z & \Omega_y \\ \Omega_z & 1 & -\Omega_x \\ -\Omega_y & \Omega_x & 1 \end{bmatrix} \quad (2.38)$$

where

$$\Omega_x = n_1\theta; \quad \Omega_y = n_2\theta; \quad \Omega_z = n_3\theta \tag{2.39}$$

Hence the small motion equation can be written as,

$$x' = R_a\ x + T \tag{2.40}$$

If the object point before and after motion are separated by unit time , the image velocity can be easily computed. Let the focal length be unity in the perspective imaging equation (2.7). Let $(\alpha,\ \beta)$ be the velocity of an image point.

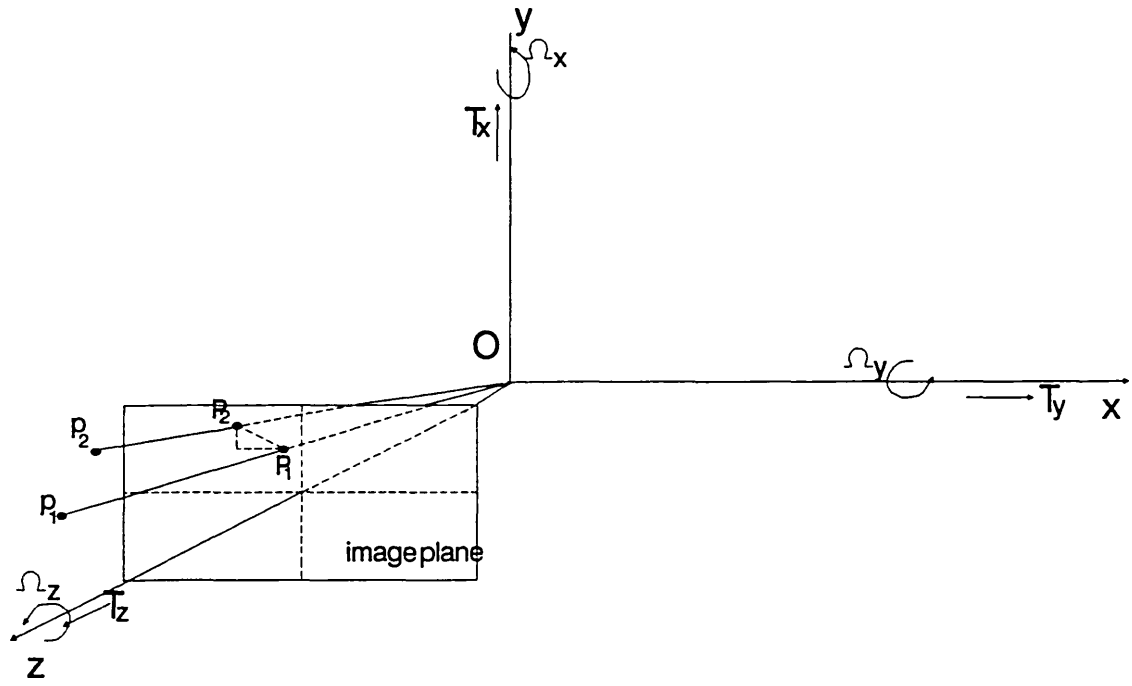


Figure 2.4

Geometry for optical flow based motion estimation.

Using equations (2.7) and (2.38), we get the image velocity as,

$$\alpha = X'-X = -\Omega_x XY+\Omega_y(1+X^2)-\Omega_z Y+(T_x-T_z X)/z \quad (2.41\ a)$$

$$\beta = Y'-Y = -\Omega_x(1+Y^2)+\Omega_y XY+\Omega_z X+(T_y-T_z Y)/z \quad (2.41\ b)$$

It can be seen that (α,β) can be decomposed as a sum of rotational and translational components.

$$(\alpha, \beta) = (\alpha_R,\beta_R)+ (\alpha_T,\beta_T)$$

where,

$$\alpha_R=-\Omega_x XY+\Omega_y(1+X^2)-\Omega_z Y \quad (2.42a)$$

$$\alpha_T=(T_x-T_z X)/z \quad (2.42b)$$

$$\beta_R=-\Omega_x(1+Y^2)+\Omega_y XY+\Omega_z X \quad (2.42c)$$

$$\beta_T=(T_y-T_z Y)/z \quad (2.42d)$$

The translational vector is defined by a unit vector $U = \frac{T}{r}$ where, r is the length of T. Once the motion parameters are recovered , it is possible to estimate the relative depth, z(x,y)/r, corresponding to each pixel (X,Y) where a flow vector is defined, for $r \neq 0$. In [46,47] , a flow field induced by a rigid motion of a planar patch is considered. Let the patch be represented as,

$$ax + by + cz=1 \quad (2.43)$$

Using the notation,

A =(a, b, c); L= (l,m,n) and L= r A and using equation (2.7) with F=1 we get,

$$\frac{r}{Z}= l X+ m Y+ n .$$

Substituting the unit translation vector and the above equation for (α,β) in (2.41)

we can easily get,

$$\alpha = k_1 + k_2X + k_3 Y + k_7X^2 + k_8XY \quad (2.44\ a)$$

$$\beta = k_4 + k_5 X + k_6 Y + k_7XY + k_8 Y^2 \quad (2.44\ b)$$

where,

$$k_1 = \Omega_y + nU_x \quad (2.45\ a)$$

$$k_2 = l\ U_x - nU_z \quad (2.45b)$$

$$k_3 = -\Omega_x + mU_z \quad (2.45\ c)$$

$$k_4 = -\Omega_x + nU_y \quad (2.45\ d)$$

$$k_5 = \Omega_z + lU_y \quad (2.45\ e)$$

$$k_6 = m\ U_y - nU_z \quad (2.45\ f)$$

$$k_7 = \Omega_y - lU_z \quad (2.45\ g)$$

$$k_8 = -\Omega_x - mU_z \quad (2.45\ h)$$

Adiv called the above set of eight equations the Ψ transformation. They describe a 2-D motion in the image plane by the eight parameters. If we have a collection of flow vectors α and β it is possible to get the 3-D motion parameters. A similar set of equations were introduced by A.Waxman et al [48]. They also analyzed the optic flow induced by a planar patch. However, they used a Taylor's series expansion to derive equations similar to the above equations. Adiv [47] extensively studied the ambiguities of interpretation of the optic flow induced by planar patches. He mentioned several factors responsible for the ambiguity of interpretation. Among those the following are important ones: (a) the noise level in pixels are high, (b) the object is far away (c) the flow field is sparse and (d) the noise values in adjacent flow vectors are highly correlated.

In [49], Mitiche discussed a method of deriving structure and motion of a

rigid object from optical flow. He proposed an approach in which functions other than the luminance functions like R-G-B signals can be used. In [50] , the author addressed the problem of using two distinct views taken from unknown viewpoints of a number of points for which optical flow was known, to derive the position of the points in space , their motion , and the relative displacement of the viewing systems. The viewing systems were assumed immobile and the object were moving. He proposed a linear method which used both the point correspondence information and optical flow information. With four or more points he determined the displacements of the view points. Once this displacement is determined the position and their motion were easily determined.

Kanatani [51,52] discussed two methods of deriving the structure from motion using planar patch surfaces. In [51] he considered the problem under orthographic projection and in [52] he considered the perspective case. Both the methods use the optical flow of a planar patch. He emphasized the geometrical invariance properties of optical flows in both the cases. The methods are applicable for small rotation only.

2.4 COMPARISON OF FEATURE BASED METHODS AND OPTIC FLOW BASED METHODS:

In sections 2.2 and 2.3 we have discussed several methods of motion estimation and hence the structure of moving objects. It is seen in section 2.2 that the FBA methods require that correspondence be established between a sparse set of features detected in one image with that of a set of features detected in the second image. This correspondence process is often very difficult to achieve due to occlusion, and other problems. An elaborate discussion on feature detection and

correspondence analysis is given in chapter-5. A few results also are given for real image sequences. When multiple objects are present in a scene , the situation is very complicated. Besides, if a large number of image sequences are considered the feature detection and feature correspondence operations may be computationally costly. Optic flow methods do not suffer from this type of problem. FBA methods are slower due to a large amount of pre-processing. If the features are more complex only a few possible candidates satisfy the conditions and hence matching is easy.

On the other hand, computation of optic flow requires finding derivatives both in the spatial domain and in the time domain. Real images are often noisy. Finding the derivatives is a noise augmenting process. The higher the order of the derivative, the higher the noise is. Even though several closed form solutions are available, the OBA methods do not produce useful results due to this sensitivity to noise. [53].

As a final remark we would like to say that due to a large number of publications in motion analysis it is not possible for us to provide a complete review of the field. Hence, the references cited are only representative. A good review on motion analysis is given in [7], where further references can be found.

PART-1

POINT BASED MOTION PARAMETER ESTIMATION

CHAPTER THREE

MOTION ESTIMATION USING LINEAR ALGORITHM AND IMPROVING THE TRANSLATION PARAMETERS.

3.1 INTRODUCTION:

Three dimensional motion parameter estimation is an important step due to its various applications which include object tracking and robot motion. The application of image sequence analysis and motion estimation was discussed extensively in Chapter-1. However, deriving motion from the two dimensional image sequences is often difficult. This is due to the nature of the problem itself. We try to estimate the 3-D parameters from the available 2-D data. Since, we have only partial information regarding the motion , quite often the algorithms proposed are nonlinear. Only a few linear algorithms have been proposed so far, even though the research area is almost a decade old.

Linear algorithms have several advantages. They are

(1) the structure of the solution set of linear equations is clear.

(2) In most cases they give a unique solution.

(3) Linear algorithms are faster.

However , in the presence of noise , a linear algorithm may not be an optimal one. On the other hand, non-linear algorithms may give an optimal solution in presence of noise. They are iterative in nature and hence computationally slower. They require some sort of an initial guess to start the iteration. If the initial guess is wrong, the algorithm may not converge at all or it may converge to a local minimum.

Huang's group is one of the first to work extensively on the three dimensional motion parameter estimation problem. Previous formulations on motion parameter estimation made a few assumptions like (a) fixed axis assumption (b) parallel projection (c) translation only etc., These assumptions are satisfied by only a few kinds of motion , of course. In the previous chapter we have seen the different methodologies of motion parameter estimation that are currently being used. Most of the methods proposed are non-linear in nature. In this chapter we critically analyze an algorithm proposed by Tsai and Huang [8]. This is for several reasons. Their algorithm is a linear one and one of a few which deal with noise performance, though the noise analysis is not complete. The performance of the algorithm is good and robust for a general motion when the data is free from noise. The algorithm is simple to implement and to illustrate.

We are mainly interested in the noise performance of the algorithm in various situations like (1) small translation, (2) medium translation and (3) large translation. It is to be noted here that we are analyzing the performance of the algorithm for different translation situations alone. This is because the error in the translation parameter estimation is very large compared to the error in the rotation parameter estimation. This observation will soon be clear from the simulation experiments in the next section.

To help the reader, the actual linear algorithm is first presented briefly. Then the modified algorithm which improves the translation parameter estimation is presented in section 3.3. A comparison of the performance of both algorithms is analyzed for the three cases mentioned above and a noise analysis is made. The performance of the algorithm for a real image sequence is given in chapter 5. The basic block diagram is shown in Figure 3.1 , the shaded blocks being used when

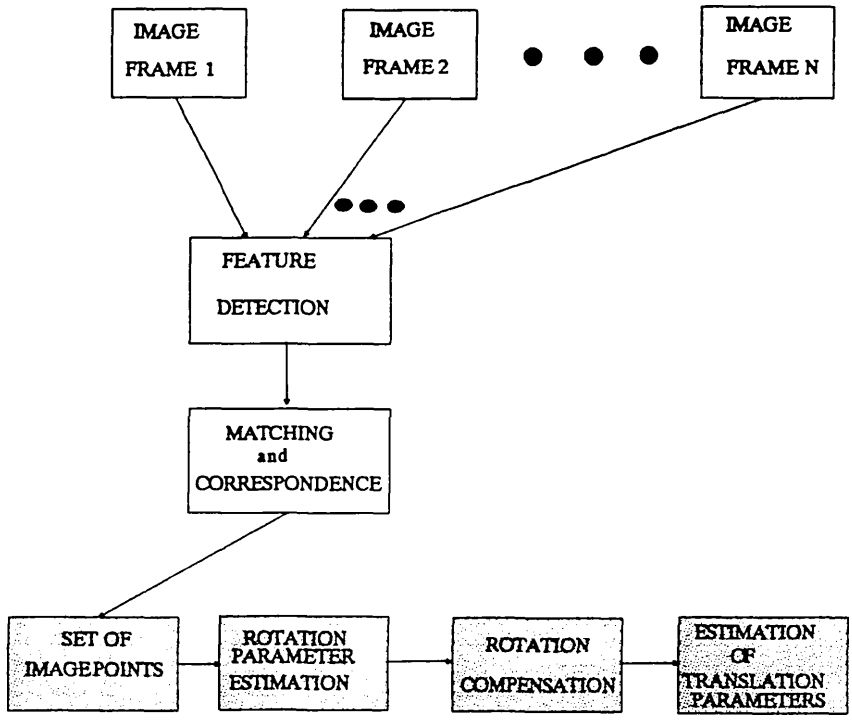


Figure 3.1
Basic block diagram of motion estimation.

using synthetic image points.

We assume rigid object motion only throughout the thesis. For synthetic image points we assume the correspondence of image points between frames is known . Later in chapter 5 we show how this correspondence is achieved for real image sequences.

3.2 TSAI AND HUANG'S LINEAR ALGORITHM (THLAM):

In this section we present Tsai and Huang's linear(THLAM) algorithm as it was presented originally. For a detailed description , the reader is advised to refer to [8].

The basic geometry of the problem is as shown in Figure 3.2. Let p_i (x_i , y_i , z_i) be a point on the object before motion , and p'_i (x'_i , y'_i , z'_i) be the same point after motion . Let P_i(X_i , Y_i) and P'_i(X'_i, Y'_i) be the image coordinates of the 3-D points after projecting the 3-D points on to the image (2-D) plane. We take central or perspective projection.

3.2.1 Formulation:

Our goal is to find the motion parameters due to the motion that actually takes place during time t_1 to t_2 , from a given set of corresponding image points in the two frames one taken before the motion and the other after motion.

Any 3-D rigid body motion can be decomposed into a rotation by an angle θ around an axis through the origin with directional cosines n_1, n_2 and n_3 and a

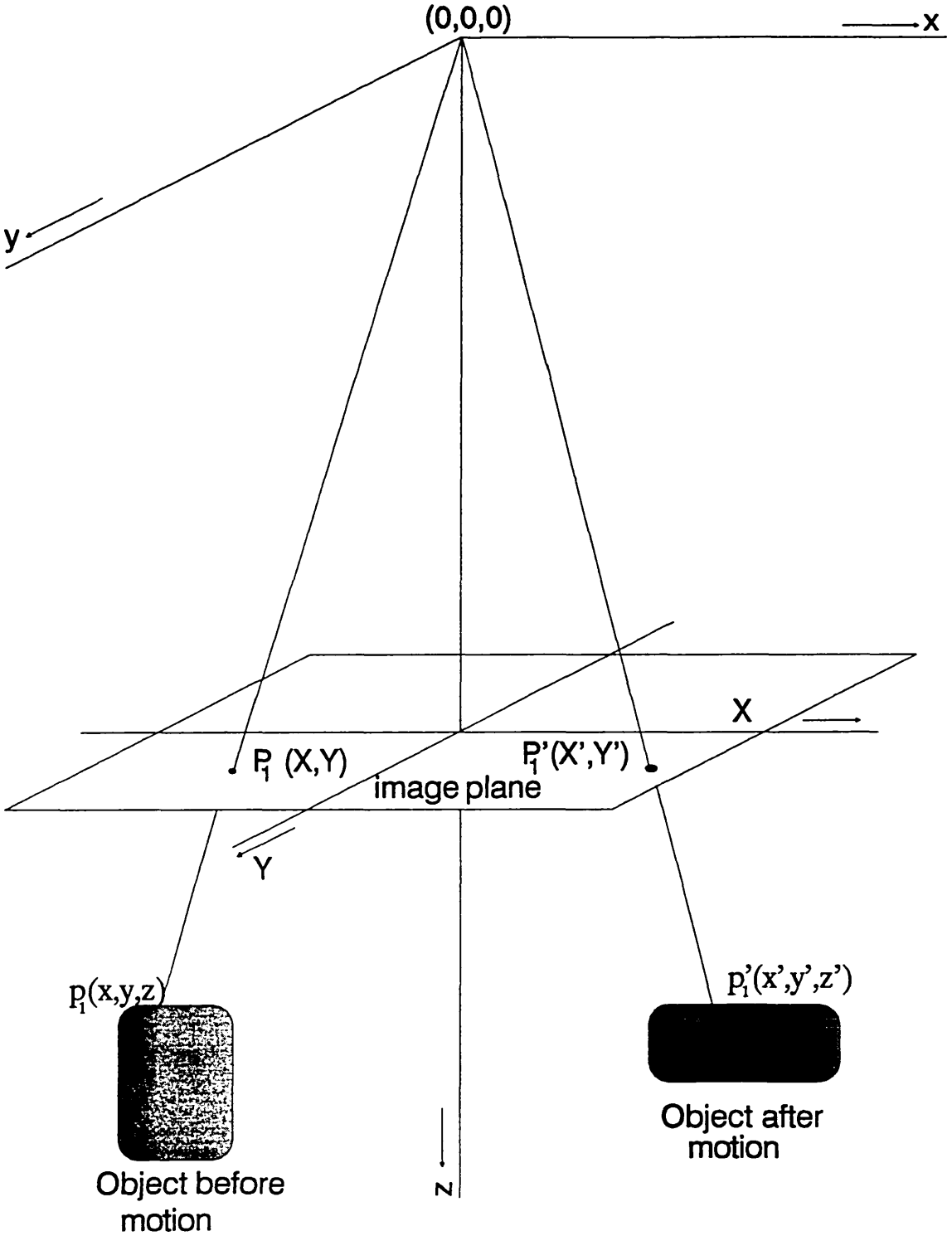


Figure 3.2
Basic geometry for object motion.

translation T (t_x, t_y , t_z)$^\top$ in the three spatial directions. Hence, the 3-D motion is given by,

$$\mathbf{x'} = \mathbf{R}\,\mathbf{x} + \mathbf{T} \tag{3.1}$$

where,

$\mathbf{x} = (\, x_1, y_1, z_1)^\top$ and $\mathbf{x'} = (\, x_1', y_1', z_1')^\top$ are the coordinates of a 3-D point before and after motion respectively.

$$\mathbf{R} = \begin{bmatrix} n_1^2+(1-n_1^2)\mathrm{Cos}\theta & n_1n_2(1-\mathrm{Cos}\theta)-n_3\mathrm{Sin}\theta & n_1n_3(1-\mathrm{Cos}\theta)+n_2\mathrm{Sin}\theta \\ n_1n_2(1-\mathrm{Cos}\theta)+n_3\mathrm{Sin}\theta & n_2^2+(1-n_2^2)\mathrm{Cos}\theta & n_2n_3(1-\mathrm{Cos}\theta)-n_1\mathrm{Sin}\theta \\ n_1n_3(1-\mathrm{Cos}\theta)-n_2\mathrm{Sin}\theta & n_2n_3(1-\mathrm{Cos}\theta)+n_1\mathrm{Sin}\theta & n_3^2+(1-n_3^2)\mathrm{Cos}\theta \end{bmatrix} \tag{3.2}$$

where,

n_1,n_2, n_3 are the direction cosines of the rotation axis and θ is the rotation angle.

The rotation matrix is an orthogonal matrix in which $|\mathbf{R}| = 1$.

It can be written in terms of its elements as below:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.3}$$

The image coordinates for point P_i and P_i' are given by,

$$X_i = F\frac{x_i}{z_i} \qquad Y_i = F\frac{y_i}{z_i}$$

$$X_i' = F\frac{x_i'}{z_i'} \qquad Y_i' = F\frac{y_i'}{z_i'} \tag{3.4}$$

The focal length F, can be assumed unity. Using (3.1) ,(3.3) and (3.4) we can derive,

$$X_i' = \frac{(r_{11} X_i + r_{12} Y_i + r_{13})\, z_i + t_x}{(r_{31} X_i + r_{32} Y_i + r_{33})\, z_i + t_z} \tag{3.5a}$$

$$Y_i' = \frac{(r_{21} X_i + r_{22} Y_i + r_{23})\, z_i + t_y}{(r_{31} X_i + r_{32} Y_i + r_{33}) z_i + t_z} \tag{3.5b}$$

From equation (3.5a) we get,

$$z_i = \frac{(t_x - t_z X_i')}{[\, X_i' (\, r_{31} X_i + r_{32} Y_i + r_{33}) - (\, r_{11} X_i + r_{12} Y_i + r_{13})]} \tag{3.6a}$$

Similarly from equation (3.5b) we get,

$$z_i = \frac{(t_y - t_z Y_i')}{[Y_i' (\, r_{31} X_i + r_{32} Y_i + r_{33}) - (\, r_{21} X_i + r_{22} Y_i + r_{23})]} \tag{3.6b}$$

Equating the right hand sides of (3.6a) and (3.6b) we get

$$\begin{bmatrix} X_i' & Y_i' & 1 \end{bmatrix} \mathbf{E} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = 0 \tag{3.7}$$

where,

$$\mathbf{E} \triangleq \begin{bmatrix} t_x r_{21} - t_y r_{31} & t_x r_{22} - t_y r_{32} & t_x r_{23} - t_y r_{33} \\ t_x r_{31} - t_z r_{11} & t_x r_{32} - t_z r_{12} & t_x r_{33} - t_z r_{13} \\ t_y r_{11} - t_x r_{21} & t_y r_{12} - t_x r_{22} & t_y r_{13} - t_x r_{23} \end{bmatrix} \tag{3.8}$$

$$\mathbf{E} \triangleq \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}$$

Assume the 3-D object is twice as far away as the initial position and has a translation velocity twice as that of the original, in all three directions respectively. In this case the translation vector T is doubled and equation (3.7) still holds . This means that there is a common scale factor for translation which cannot be determined. The scale factor can be assumed to be unity, without loss of generality. The elements in **E** are called the **'essential parameters'** as the actual motion parameter can be determined uniquely , given **E**.

Equation (3.7) can be expanded for an image point, before and after motion in terms of the essential parameters as follows,

$$X_1'X_1\, e_1 + X_1'Y_1\, e_2 + X_1'\, e_3 + Y_1'X_1 e_4 + Y_1'Y_1\, e_5 + Y_1' e_6 + X_1 e_7 + Y_1 e_8 = -1 \tag{3.9}$$

Hence if we have several , say eight corresponding image points, we have a set of linear equation as follows,

$$[\mathbf{M}] \begin{bmatrix} e_1 \\ e_2 \\ \cdots \\ \\ \cdots \\ e_8 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ \cdots \\ \\ \cdots \\ -1 \end{bmatrix} \tag{3.10}$$

where **M** is given by,

$$[\mathbf{M}]= \begin{bmatrix} X_1'X_1 & X_1'Y_1 & X_1' & Y_1'X_1 & Y_1'Y_1 & Y_1' & X_1 & Y_1 \\ X_2'X_2 & X_2'Y_2 & X_2' & Y_2'X_2 & Y_2'Y_2 & Y_2' & X_2 & Y_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ X_7'X_7 & X_7'Y_7 & X_7' & Y_7'X_7 & Y_7'Y_7 & Y_7' & X_7 & Y_7 \\ X_8'X_8 & X_8'Y_8 & X_8' & Y_8'X_8 & Y_8'Y_8 & Y_8' & X_8 & Y_8 \end{bmatrix} \quad (3.11)$$

Hence solving the linear system in (3.10) will give the values for the essential parameters e_1, e_2,... e_8. However , there are some considerations on the actual 3-D object points when solving the above set of equations. Some of the configurations, such as the vertices of a cube, may be degenerative [18]. We avoid this degeneracy problem by looking at the rank of the matrix **M**, while solving the above system. If the rank is ≤ 7 , then the system of equations are linearly dependent. In such cases the solution to the above system is not unique.

In the following we discuss the uniqueness of **R** given **E**.

Equation (3.8) can be decomposed as,

$$\mathbf{E} = \begin{bmatrix} t_z & 0 & 0 \\ 0 & t_x & 0 \\ 0 & 0 & t_y \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{R} -$$

$$\begin{bmatrix} t_y & 0 & 0 \\ 0 & t_z & 0 \\ 0 & 0 & t_x \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}$$

$$\mathbf{E} = \begin{bmatrix} 0 & t_z & 0 \\ 0 & 0 & t_x \\ t_y & 0 & 0 \end{bmatrix} \mathbf{R} - \begin{bmatrix} 0 & 0 & t_y \\ t_z & 0 & 0 \\ 0 & t_x & 0 \end{bmatrix} \mathbf{R}$$

$$= \begin{bmatrix} 0 & t_z & -t_y \\ -t_z & 0 & t_x \\ t_y & -t_x & 0 \end{bmatrix} \mathbf{R}$$

$$\triangleq \mathbf{GR} \tag{3.12}$$

It is shown in [8] that **G** is always singular and contains multiple singular values . However, because of the special structure(skew-symmetric) of **G**, it is shown that once **E** is given **G** and **R** are unique. The actual motion parameter can be uniquely determined given **E** by taking the Singular Value Decomposition (SVD) of **E** without having to solve the non-linear equations.

Theorem 1:

Let the SVD of $\mathbf{E}$ be given by

$$\mathbf{E} = \mathrm{U}\ \Lambda\ \mathrm{V}^{\top} \tag{3.13}$$

where

U and V are ortho-normal matrices and Λ the eigen value matrix.

Then, given $\mathbf{E}$ to within a scale factor , there are two solutions for the rotation matrix

$$\mathbf{R}_1 = \mathrm{U} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & \mathrm{s} \end{bmatrix} \mathrm{V}^{\top} \tag{3.14}$$

or

$$\mathbf{R}_2 = \mathrm{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & \mathrm{s} \end{bmatrix} \mathrm{V}^{\top} \tag{3.15}$$

where s = det U . det V and one solution for the translation (up to a scale factor) is

$$\begin{bmatrix} \mathrm{t}_x \\ \mathrm{t}_y \\ \mathrm{t}_z \end{bmatrix} = \alpha \begin{bmatrix} \phi_1^{\top}\ \phi_2\ /\ \phi_2^{\top}\phi_3 \\ \phi_1^{\top}\phi_2\ /\ \phi_1^{\top}\ \phi_3 \\ 1 \end{bmatrix} \tag{3.16}$$

where ϕ_i is the i^{th} row of $\mathbf{E}$, i=1,2,3 and α is some scale factor. Only one of the solutions of $\mathbf{R}$, together with appropriate sign for α will yield positive z and z'. The proof of the above theorem is omitted here and can be seen in [8]. Having found the rotation matrix $\mathbf{R}_1$ and $\mathbf{R}_2$, the actual rotation parameters namely, the rotation angle, the direction cosines n_1, n_2 and n_3 can be found easily. There are two methods available for finding the actual parameters and these are

explained in Appendix A-1.

3.2.2 Simulation studies and results:

Data generation: The 3-D data points are selected from within a cube of size 10 units, randomly. The depth of the points is varied from 2 units to 10 units. Having selected a number of 3-D points , they are moved according to equation (3.1). A known rotation and translation motion are assumed. The resulting coordinates are those of the object points after motion. The object points before and after motion are then projected on to the image plane to get the image coordinates(2-D points).

Addition of noise: In our simulations we have added noise having a uniform probability density function with a range [– N , + N] , to each of the image points , where N, the range of the noise, is set by the user. This may on occasions cause the percentage of simulated error to be more than 30%, which is more than the perturbation effects in [8]. In the actual case, the noise is due to the digitization, quantizing effects etc., We cannot have infinite resolution; hence the rounding of the coordinates also make up for the noise.

In Table T.3.1 we show the performance of this algorithm for different amounts of noise. From Table T.3.1 we can see that increasing the noise decreases the performance of the linear algorithm. The reference parameters are given in bold letters. The error observed in estimating the motion parameters is not unexpected, since , whenever noise is present , there will be error in estimating the essential parameters themselves.This is because the Right hand side and the Left hand side of equation 3.9 are not equivalent under noisy conditions. Besides, we

TABLE –T 3.1

Effect of noise on motion parameters in THLAM.

Noise (N)	t_x/t_z	t_y/t_z	α (Deg)	β (Deg)	γ (Deg)	θ (Deg)
0.001	5.1733	3.1580	29.0307	80.3230	62.9199	43.3980
0.005	4.5851	20.8320	23.2391	80.7805	68.8062	36.0952
0.01	2.8995	−6.0102	12.2472	79.2565	84.1895	26.5893
0.05	1.3115	−3.1769	71.7314	89.2056	161.7129	32.6941
0.1	2.3413	13.4568	26.3889	78.8461	113.5879	30.8108
REF	5.0000	2.5000	30.0000	80.0000	62.0385	45.0000

can see the effects of noise on translation parameter estimation. In some cases the direction of the translation itself changes. It is to be mentioned here that not only the rotation and translation parameters are affected, but also, the actual orientation of the rotation axis , that is, the direction cosines of the rotation axis. This is an important observation as this will be further emphasized in the modified algorithm. Initially, we tested our simulation method with 8 corresponding points. Table T.3.2 shows the effect of the algorithm on varying the number of correspondence points. We have also shown the mean and the standard deviation of the parameters for a number of simulation runs. In general , the performance improves as the number of corresponding points are increased. This is well seen in the estimated rotation angle and the translation parameters. Also the standard deviation of the translation parameter values are large, implying the reliability of the experiment in this case is poor. Also the results are far away from their mean values. When the number of corresponding points are more than 8 , equations (3.10) are over determined and hence we use a least square algorithm.

As the translation parameters are found directly from the elements of the **E** matrix , the error in the **E** matrix affects the translation parameter estimation, largely. The rotation parameters are, however, estimated through the SVD procedure. It seems, the SVD procedure to some extent compensates for the errors in the **E** matrix.

The major short-coming of this method is that it is highly sensitive to noise, and large errors in the estimates are observed in the presence of even small amounts of noise. Also it is seen that the errors in estimating the translation parameters are greater than that of the error in estimating the rotation parameter.

TABLE – T 3.2

Effect of increasing the number of points in THLAM.

INPUT:

$\alpha = 52^\circ$ $\beta = 75^\circ$ $\gamma = 41.0123^\circ$

$\theta = 78^\circ$

$t_x = 12$ unit $t_y = 10$ unit $t_z = 1$ unit.

Noise = (−0.001, +0.001)

	N=8	N=10	N=12	N=14	N=16
Angle of (1)	77.7849	77.6766	77.9344	77.8052	77.9926
rotation (2)	78.1111	76.6628	78.2658	78.1621	77.7860
θ (3)	78.2974	80.3888	78.7878	77.8807	78.0587
mean	*78.06447*	*78.24273*	*78.32933*	*77.94933*	*77.94577*
std.deviation	*0.21181*	*1.57293*	*0.35128*	*0.15357*	*0.11615*
(1)	16.1321	15.2118	11.6482	13.6703	10.5431
translation (2)	16.5850	10.7275	12.5836	13.3381	11.2582
t_x/t_z (3)	12.0155	24.1241	14.0636	11.7172	13.8487
mean	*14.91087*	*16.68780*	*12.76513*	*12.90853*	*11.8833*
std.deviation	*2.05567*	*5.56783*	*0.99440*	*0.85325*	*1.42006*
(1)	8.8221	8.9619	10.6190	9.1599	11.0432
translation (2)	6.4968	8.9035	8.4162	12.2926	9.8309
t_y/t_z (3)	10.4333	7.2769	9.1021	10.0290	9.8887
mean	*8.58407*	*8.38077*	*9.37910*	*10.49383*	*10.25427*
std.deviation	*1.61586*	*0.78092*	*0.92037*	*1.32048*	*0.55836*

It was pointed out in [8] that, " when the motion is small, (X_i' , Y_i') is close to (X_i, Y_i) and therefore the columns of the coefficient matrix in (3.11) become nearly dependent , causing the condition number of the matrix and the error for the essential parameters to be very large." The condition number of a matrix is defined as the ratio of the greatest to the least eigen values. However, it is not clear why the translation estimation error is more than that of rotation.

In the next section we present an improved method for the estimation of the translation parameters. The basic idea is to use the less sensitive rotation parameters found from the Linear algorithm to compensate for the rotation and transform the points to an intermediate set of virtual points in 3-D space where the data appears to be from a pure translation. This way we can reduce the error in the estimation of the translation parameters. We should bear in mind that the THLAM algorithm is limited to a general motion situation in which $t_z \neq 0$. Similarly the modified algorithm is limited to this condition. This is because the rotation parameters are found from the THLAM algorithm.

3.3 AN IMPROVED METHOD FOR ESTIMATING THE TRANSLATION PARAMETERS:

As stated already, in this modification we will be using the less sensitive rotation parameter estimated from the linear algorithm. The formulation is as given below.

3.3.1 Formulation:

Let us take the basic motion equation in (3.1).

$$x' = \mathbf{R}\, x + T \tag{3.1}$$

If we pre-multiply (3.1) by $\mathbf{R}^T$ and rearrange we get

$$\mathbf{R}^T \mathrm{T} = \mathbf{R}^T \mathbf{x}' - \mathbf{x} \tag{3.17}$$

Expanding the above equation we obtain,

$$\begin{aligned} r_{11}t_x + r_{21}t_y + r_{31}t_z &= r_{11}x' + r_{21}y' + r_{31}z' - x = U \\ r_{12}t_x + r_{22}t_y + r_{32}t_z &= r_{12}x' + r_{22}y' + r_{32}z' - y = V \\ r_{13}t_x + r_{23}t_y + r_{33}t_z &= r_{13}x' + r_{23}y' + r_{33}z' - z = W \end{aligned} \tag{3.18}$$

which may be written as

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.19}$$

where C_{ij} is the cofactor of the ij th element of the matrix formed by the coefficients of r_{ij} in (3.18). However, the above equations represent a relation in 3-D space coordinates only. As far as the image of the 3-D objects is concerned , all we have are the image coordinates of a few feature points of the object in the scene in two image frames. Hence solving (3.18) is not straightforward. We now give a method for finding the translation parameters(up to a scale factor) from the given image coordinates.

Let $A_1(X_1,Y_1)$ $(A_1'(X_1',Y_1'))$ and $A_2(X_2,Y_2)$ $(A_2'(X_2',Y_2'))$ be two corresponding pairs of image points before (after) motion.

For image points A_1 and A_1' we relate the space coordinates as

$$\begin{aligned} X_1 &= x_1/z_1 & X_1' &= x_1'/z_1' \\ Y_1 &= y_1/z_1 & Y_1' &= y_1'/z_1' \end{aligned} \tag{3.20}$$

assuming that the focal length of the imaging system is unity as before . We can

write similar expressions for A_2 and A_2'.

Using equation (3.18) we write for points A_1 and A_1',

$$U_1 = r_{11} x_1' + r_{21} y_1' + r_{31} z_1' - x_1$$

and substituting equations (3.20) we get,

$$U_1 = z_1' \sigma_1 - z_1 X_1 \tag{3.21 a}$$

where

$$\sigma_1 = r_{11} X_1' + r_{21} Y_1' + r_{31} . \tag{3.21 b}$$

Writing in a similar way the expression for V_1 is

$$V_1 = z_1' \sigma_2 - z_1 Y_1 \tag{3.22 a}$$

where

$$\sigma_2 = r_{12} X_1' + r_{22} Y_1' + r_{32} . \tag{3.22 b}$$

And the expression for W_1 is

$$W_1 = z_1' \sigma_3 - z_1 \tag{3.23 a}$$

where

$$\sigma_3 = r_{13} X_1' + r_{23} Y_1' + r_{33} . \tag{3.23 b}$$

Now from (3.19)

$$t_x = C_{11} U_1 + C_{12} V_1 + C_{13} W_1$$

and substituting for U_1,V_1 and W_1 we obtain,

$$t_x = \lambda_1 z_1' - \lambda_2 z_1 \tag{3.24}$$

where

$$\lambda_1 = C_{11} \sigma_1 + C_{12} \sigma_2 + C_{13} \sigma_3$$

$$\lambda_2 = C_{11} X_1 + C_{12} Y_1 + C_{13}$$

Similarly for t_y and t_z ,

$$t_y = \mu_1 z_1' - \mu_2 z_1 \tag{3.25}$$

$$t_z = \nu_1 z_1' - \nu_2 z_1 \tag{3.26}$$

where

$$\mu_1 = C_{21}\,\sigma_1 + C_{22}\,\sigma_2 + C_{23}\,\sigma_3$$

$$\mu_2 = C_{21}\,X_1 + C_{22}\,Y_1 + C_{23}$$

$$\nu_1 = C_{31}\,\sigma_1 + C_{32}\,\sigma_2 + C_{33}\,\sigma_3$$

$$\nu_2 = C_{31}\,X_1 + C_{32}\,Y_1 + C_{33}$$

Since we are using perspective projection we can find the translation parameters only up to a scale factor; hence solving we get

$$t_x/t_z = B_1\,t_y/t_z + B_2 \tag{3.27}$$

where

$$B_1 = \left(\frac{\lambda_1\nu_2 - \nu_1\lambda_2}{\nu_2\mu_1 - \nu_1\mu_2}\right) \text{ and } B_2 = \left(\frac{\lambda_2\mu_1 - \lambda_1\mu_2}{\nu_2\mu_1 - \nu_1\mu_2}\right)$$

Using A_2 and A_2' we get another equation

$$t_x/t_z = B_3\,t_y/t_z + B_4 \tag{3.28}$$

(3.27) and (3.28) may be solved to give

$$t_x/t_z = (\,B_1\,B_4 - B_2\,B_3\,)/\,(B_1 - B_3) \tag{3.29a}$$

$$t_y/t_z = (\,B_4 - B_2\,)/\,(B_1 - B_3) \tag{3.29.b}$$

where

B_j's are functions of r_{ij} and image points.

Thus using two points in two frames we can find the translational parameters.

From the above equations it is clear that, given the rotation parameters it is possible to compensate for the rotational component and treat the intermediate virtual points as that caused by pure translation. Though it is possible to solve for the translation parameter with two points in two frames, we made use of all the eight points, to give greater accuracy.

3.3.2 Simulation and experimental results:

In order to evaluate the performance of the linear and the modified method we conducted several sets of simulation experiments.The data generation is as explained previously. In Table T.3.3 , we compare the translation parameter estimates obtained by the linear method with those obtained by the modified method. Here the rotation angle and the noise is kept constant and the experiment is repeated for several sets.It can be seen from Table T.3.3 that the performance of the modified algorithm is better than the linear method. Also we can see a consistent performance in the modified method. In order to compare the improvement for various rotation angles we conducted several sets of simulation experiments from 10 to 60 degrees of rotation (Table T.3.4). In most of the cases the modified method improves the translation parameter estimation. For the translation parameter $\frac{t_x}{t_z}$, an improvement of 76% is observed and for $\frac{t_y}{t_z}$ about 90% improvement is achieved.

We have also performed simulation experiments for increasing the number of corresponding points. The results are as shown in Table T.3.5. The rotation parameters are the same as shown in Table T.3.2. The modified algorithm improves the translation parameters. Also the standard deviation in the modified case is much smaller than the linear method, suggesting the results for the modified method is robust. We feel about 12 points are optimum for a better motion estimation. Increasing the number of points further does not increase the performance.

Having found that the modified method performs better, we would like to know for how much percentage of error in rotation the modified method performs better. For that, we conducted the simulation from 5 to 60 degrees. To analyze

TABLE – T 3.3

Comparison between THLAM and the modified method.

Noise level 0.01 fixed.
Rotation angle 10 Degrees fixed.

SET	LINEAR METHOD		MODIFIED METHOD	
	t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	2.4728	3.6242	1.7677	3.2180
2	6.2931	−14.6454	2.2082	2.8806
3	1.5546	3.8349	1.7037	2.3016
4	4.0023	38.6481	1.7614	2.5124
5	1.8171	2.1121	2.1232	2.6337
REF	2.0000	3.0000	2.0000	3.0000

TABLE – T 3.4

Effect of rotation angle on THLAM and the modified method.

Noise level 0.01 fixed.

ROTATION ANGLE	SET	LINEAR METHOD		MODIFIED METHOD	
		t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
20	1	2.0106	3.1466	1.9331	2.9544
	2	3.0197	21.3184	2.9279	3.6291
	3	2.0483	4.1615	2.2896	3.2145
	4	1.9964	3.5106	2.2106	3.0670
30	1	2.0489	2.7748	1.9797	3.1311
	2	3.0712	7.8837	2.9621	3.3984
	3	2.2176	8.0471	2.3237	3.0360
	4	1.7810	2.2947	1.9700	3.1225
45	1	1.9650	2.7892	2.4110	4.0403
	2	2.5091	3.6035	2.3113	3.0304
	3	0.3301	0.9598	0.7252	2.1152
	4	3.3233	174.4913	2.2255	2.0566
60	1	1.3005	1.2815	1.3853	2.4201
	2	1.4748	2.2058	1.5993	2.4527
	3	2.3443	−1.9673	2.6108	0.8199
	4	2.5006	3.3452	1.9604	2.6394
REF		2.0000	3.0000	2.0000	3.0000

Comments: For t_x/t_z an improvement is seen in 16 out of 21 cases (76%)
For t_y/t_z an improvement is seen in 19 out of 21 cases (90%)

Note: The simulation results of Table–T 3.3 is also taken in the above percentage calculation.

TABLE – T 3.5

Effect of increasing the number of points on modified method.

INPUT:
$\alpha = 52^{\circ}$ $\beta = 75^{\circ}$ $\gamma = 41.0123^{\circ}$
$\theta = 78^{\circ}$
$t_x = 12$ unit $t_y = 10$ unit $t_z = 1$ unit.
Noise = (−0.001, +0.001)

	N=8	N=10	N=12	N=14	N=16
Angle of (1)	77.7849	77.6766	77.9344	77.8052	77.9926
rotation (2)	78.1111	76.6628	78.2658	78.1621	77.7860
θ (3)	78.2974	80.3888	78.7878	77.8807	78.0587
mean	*78.06447*	*78.24273*	*78.32933*	*77.94933*	*77.94577*
std.deviation	*0.21181*	*1.57293*	*0.35128*	*0.15357*	*0.11615*
translation (1)	11.23204	11.08623	12.01908	11.53064	12.20681
(modified) (2)	12.01573	13.01615	11.93732	12.12969	11.03116
t_x/t_z (3)	12.11865	8.50831	12.21090	12.01656	11.39240
mean	*11.78881*	*10.87023*	*12.05577*	*11.89230*	*11.54346*
std.deviation	*0.39593*	*1.84665*	*0.11416*	*0.29875*	*0.49170*
translation (1)	9.27550	9.15607	10.02783	9.56349	10.19862
(modified) (2)	9.89963	10.84826	9.92496	10.09764	9.22562
t_y/t_z (3)	10.06898	6.95154	10.09837	10.02778	9.45955
mean	*9.74804*	*8.98529*	*10.01705*	*9.89630*	*9.62793*
std.deviation	*0.34121*	*1.59406*	*0.07120*	*0.23706*	*0.41469*

Note : See Table– T 3.2

the feasibility of this method we have conducted several simulations for three cases.

Case 1: small translation. (typical values used in the experiments are t_x= 0.1 unit, t_y = 0.1 unit and t_z = 1.0unit) .

Case 2: medium translation (typical values used in the experiments are t_x= 8 units , t_y = 6 units and t_z= 2.0 units).

Case 3: large translation (typical values used in the experiments are t_x = 12 units, t_y = 10 units , and t_z = 1.0 unit).

For each of the set, we calculated the percentage error in theta, the rotation angle, estimated from the linear algorithm. We have also calculated the percentage error in estimating the translation parameters for both methods. This error is a function of rotation angle, the direction cosines of the rotation axis (α, β and γ) and the amount of noise present in the input data itself. This functional dependence on different variables made it difficult to have a precise formula which can be applied to compensate for the rotation. However, we have compared the performance of the two methods with respect to the rotation angle error alone.

A few results are tabulated in Tables (T.3.6–T.3.14) for each of the three cases mentioned above. The error in the estimated translation parameter increases with the error in the rotation angle error. When the rotation angle error percentage is within ±5% the modified method in general performs better than the linear method.This error percentage range also varies for the type of translation. From the graphs it is clear that the modified method performs well in all the cases. The error percentage in the estimated parameters are also not as high as in the case of linear translation parameter estimation. Even this failure to perform ideally may be due to the errors in the direction cosines of the rotation

TABLE –T 3.6

Performance of THLAM and the Modified method for (small translation case: θ=5 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 5.00^{\circ}$ $t_x = 0.10$ $t_y = 0.10$ $t_z = 1.00$

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	66.1281	65.4545	35.4473	4.8207	-2.9501	-6.4768	0.2714	-0.0469
				(-3.586)[1]	(-3050.1)	(-6576.8)	(171.41)	(-146.88)
2	46.0256	71.1846	49.9592	3.9555	4.1067	2.4212	0.2893	0.5260
				(-20.89)	(4006.7)	(2321.2)	(189.32)	(426.00)
3	47.5661	67.1877	51.0961	7.5394	-0.5793	-0.5056	0.5320	0.3186
				(50.788)	(-679.3)	(-605.6)	(431.96)	(218.64)
4	36.1237	77.9372	56.5475	6.7588	0.8242	1.8948	0.3230	0.9524
				(35.176)	(724.2)	(1794.8)	(223.0)	(852.4)
5	39.6104	73.4930	55.1974	5.5192	0.8138	0.7130	0.2119	0.4759
				(10.384)	(713.8)	(613.0)	(111.92)	(375.9)
6	26.9888	80.2389	65.1049	6.4905	0.6655	0.4923	0.4604	0.7298
				(29.81)	(565.0)	(392.3)	(360.42)	(629.8)
7	35.6314	76.2696	57.8577	6.6056	0.5537	0.3060	0.6675	0.5862
				(32.112)	(453.7)	(206.0)	(567.45)	(486.16)
8	51.4965	57.4763	55.3446	5.6839	0.2818	0.1936	0.0879	0.2544
				(13.678)	(181.8)	(93.6)	(-12.11)	(154.41)
9	30.6407	85.3672	59.7873	7.4646	0.1261	-0.1628	-0.0291	-0.0458
				(49.292)	(26.1)	(−262.8)	(−129.1)	(−145.8)
10	106.4889	78.6815	20.1865	3.8221	0.0156	0.0136	0.1270	-0.1588
				(-23.558)	(-84.4)	(-86.4)	(26.95)	(-258.77)

[1] all values denoted in brackets in this table and the following tables represent the relative error percentage of the corresponding parameters.

TABLE – T 3.7

Performance of THLAM and the Modified method for (small translation case: θ=15 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 15.00^{\circ}$ $t_x = 0.10$ $t_y = 0.10$ $t_z = 1.00$

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	111.5368	81.5837	23.2785	11.4947	-1.2026	-0.2496	0.1933	-0.1732
				(-23.37)	(-1302.6)	(-349.6)	(93.9)	(-27.32)
2	61.9988	62.6781	41.0392	14.4305	-2.3652	-6.9503	0.3244	0.0022
				(-3.795)	(-2465.2)	(-7050.3)	(224.41)	(-97.78)
3	58.8794	63.3697	43.1674	13.6280	4.2141	2.3228	0.0944	0.0458
				(-9.147)	(4114.1)	(2222.8)	(-5.63)	(-54.25)
4	51.8154	70.8571	44.4099	14.1664	1.5728	2.0257	0.7822	0.6101
				(-5.557)	(1472.8)	(1925.7)	(682.16)	(510.1)
5	56.0684	57.7361	50.5671	17.9607	-0.0623	0.0361	0.0651	0.0167
				(19.738)	(-162.3)	(-63.9)	(-34.87)	(-83.31)
6	53.3870	65.0056	46.9629	17.3516	-0.6107	-0.5677	0.7982	0.5833
				(15.677)	(-710.7)	(-667.7)	(698.15)	(483.28)
7	49.2681	68.2959	48.5930	15.8242	0.6070	0.4338	0.5197	0.7764
				(5.495)	(507.0)	(333.8)	(419.69)	(676.40)
8	48.0165	68.7632	49.5254	15.2971	0.5221	2.8124	0.2966	1.3979
				(1.981)	(422.1)	(2712.4)	(196.63)	(1297.97)
9	51.4824	69.2456	45.7677	14.6747	1.2655	1.4578	0.8273	0.9129
				(-2.169)	(1165.5)	(1357.8)	(727.34)	(812.92)
10	43.9271	65.9245	55.8663	16.4454	1.5038	1.0519	0.7799	0.7724
				(9.636)	(1403.8)	(951.9)	(679.97)	(672.37)

TABLE –T 3.8

Performance of THLAM and the Modified method for (small translation case: θ=25 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 25^{\circ}$ $t_x = 0.100$ $t_y = 0.100$ $t_z = 1.00$

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	57.2157	52.5605	54.4991	29.3716	-0.0187	-0.0460	-0.0153	-0.2633
				(17.486)	(-118.7)	(-146)	(-115.29)	(163.23)
2	60.4221	62.7234	42.3419	24.3966	-2.3228	-12.7770	0.4458	0.0766
				(-2.4136)	(-2422.8)	(-12877)	(345.75)	(-23.39)
3	59.8630	62.8333	42.7367	23.6145	3.0177	1.9469	-.0132	-.0123
				(-5.542)	(2917.7)	(1846.9)	(-113.18)	(-112.34)
4	64.3871	57.4960	43.6030	23.5005	-0.7343	-1.3979	0.0792	0.0632
				(-5.998)	(-834.3)	(-1497.9)	(-20.79)	(-36.82)
5	51.9457	68.3485	45.9221	29.2778	-0.3183	-0.5623	-0.0016	0.6850
				(17.111)	(-418.3)	(-662.3)	(-101.63)	(585.0)
6	54.7760	64.1739	46.2868	26.1930	-0.4516	-0.3814	0.1735	0.2111
				(4.772)	(-551.6)	(-481.4)	(73.54)	(111.13)
7	53.2475	65.8313	46.4710	25.4021	0.6551	0.5493	0.4105	0.7223
				(1.61)	(555.1)	(449.3)	(310.51)	(622.3)
8	54.6328	65.8582	45.1320	24.4182	1.3939	2.0513	0.7727	1.0239
				(-2.3272)	(1293.9)	(1951.3)	(672.69)	(923.94)
9.	51.3079	66.9005	47.5655	26.3483	0.7364	0.8353	0.2571	0.4518
				(5.3932)	(636.4)	(735.3)	(157.1)	(351.82)
10	50.4337	64.7217	50.0727	25.5155	1.4463	1.5032	0.5659	0.8359
				(2.062)	(1346.3)	(1403.2)	(465.86)	(735.93)

TABLE –T 3.9

Performance of THLAM and the Modified method for (medium translation case: θ=5 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 5^{\circ}$ $t_x = 8.00$ $t_y = 6.00$ $t_z = 2.00$

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	67.7390	57.9686	40.6759	7.0967	4.4432	2.5768	3.9073	2.7010
				(41.934)	(11.08)	(-14.107)	(-2.318)	(-9.968)
2	58.0372	50.1689	56.1982	5.0427	3.6865	2.5873	3.8252	2.9480
				(0.854)	(-7.838)	(-13.757)	(-4.370)	(-1.734)
3	56.5588	57.2706	50.5357	5.2048	4.2766	2.8280	3.9723	3.0503
				(4.096)	(6.915)	(-5.733)	(0.692)	(1.678)
4	49.8515	49.6776	65.9913	6.1043	5.1864	2.4643	3.7998	3.1484
				(22.086)	(29.66)	(-17.857)	(-5.005)	(4.946)
5	58.1114	60.6032	46.1466	5.5979	5.0440	2.7284	4.2158	3.1017
				(11.958)	(26.1)	(-9.053)	(5.396)	(3.389)
6	60.4433	64.5716	40.8429	4.8498	4.2924	3.1446	4.2045	3.0692
				(-3.004)	(7.31)	(4.82)	(5.111)	(2.308)
7	59.7839	50.3752	54.3319	5.6545	3.6891	2.4470	3.7170	2.8355
				(13.09)	(7.773)	(-18.433)	(7.076)	(-5.485)
8	47.2450	55.8719	61.7267	6.1423	5.5272	3.1228	4.0289	3.2299
				(22.846)	(38.18)	(4.093)	(0.722)	(7.665)

TABLE –T 3.10

Performance of THLAM and the Modified method for (medium translation case:θ=15 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 15.00^{\circ}$ $t_x = 8.00$ $t_y = 6.00$ $t_z = 2.00$

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	61.3591	61.2431	42.7746	15.1808	4.2303	2.8087	3.9971	2.8687
				(12.05)	(5.76)	(-6.38)	(-0.07)	(-4.38)
2	60.3768	62.4675	42.5909	14.5858	3.4402	3.4298	3.9618	2.8769
				(-2.76)	(-13.99)	(14.33)	(-0.96)	(-4.10)
3	56.2663	61.5151	47.0561	14.7994	3.7773	2.9305	4.0084	3.0173
				(-1.34)	(-5.57)	(-2.32)	(0.21)	(0.58)
4	70.0230	55.1406	41.7504	18.0218	6.9601	1.8753	3.4479	2.4419
				(20.15)	(74.00)	(-37.49)	(-13.80)	(-18.60)
5	48.0992	78.5826	44.1520	20.4480	12.1848	7.3407	6.2585	4.7284
				(36.32)	(204.62)	(144.69)	(56.46)	(57.61)
6	58.3084	65.4147	42.0777	17.7556	4.7131	2.6296	4.5886	3.3277
				(18.37)	(17.83)	(-12.35)	(14.72)	(10.92)
7	60.5902	60.4879	44.0714	15.1606	3.9964	3.2463	3.9628	2.9490
				(1.07)	(-0.09)	(8.21)	(-0.93)	(-1.70)
8	65.6123	48.5625	51.2651	17.3420	2.4203	2.8642	3.1023	2.4038
				(15.61)	(-39.49)	(-4.53)	(-22.44)	(-19.87)
9	56.0815	68.7897	41.6851	14.1089	5.1044	2.9386	4.7212	3.4322
				(-5.94)	(27.61)	(-2.05)	(18.03)	(14.41)

TABLE –T 3.11

Performance of THLAM and the Modified method for (medium translation case: θ=25 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 25^{\circ}$ t_x= 8.00 t_y = 6.00 t_z= 2.00

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	61.0466	61.2052	43.0712	25.1331	4.1298	2.8175	4.0660	2.8912
				(0.53)	(3.25)	(-6.42)	(1.65)	(-3.63)
2	60.0466	57.7488	46.6081	24.9843	3.3832	2.3709	3.5438	2.6678
				(-0.06)	(-15.42)	(-20.97)	(-11.40)	(-11.07)
3	63.3984	56.9087	44.9198	23.9455	4.7751	2.5385	3.3684	2.4400
				(-4.22)	(19.38)	(-15.38)	(-15.79)	(-19.67)
4	63.5439	51.4684	49.9836	27.3020	2.2551	2.6674	3.0587	2.3942
				(9.21)	(-43.62)	(-11.09)	(-23.53)	(-20.19)
5	60.9559	56.7052	47.1240	25.3859	3.3477	1.8335	3.7072	2.7810
				(1.54)	(-16.31)	(-38.88)	(-7.32)	(-7.30)
6	60.3576	61.6503	43.2858	24.5973	3.5905	3.6945	4.1579	3.0466
				(-1.61)	(-10.24)	(23.15)	(3.95)	(1.55)
7	57.8836	60.4684	46.4672	24.8305	3.5520	3.1058	4.0970	3.0906
				(-0.68)	(-11.20)	(3.53)	(2.43)	(3.02)
8	53.2828	70.0476	43.5033	29.3988	8.1425	5.4023	6.3645	4.8713
				(9.60)	(103.56)	(80.08)	(59.11)	(62.38)
9	66.9093	54.1454	44.8219	27.2985	5.4347	1.7455	3.2838	2.3556
				(9.19)	(35.87)	(-41.82)	(-17.91)	(-21.48)

TABLE –T 3.12

Performance of THLAM and the Modified method for (large translation case: θ=5 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 5.00^{\circ}$ t_x= 12.0 t_y = 10.0 t_z= 1.00

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	65.4564	58.8469	41.5639	7.4695	23.9519	6.8473	12.0327	9.8198
				(49.39)	(88.12)	(-31.527)	(0.272)	(-1.802)
2	59.1166	70.4366	37.7964	4.5909	13.8339	16.4146	13.7859	11.2667
				(-8.182)	(15.283)	(64.146)	(14.883)	(12.668)
3	55.4281	44.9232	65.1527	6.7167	18.6950	4.5798	9.1374	8.0789
				(34.334)	(55.792)	(-54.202)	(-23.855)	(-19.211)
4	56.3784	53.0652	54.7971	5.4338	12.6049	7.2598	11.9771	10.1739
				(8.676)	(5.041)	(-27.402)	(-0.191)	(1.739)
5	59.3710	59.3493	46.1156	4.7912	12.9725	9.2523	12.4809	10.3457
				(-4.176)	(8.104)	(-7.477)	(4.007)	(3.457)
6	61.3128	51.3008	52.0224	5.5310	8.1676	7.0745	10.6585	8.9206
				(10.62)	(-31.937)	(-29.255)	(-11.179)	(-10.795)
7	56.3145	63.2898	45.5529	5.9304	37.2955	7.7184	14.4128	11.8248
				(18.608)	(210.816)	(-22.816)	(20.106)	(18.248)
8	56.8913	63.2450	45.0587	4.8226	13.1004	9.6741	12.8914	10.7564
				(-3.548)	(9.17)	(-3.259)	(7.428)	(7.564)
9	56.2503	61.3646	47.1977	5.4692	37.4025	7.4390	13.1479	10.9269
				(9.384)	(211.688)	(-25.61)	(9.566)	(9.269)
10	56.1523	60.0087	48.4520	8.0601	-20.0166	4.0953	4.6938	3.9041
				(61.202)	(-266.81)	(-59.047)	(-60.885)	(-60.959)

TABLE –T 3.13

Performance of THLAM and the Modified method for (large translation case: θ=15 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 15.00^{\circ}$ t_x= 12.0 t_y = 10.0 t_z= 1.00

NOISE:

-0.01 to 0.01(random)

SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	62.4909	59.4161	43.4075	17.7312	26.1727	6.5614	12.1461	9.8823
				(18.208)	(118.106)	(-34.386)	(1.218)	(-1.177)
2	58.8278	56.6478	49.0346	14.9413	10.0653	7.8890	10.8963	9.3051
				(0.3913)	(13.874)	(-50.331)	(16.953)	(-15.193)
3	59.1361	60.8094	45.0594	15.2648	18.3454	8.6419	13.5804	11.2082
				(1.765)	(52.878)	(-13.581)	(13.170)	(12.082)
4	58.5842	61.2633	45.1634	15.8805	29.7716	7.5812	13.3929	10.9508
				(5.87)	(148.097)	(-24.188)	(11.607)	(9.508)
5	54.1106	68.4069	43.8016	26.2638	-3.2814	1.9750	2.3067	1.9515
				(75.092)	(-127.345)	(-80.25)	(-80.777)	(-80.485)
6	61.2462	55.9211	47.6037	15.5309	14.8771	5.7918	9.5461	7.9654
				(3.539)	(23.976)	(-42.082)	(-20.449)	(-20.346)
7	59.6956	56.4231	48.4740	15.2409	8.3401	7.9711	10.6663	8.9999
				(1.606)	(-30.499)	(-20.289)	(-11.114)	(-10.001)
8	59.2941	59.3469	46.1853	15.8633	29.1843	7.3721	7.7831	6.4506
				(5.755)	(143.203)	(-26.279)	(-35.141)	(-35.497)
9	58.0480	60.2990	46.4649	16.6461	42.3202	5.5799	3.8828	3.1846
				(10.974)	(252.668)	(-44.281)	(-67.648)	(-68.155)
10	59.8796	62.7390	42.7992	14.5450	11.6942	16.0107	13.1614	10.8211
				(-3.033)	(-2.548)	(60.107)	(9.679)	(8.211)

TABLE –T 3.14

Performance of THLAM and the Modified method for (large translation case: θ=25 °)

INPUT:

$\alpha = 60.00^{\circ}$ $\beta = 60.00^{\circ}$ $\gamma = 45.00^{\circ}$

$\theta = 25^{\circ}$ $t_x = 12.00$ $t_y = 10.00$ $t_z = 1.00$

NOISE:

-0.01 to 0.01(random)

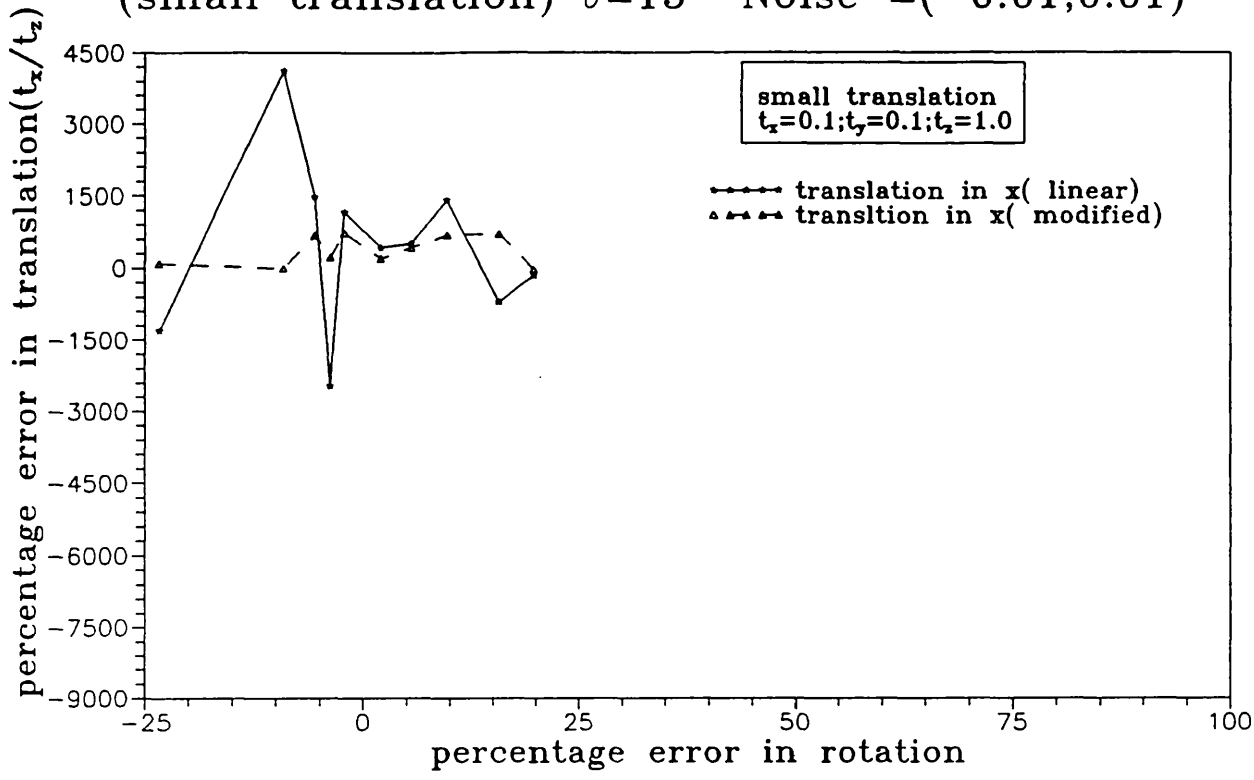
SET	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_x/t_z	t_y/t_z	t_x/t_z	t_y/t_z
1	58.9070	60.1465	45.8302	25.8447	14.4114	7.4288	12.7384	10.6558
				(3.379)	(20.095)	(-25.712)	(6.154)	(6.558)
2	61.7152	59.4114	44.0537	28.0874	26.7220	6.2595	13.5898	11.0200
				(12.350)	(122.683)	(-37.405)	(13.248)	(10.200)
3	59.5920	56.5520	48.4456	26.0316	8.6744	5.6101	10.3824	8.8304
				(4.126)	(-27.713)	(-43.899)	(-13.480)	(-11.696)
4	59.4014	57.4666	47.7730	24.5523	8.8531	8.3471	10.7862	9.2355
				(-1.791)	(-26.224)	(-16.529)	(-10.115)	(-7.646)
5	58.9758	59.4403	46.3839	24.9175	14.1277	8.6447	13.0279	10.9452
				(-0.33)	(17.731)	(-13.553)	(8.566)	(9.452)
6	53.5284	66.8507	45.4521	35.9194	-4.6297	1.8248	-0.6880	-0.5527
				(43.678)	(-138.581)	(-81.752)	(-105.733)	(-105.53)
7	59.4636	57.6837	47.5209	25.1968	9.0267	8.3129	10.0365	8.6517
				(0.787)	(-24.778)	(-16.871)	(-16.362)	(-13.483)
8	59.4890	59.2573	46.0930	25.1124	17.5973	8.9364	11.8606	9.9147
				(0.450)	(46.644)	(-10.636)	(-1.162)	(-0.853)
9	59.8546	56.4967	48.2666	25.6613	16.1826	4.5917	9.0987	7.7971
				(2.645)	(34.855)	(-54.083)	(-24.177)	(-22.030)

axis. In all our simulations noise is kept at a constant value of 0.01. It should also be noted that if the noise in the input data is more, the error in the estimated rotation angle will be more (See Table T.3.1) which in turn affects the estimated translation motion parameters.

Figures (3.3,–3.5) show the effect of error in theta on the translation for Case-1. Figure (3.6) shows the overall performance of the motion estimation algorithms when the rotation angle is varied from 5° to 60°. In figures 3.6, 3.10 and 3.14 the different shaded areas represent the different error regions. We have used absolute values of different error percentages in these figures. A larger area means a large number of simulation experiments yield this result. From the figures it is clear that the modified algorithm works better than the linear algorithm when the error in the rotation angle is within ± 5%. Similar graphs are shown for the other two cases. Figures (3.7,3.8,3.9 and 3.10) show the effect for medium translation. Here Figure (3.10) shows the overall picture for the rotation angles from 5° to 60°. Figures (3.11, 3.12, 3.13 and 3.14) show the effect for large translation. As in the previous case Figure 3.14 shows the overall effect when the rotation angle is varied from 5° to 60° These show the variation of relative percentage error in translation with the relative percentage error in theta , the angle of rotation. It can be seen in both the cases (t_x/t_z and t_y/t_z) that the modified method performs better than the linear method.

In Case-1, a large error in the estimated parameters of the rotation angle and translation is observed. This may be due to the fact that when the translation is very small , the resulting motion will be predominantly rotation. Hence the image points before and after motion are very close. As we have already seen that when the motion is small , (X_i' , Y_i') is close to (X_i, Y_i) and hence the error for

Figure 3.3
Effect of error in theta on translation
(small translation) θ=15° Noise =(−0.01,0.01)



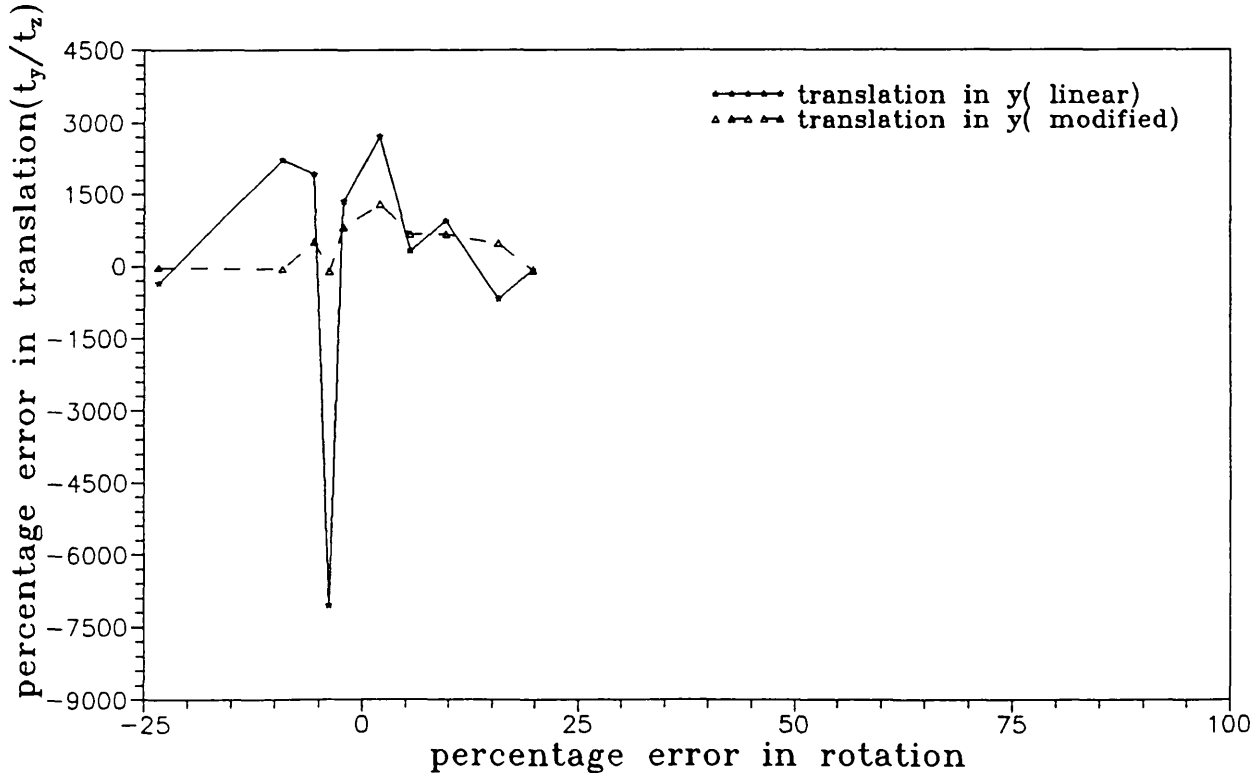
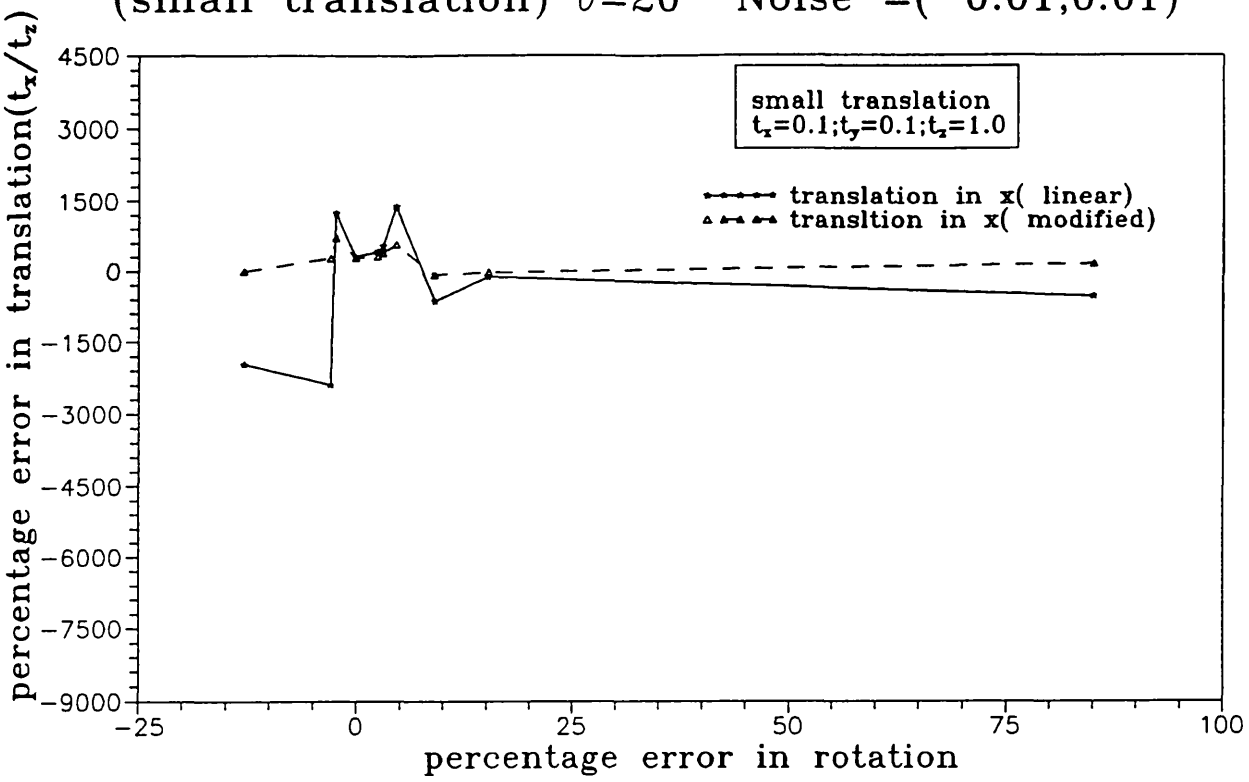


Figure 3.4
Effect of error in theta on translation
(small translation) θ=20° Noise =(−0.01,0.01)



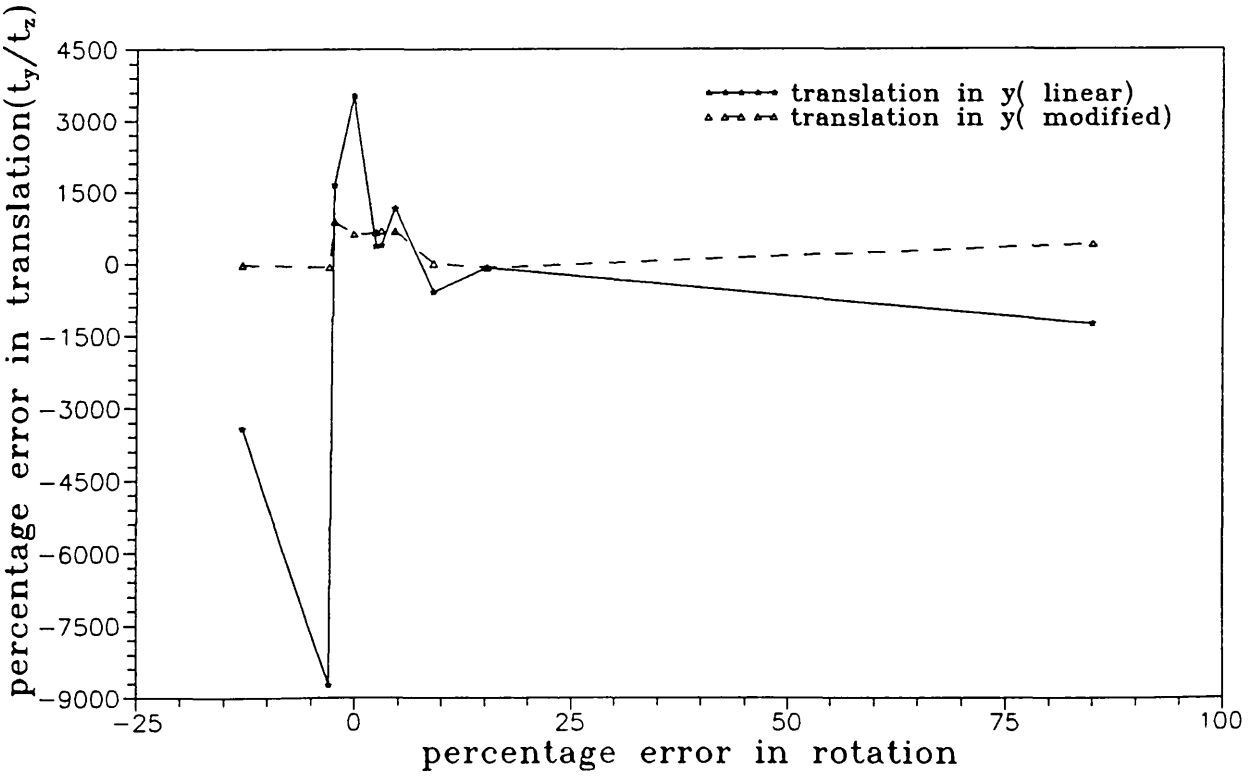
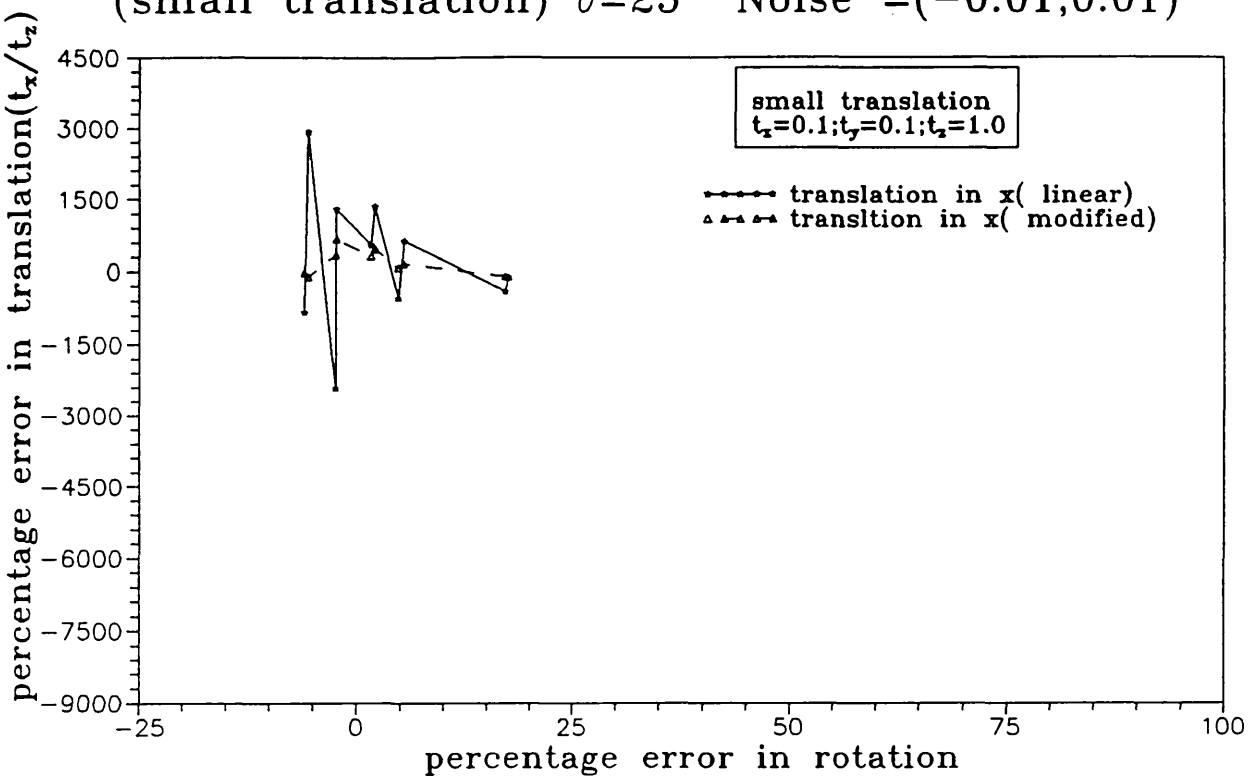


Figure 3.5
Effect of error in theta on translation
(small translation) θ=25° Noise =(−0.01,0.01)



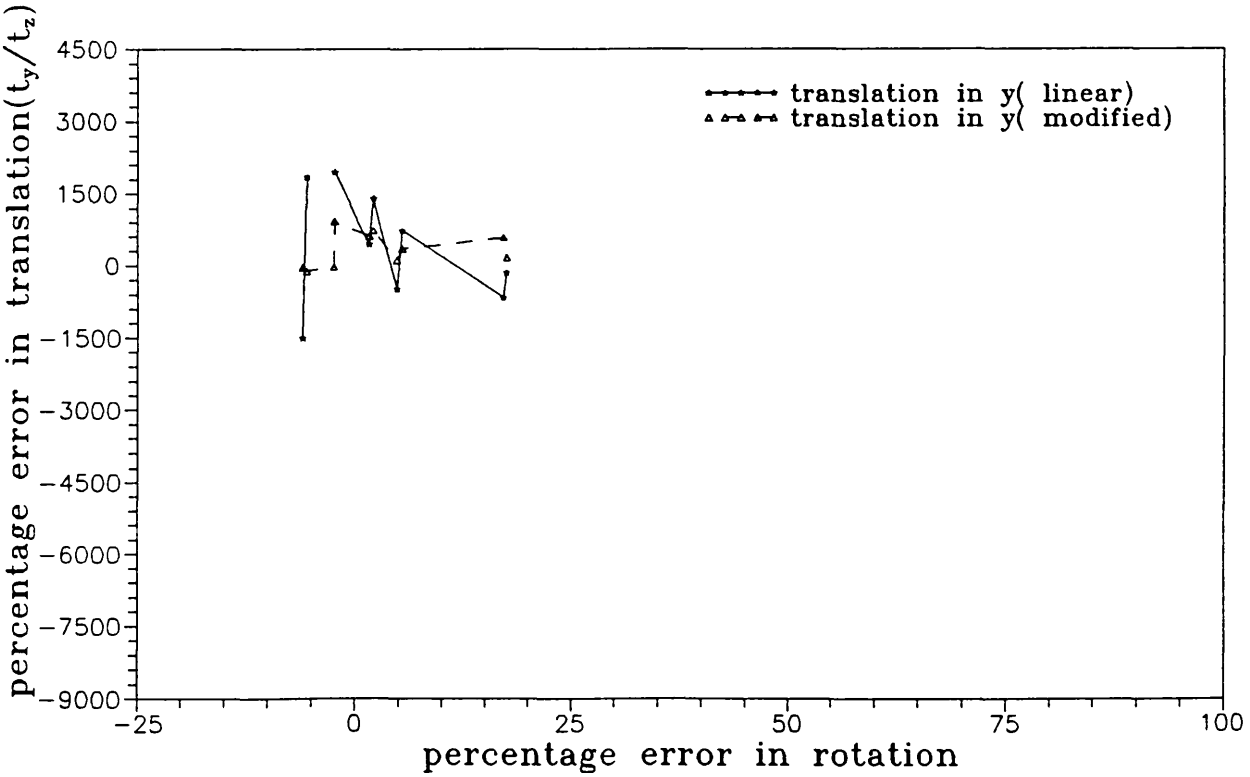
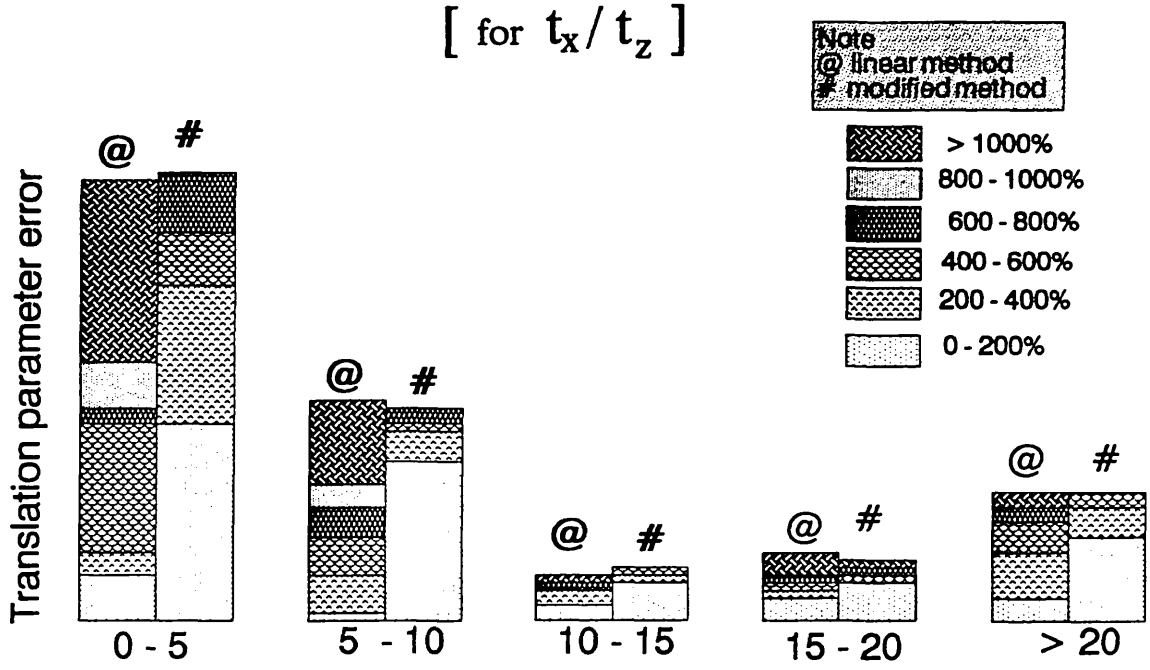


Figure 3.6 Overall performance of motion estimation

(Small translation)

[for t_x / t_z]



[for t_y / t_z]

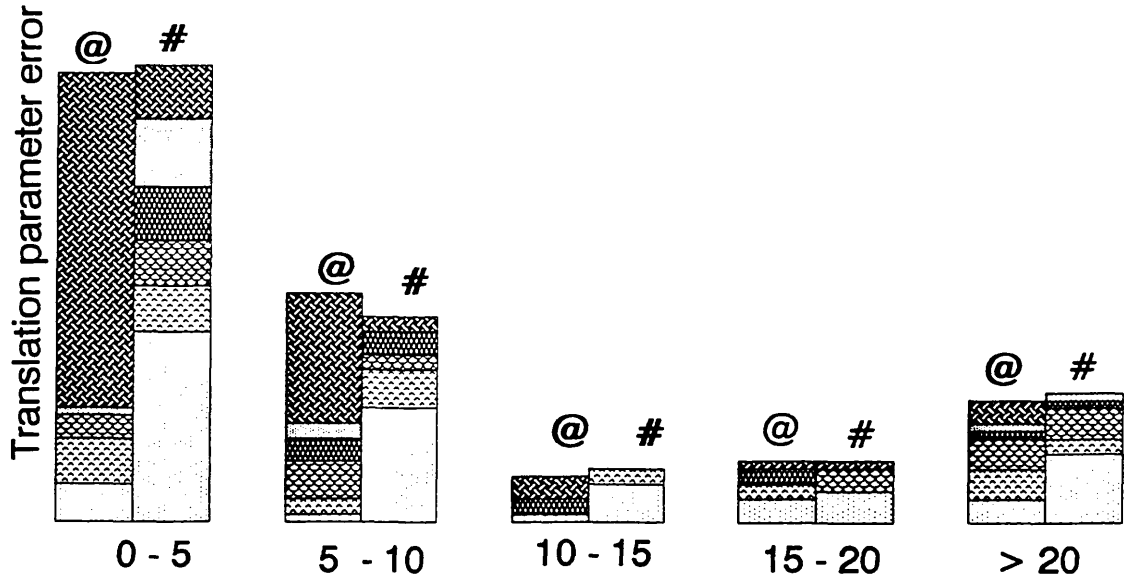


Figure 3.7
Effect of error in theta on translation
(medium translation) θ= 15° Noise =(−0.01,0.01)

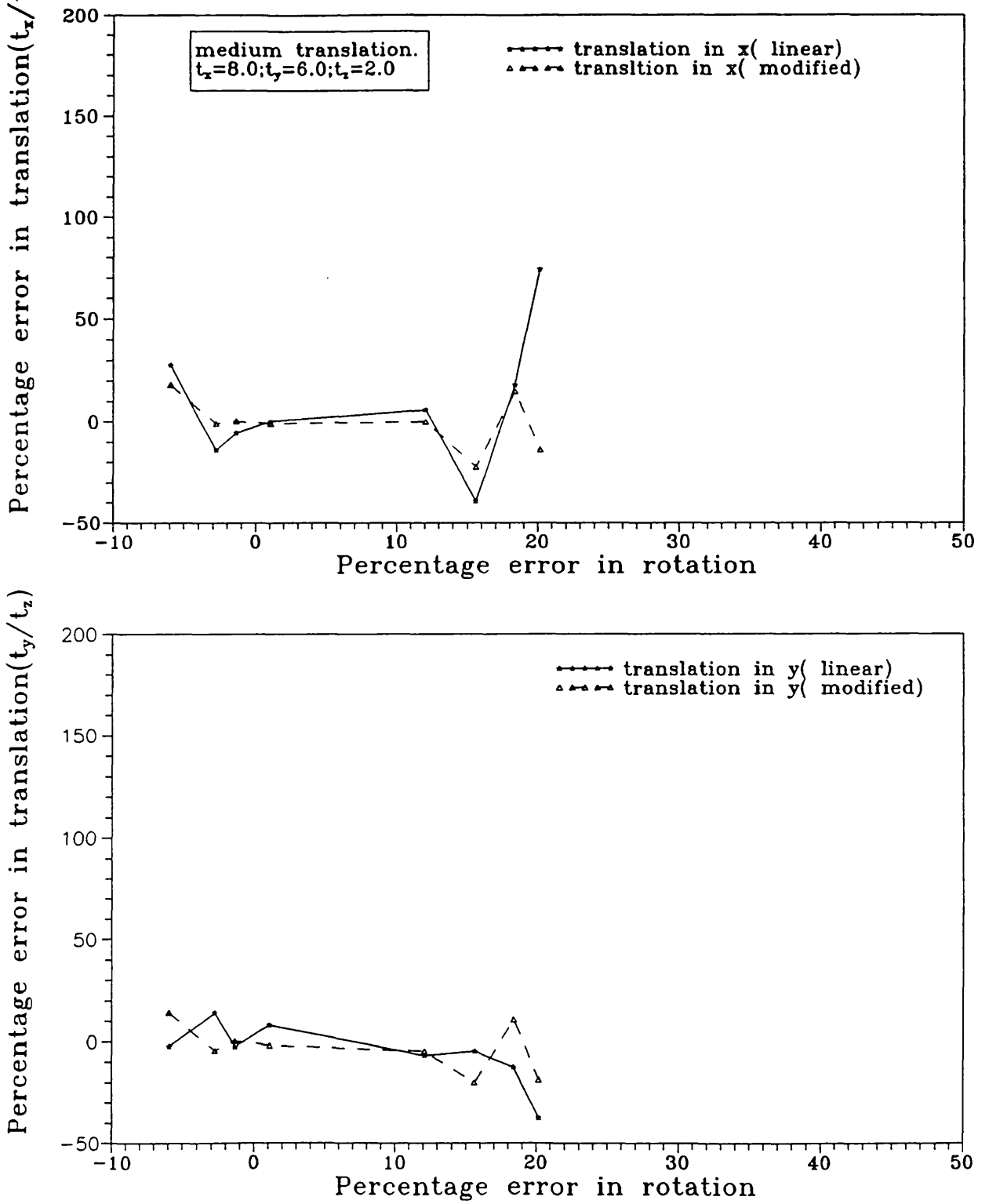
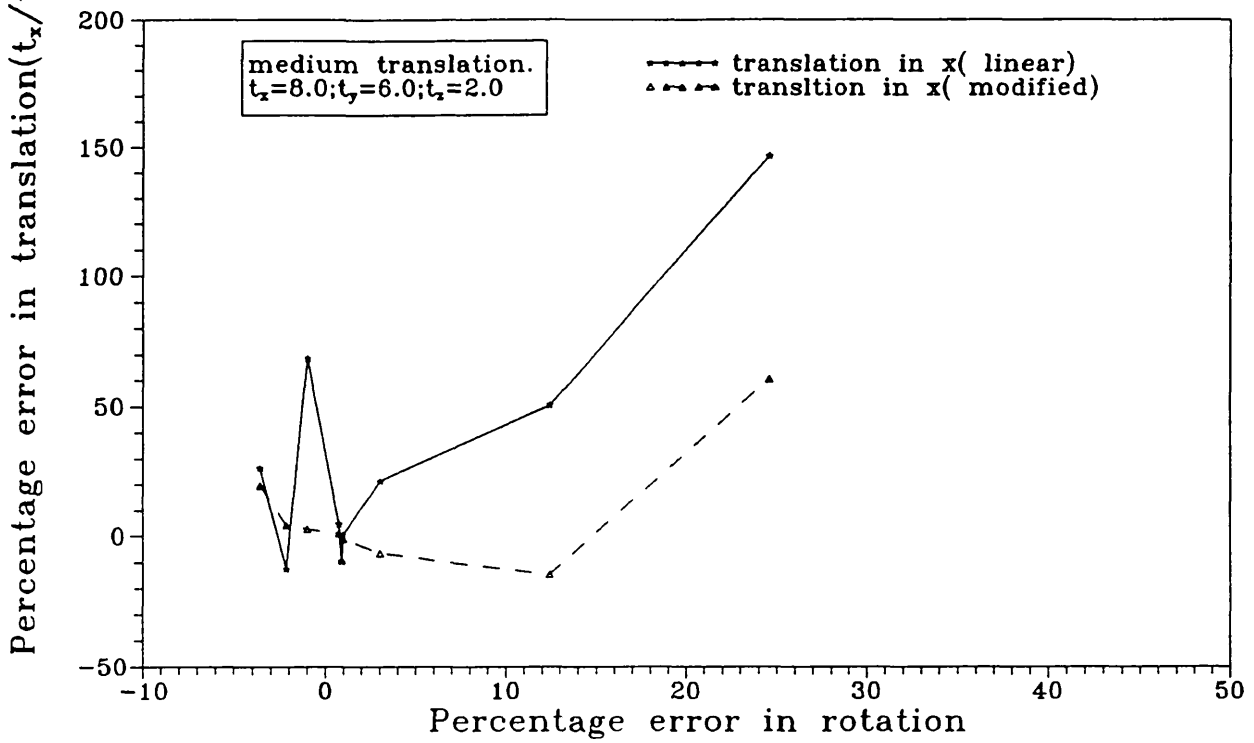


Figure 3.8
Effect of error in theta on translation
(medium translation) θ= 20° Noise =(−0.01,0.01)



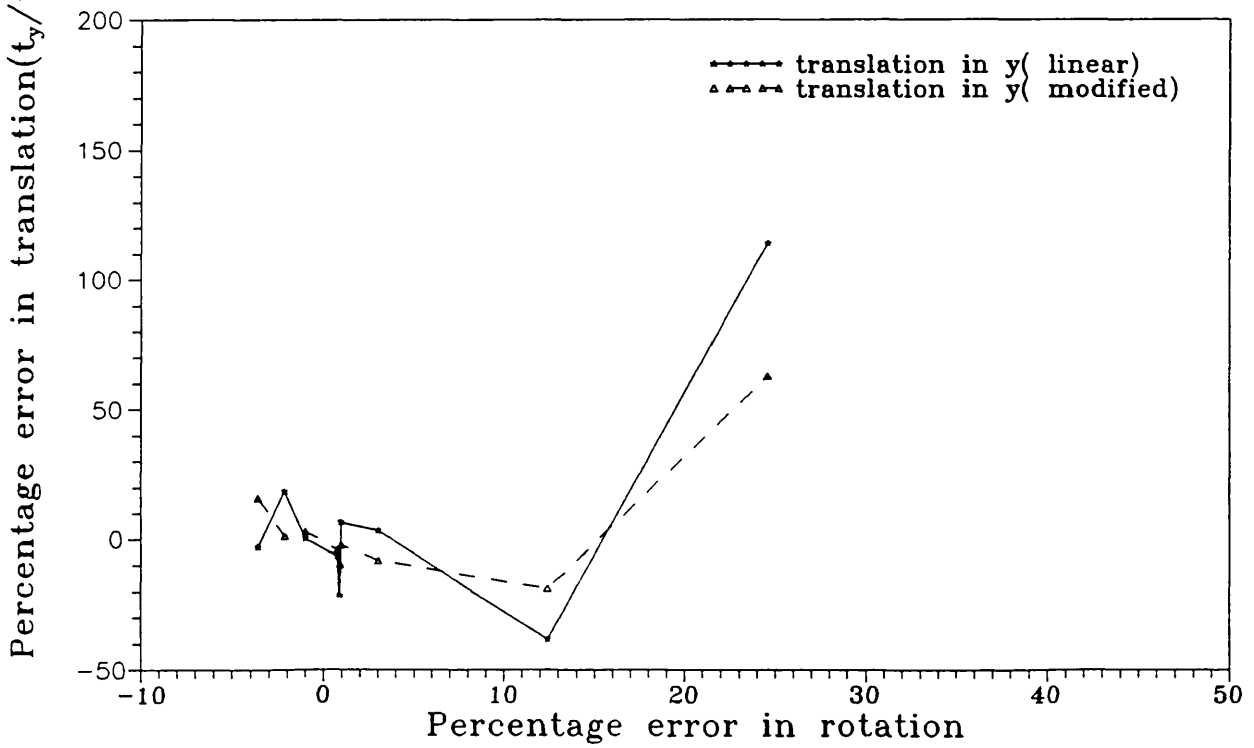
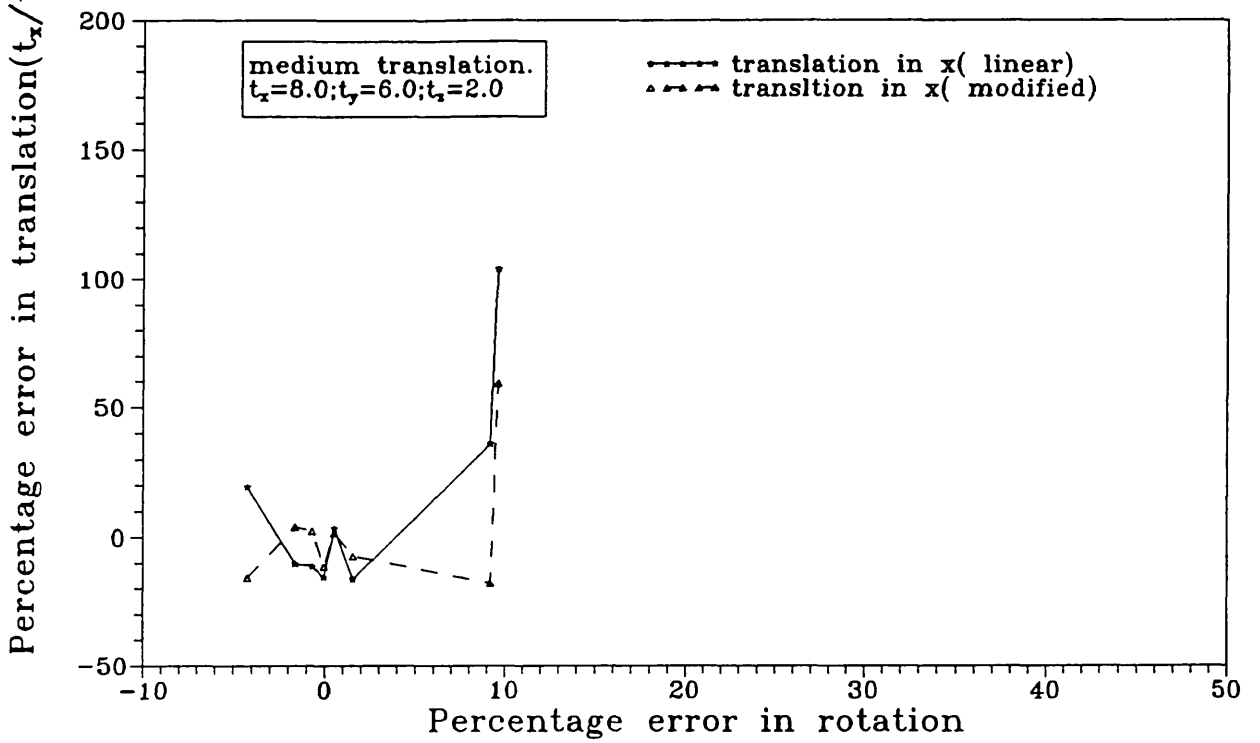


Figure 3.9
Effect of error in theta on translation
(medium translation) θ= 25° Noise =(−0.01,0.01)



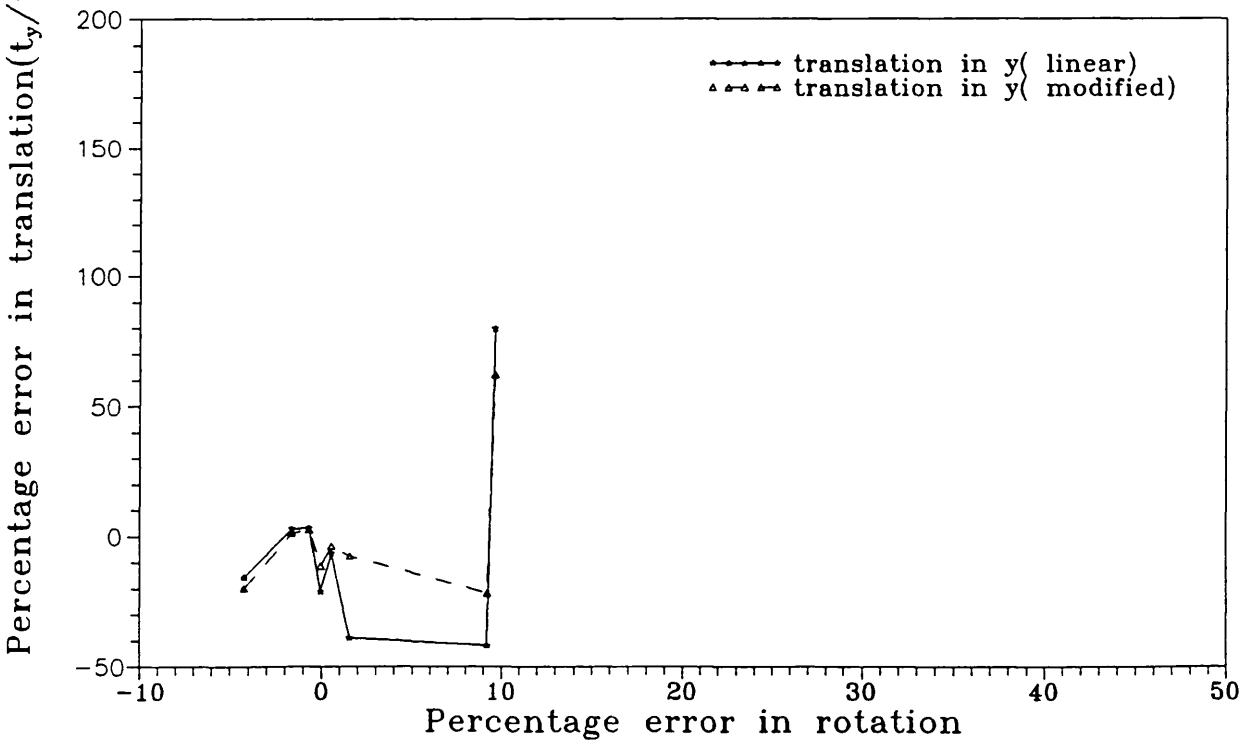


Figure 3.10 Overall performance of motion estimation

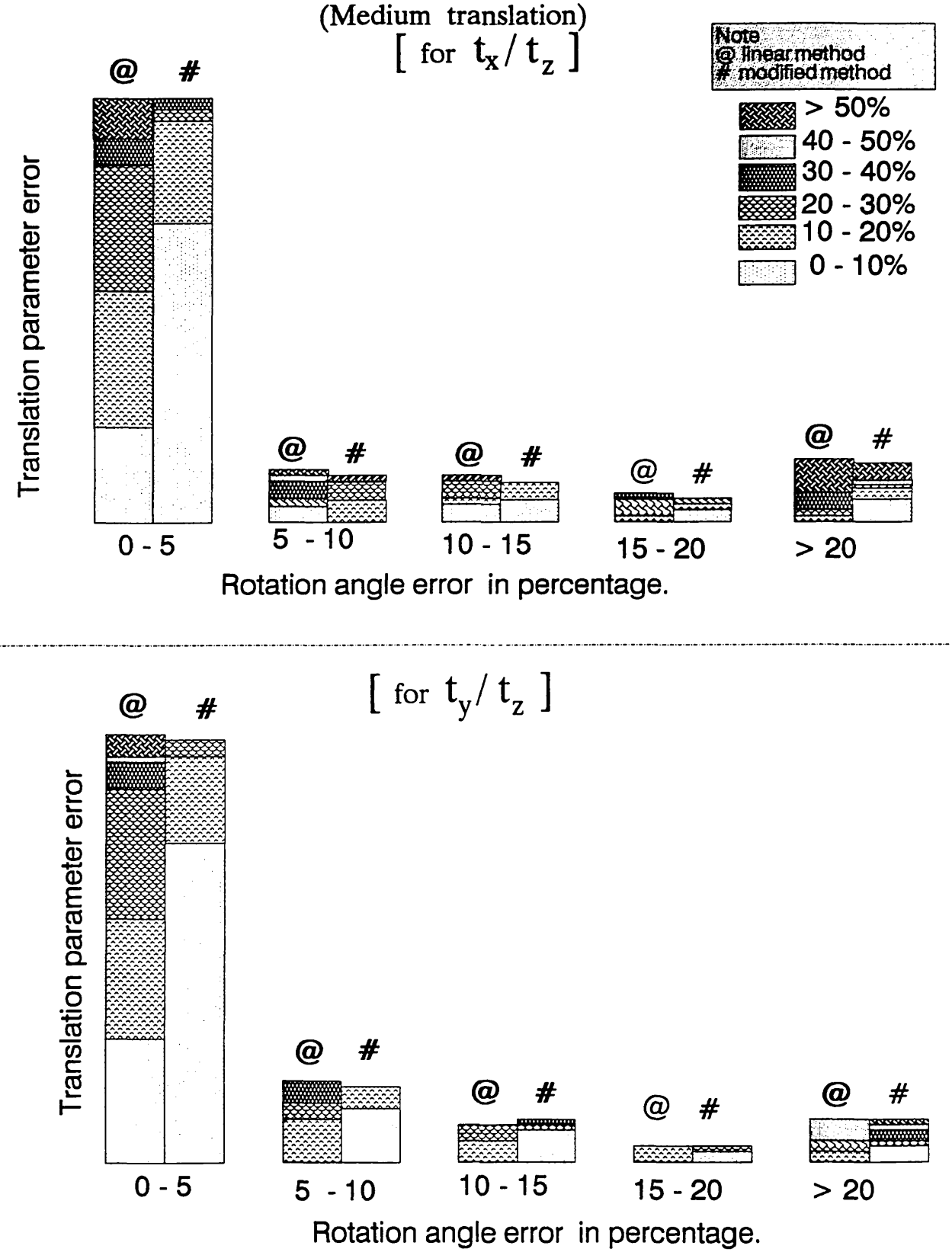


Figure 3.11
Effect of error in theta on translation
(large translation) θ= 15° Noise =(−0.01,0.01)

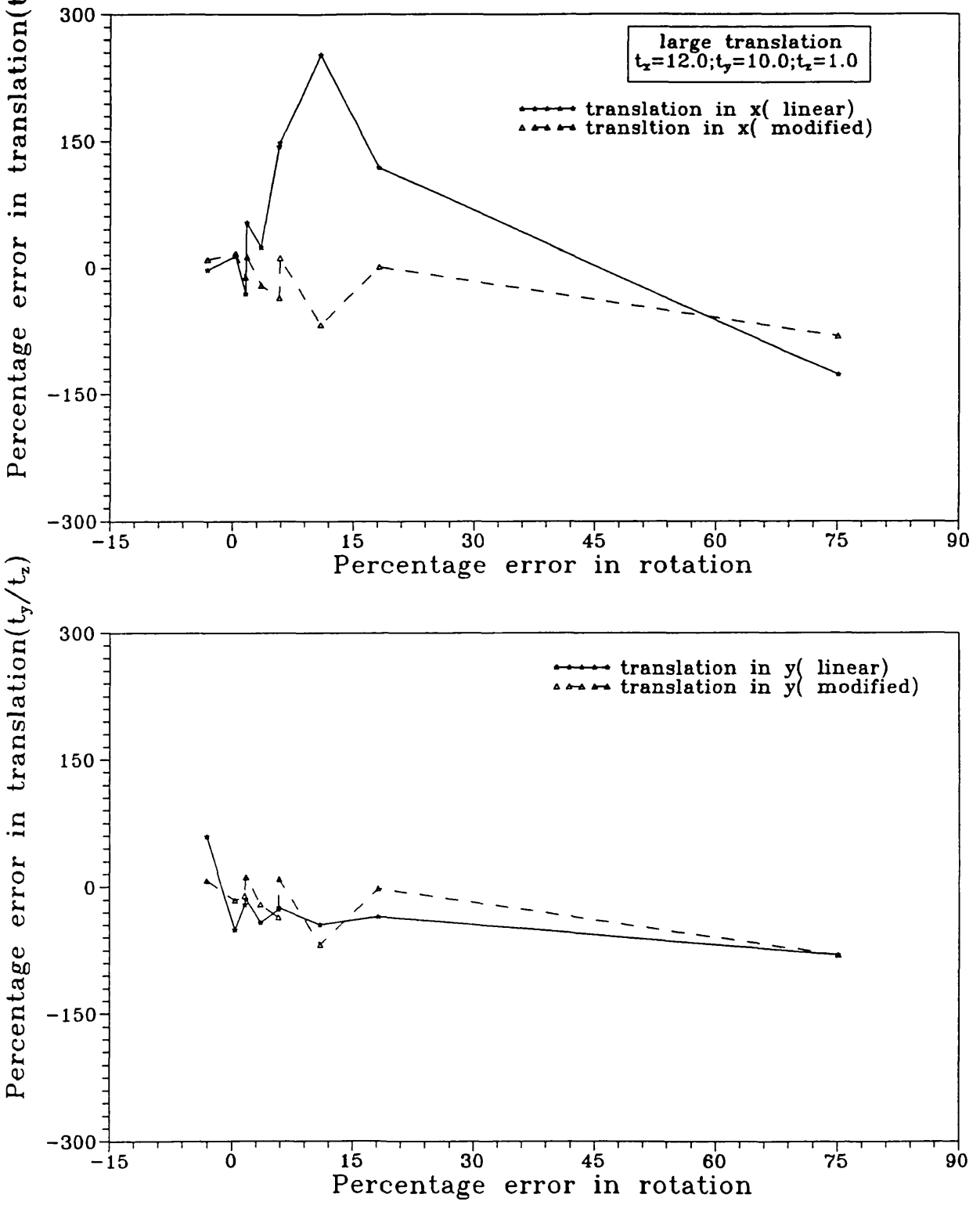


Figure 3.12
Effect of error in theta on translation
(large translation) $\theta = 20^{\circ}$ Noise =(−0.01,0.01)

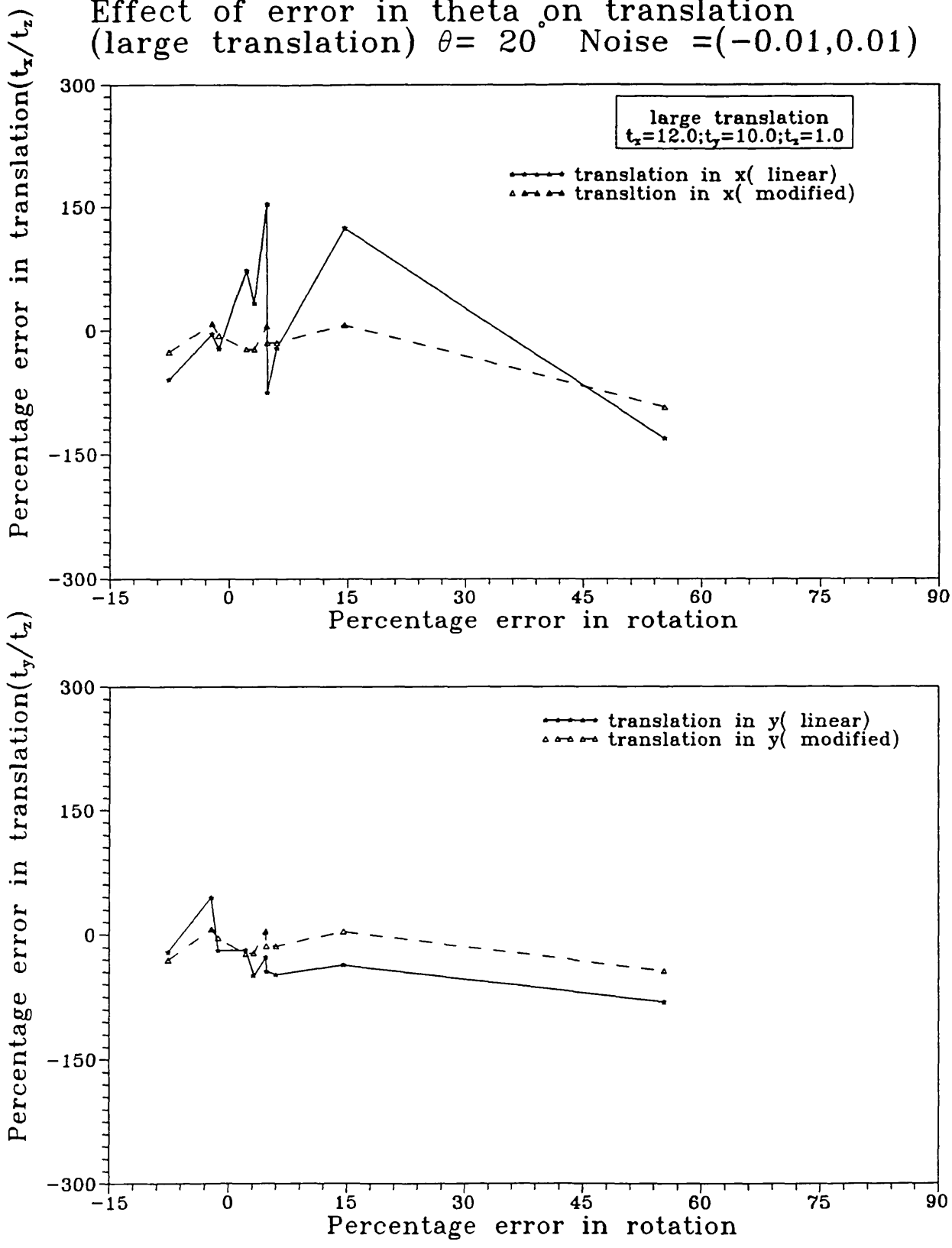


Figure 3.13
Effect of error in theta on translation
(large translation) θ= 25° Noise =(−0.01,0.01)

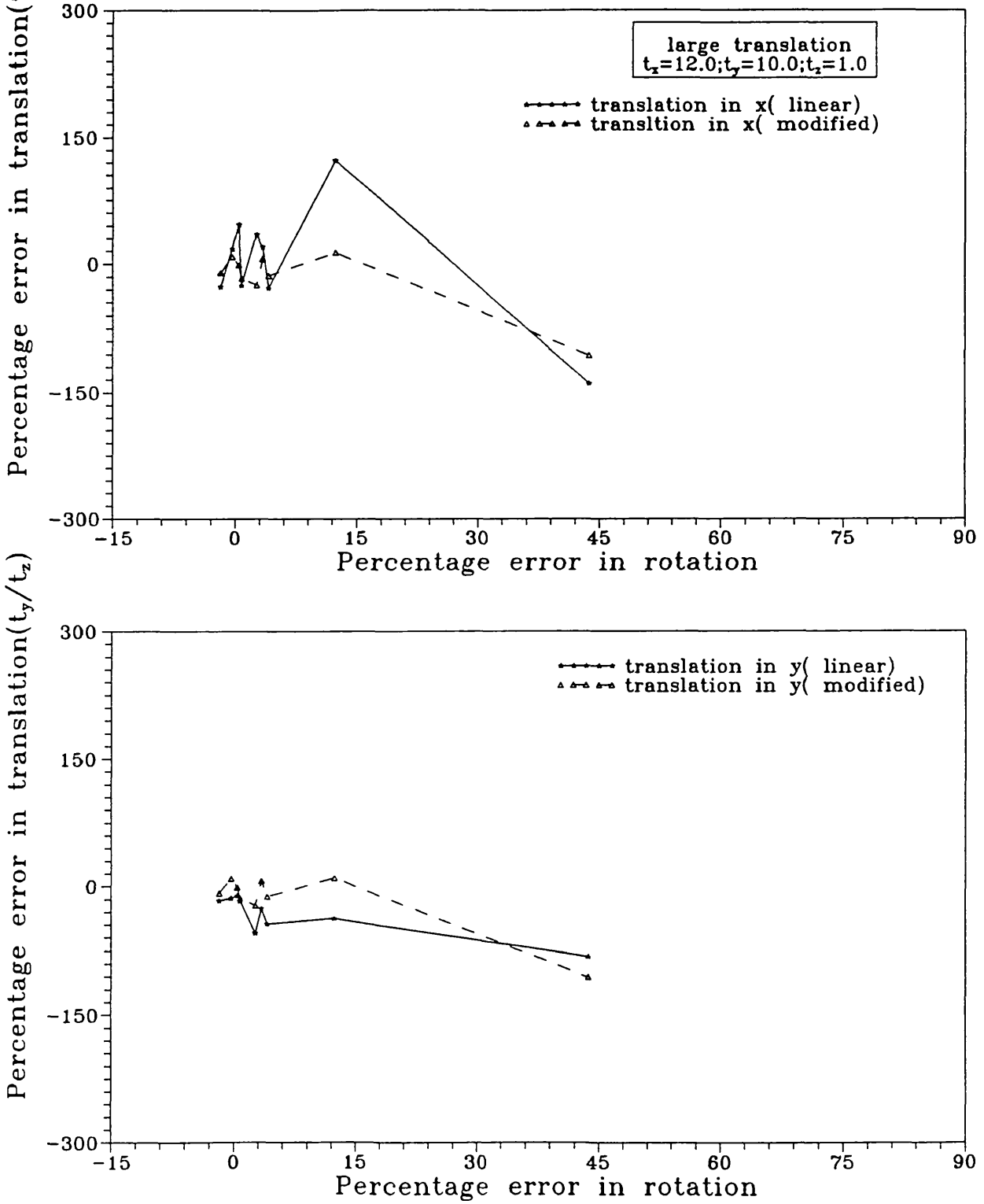
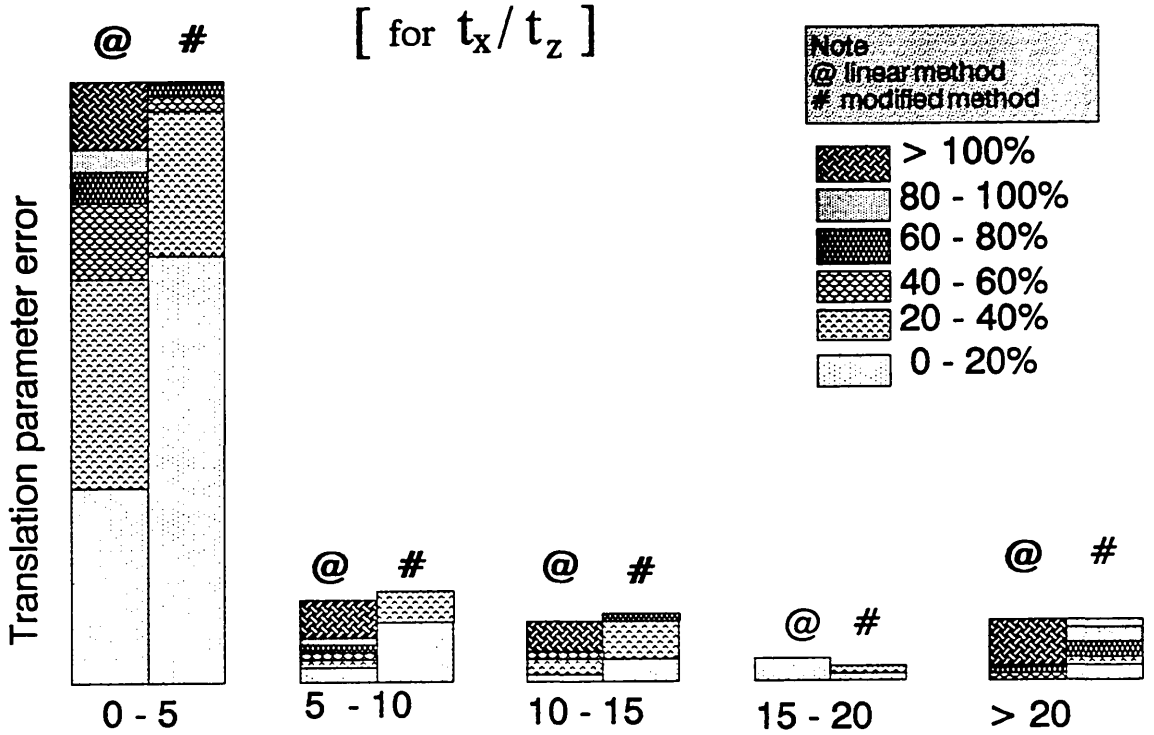
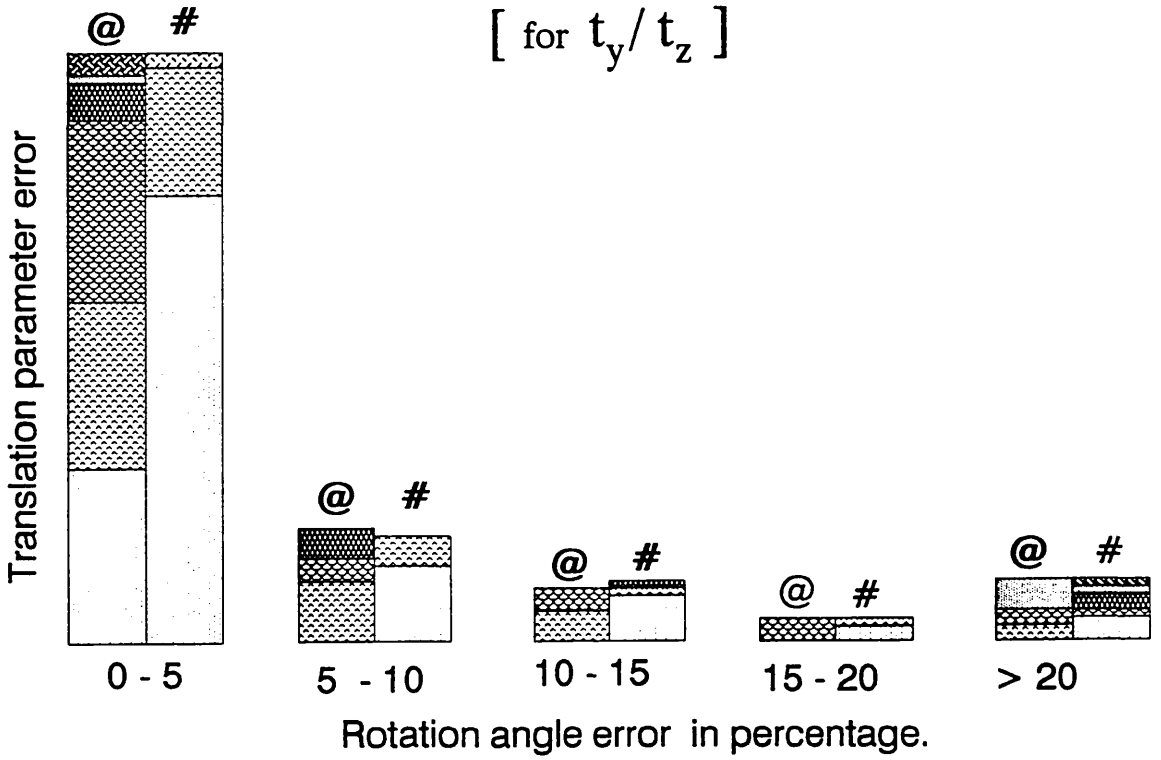


Figure 3.14 Overall performance of motion estimation

(Large translation)





this small translation case is very large. In Case -3 , where the translation motion is large, the relative error percentages for the parameters are more. In this case the contribution due to rotation is less and hence the compensation achieved may also be less. This may be the reason for a poorer performance here.

In [54] Huang et al, discuss a method in which the linear method is modified and applied for degenerate cases also. However, the number of solutions under these conditions is more than one. For instance, when the rank of the M matrix is 7, the number of solutions to motion is at most 3.

We have already seen that the performance of the linear method largely depends on the E matrix values. Recently, Huang et al [55], showed that a necessary and sufficient condition for a 3 x 3 matrix to be decomposable is that " one of the singular values is zero and the other two are equal." These conditions are given as follows:

Necessary condition:

$$E_1 \cdot E_2 \times E_3 = 0.$$

Sufficient condition:

$$\| E_1 \times E_2 \|^2 + \| E_2 \times E_3 \|^2 + \| E_3 \times E_1 \|^2 = \frac{1}{4}\left(\| E_1 \|^2 + \| E_2 \|^2 + \| E_3 \|^2 \right)^2$$

where,

$E_i (i=1,2,3)$ = column vectors of the E matrix,

$\cdot$ and $\times$ represent the dot and cross products respectively.

The first condition means that the determinant of E = 0. Hence the number of eigen values is two.

If one imposes the above decomposability conditions for E while solving equation (3.7) , a better estimation can be achieved.

As a remark, we say, that if we use a better estimation method for the

rotation parameter [56] then we may expect the modified method to give even better performance. One can use the regularization method proposed by Gerard Medioni et al, [57] which gives improved estimation of rotation angle and improve the translation parameter as above. However, the regularization method is valid only for a small rotation angle.

CHAPTER FOUR

IMAGE ACQUISITION

4.1 INTRODUCTION:

The goal of the present work is to estimate the 3-Dimensional motion of an object/objects from its 2-D images. Hence, it is essential to acquire moving object/objects as an image sequence. All the images used for our experiments were obtained from the Centre For Robotics (CFR) of the Imperial College of Science , Technology and Medicine.(ICSTM). As it is often difficult to acquire objects moving smoothly in an outdoor scene , we have got all our images from an indoor scene. Instead of moving the objects uniformly in the scene, a camera attached to the arm of a Robot called MERLIN (Modular Expandable Robot LINe), is moved under the control of a computer. The MERLIN system [58] is as shown in Figure 4.1. It has two types of motion: (1) jointed motion and (2) continuous motion. The robot can be operated in both the modes. Commands can either be given through AR–BASIC programming language or through a joy-stick.

We assumed rigid objects for our motion analysis. All the objects used are rigid objects, having mostly sharp edges and corners. The objects are placed on a Table in front of a CCD camera. The lighting is adjusted so that the camera did not saturate throughout the experiments. Also care was taken, so that the effects of external lighting and the effect of reflections from other sources are kept minimum. This is essential as will be seen in Chapter 5.

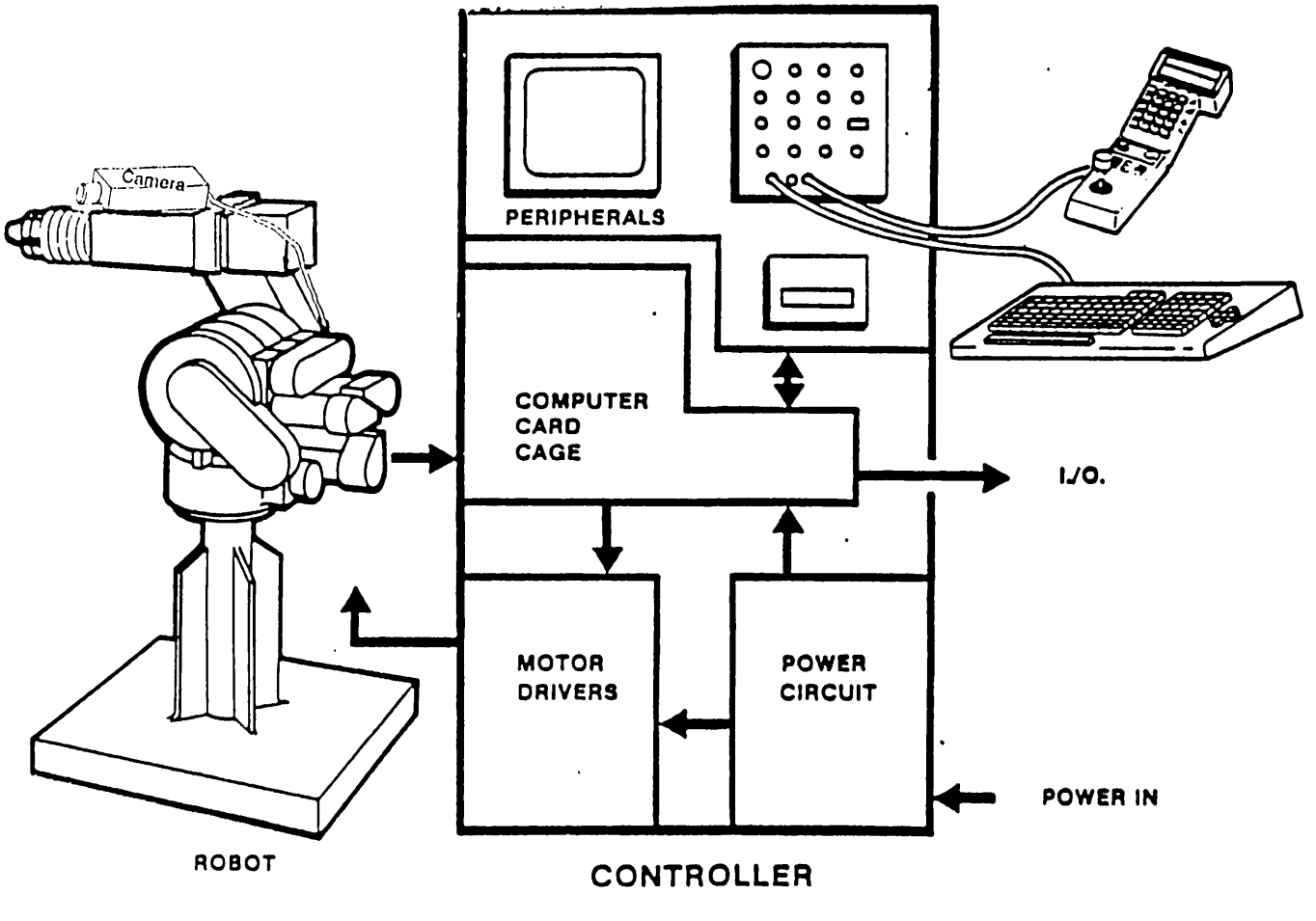


Figure 4.1
The MERLIN system

4.1.1 Calibration of the Robot:

First power was applied to the robot and after ensuring that the ROBOT is ON , the following command was issued to the robot from AR–BASIC.

> CALIBRATE PWR

This enabled the joystick in a joint -coordinated mode and allow us to position each of the robot's six axes. The six axes are X,Y,and Z axes; the rotational axes roll, pitch and yaw. Each of the axes are marked with a calibration mark. Using the joystick the robot arm is slowly moved to nearly the calibration marks. When the robot is with in $3\frac{1}{2}^{\circ}$, pressing the function key F_1 moves the robot to its exact calibration position. In this position the origin is at the center of gravity of the robot.

Once power up calibration is performed , the robot is ready to move to various positions(points). These points can be recorded. Whenever we want to store a new point , the point will first be named and stored by the following command

> TEACH _ ABS PT1 ; PT1 is the name of the first position.

> MOVE TO PT1

The above command is used to move the robot arm to the point PT1 from any point where the robot is positioned before, for instance from the calibration position. The positions of all the points are stored in a file called Position Definition File (PDL)

> LIST $ PDL command will give a list of points.

Global Frame:

The one frame that never changes is called the global frame. It is located in

the center of the robot and is based on three axes: X,Y and Z. It is shown in Figure 4.2.

Camera:

The CCD camera attached to the robot arm is focused manually. Care was taken so that the object was in focus throughout the motion sequence. The image taken by this arrangement was 512 x 512 pixels, each pixel having 16 bit gray levels. The focal length of the camera is 8 mm and the field of view is 12 mm^2. These values are taken into consideration in the motion estimation algorithm while using the real image sequences.

The camera calibration was not done as it was not available. The center of the projection should be at the camera center. As we could not locate the center of the camera precisely , the motion was performed with respect to the robot center in the global frame of reference itself.

4.1.2 Intermediate process :

The actual implementation of the various algorithms , to be explained later is performed in the Centre for Remote Sensing(CRS) facility of the ICSTM. The image processing facility at the CRS was different. The frame store at the CRS takes only 8 bits per pixel; hence the images taken from the CFR is converted in to an eight bit format by software. During this process one line, either at the top or at the bottom is lost, due to the software difficulty.

4.1.3 Pre processing :

After the successful image transfer, it was seen that the recovered image still had some noise. Hence the images were filtered using a MEAN filter which was available as a software package at the CRS. The window size is 3 x 3.

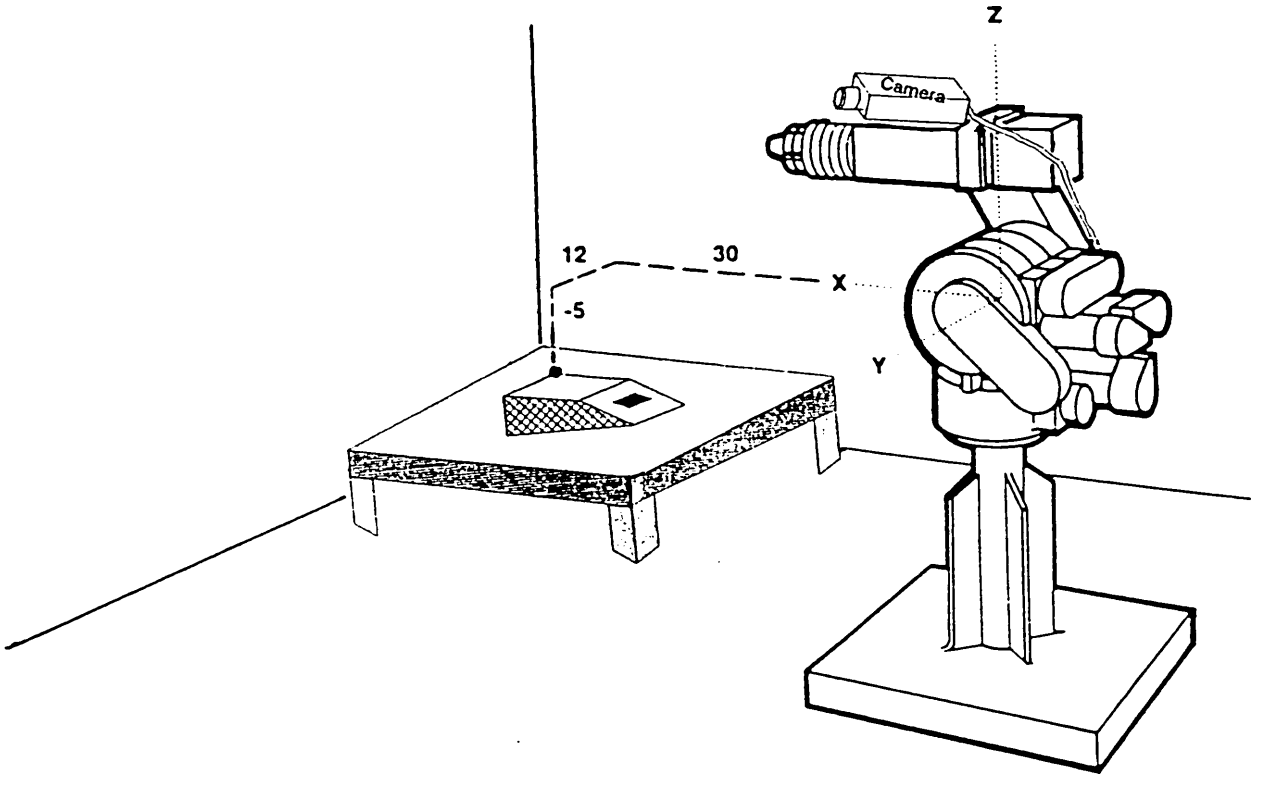


Figure 4.2
Points defined by Merlin's Global Frame.

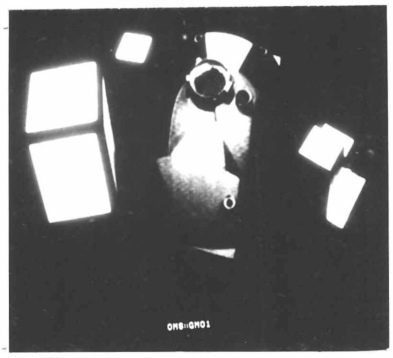
Only these filtered images are used for further processing. The feature detection and the correspondence analysis is performed on 256 X 256 images only. This is mainly to reduce the computation time involved in these processes.

We have acquired two sets of image sequences; each set comprising 6 images. In the first set the camera attached to the robot arm is given a small motion and in the second set a large motion is given. In small motion case we used a translation of 0.1 inches for t_x, t_y and t_z and a rotation of 0.1° with respect to a rotation axis. As far as large motion is concerned, we gave a translation of 0.4 inch for t_x, t_y and t_z and a rotation of 1° with respect to a rotation axis.The rotation axis for the robot is defined by the roll(ϕ) , pitch(ν) and yaw(ψ). The corresponding rotation matrix[59] is given by,

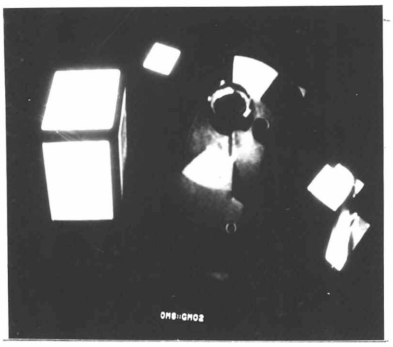
$$\mathbf{R} = \begin{bmatrix} \cos\nu\cos\psi & -\cos\nu\sin\psi & \sin\nu \\ \sin\phi\sin\nu\cos\psi+\cos\phi\sin\psi & -\sin\phi\sin\nu\sin\psi+\cos\phi\cos\psi & -\sin\phi\cos\nu \\ -\cos\phi\sin\nu\cos\psi+\sin\phi\sin\psi & \cos\phi\sin\nu\sin\psi+\sin\phi\cos\psi & \cos\phi\cos\nu \end{bmatrix} \tag{4.1}$$

The two sets of moving images are as shown in figures 4.3–4.4. The actual orientation and the motion of the camera are given in Table T.4.1 and Table T.4.2. It should be mentioned here that the represented quantities are with respect to the center of gravity of the robot and not with respect to the lens center.

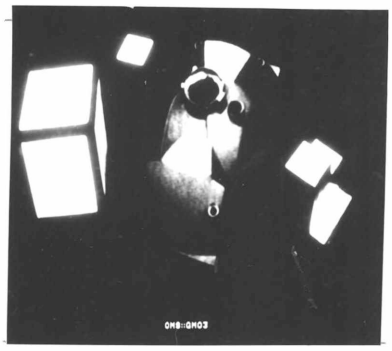
In the following chapter we discuss a method of feature detection and correspondence analysis for these image sequences.



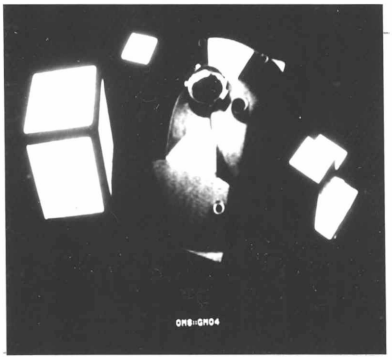
FRAME 1



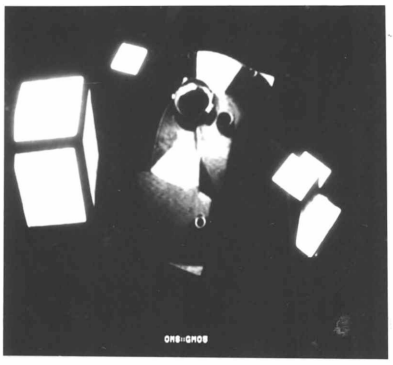
FRAME 2



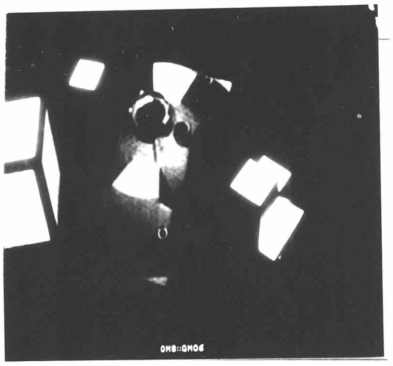
FRAME 3



FRAME 4

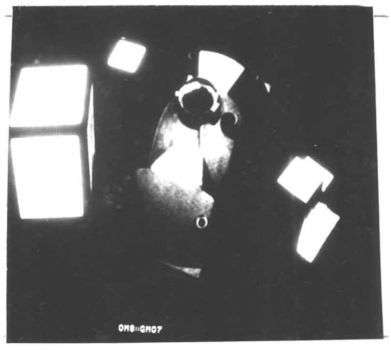


FRAME 5

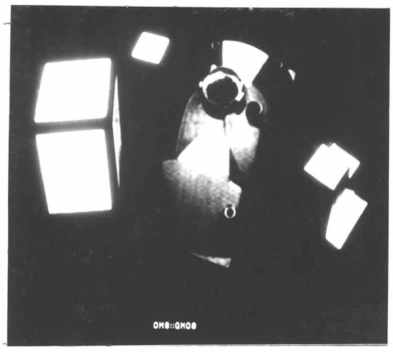


FRAME 6

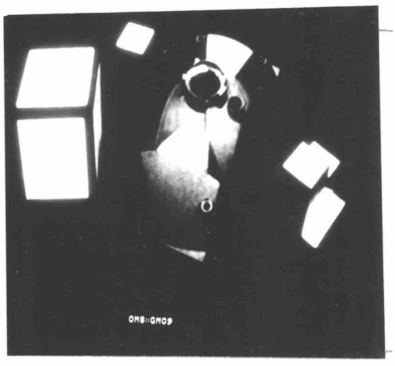
Figure 4.3 Objects Motion sequence: large motion.



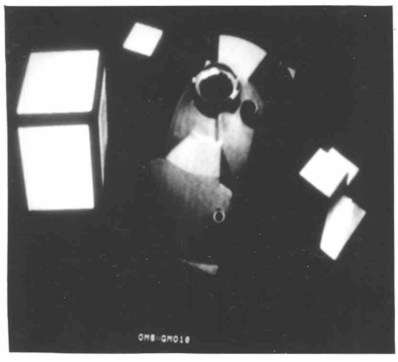
FRAME 7



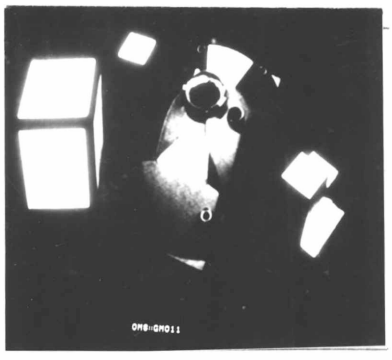
FRAME 8



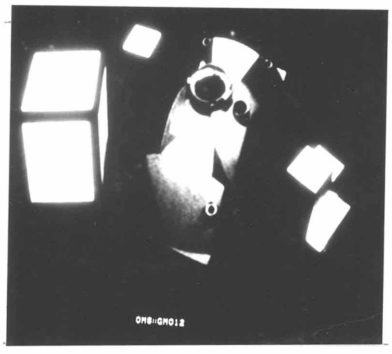
FRAME 9



FRAME 10 (A)



FRAME 11(B)



FRAME 12 (C)

Figure 4.4 Objects Motion sequence: small motion.

TABLE–T 4.1

The Orientation and motion of the Camera during [OMS] acquisition. (large motion)

Frame	x (inches)	y (inches)	z (inches)	roll (degrees)	pitch (degrees)	yaw (degrees)
GMO1	20.029	−0.816	1.618	−1.0	39.0	37.0
GMO2	20.429	−1.216	2.018	−0.635	38.389	37.385
GMO3	20.829	−1.616	2.418	−0.252	40.328	37.75
GMO4	21.229	−2.016	2.818	+0.121	41.065	38.124
GMO5	21.629	−2.416	3.218	+0.497	41.739	38.502
GMO6	22.029	−2.816	3.618	+0.867	42.464	36.884

TABLE–T 4.2

The Orientation and motion of the Camera during [OMS] acquisition. (small motion)

Frame	x (inches)	y (inches)	z (inches)	roll (degrees)	pitch (degrees)	yaw (degrees)
GMO7	20.029	−0.816	1.618	−0.965	38.958	37.052
GMO8	20.129	−0.916	1.718	−0.934	39.036	37.093
GMO9	20.229	−1.016	1.818	−0.879	39.113	37.112
GMO10	20.329	−1.116	1.918	−0.848	39.191	37.152
GMO11	20.429	−1.216	2.018	−0.811	39.268	37.188
GMO12	20.529	−1.316	2.118	−0.779	39.344	37.228

CHAPTER FIVE

FEATURE DETECTION AND CORRESPONDENCE ANALYSIS

5.1 INTRODUCTION:

Image sequence analysis is widely used to determine structure from motion, shape and shading. An important first step in most of these applications is, the identification of feature points (FEATURE DETECTION) from a given image frame. The next important step is to identify similar feature points in the given image frames which represent the same object points in 3-D space (CORRESPONDENCE ANALYSIS). The second problem is by no means an easy one to solve. Several methods exist for feature extraction whereas only a few algorithms have been proposed for the second. Feature detection and Correspondence analysis form the low level vision process in the "Structure from Motion" problems. In this chapter we analyze some aspects of these low level vision process modules. We use the terms ' tokens ' and ' features ' equivalently throughout this chapter.

The following are some of the problems faced, when dealing with the correspondence of object points from multiple image frames.

1. Noise: The noise present inherently in the image frames make it difficult to detect the feature points accurately. This noise is due to the camera non-linearities , quantization and other digitization processes.

2. Occlusion: The feature points identified in one image may not be available in the next image due to occlusion by other faces of the moving object [60]. Occlusion is the most acute problem in all the token based approaches.

3. Appearance: Some features may appear anew because the motion of an object uncovers some tokens.

4. Disappearance: Some of the feature points go out of the scene and may not be present in the next image frame, due to motion.

5. Lighting: The illumination characteristics of a few image frames are normally assumed to remain the same. This is known as ' *constancy of illumination* '. However, in reality, the illumination characteristics change between frames. Due to this variation , the thresholds selected (which are inevitable while identifying the features from an image) for one frame may not be the optimum for another frame and hence the detection of identical features in two image frames will be difficult.

In our analysis, we assume the constancy of illumination. Our image acquisition procedure suits this assumption. We take a laboratory Objects Motion Sequence [OMS]. We can set up the scene to have the same illumination characteristics by adjusting the lighting condition properly and also avoiding reflections from other sources. Hence, the constancy of illumination is a valid assumption, as far as our set up is concerned.

Occlusion of feature points will still be a problem. This will be tackled to some extent in our relaxation based matching algorithm. The matching algorithm will identify the features only if they have a high confidence of matching. Since, the occluded points will not have a potential match, the match merit will be low for such points and will not be detected. These points will be dropped from further analysis. There is a problem, though, of missing some of the true feature points. This will always be there in all the correspondence based approaches.

In section 5.2 we discuss a method of feature point detection and a way to

improve the detection using the temporal information when image sequences are available. A shuffling scheme is suggested to prune the feature points detected in two different frames. This scheme is faster and improves the elimination of false corner points in different image frames. An account of this algorithm and some results are presented in section 5.3. We use a ' relaxation ' based method for feature point matching. This is discussed in section 5.4. The features derived from the feature detection module are passed on to this matching algorithm. Some experimental results based on the above algorithm for an image sequence are given in section 5.5. The basic flow diagram for motion estimation is shown in figure 5.1.

5.2 FEATURE POINT DETECTION:

The first step in all token based approaches, is identifying a set of tokens (features) from a given image sequence. A feature may be a single pixel, a group of pixels that form a detectable edge, a group of pixels having similar brightness or colour, or a group of pixels that form a straight line of a curved segment. Single pixel features are termed ' interest points', a term first used by Moravec [61,62] in his mobile rover CART experiments. He used a directional variance for identifying the features. Sum of squares of differences of intensities of pixels adjacent in each of the four directions (vertical , horizontal and two diagonals) over each window is calculated . The minimum of these sums is taken as the directional variance. Features are chosen where the interest measure has a local maximum in a small neighbourhood. He used a 3x3 window for finding the directional variance. Dreschler [63] subsequently , used a modified form of Moravec's operator for feature detection. Other interest points are ' corner '

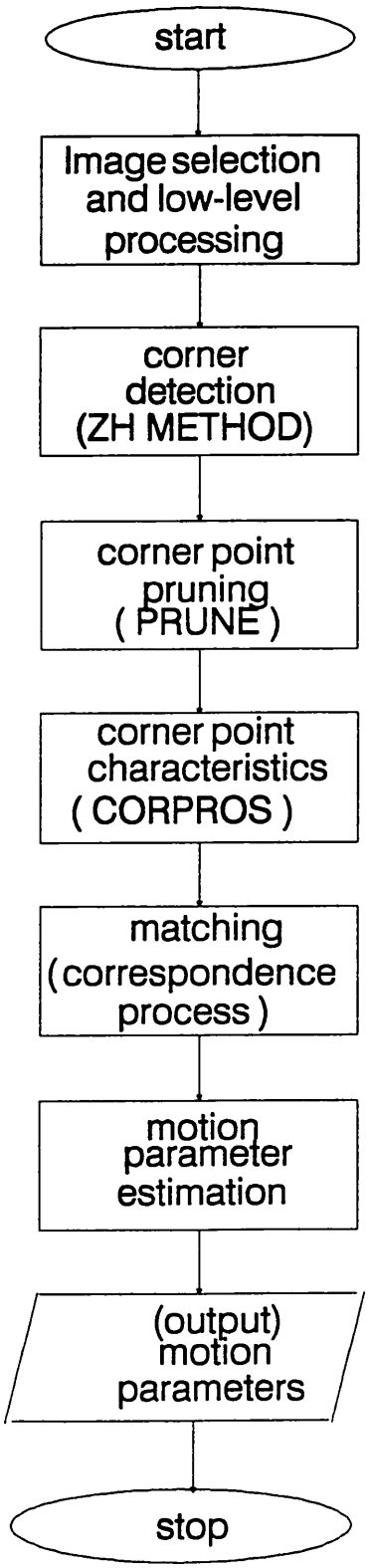


Figure 5.1 Flow chart for general motion estimation algorithm

points, the centroids of regions etc., Line features were used in [32,33,34] for estimating the motion parameters. A robust method of detecting lines and matching the lines from a given image pair was given in [64]. Price [65] used curved segments for object identification. Regional properties are not much utilized among computer vision scientists. We use a combination of regions and moments as features in our new proposal for motion parameter estimation. This is dealt with, in Part 2 of the thesis.

A feature point should have the following properties so that it can be used for motion estimation problems.

(1) It should be a highly discriminative pixel among the others in a small neighbourhood,

(2) It should be easily identified and should take very little computation time for identification.

We use 'corners' as feature points in our point correspondence based motion estimation algorithms. A corner is a point at which the rate of change of the directional derivative is a maximum. There are several methods of finding corner points from a given image.

Corner detectors are of two kinds . The first kind is based on a prior segmentation and uses the segmented image to detect a corner employing a chain coding method. The drawback with this method is that it is prone to errors in segmentation. The second kind of corner detector uses the gray values and some local operators to detect the ' corners '. These methods are known as ' gray level corner detectors'. The main advantage of these methods is that their performance is not dependent on the success or failure of earlier segmentation.

The earliest attempt in using the gray levels to detect feature points was made by Beaudet [66]. His DET operator was used to detect the saddle points at which the gray level function f(X,Y) is neither a maximum nor a minimum.

5.2.1 Cubic facet model:

Haralick [67] advocated the idea of fitting a bicubic polynomial surface over a small neighbourhood of the gray level image. According to him " any property associated with a pixel or a neighbourhood of pixels should be evaluated by relating it to the property of the corresponding gray level surface f(X,Y) which underlies the neighbourhood ". Following their lead, we fitted a bicubic polynomial surface in a small neighbourhood of pixel points. The bicubic polynomial is given by

$$f(X,Y) = k_1 + k_2X + k_3Y + k_4X^2 + k_5XY + k_6Y^2 + k_7X^3 + k_8X^2Y + k_9XY^2 + k_{10}Y^3 \quad (5.1)$$

where f(X,Y) represents the gray value of a point P at (X,Y). A small window of 5x5 pixel size is selected and the bicubic polynomial is fitted over the window. Pixels within this window are represented as shown below. This arrangement

reduces the computation time in finding the coefficients of the bicubic polynomial.

−2,−2	−2,−1	−2,0	−2,+1	−2,+2
−1,−2	−1,−1	−1,0	−1,+1	−1,+2
0,−2	0,−1	0,0	0, +1	0, +2
+1,−2	+1,−1	+1,0	+1,+1	+1,+2
+2,−2	+2,−1	+2,0	+2,+1	+2,+2

The squared error is given by,

$$e^2 = \sum_{X=-2}^{+2} \sum_{Y=-2}^{+2} [(k_1 + k_2X + k_3Y + k_4 X^2 + k_5 XY + k_6 Y^2 + k_7X^3 +$$

$$k_8X^2Y + k_9 XY^2 + k_{10}Y^3) - f(X,Y)]^2 \qquad (5.2)$$

The expressions for the coefficients k_i are given in the appendix of [68].

A measure known as the ' cornerness value ' is used to assess the characteristic of a pixel to be a corner point. This is based on the change in the gradient direction at a pixel. A higher value of cornerness means a sharp change in the gradient direction and hence the presence of a corner. Because of quantization, it is possible to get more corner points than are actually present in the image, some of which may not be true corner points. However, in dynamic scene analysis we require some time-varying features to be detected. Hence, we are interested in finding moving corners rather than static corners.

5.2.2 Time-varying Cornerness:

If a sequence of images contains a moving car, for example, then the corners associated with the car only move or change in their relative positions in subsequent images. These are known as time-varying corners. The above is true only when the camera is assumed stationary. Other corners present in the background of the scene are static corners and are not useful to us. When evaluating the characteristic of a time-varying corner, we use a measure known as ' time-varying cornerness ' value [69] .

It is defined as,

$$K_t = K_s \ \nabla_t \tag{5.3}$$

where,

K_t = Time-varying cornerness value

K_s = static cornerness value

∇_t = temporal derivative of a pixel

When a large number of frames is used, a simple temporal derivative is used , obtained by averaging the absolute values of the temporal derivatives between each adjacent pair of frames.

That is,

$$\nabla_t = \frac{\sum_{i=1}^{N-1} | F_{i+1}(p,q) - F_i(p,q) |}{(N-1)} \tag{5.4}$$

N = Number of frames,

$F_j(p,q)$ = intensity of the (p,q) th pixel in the j^{th} frame.

5.2.3 Comparison of corner detectors:

We have compared the performance of a few gray level corner detectors when a large number of frames is used. For all the corner detectors we first fit the cubic facet model as explained earlier and find the static cornerness value. Then for a given number of frames , the temporal derivative is found. Using the static cornerness value and the temporal derivative, the time- varying cornerness value is calculated according to equation 5.3. We compared Beaudet's DET [66] operator, Kitchen and Rosenfled's (KR) [70] corner detector, Fang and Huang's (FH) [71] corner detector and Zuniga and Haralick's [72] corner detector (ZH). We have tested the performance of the various corners for (a) increasing the number of frames and (b) varying the threshold while identifying the corners and the edge finding. As the threshold (CPT) value increases the number of corners detected decreases. Eventhough the KR operator detects fewer corners, it is not suitable as it fails to track the true corners well. Having a large number of corners from a corner detector will make the other operations like matching etc., difficult and time consuming. Hence, we look for a corner detector which gives a small number of corners and at the same time tracks true corners well [73]. Based on the performance of the corner detectors, we have selected the ZH corner detector operator for our feature detection module. From figures 5.2a–5.2d, we conclude that the ZH operator yields a better performance for time-varying corner detection than the KR and FH operators. figure 5.3 represent one image frame in the sequence "UMAS". In figures 5.4,5,5 and 5.6 we show the performance of corner operators KR, FH and ZH respectively. From these figures we see the ZH operator tracks more true corners than the other corner detectors. In the following we see some aspects of

Figure 5.2–a
Performance of FH operator.

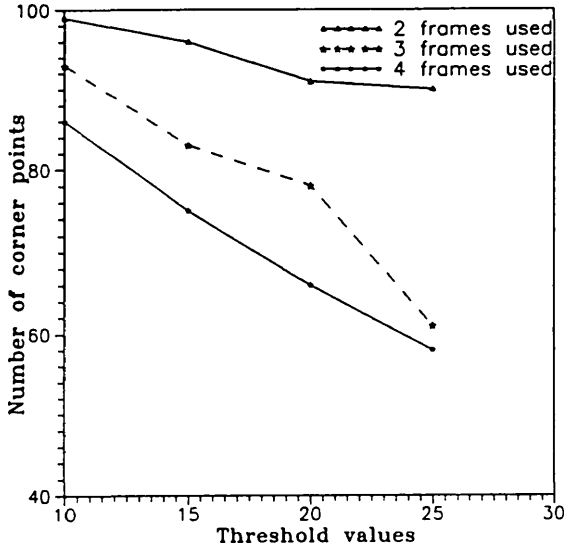


Figure 5.2–b
Performance of KR operator.

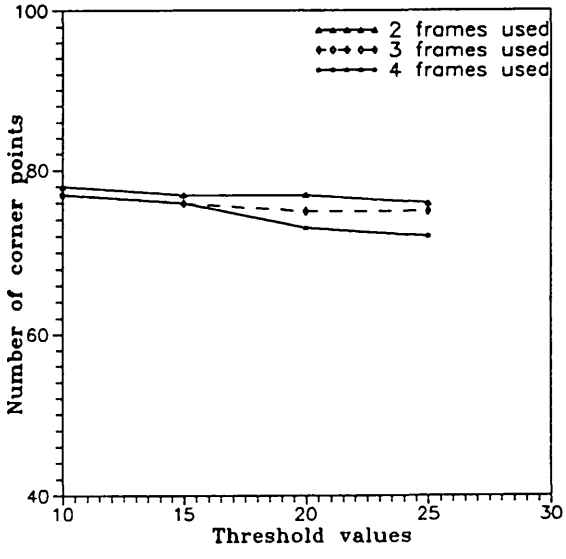


Figure 5.2–c
Performance of ZH operator.

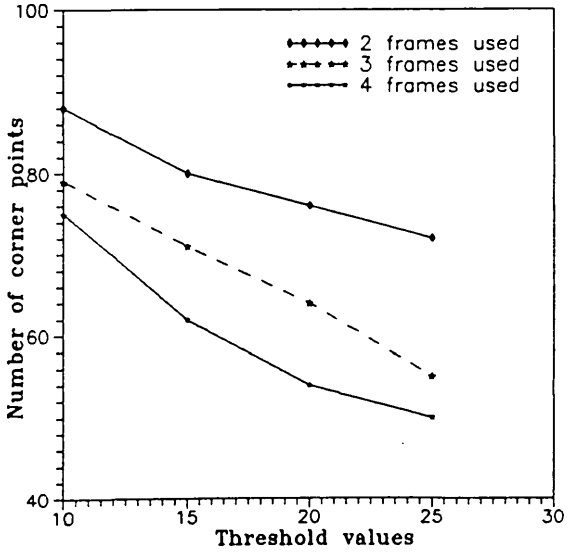
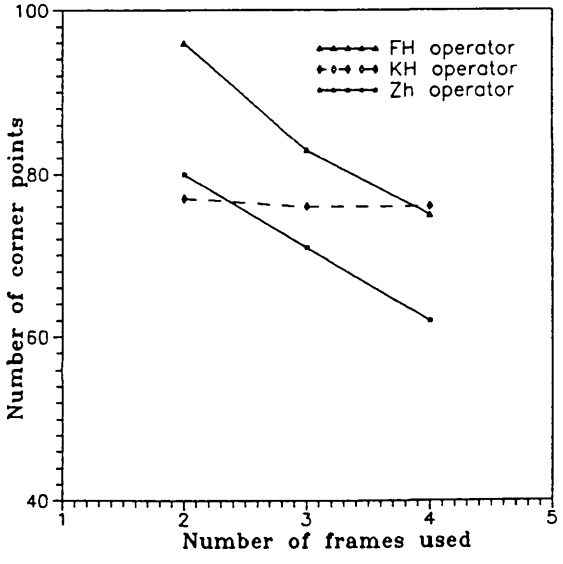


Figure 5.2–d
Relative performance of
corner operators.



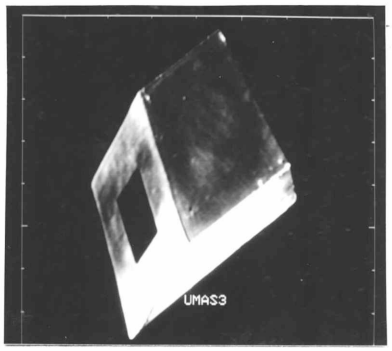


Figure 5.3

One frame in Sequence "UMAS"

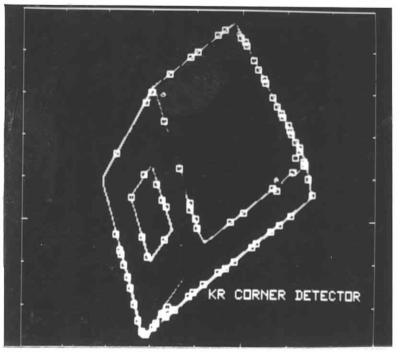


Figure 5.4

Result of KR operator on "UMAS"

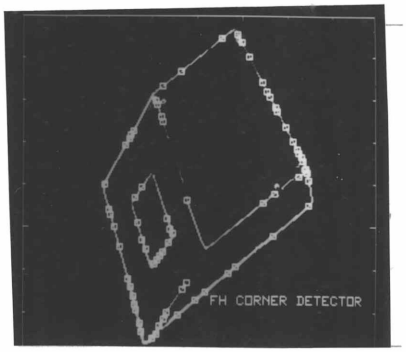


Figure 5.5

Result of FH operator on "UMAS"

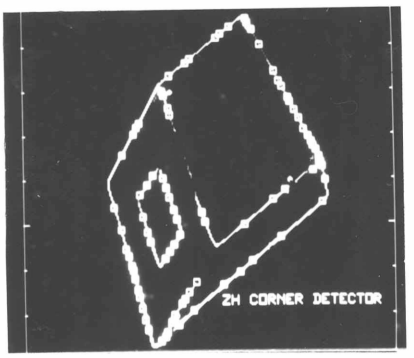


Figure 5.6

Result of ZH operator on "UMAS"

Note:

Number of frames used: 3

Threshold for corner detection:15

the ZH operator.

5.2.4. Zuniga and Haralick corner detector:

Let the gradient direction of a pixel point in the image be given by θ(X,Y) which is given by,

$$\tan \theta = \frac{f_x}{f_y} \tag{5.5}$$

where,

f_x and f_y are the gradients in the direction X and Y respectively.

Let θ_α(X,Y) be the first directional derivative of θ(X,Y) in the direction α. This can be computed as follows:

Consider a line passing through the origin in the direction α. An arbitrary point on the line is given by ($\rho \cos \alpha$, $\rho \sin \alpha$) and the gradient at that point is given by

$$\theta'(\rho) = \frac{f_y.f_{xx} - f_x.f_{yy}}{f_x^2 + f_y^2} \tag{5.6}$$

where f_{xx} , f_{yy} are second order partial derivatives of the function f(X,Y) and ρ is the distance between the center pixel and the arbitrary point. Using the bicubic polynomial function approximation at the point ($\rho \cos \alpha$, $\rho \sin \alpha$) and evaluating the directional derivative of tanθ, at the origin (0,0), ($\rho \longrightarrow 0$ and $\alpha \longrightarrow 0$) that is at the center pixel in the 5x5 window they obtained,

$$K_s = \frac{-2(k_4k_3^2 + k_6k_2^2 - k_2k_3k_5)}{(k_2^2 + k_3^2)^{\frac{3}{2}}} \tag{5.7}$$

The center point (0,0) is declared to be a corner point if

(a) (0,0) is an edge point,

(b) the magnitude K_s exceeds some set threshold value.

We have developed an algorithm for detecting the corners when a few image sequences are given. The algorithm is flexible and any of the above corner detection methods can be selected easily by the user. Besides, the number of frames needed to calculate the time-varying cornerness is also user dependent. We have used a maximum of up to four frames for finding the time-varying cornerness value. This is due to the present hardware limitation of the facilities. The steps of the algorithm are as follows:

Algorithm:

1. A small window of size (5x5) is selected and a polynomial surface is fitted over this window.

2. The static cornerness value (K_s) is found using equation 5.7

3. The temporal derivative corresponding to the center pixel in the window is evaluated.

4. The time-varying cornerness value (K_t) is calculated using equation 5.3

5. If the time-varying cornerness value exceeds a set threshold ***and*** also the center pixel is an edge pixel , then the corresponding center pixel is declared as a time varying corner, otherwise the window is moved to the next pixel and the steps 1,2... 5 are repeated until the last pixel is encountered.

In our implemented algorithm we do not fit the polynomial surface to all the points since this will be computaionally costly. If all the points in a given window are zero or are of equal value, the fitted surface will be flat and hence there will not be any corners. This means that there is no corner point in that neighbourhood. Hence in such cases, we do not fit the polynomial surface to that

center pixel. In fact, we find the mean and variance for the 5x5 window pixels. If the mean is zero and/or the variance is within some desired value (VTHRES) then we say the corresponding center pixel is not a corner pixel, and we move on to the next pixel. In this way the computation time is very much reduced. Typically the computation time in VAX 11/780 is about 0.8 seconds of the CPU time. It should be pointed out that the above CPU time includes pruning of the corner points (PRUNE subroutine) to be explained in the next section and deriving the corner point characteristics (CORPROS subroutine) which calculates some of the characteristics of the corner points like sharpness, contrast and other things , for future use in correspondence matching.It should be mentioned here that the results presented in figures 5.3–5.6 , while comparing the performance of corner detectors , however, did not use the above modification. Hence, the number of corners detected are large in number.

In our image analysis experiments , we first detect the edges using a 3x3 Sobel operator [74]. Selection of Sobel operator is only for simplicity. However, other edge detection methods like the Canny operator [75] can be tried. One of a Sobel operated image for the sequence OMS is shown in figure 5.7. Our aim was to derive the edges and hence we used a simple operator like the Sobel operator. After Sobel edge detection a non-maximum suppression is performed. The cubic facet model is fitted to the original image and if the cornerness value exceeds a certain threshold we check whether the corresponding pixel in the edge image is an edge point; if so the value of K_s is calculated. We have tried two experiments: (a) keeping the EDTHR (edge threshold) constant and varied the CPT(corner point threshold) and (b) keeping the CPT constant and varying the EDTHR. Some of the results are shown in figures 5.8 and 5.9. Based on our experiments we

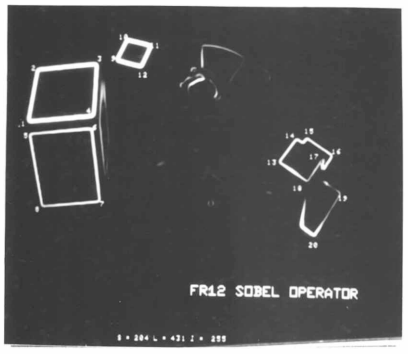
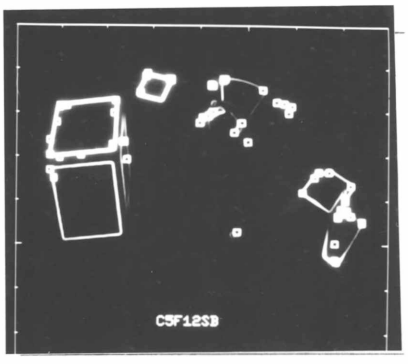
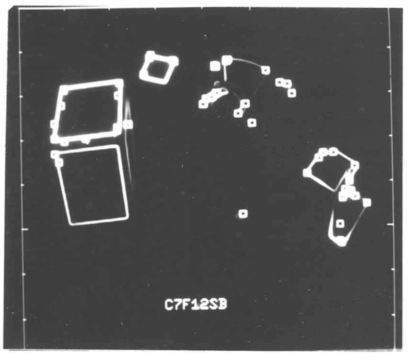


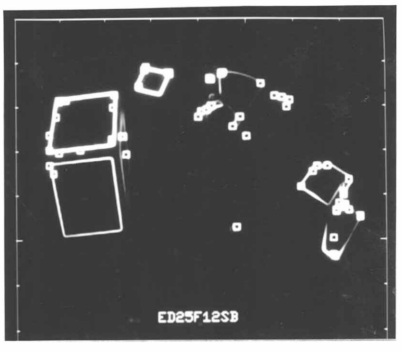
Figure 5.7 'Sobel' operated image:Frame-1[OMS]

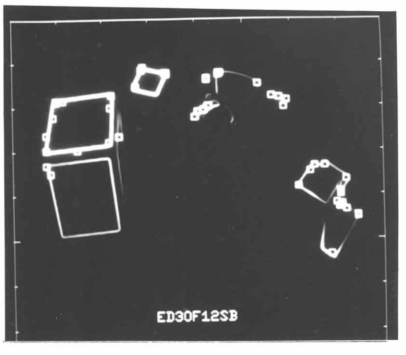




(A) CPT=5 (B) CPT = 7

Figure 5.8 Varying corner point threshold





(A) EDTHR = 25 (B) EDTHR = 30

Figure 5.9 Varying edge threshold.

used a value of CPT =5 and EDTHR=25 for the rest of our experiments. This seem to give a good result. In figures 5.8 and 5.9. corner points are superimposed on the edge detected image. The detected corners are, of course, time-varying corners. In most of the cases we used two frames for finding the time varying corners. Due to the discontinuity problems in the edge detection we get more corner points. In some cases we loose the actual corner points also. This may be due to the illumination characteristics of the images and also the thresholds used for finding the time-varying corners. Besides, if the sharpness change is within one pixel , then the corresponding corner point is surely missed. In such circumstances a better edge detection method should be used. Sub-pixel level edge detection may also be tried.

All the above corner detectors are basically the same in principle. There are many other corner detectors reported. Medioni and Yasumoto [76] used cubic B - splines for corner detection and curve representation. Wu and Rosenfeld [77] used projection of the gray values along the axes to get corner points. In [78] a method based on the distribution of gray levels is used to detect corners. Though there are many corner detection algorithms, still , it is very difficult to identify the true corners in image frames. Recently, in [79] an optimal corner detector is reported. The authors used an optimization theory to detect corners. It is clear from the plethora of corner detection methods, that feature point detection is an important and foremost step in all image analysis algorithms.

5.3 A SHUFFLING SCHEME TO PRUNE CORNER POINTS:

We have discussed a method of detecting the feature points from a given

image sequence in the previous section. Due to the quantization effects , the detected edges will not be of one pixel width and there will be discontinuity in the edges. These discontinuities in the digital edges may give rise to ' corner points' which are not actually present in the scene. These corner points are called ' noise corners '. The number of corners detected from a given image sequence will be more because of these noise corners. Hence we need to prune (PRUNE subroutine) the corner points. There are two disadvantages in having all the corner points. They are,

(1) the computation time for the correspondence analysis and other processes will be more,

and (2) matching the feature points may be erroneous.

We have devised a shuffling scheme to eliminate the noise corner points. The algorithm is an iterative one and it takes very little computation time.

The main assumptions of the shuffling scheme are with respect to the orientation of corner points and the length between corner points. We start from an arbitrary corner point say #1(See figure 5.10-a). We find (a) the orientation with respect to #1 of all other corner points and (b) the length between the corner point #1 and all other corner points. The orientation and the length are stored in an array. If the orientation between corner point #k (k=2,3... N; N being the total number of corner points detected) and the corner point #1 is within some threshold say, ATHE and the length between the corner point #k and the point #1 is within some threshold value say, LTHE, then this point is dropped.

In figures (5.10 a,b,c) the LTHE is represented as a circular region with dotted lines. Also ATHE is shown as shaded regions in these figures. This

comparison process is repeated for all other points with respect to #1. For every corner point dropped , the total number of corner points is decremented by one. After all these pairs are compared, the corner points are shuffled and the starting corner point #1 is moved to some other one. Again the orientation and the length with respect to this new starting corner point is calculated and again the comparison is made as explained above.(see figures 5.10.b and 5.10.c). This process is repeated until all the corner points take the role as the starting corner point once. The dropped feature points are shaded in subsequent figures. Also, the actual number assigned to the corner points in the previous comparisons are given in brackets. It can be seen in figure 5.10.c that the corner point #2 is within the LTHE but it is not dropped as the ATHE threshold is not satisfied at this point. In figures 5.10a–c we used a value of 30° and 1.5 cm for ATHE and LTHE respectively for illustration purpose. In the actual images the LTHE/ATHE used are different.

Now, the total number of remaining corner points are compared with the required number of corner points (set by the user; we always , in our experiments, set this to 25). If the remaining corner points are greater than this, then ATHE/LTHE are incremented by a small step. The whole process is repeated once more. This iteration is repeated a set number of times, determined by the user. Even after the required number of iterations , if the number of corner points remaining are too many, then either we can take them as they are, or we can increase the thresholds of the angle and the length so that the required number of corner points are ultimately met by further iterations. From the above discussion it can be seen that we assume the corner points in a given 3-D object are not very near and also the relative orientations are not very close. Hence some

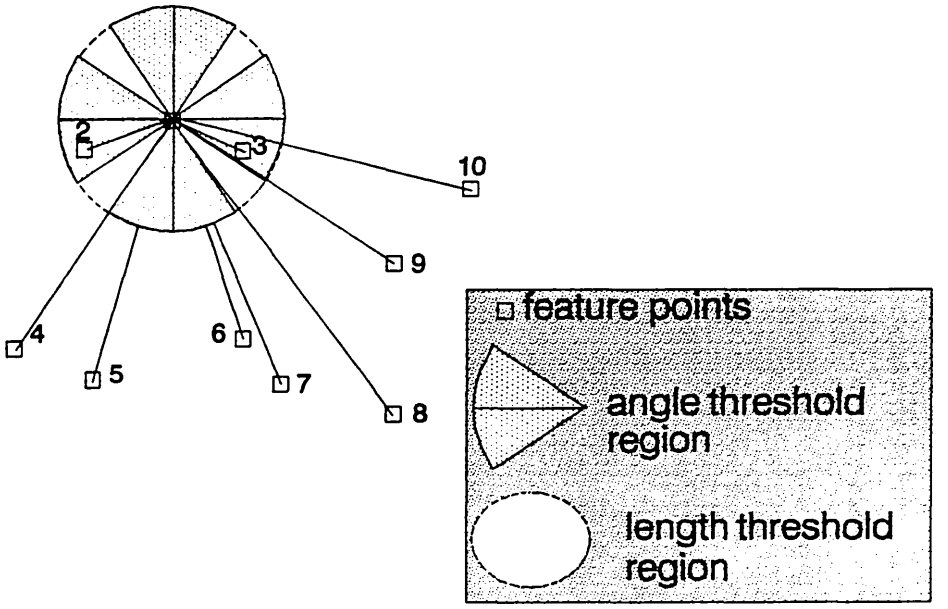


Figure 5.10 - a: Pruning the corner points: stage 1.

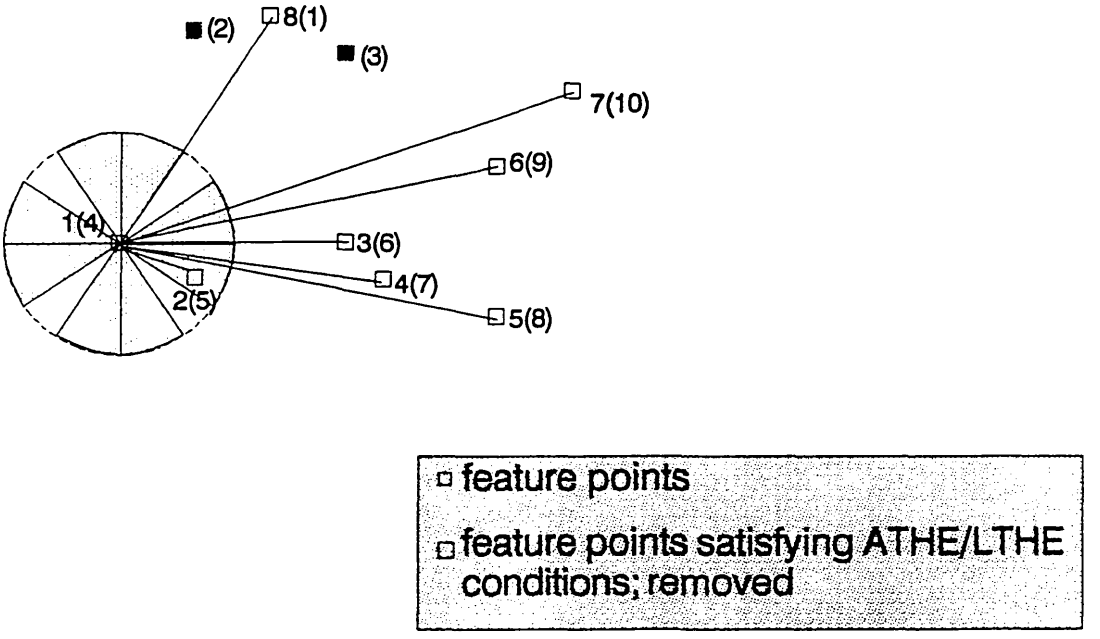


Figure 5.10 - b: Pruning the corner points: stage 2.

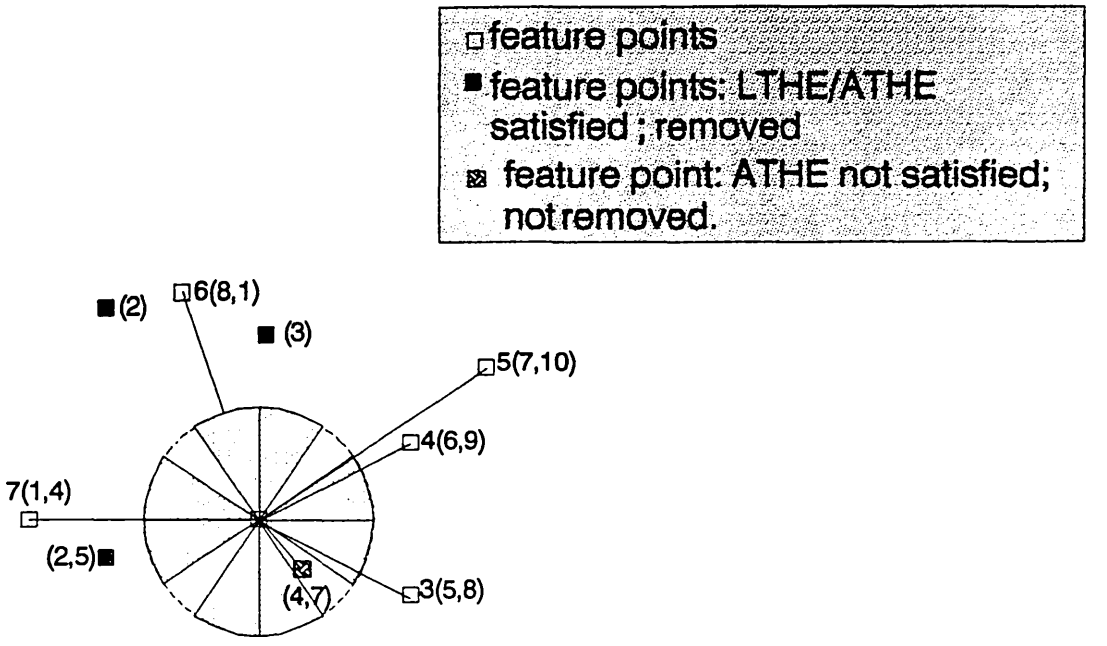


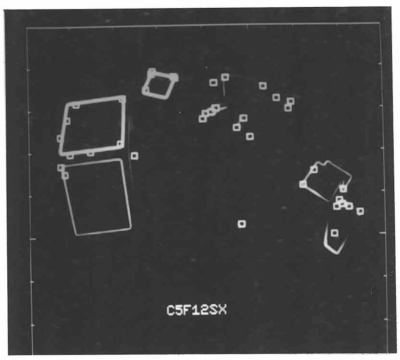
Figure 5.10 - c: Pruning the corner points:stage-3.

closeby corners may not be detected by the algorithm even if they are actually present due to the above limited assumptions. This is the main drawback in the algorithm. In our OMS images, unfortunately, due to the painting of one of the blocks , we do get such closeby corners.(see figure 5.8 and 5.9) If the shape of the 3-D object changes , then this may lead to some errors. We do not consider such non-rigid objects.

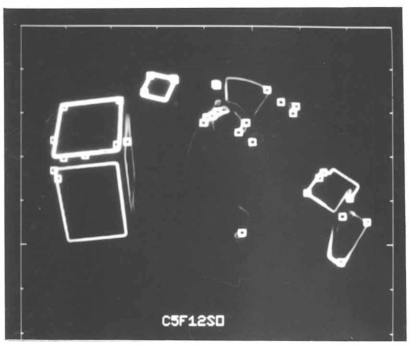
5.3.1 Sorting of the corner points:

If a corner point, taking the role as the starting corner point is chosen arbitrarily, we observed that some of the true corner points are missed. This drawback can however be avoided, if we sort the corner points before passing the corner points on to the PRUNE algorithm. The corner points are sorted with respect to the (0,0) pixel which is positioned at the top left-hand corner of the image. Based on the distance from the origin to the corner points, the corner points are sorted in increasing order. The sorted corner points are fed to the PRUNE algorithm which will take the starting corner point in the sorted order. Sorting in most of the cases improved the detection of true corner points. Also some close-by corner points are not dropped by the pruning algorithm. There are , however, some exceptions to this performance.

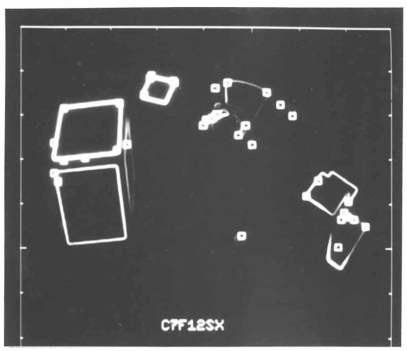
In figures 5.11 and 5.12 we show the performance of the pruning algorithm (a) without sorting and (b) with sorting. Figure 5.11 shows the images where EDTHR is kept constant and CPT is varired. Figure 5.12 shows the images where the constant CPT is kept constant and EDTHR is varied. From the figures we see sorting improves pruning and thereby improves the performance. For the rest of the analysis we use only the sorted corner points for the pruning algorithm. The performance of the corner detection and pruning algorithms are shown in



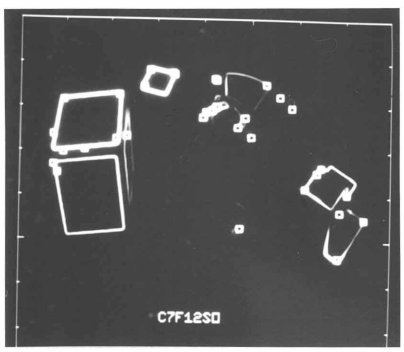
Without sorting (CPT=5)



With sorting (CPT=5)



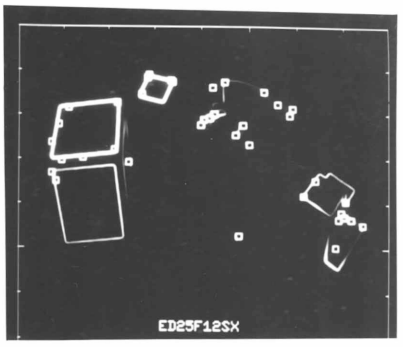
Without sorting (CPT=7)



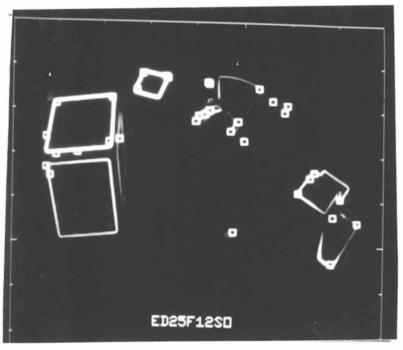
With sorting (CPT=7)

Figure 5.11

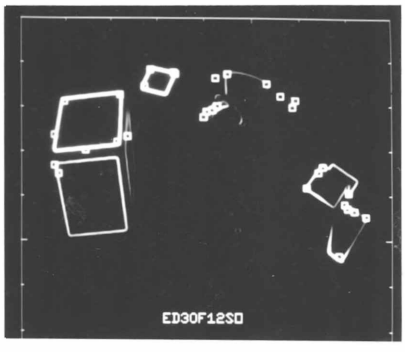
Performance of Pruning algorithm: with and without sorting.



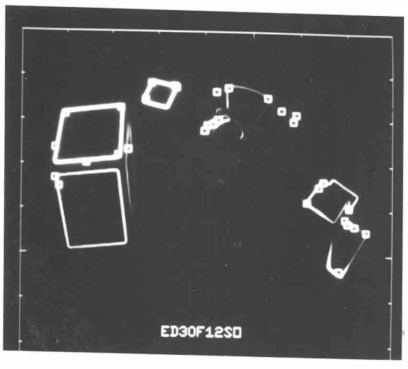
Without sorting (EDTHR=25)



With sorting (EDTHR=25)



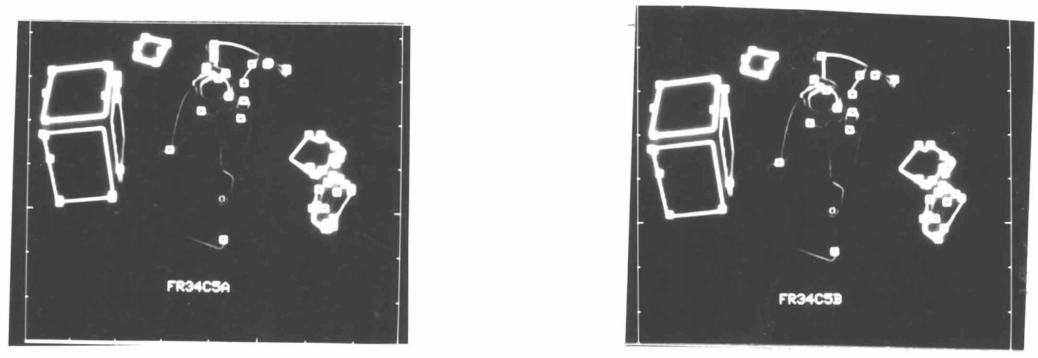
Without sorting (EDTHR=30)



With sorting (EDTHR=30)

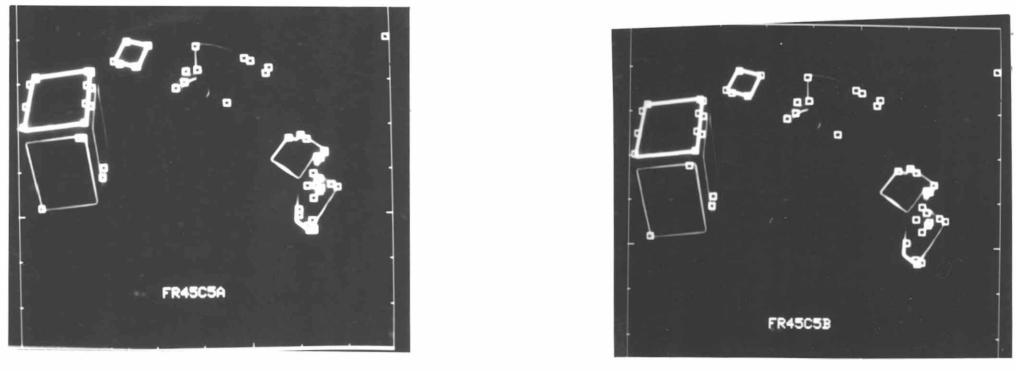
Figure 5.12

Performance of Pruning algorithm: with and without sorting.



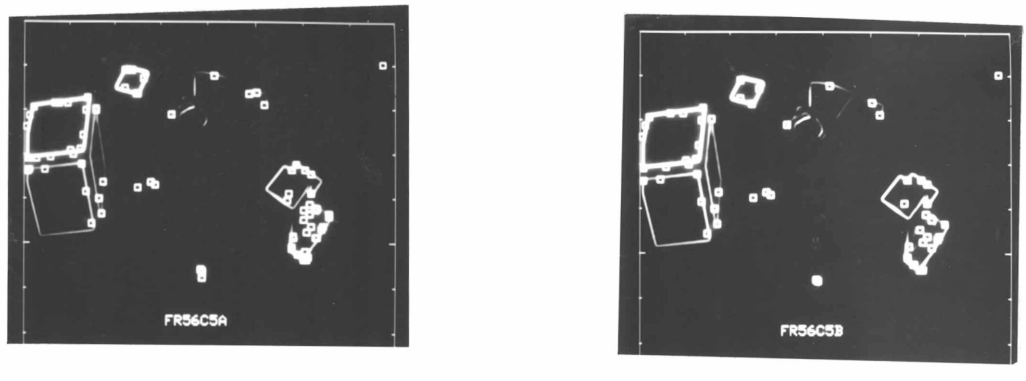
(A) (B)

FRAMES 3 AND 4



(A) (B)

FRAMES 4 AND 5

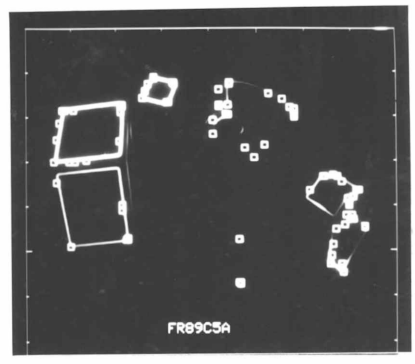


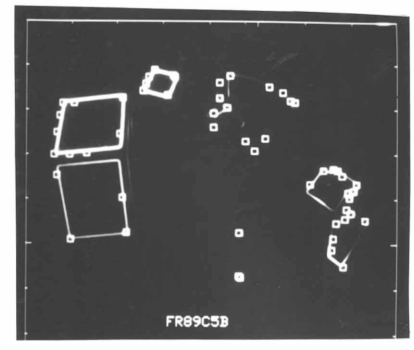
(A) (B)

FRAMES 5 AND 6

(A) Before Pruning (B) After Pruning

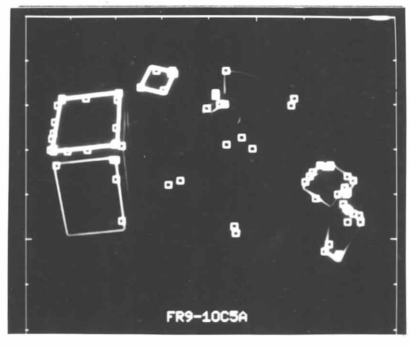
Figure 5.13 Results of corner detection and Pruning algorithms: large motion case

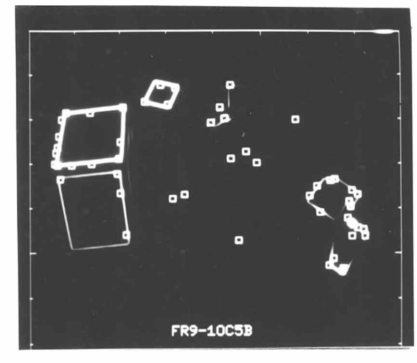




(A) (B)

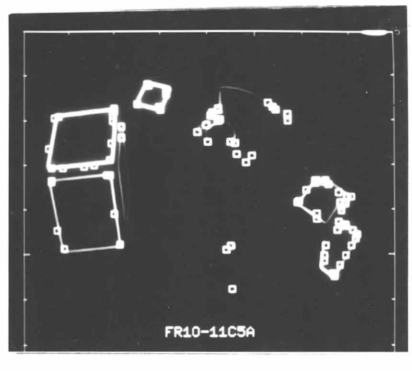
FRAMES 8 AND 9

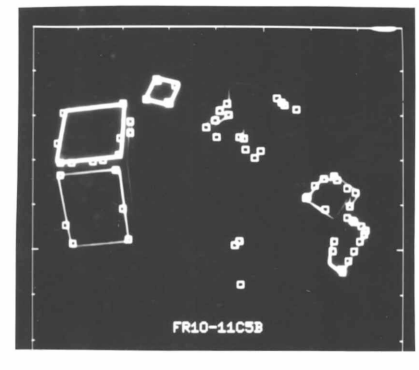




(A) (B)

FRAMES 9 AND 10





(A) (B)

FRAMES 10 AND 11

(A) Before Pruning (B) After Pruning

Figure 5.14 Results of corner detection and Pruning algorithms: small motion case

figures 5.13 and 5.14 for a few images in OMS sequence. Figure 5.13 represents large motion sequence results. We show here, the performance for 3 frames only. Similarly figure 5.14 shows the performance for the small motion case.

After the corner points have been detected using the pruning algorithm, we need to derive some characteristics of the corner points, which are needed for the correspondence analysis. Some of the characteristics of the corner points are (1) Slope at the corner point (2) contrast (3) sharpness (4) cornerness and (5) time- varying cornerness. Some of them are defined as follows,

$$\text{slope} = \tan^{-1}\left(\frac{g_x}{g_y}\right)$$

where g_x and g_y are the partial derivatives of a corner point in the X and Y directions respectively.

The contrast and sharpness at the corner point are given by,

$$\text{contrast} = \sqrt{g_x^2 + g_y^2}$$

$$\text{sharpness} = \tan^{-1}\left(\frac{\theta_x}{\theta_y}\right)$$

where,

$$\theta_x = \frac{(g_{xy}.g_x - g_{xx}.g_y)}{(g_x^2 + g_y^2)}$$

$$\theta_y = \frac{(g_{yy}.g_x - g_{xy}.g_y)}{(g_x^2 + g_y^2)}$$

g_{xx}, g_{xy} and g_{yy} are second order partial derivatives.

Cornerness and time-varying cornerness are as defined in sections 5.2.4 and 5.2.2 respectively. The subroutine CORPROS in our corner finding algorithm finds these characteristics. A flow chart for the algorithm is given in figure 5.15. In the

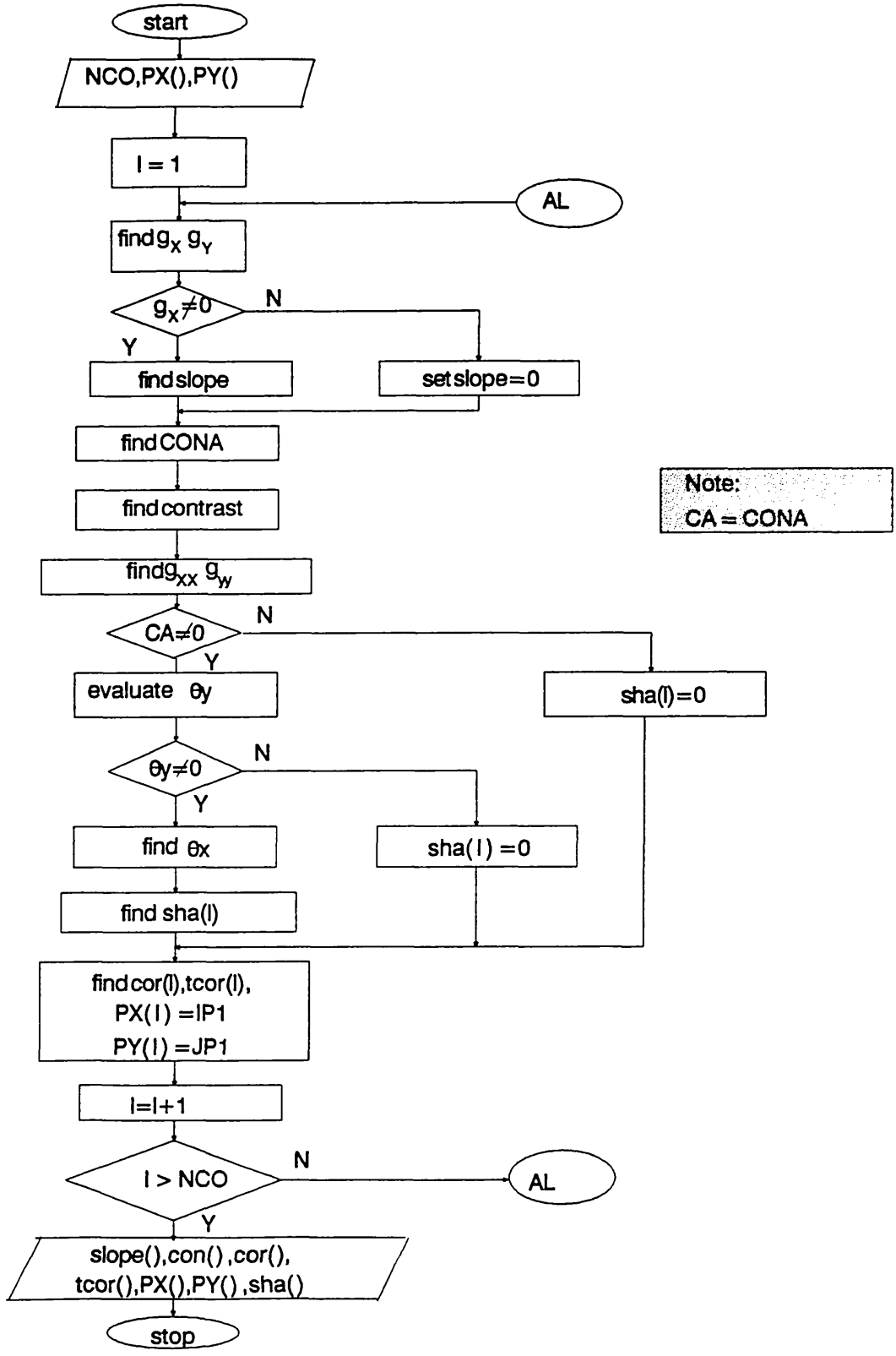


Figure 5.15: Flow chart for CORPROS subroutine

flow chart NCO represent the number of corner points detected by the PRUNE routine. The positions in the image plane of the corner points are also passed on to this subroutine by the PRUNE module. The actual image frames are transparent to all the subroutines, hence the partial derivatives at a corner point can be calculated. $(g_x^2+g_y^2)$ is denoted as CONA in the figure 5.15. The outputs like slope, contrast, sharpness, cornerness and time-varying cornerness of the CORPROS are stored in separate arrays for future use.

5.4 CORRESPONDENCE ANALYSIS:

Having derived a set of features in two image frames , the next step is to match these two sets so as to get corresponding features which represent the same object points before and after motion. This is known as *correspondence analysis.*

Ullman [80] in his highly acclaimed work suggested that " the correspondence is a low-level, autonomous process. It depends on spatio-temporal and similarity parameters between elementary tokens in the image , but not on a higher level process such as recognition". He studied extensively the correspondence process in relation to 3-D scene interpretation. He used various measures like distance between the features, length of the features etc., for identifying the correspondence between the tokens.

Let us see the correspondence process by taking a simple object like the one shown in figures 5.16a and 5.16b. The small square blocks are the corner points that are detected in the first image. Similarly in figure 5.16b the corner points detected in the second image are circled. The corner points are numbered 1,2,3 and 4 in figure 5.16a and a,b,c and d in figure 5.16b. Thus in correspondence analysis we would like to match the point #1 to exactly one of the corner points

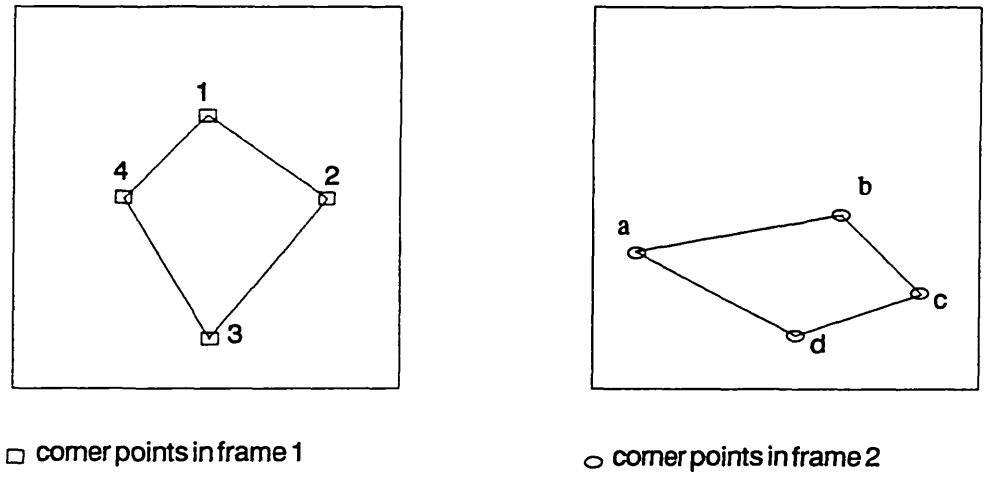


Figure 5.16
An example for correspondence process .

in the set {a,b,c,d}. If #1 is matched with #c then #2 will match to #d, #3 will match to #a, and #4 will match to #b. This is based on a distance measure. This can be seen directly from the figure 5.16. In reality , the situation is not as simple as the one considered above. This is because of the various reasons already pointed out earlier in this chapter. Hence many constraints are to be used in correspondence analysis.There are two basic approaches: (1) the ionic or template models and (2) structural models [81].

In template models , a subimage of size $\Delta x.\Delta y$ in frame F_1 that has a possible moving object is first detected and an iconic representation for the same is constructed.

Let $F_1 = f(x, y) \quad x_1 \leq x \leq x_2, \; y_1 \leq y \leq y_2$ be a frame at $t=t_1$

and $F_2 = f(x, y) \quad 0 \leq x \leq n, \; 0 \leq y \leq m$ be a frame at $t=t_2$

$\Delta x = x_2 - x_1; \quad \Delta y = y_2 - y_1$

Then the second frame is searched fully for a match. Some kind of a match merit is needed to correctly identify the match. Some of the match measures [82] that are normally used are defined as follows.:

(a) Normalised cross correlation.(similarity measure) .

$$C(x,y) = \frac{P(x,y)}{Q(x,y)} \tag{5.8}$$

where,

$$P(x,y) = \sum_{i=0}^{\Delta y} \sum_{j=0}^{\Delta x} F_1(x_1+ i, y_1+j) \, F_2(x+ i , y+ j) \tag{5.9}$$

$$Q(x,y) = \sqrt{\sum_{i=0}^{\Delta y} \sum_{j=0}^{\Delta x} F_1(x_1+ i, y_1+j)^2 \, F_2(x+ i , y+ j)^2} \tag{5.10}$$

(b) Sum of the absolute differences (difference measure):

$$A\ (\ x\ ,\ y) = \sum_{i=0}^{\Delta y} \sum_{j=0}^{\Delta x} |\ F_1(x_1+\ i,\ y_1+j\ ) - F_2(x+\ i\ ,\ y+\ j)\ | \qquad (5.11)$$

(c) Sum of squared differences (difference measure):

$$S\ (x\ ,\ y) = \sum_{i=0}^{\Delta y} \sum_{j=0}^{\Delta x} [\ F_1(x_1+\ i,\ y_1+j\ ) - F_2(x+\ i\ ,\ y+\ j)\ ]^2 \qquad (5.12)$$

It can be seen from the above equations that to find a match corresponding to a sub image F_1 we need to compute the match measure at each pixel in frame F_2. Computationally this will be costly. Hence the following method is used. A point in the frame F_2, which is on an identical position as that of the interest point in frame F_1, is selected. Then computation of the match merit and the search for a correct match is performed.This search is limited to some fixed distances in both the X and Y directions around the selected pixel in F_2. In other words, the maximum displacements in the respective directions are assumed to be within some known values. This puts a restriction on the 3-D motion indirectly. There are two problems with this approach: (1) the correct match may not be the one of maximum similarity due to noise and quantization (2) the computational cost involved in the search is high. Earlier methods of matching used this conventional approach. However, some parallel schemes are now available, to reduce the computational cost [83]. The main drawback with the parallel method is that it is not suitable when a large rotation and translation motion are present in the object.

In the structural matching technique, first the features (tokens) from two time-sequential images are to be found, using some low level processing. Then the matching process constructs a matching from a set of tokens in one image, to a set

of tokens in the second image. In this type, the matching performance depends on the attributes of the tokens. The attributes may be the contrast surrounding a feature pixel, the slope of a line passing through the pixel, cornerness of the pixel etc., We use this structural model for our matching scheme.

5.4.1. Relaxation based token matching:

" Relaxation " is an iterative approach to match the tokens derived from two image frames, using some of the characteristics of the tokens for local support. Relaxation has been widely used in recent times for multispectral pixel classification(segmentation), matching and other purposes in image analysis [82,84,85,86].

Barnard and Thompson [87] used a probabilistic approach to match the disparity vectors, which arise due to the differences between frames. Ranade and Rosenfeld [88] proposed a simple scheme for matching two feature sets derived from two images. Ibison and Zapalowski [89] used a relaxation labelling scheme similar to Ranade et al, [88] with a modified symmetric constraint to solve the correspondence problem. Most of the above methods use only two frames at a time and assume the rigidity constraint in some way or other. Recently, Sethi and Jain [90] proposed a scheme which uses more than two frames. They used the smoothness of motion criterion as opposed to rigidity criterion. According to them " smoothness of motion is a more powerful criterion than the rigidity constraint". Their method is also a relaxation based one.

We use a variant of Ranade and Rosenfelds' method. This method is chosen mainly because of its simplicity. It is also powerful and computationally less costly.

Assume there are two sets of feature points $\mathcal{P}$, $\mathcal{Q}$ derived from a given

image sequence. Let $\mathcal{P}$ be $P_1(X_1,Y_1)$, $P_2(X_2, Y_2)$ $\cdots$ $P_m(X_m,Y_m)$ and $\mathcal{Q}$ be $Q_1(X_1, Y_1)$, $Q_2(X_2, Y_2)$, $\cdots$ $Q_n(X_n, Y_n)$ where ' m ' and ' n ' are some positive integers. (X_i, Y_i) represents the feature points in a 2-D space. We also have some of the characteristics of the features like the sharpness, contrast, slope , cornerness and the time-varying cornerness. Our goal is to match P_1, P_2,$\cdots$ P_m to Q_1, Q_2,$\cdots Q_n$.

Let the orientation , sharpness, contrast, cornerness and time-varying cornerness of the feature set $\mathcal{P}$ be,

θ_i^p, S_i^p, C_i^p, CO_i^p and TCO_i^p where $1 \leq i \leq m$ respectively.

Similarly for the set $\mathcal{Q}$ we have,

ϕ_j^q, S_j^q, C_j^q, CO_j^q and TCO_j^q where $1 \leq j \leq n$ respectively.

Initial Figure Of Merit (IFOM):

Here we assign to each pair (P_i, Q_j) that succeed the following conditions, a figure of merit.

Condition 1: If the relative sharpness and contrast of a pair (P_i, Q_j) is less than or equal to a given threshold.

$$\text{i.e.,} \quad R_s = \frac{|S_i^p - S_j^q|}{\frac{1}{2}|S_i^p + S_j^q|} \leq T_s \tag{5.13}$$

$$R_c = \frac{|C_i^p - C_j^q|}{\frac{1}{2}|C_i^p + C_j^q|} \leq T_c \tag{5.14}$$

where

R_s and R_c represent the relative sharpness and contrast respectively.

T_s and T_c the thresholds of sharpness and contrast.

Condition 2: If the absolute difference of time-varying cornerness(D_t) of a pair (P_i, Q_j) is less than a given threshold (T_t).

$$D_t = | TCO_i^p - TCO_j^q | \leq T_t \tag{5.15}$$

Condition 3: If the absolute distance(DIS) between the points P_i and Q_j exceeds some distance threshold (T_δ)

$$DIS = | P_i - Q_j | \geq T_\delta \tag{5.16}$$

Condition 4: If the difference in orientation (D_O) exceeds some orientation threshold (T_O)

$$D_O = | \theta_i^p - \phi_j^q | \geq T_O \tag{5.17}$$

When the lighting condition and other external factors do not change the relative contrast and sharpness changes will be very small. Also if the motion between the frames is not very large, the change in the time varying cornerness will be nearly the same for the two sets of corner points and hence the condition 2.

Condition 3 puts a bound on the amount of motion that is allowed to take place between the image frames. Similarly condition 4 puts a bound on the amount of rotation that is allowed between the frames. Condition 3 and 4 are optional and are not used in our algorithms. If the number of candidate points satisfying conditions 1 and 2 is large then conditions 3 and 4 can be used, which will reduce the number of candidate points for further processing and reduce the computational cost.

The initial figure of merit is equal for all the candidate points, and is set to an arbitrary value of 0.02 , which will survive the first two conditions. This is due to the fact that initially any point from set 1 can be matched with any other point in set 2. Now we determine the support given to the pair (P_i, Q_j) by a given pair (P_h , Q_k). The pair (P_h , Q_k) supports the pair (P_i, Q_j) if the

distance of P_h relative to P_i is close to the distance of Q_k relative to Q_j and also the relative differences in orientation is within some desired threshold. The conditions for the support function is given by,

(a) displacement:

$$|\sqrt{(\ PX_h - PX_i)^2 + (\ PY_h - PY_i)^2\ } - \sqrt{(\ QX_k - QX_j)^2 + (\ QY_k - QY_j)^2\ }| < \text{THRES1} \tag{5.18}$$

(b) orientation :

$$|\ (\theta_h^p - \theta_i^p) - (\phi_k^q - \phi_j^q)\ | < \text{THRES2} \tag{5.19}$$

If both displacement and orientation condition are satisfied then the pair (P_h, Q_k) supports the pair (P_i, Q_j). However, the amount of support will vary for different (P_h , Q_k) s'. In our algorithm , we have added an amount proportional to the distance between (P_h , Q_k) and (P_i, Q_j) to the initial figure of merit. That is,

Figure Of Merit (FOM) = Initial Figure Of Merit + Support due to (P_h , Q_k)

$$F_{ij}(h,k) = \text{IFOM} + \frac{1}{1+(\ \delta_{ij}(\ h,k))^2} \tag{5.20}$$

where, δ_{ij} (h , k) is the distance between (P_h , Q_k) pair and (P_i, Q_j) pair. From equation (5.20) it can be seen that if (P_h , Q_k) is close to (P_i, Q_j) then the support is more. However, in any case the F_{ij} value will not be greater than one as we do not take the case where the δ_{ij} will be zero. δ_{ij} will be zero only when (P_h , Q_k) coincides with (P_i, Q_j) . For such points we do not calculate

the support as it is meaningless to calculate the support for a point by itself.

Since many Q_k's will lie close to each P_h, we take the maximum value of $F_{ij}(h,k)$ which is the one closest to the wanted P_h,

i.e. $\max_{k \neq j}[F_{ij}(h,k)]$

The total support for (P_i, Q_j) can be obtained by taking the average of all the contributions for P_h's

$$S^{[0]}(P_i, Q_j) = \frac{1}{(m-1)} \sum_{h \neq i} [\max_{k \neq j} \{ F_{ij}(h,k)\}] \quad (5.21)$$

where,

m is the number of features present in the first frame.

This gives the initial support for P_i and Q_j. This support can be recomputed using the figure of merit and the initial support using the following updating rules [88]:

$$S^{[1]}(P_i, Q_j) = \frac{1}{(m-1)} \sum_{h \neq i} [\max_{k \neq j} \{ \min (F_{ij}(h,k) , S^{[0]}(P_h, Q_k))\}] \quad (5.22)$$

$$S^{[r]}(P_i, Q_j) = \frac{1}{(m-1)} \sum_{h \neq i} [\max_{k \neq j} \{ \min (F_{ij}(h,k) , S^{[r-1]} (P_h, Q_k))\}] \quad (5.23)$$

After each iteration a confidence measure [91] is calculated using the best and next best matches. The maximum value of the $S^{[r]}(P_i, Q_j)$ will give the best figure of merit M_{ij} for matching P_i and Q_j.

Confidence measure: Let M_{ij1} and M_{ij2} be the figure of merits for the best and the next best matches at a given iteration. Let them be (P_i, Q_{j1}) and (P_i, Q_{j2}) and let $\overline{M}$ be the average of non-zero M_{ij1}'s. Now we define the confidence C_{ij1} in the pair (P_i, Q_{j1}) as follows.

If $M_{ij1} < \alpha \overline{M}$, where $0<\alpha \leq 1$ is a parameter, then ,

$C_{ij1} = 0.$

If $M_{ij1} \geq \alpha \ \overline{M}$ then

$$C_{ij1} = \frac{M_{ij1}}{M_{ij2}}$$

If $M_{ij2}=0$ then,

$C_{ij1} = \infty$ (or a large number)

We have chosen a value of 0.01 for α in our experiments.

The pairs having a high confidence value , other than ∞ (a large number of value 500 is kept in our experiments) are selected as the matched pairs. In some cases a point in a set $\mathcal{Q}$ is assigned to two different points in the set $\mathcal{P}$. This is inevitable and in such cases ' first come first served ' rule is used. Hence the second point will be made to take the next best match available to it. In such cases, where the next best match is also assigned to some other point,the point will be dropped.

Once a set of eight or more correspondences match points have been found as explained in the above algorithm we can use the formulation presented in Chapter 3 to estimate the motion parameter of the object.

When multiple objects are moving in a scene, the situation is slightly more complicated. Here we assume that each of the objects move with different velocities. As before the corresponding points are found using the above algorithm. Then the corresponding points having nearly the same displacements are clustered together. If the scene contains ' N ' different objects moving with N different velocities, then , ideally we will have N clusters. These clusters are separated and the points belonging to the same cluster will be from the same

object. Hence these points are used in the motion parameter estimation algorithm to give the motion parameters of that object. Similarly we can find the motion parameters of all the objects present in the scene. It should be emphasised here that in all the above cases, it is assumed the objects are rigid and undergoing a rigid motion. Non- rigid motion is different and the analysis of matching is also different. Goldof, Lee and Huang [92] recently discussed non-rigid object motion and correspondence in their paper.

It should be pointed out here, that although our images contain multiple objects, they are not moving independently . Hence, the above correspondence method is applicable. In our image acquisition procedure , we moved the camera and hence all the objects are treated as single object.

The methods described above are tested for different real images. Some of the results are shown in Table T.5.1– T.5.4. We used 8 and 10 corner points which are selected from the results of the relaxation based matching algorithm, based on match merit values. In order to identify time-varying corners we have to use atleast two consequetive frames. Two such pair of frames are used for matching. In the tables we have shown the frame pairs which we have used for motion parameter estimation. Although we use three frames here, it can be seen that the motion parameter estimation is for two adjacent frames. In the tables we give two sets of results for each matched pairs. The two sets are selected from the matched corner points. The main drawback is that we cannot compare these results with the actual motion that took place during our image acquisition procedure. This is due to the lack of knowledge of the ground truth. The other problem is due to the camera calibration. The Robot is calibrated initially as per the manufacturers specification as explained in Chapter 4. The origin of the Robot

TABLE –T 5.1

Results of THLAM and the Modified method for OMS: algorithm selected corner points (large motion case- N=8).

No	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
*fr*12–23.								
1	58.0899	57.6354	48.7901	26.2728	2.7088	2.1588	0.1170	0.6752
2	37.8440	61.0438	67.8620	31.9733	−0.0779	−0.0274	0.0427	−0.0133
*fr*23–34.								
1	53.6633	54.6342	124.0746	5.9076	−3.4193	0.2193	0.5633	1.2298
2	87.5491	64.7525	154.6169	5.7636	−7.4455	0.2433	0.1167	0.2952
*fr*34–45.								
1	45.3767	90.9786	44.6401	13.1768	1.3351	7.6843	0.9356	0.6812
2	128.0980	124.2025	123.4183	13.3633	-0.2269	-0.1384	-0.1060	-0.2247
*fr*45–56.								
1	156.4609	100.0771	111.0391	62.6916	-0.8869	-0.3092	0.1964	1.0934
2	163.3571	106.2521	86.5108	64.8600	-0.6080	-0.6305	-0.3507	0.1866

TABLE –T 5.2

Results of THLAM and the Modified method for OMS: algorithm selected corner points (large motion case- N=10).

No	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
*fr*12–23.								
1	118.8749	140.5414	114.4031	39.1478	-0.4055	-0.6126	1.0109	0.9541
2	34.1901	72.8088	61.4495	52.2488	-2.7092	0.6910	0.5046	0.4393
*fr*23–34.								
1	133.5923	109.2142	130.1789	37.7419	-0.1139	0.3469	0.1366	0.8328
2	20.2134	70.3374	94.5028	27.3973	-0.2860	0.1614	-0.0030	-0.1166
*fr*34–45.								
1	66.5488	151.4311	105.3766	170.5194	0.3411	-1.5587	2.2244	2.9819
2	60.5959	130.2312	115.5025	162.6588	12.3330	-9.8576	0.3870	1.9116
*fr*45–56.								
1	111.5114	136.7151	125.4036	29.2604	-0.0631	-0.7500	0.5701	1.7808
2	162.8916	107.1059	89.7155	63.4752	-0.6296	-0.5967	-0.1827	0.3399

TABLE –T 5.3

Results of THLAM and the Modified method for OMS: algorithm selected corner points (small motion case- N=8).

No	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
*fr*78–89.								
1	89.3985	91.1192	1.2706	0.7064	0.4875	1.9644	1.1454	1.7583
2	126.7119	118.9771	129.6953	8.1029	0.0702	-0.8832	0.7240	0.9880
*fr*89–9*A*.								
1	147.9403	91.0056	122.0401	27.6349	-0.6983	-0.2098	0.3098	1.3417
2	137.0520	70.7763	126.6192	7.9317	0.5999	0.2368	0.3099	1.8290
*fr*9*A*–*AB*.								
1	172.1656	88.9984	82.2307	9.3894	7.0514	-0.9642	-0.2930	0.1045
2	42.1524	81.7389	49.0391	6.7467	1.5804	1.7107	0.7531	1.2026
frAB–*BC*.								
1	88.6460	55.8720	145.8376	31.9369	-5.7852	-2.3003	0.6929	-0.5558
2	62.9439	27.0561	90.0159	9.0237	1.6733	22.2149	1.3602	2.2206

TABLE –T 5.4

Results of THLAM and the Modified method for OMS: algorithm selected corner points (small motion case- N=10).

No	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
*fr*78–89.								
1	92.8635	137.0450	132.8116	7.6641	0.0108	0.0854	-0.3354	0.8502
2	123.4900	138.8202	111.0532	32.1284	-0.0520	-0.1025	2.0165	2.1860
*fr*89–9*A*.								
1	67.7921	25.7489	77.6335	3.2999	1.4353	0.3888	-0.0727	-0.0030
2	133.4391	114.6636	126.4571	3.7524	0.4036	1.0922	0.3098	1.3417
*fr*9*A*–*AB*.								
1	84.7036	139.2723	130.2334	19.1396	-0.6209	-0.8054	-0.5300	0.9028
2	118.8778	127.5629	128.9456	14.9456	-0.0073	-0.0080	0.5416	1.6421
frAB–*BC*.								
1	162.7817	106.7115	94.0291	45.7443	-0.2005	-0.5792	0.4842	0.7222
2	55.9700	143.6694	78.7880	40.1492	0.6201	-1.3322	2.0208	1.3203

and the camera centre is different. This posed several problems in having a same axis of rotation. As we have given unit translation motion in each of the coordinate axes while acquiring the image sequences (see Table T.4.1–T.4.4), we may expect the scaled translation of t_y/t_x and t_z/t_x to be unity. From the tables T.5.1 –T.5.4 we observe that a large number of estimated values in case of the modified method have less error compared to the linear method (THLAM). When the number of feature points is increased the performance slightly detoriates. This is due to the errors in finding the correct match points.

We have also tested both the THLAM and the modified methods for a set of hand-picked orner points by a human being. The hand picked points are fed to the motion estimation algorithm and the results are obtained. We obtained the motion parameters for a varying number of hand-picked corner points. In figure 5.7 we numbered the hand-picked corner points. We used 8.,12,16 and 20 corner points for motion parameter estimation. The results are given in Tables T.5.5–T.5.8. The image pair numbers are given in Hexadecimal number representation in these tables. We reitarate here that increasing the number of corner points does not increase the performance , an obsevation made in the synthetic images also.

As a remark we say, that we should be careful in selecting the various thresholds. At the moment it is a trial and error method. As we use many threshold at various stages, the overall performance of the algorithm may be affected.

TABLE –T 5.5

Results of THLAM and the Modified method for OMS: Hand-picked corner points (N=8).

fram	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
Large Motion sequence:								
1-2	105.6466	64.0143	30.9642	4.5009	0.6205	0.3656	-1.6654	-0.7935
2-3	79.9129	10.6311	93.3223	2.9834	0.4659	0.1242	0.2460	0.9113
3-4	98.2663	9.6361	86.3759	1.3673	0.2664	0.5603	0.8824	2.0317
4-5	91.5107	9.9301	80.1878	1.6748	0.6975	0.1821	0.6491	1.9253
5-6	96.9368	20.5436	70.7626	6.1136	-0.3037	0.0125	0.1771	1.3254
Small Motion sequence:								
7-8	96.0540	38.9906	51.6626	0.8469	0.7529	0.4629	0.3307	0.7751
8-9	109.3476	65.2762	147.7536	8.6041	-4.5221	-0.2380	0.6974	-0.3138
9-*A*	64.4424	51.7200	49.0186	1.3338	0.0923	0.3863	0.30447	0.2425
A-B	82.9284	7.8807	93.4605	2.0491	-0.9813	0.3033	1.8144	-0.7879
B-C	154.3907	85.8383	115.2201	0.4923	0.0695	0.5599	-0.1438	0.0188

TABLE –T 5.6

Results of THLAM and the Modified method for OMS: Hand-picked corner points (N=12).

fram	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
Large Motion sequence:								
1-2	106.0285	61.5566	33.4328	4.3915	0.6347	0.4256	0.0204	0.0203
2-3	142.8430	56.5543	104.3056	5.8040	1.8727	-1.7871	-0.3037	0.6789
3-4	94.4407	7.9857	83.3762	1.4622	0.3275	0.5043	0.8024	2.0501
4-5	118.6977	29.0501	85.8640	1.5494	0.9921	0.2047	0.3777	1.5177
5-6	94.9389	14.7405	76.1469	5.7877	-0.3654	0.0278	0.0746	1.6296
Small Motion sequence:								
7-8	34.9179	70.0761	62.6186	5.5222	-0.5050	3.7546	0.2237	0.3508
8-9	25.4393	70.0948	74.8161	4.1972	23.1035	15.1512	0.0845	0.1271
9-*A*	72.0520	49.2478	46.2088	1.7875	0.2511	0.4393	0.5720	1.5522
A-B	49.3835	66.8996	49.4714	0.5993	0.6136	0.8579	0.4247	1.1156
B-C	89.1689	33.1249	56.8883	0.7570	0.3699	0.5028	0.6853	1.5962

TABLE –T 5.7

Results of THLAM and the Modified method for OMS: Hand-picked corner points (N=16).

fram	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
Large Motion sequence:								
1-2	116.3831	59.4773	42.4416	3.6587	0.7426	0.5741	0.5898	1.0399
2-3	68.1920	61.4870	142.7797	3.3673	0.2347	-2.2387	0.4876	0.0731
3-4	109.5983	28.2903	70.4379	2.1643	0.9232	0.2942	0.3387	1.2259
4-5	129.9069	39.9253	88.9804	1.6996	1.0444	0.2327	0.3179	1.2750
5-6	88.4880	20.2723	69.7891	5.0684	0.6011	0.4432	0.7562	1.4502
Small Motion sequence:								
7-8	27.3683	66.3572	77.0129	5.8781	0.3982	0.7057	0.1680	0.1338
8-9	54.6060	50.0373	59.8680	0.7298	0.5528	1.0490	-01245	-0.0820
9-*A*	48.7500	55.7267	60.1235	1.7184	0.9279	0.7917	0.58441	0.5318
A-B	36.0674	59.2598	73.0145	0.8940	0.6445	0.9488	-0.0522	-0.0298
B-C	70.5918	38.7868	57.9252	0.8817	0.6162	0.8421	1.0821	0.8871

TABLE –T 5.8

Results of THLAM and the Modified method for OMS: Hand-picked corner points (N=20).

fram	α	β	γ	θ	LINEAR METHOD		MODIFIED METHOD	
					t_y/t_x	t_z/t_x	t_y/t_x	t_z/t_x
Large Motion sequence:								
1-2	115.4259	54.6138	46.1276	3.7552	0.9839	0.4660	0.5690	1.0032
2-3	63.0516	68.7151	144.5039	3.5058	0.2193	-3.6323	1.1742	1.2347
3-4	113.8470	23.8995	88.5082	1.6818	1.3322	0.1732	0.1899	1.4503
Small Motion sequence:								
7-8	18.8514	74.9202	78.9527	2.0760	3.1354	0.9365	-0.0160	0.2947
8-9	74.1086	41.4060	52.9833	0.8286	0.4190	0.9574	0.0429	-0.0416
9-*A*	40.6931	56.5146	78.7835	1.1912	2.6915	0.3134	0.1992	0.1565
A-B	32.6390	59.7268	78.9475	2.1282	7.8289	0.1543	0.1047	0.0532
B-C	97.2447	36.4342	54.5244	0.5172	0.6445	0.9884	0.6471	0.5530

PART-2

REGION BASED MOTION PARAMETER ESTIMATION

CHAPTER SIX

INTRODUCTION AND REVIEW OF MOMENTS

6.1 INTRODUCTION:

Motion parameter estimation requires some kind of features which are invariant to translation, rotation and scale change to be detected from the images as a first step. In part I of the thesis we have seen 'corners' as feature points in the motion parameter estimation. However, there are various other features like contours , lines etc., that can be used as features. In part II of this thesis , an attempt is made to use moments as features to estimate the motion parameters. Moments are kinds of regional features as they are obtained using a large number of points, usually from a surface of an object. In the following section we present some aspects of moments and moment invariants(MI). In Chapter 7 we show how the correspondence problem for regions is solved using moment invariants. Chapter 8 deals with the motion parameter estimation using moments.

6.2 MOMENTS AND MOMENT INVARIANTS :

One of the most important tasks of a visual system is to identify (recognize) an object irrespective of its position (orientation), size etc., Human vision does this marvelously. However, it is still not clear how the vision system recognize the objects from large classes of objects. The visual system uses many sources of information [multisensor fusion], including a vast prior knowledge base which human beings have learnt from their experience. Having said that, we

should accept that we have very little knowledge of how we, human beings, represent different classes of objects. A baby learns to identify objects from its images (2-D) even without actually seeing the 3-D objects. Before the baby learns what a learning process is , the baby completes the whole learning. This may be one of the reasons why scientists could not develop a good model for object recognition. It is worth mentioning here that Poggio and Edelman [93] showed that " the visual recognition of 3-D objects does not require that their 3-D shapes have to be explicitly represented in memory, only that the system has been exposed to enough views of a given object to be able to recognize another such view when it sees it". Recognition of objects also differs from person to person. This poses a serious question of using the same types of features for different classes of objects. However, recognition of 3-D objects by machine(computer recognition) can only be achieved by using a set of such features. These features should be invariant under size change, motion (including translation, rotation) and position.

6.2.1 2-D moments:

Moment invariants are a class of features which are invariant under the above said conditions. Historically moment invariants were first used by Hu [94] for recognition problem. In his classical paper , he derived a set of moment invariants that truly satisfied the invariance requirement.

The 2-D moment of order (p+q) is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p \, y^q \, f(x,y) \, dx \, dy \qquad (6.1)$$

where p,q = 0,1,2,3,...

Here it is assumed f(x,y) is continuous and that it has non-zero values only in the

finite part of the x-y plane. The whole theory on moments is based on the uniqueness theorem which is stated below.

Uniqueness theorem: The double moment sequence $\{ m_{pq}\}$ is uniquely determined by f(x,y) and conversely f(x,y) is uniquely determined by $\{ m_{pq} \}$. Hu proved the invariant properties of moments using the theory of algebraic invariants [95].

The centralised moments are given by,

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-\overline{x})^{p} (y-\overline{y})^{q} f(x,y)\, dx\, dy \qquad (6.2)$$

where $\overline{x} = \frac{m_{10}}{m_{00}}$, $\overline{y} = \frac{m_{01}}{m_{00}}$

Using the centralised moments of second and third order, a set of *seven* moment invariants were derived. They are as follows,

$$\phi(1) = \mu_{20} + \mu_{02} \qquad (6.3\ a)$$

$$\phi(2) = (\mu_{20}-\mu_{02})^2+4.\mu_{11}^2 \qquad (6.3\ b)$$

$$\phi(3) = (\mu_{30}-3.\mu_{12})^2+ (3.\mu_{21}-\mu_{03})^2 \qquad (6.3\ c)$$

$$\phi(4) = (\mu_{30}+\mu_{12})^2 +(\mu_{21}+\mu_{03})^2 \qquad (6.3\ d)$$

$$\phi(5) = (\mu_{30}-3.\mu_{12})(\mu_{30}+\mu_{12})[(\mu_{30}+\mu_{12})^2-3.(\mu_{21}+\mu_{03})^2] + (3.\mu_{21}-\mu_{03})(\mu_{21}+\mu_{03})[3.(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2] \qquad (6.3\ e)$$

$$\phi(6) = (\mu_{20}-\mu_{02})[(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2] + 4.\mu_{11}(\mu_{30}+\mu_{12})(\mu_{21}+\mu_{03}) \qquad (6.3\ f)$$

$$\phi(7) = (3.\mu_{21}-\mu_{03})(\mu_{30}+\mu_{12})[(\mu_{30}+\mu_{12})^2-3.(\mu_{21}+\mu_{03})^2]- (\mu_{30}-3.\mu_{12})(\mu_{21}+\mu_{03})[3.(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2] \qquad (6.3\ g)$$

The functions $\phi(1)$ ... $\phi(7)$ can be normalised to make them invariant under a scale change by substituting the normalised central moments η_{pq} for μ_{pq} in the

above equations; η_{pq} is defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{6.4}$$

where, $\gamma = \frac{p+q}{2}+1$.

Nevertheless , Hu used only the first two moment invariants for the problem of recognition of English alphabeticals. Yet some problems are reported with letters like ' M ' and ' W '. It is interesting to note that the same difficulty is faced with babies in their initial learning stage. Hu suggested that this problem can be tackled by using a large number of moment invariants.

Dudani et al,[96] used moment invariants for the problem of aircraft identification. They used both the boundary and the silhouette of the objects for moment invariant calculation. Boundary moments provide more information about the high frequency components (can be used for shape analysis) and the silhouette moments provide the low frequency components. First a training set of feature components are computed and stored. This is done by using a set of identification objects under various orientations and computing the moments. Usually a large database is created for generating the library. Then, given an image, a set of moment invariants are computed and compared with the library set. Two types of classification rules are used: (a) Bayes decision rule and (b) Distance weighted k nearest neighbour rule. They report a very low number of misclassifications. However, if another object is added to the library the whole set of experiments have to be repeated again and the library needs to be updated for the whole set of objects.This is computationally costly and time consuming. Besides, as the number of objects increases , the classification problems become increasingly complicated.

Wang and Hall [97] used moment invariants for matching an optical image with a radar image. They described a hierarchical method of matching two images. However, it was pointed out by Maitra [98] that the observed difference between the two sets of moment invariants are of the order of 10^{3} and 10^{4} in real terms, as Wang and Hall have tabulated the moment invariants as their logarithmic values. These errors were due to numerical inaccuracies , lack of full interpolation and contrast sensitivity of images.

Maitra [98] proposed a modification for the seven MIs which took care of scale change and contrast change in the given images. In addition to the seven MIs he also proposed a set of six moment invariants which are some functions of ϕ_i's. They are defined as below.

$$\beta\ (1) = \frac{\sqrt{\phi(2)}}{\phi(1)} \tag{6.5 a}$$

$$\beta\ (2) = \frac{\phi(3).\mu_{00}}{\phi(2).\phi(1)} \tag{6.5 b}$$

$$\beta\ (3) = \frac{\phi(4)}{\phi(3)} \tag{6.5 c}$$

$$\beta\ (4) = \frac{\sqrt{\phi(5)}}{\phi(4)} \tag{6.5 d}$$

$$\beta\ (5) = \frac{\phi(6)}{\phi(4).\phi(1)} \tag{6.5 e}$$

$$\beta\ (6) = \frac{\phi(7)}{\phi(5)} \tag{6.5 f}$$

In [99] , Teague gave a method of deriving higher order moments from

lower order moments in a simple and straight forward manner. He used a new set of moments called Zernike moments . Zernike moments performed better than the ordinary moment invariants. The author's main interest was in characterising the image by a small finite number of moments. He has shown that by incorporating higher order moments the performance of reconstruction can be improved. Reddy [100] expressed the ordinary MIs in terms of radial and angular MIs. In [101], Reeves et al, used MIs for identification of 3-D shapes. They classified the objects by using some statistical characteristics like mean, variance, skewness etc.,They have identified a few simple objects like buildings, tanks etc., In [102] a method of using aspect ratio normalisation is given. The "six aeroplane" identification problem was attempted. Using minimum Euclidian distance after 'variance balancing ' criteria of the feature vector's, they got better results compared to simple Euclidian distance classification. Variance balancing is a weighting technique to make the effect of feature elements more equal. Abu-Mostafa and Psaltis [103] have discussed the information properties of MIs in terms of loss, supression and redundancy. Ogunbona [104] used MIs for motion compensated coding. The signal to noise ratio is significantly improved when motion compensation is used. A two step matching algorithm is given for matching moving areas in the given images. He used MIs as feature vectors for segmenting the moving areas from static areas.

More recently Cyganski and Orr [105] , using tensor theory , derived MIs. In addition, they showed a way of finding the orientation parameters from MIs. It should be pointed out that the tensor method of estimating the orientation does not require any form of point correspondence. It is seen in most of the examples, that the authors solved the 2-D affine transform parameters rather than the 3-D

transform parameters. However, a set of equations are given for 3-D affine transform estimation. No noise analysis is given and also the method is not fully explored. Also, while finding the transform parameters there is an ambiguity in finding the orientation of the rotation axis. Bamieh and De Figueiredo [106] extended the tensor theory for 3-D object recognition from its 2-D projections. In both the above methods a library of standard orientation is needed. In [106] , recognition of polyhedral objects are given. Bamieh et al first stored the MI's of all the possible objects in a library. Hence, if an image of the object is given , then a wire frame of the object is extracted. From this an attributed graph is formed. The MIs of the attributed graph is compared with the library of the objects , to get the possible object in the scene.

Teh and Chin [107] discuss the problem of using digitized pixels on MIs. They suggested some numerical integration approaches for better approximation when using the digital pixels for calculating the MIs. A comparison of different types of moments and the effect of noise on image analysis problems is given in [108]. They assumed a zero mean White noise. As expected higher order moments are vulnerable to noise. However, higher order moments contain fine details. Hence, a compromise is to be made when discarding the higher order moments.

6.2.2 3 -D moments:

Extension to 3-D moment invariants was made by Sadjadi and Hall [109]. 3-D MIs were derived on similar lines to Hu [94] using algebraic invariants. They used 3-D data while finding the MIs and used these MIs for 3-D object recognition. Lo and Don [110] gave a method of identifying and positioning a 3-D object from its 3-D moments. A simple method based on Group theory is given for obtaining higher order 3-D moments. Obtaining higher order moments in the

Sadjadi and Hall [109] method is cumbersome. Lo et al [110] also gave a method of finding the orientation using the 3-D MIs .

The tensor theory of deriving the MIs is extended in [111] for medical images. They investigated the moment function method for registering 3-D test images to a standard image. They considered the problem of quantifying the left ventricular function from gated blood pool SPECT images. It is shown that MI are not particularly favourable for such tasks. They suggested the Principal axis method as a suitable candidate for performing such tasks.

From the foregoing discussion we conclude that moment invariants are important features for object recognition and shape analysis. In the following chapters we show how the moments are used to find the motion parameters of a rigid planar patch.

CHAPTER SEVEN

REGIONAL CORRESPONDENCE USING MOMENT INVARIANTS.

7.1 INTRODUCTION:

In the last chapter we have seen moment invariant functions given by Hu [94] and the moment invariant functions given by Reddy [100]. These functions are invariant to rotations, translations and scale change. Hence these can be used to identify similar patches from given image sequences. In the following we show how these functions can be used to match a rigid planar patch image in different views under noisy conditions.

7.2 MATCHING THE MOMENT INVARIANTS:

In our simulations we used both sets namely, $\phi(1)$, ... $\phi(7)$, and $\beta(1)$, ... $\beta(6)$, for matching the two images. We have performed some experiments to analyse how effective are the MIs under noisy conditions.

It should be emphazised here, that while using regions as features, the first task will be regional segmentation. Because of motion, the regions in the two images may differ in both size and orientation. It is here we will be using the moment invariants for matching the regions. From the given image sequences a set of a few regions is first segmented. Then the moment invariants of these regions are calculated as mentioned in chapter 6. A fast algorithm can be used to calculate the moment invariants [112] in case of real images. The regions having at least three or four MIs within a set threshold are then identified as ' Regions Of

Interest '(ROI). Once the ROIs are found , we say these ROIs are corresponding regions in the image frames and should have come from the same object surface.

In order to get a feel of how the moment invariants behave under noisy conditions we have performed a large set of simulations. We generated N (in our case 1000) random points which lie on a planar patch given by z=a.x+b.y+c. Then these 3-D points are given a known rotation ,θ, about an axis whose direction cosines are given by n_1, n_2 and n_3. Thus, we get two sets of points, before and after motion. In fact, we generate three sequential frames and the second frame is assumed to be the one which undergoes both the forward and the backward motion. This is because our motion estimation model requires three image frames which undergo a constant motion in 3-D. The above sets of points are then projected parallely (Orthographic) to the image plane. The MIs are calculated for the projected planar image. The selected 3-D points are from a plane whose equation is as given above. The plane is assumed to be 10 units away from the camera focal length. The x and y coordinates of the points are within 2 units and 10 units. To generate noise we have two options:

(1) adding Gaussian/Uniform random noise to each pixel
and (2) dropping a known number of pixels.
The former method is used in most of the works for noise analysis. However, the latter method is proper for our case. This is due to the image segmentation. Due to the illumination changes while acquiring the image frames, the pixels will have varying intensity values. Thus the segmentation algorithm may detect a varying number of pixels for the same segments in two image frames; hence the selection of the second method. We have varied the noise level from 0.1% to 10% that is, we randomly drop 1 to 100 pixels in 1000 points. It should be pointed out here

that the actual 3-D points are scattered all over the planar patch. However, in real cases, the segmented pixels will be confined to a small region and hence the effect of noise will be less compared with what we have reported here. In addition to that , a clever segmentation algorithm will improve the situation. From equations 6.3.a–6.3.g we can see that MIs are calculated from high order terms in X and Y. As the order of the normalised central moments increase, MI values become smaller and smaller. Hence, $\phi(2)$ ... $\phi(7)$ values will be very small and hence comparison will yield poor conclusions. Due to this problem we used $\phi(1)$, $\beta(1)$, $\beta(4)$ and $\beta(5)$ for comparing the MIs.

In the simulation experiments we generated three motion frame sequences. The second frame is assumed as the initial one. This frame is moved backwardly by a known motion to get the first frame. The third frame being the reference frame moved forwardly by the given motion. Hence we have to compare the moment invariants of frame 2 with the moment invariants of frame 1 and frame 3.

In figures 7.1–7.4 we have shown the performance of $\phi(1)$, $\beta(1)$, $\beta(4)$ and $\beta(5)$ in presence of noise. Due to the discreteness involved in the calculation the MI values in frame 1 and frame 3 deviate from their reference MI values in frame 2.

If we accept the MIs of frame 1 and frame 3 to be within a set threshold , say 10% of frame 2, we conclude that the image patches correspond to each other and are from the same surface. The accepted ±10% region is shown by two bounded lines. In figures 7.1–7.4 we can see in general up to about 4–5% of noise can be accepted. Now, the lower order moments of these segments can be used for motion estimation. In the next chapter we will discuss the formulation for deriving the motion parameters using the lower order moments.

Figure 7.1
Effect of noise on PHI–1

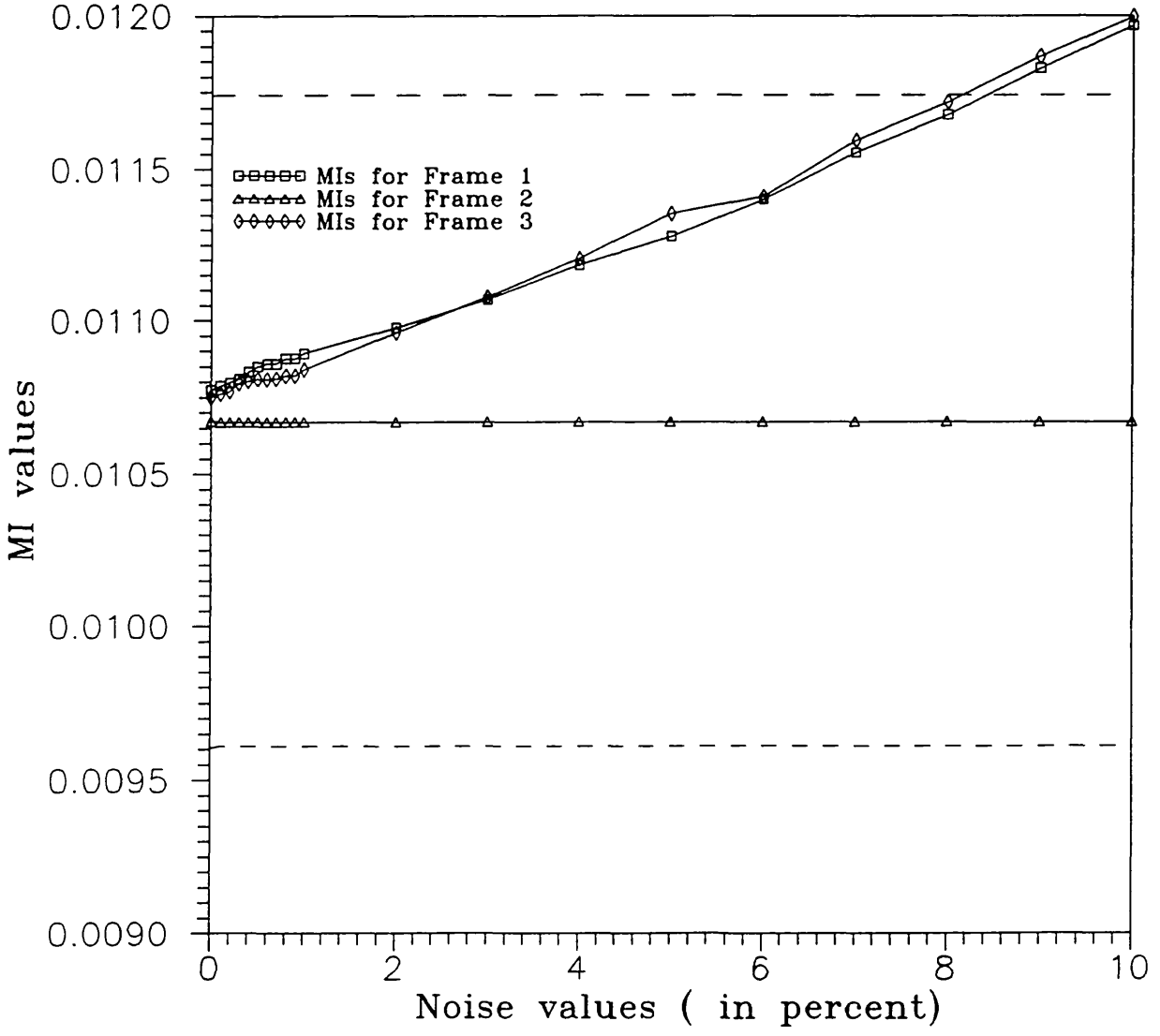


Figure 7.2
Effect of noise on BETA-1

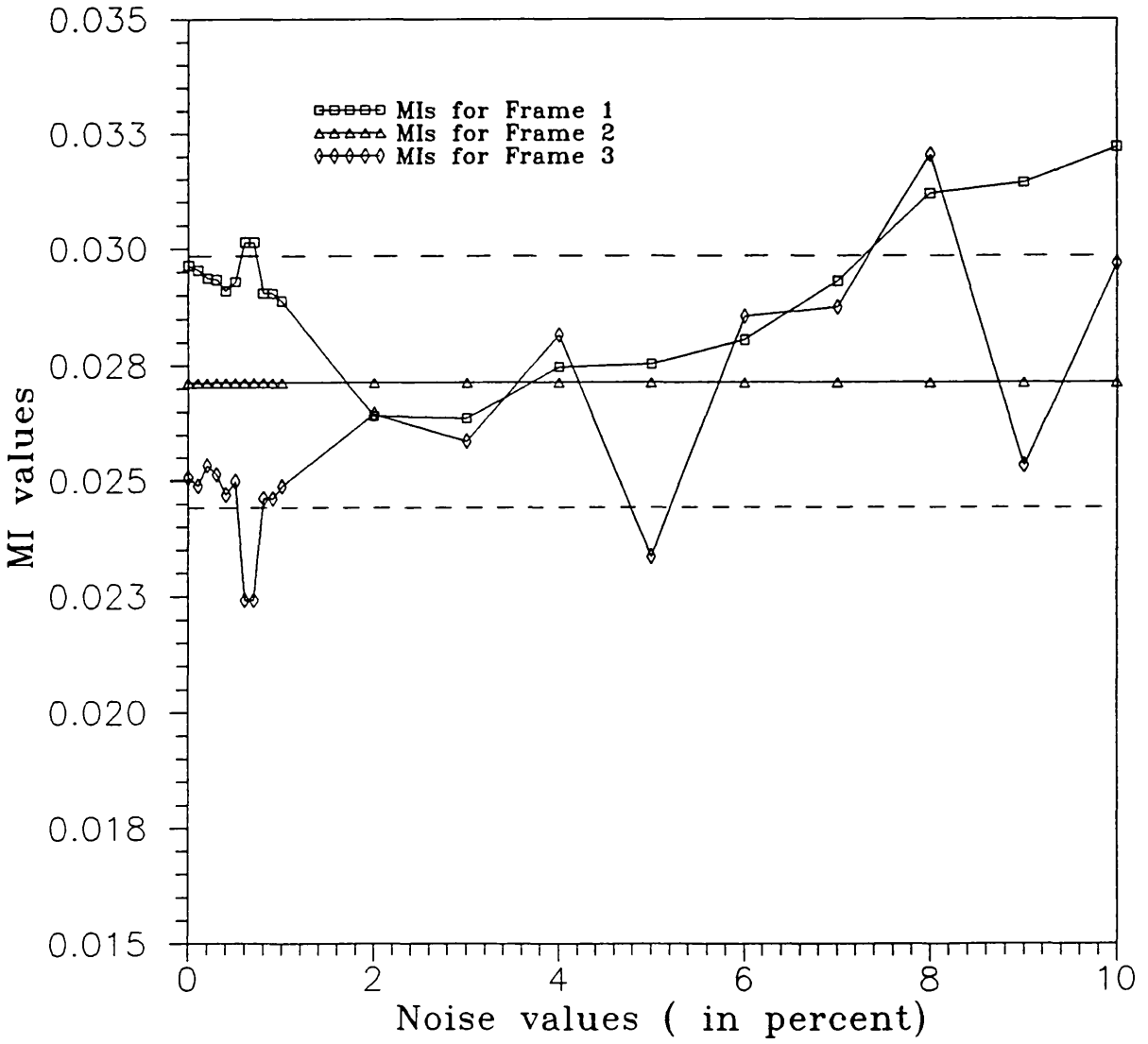


Figure 7.3
Effect of noise on BETA–4

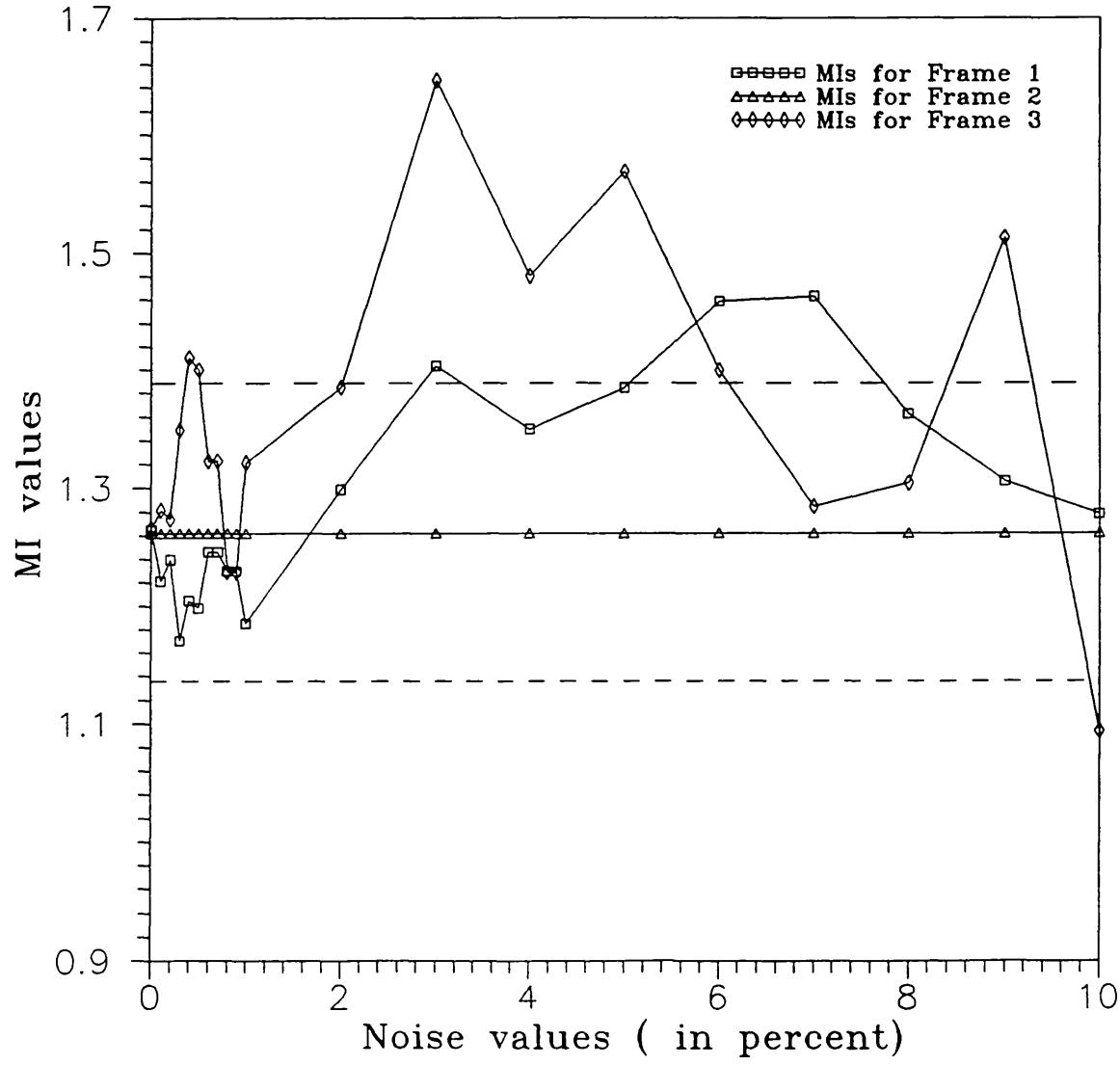
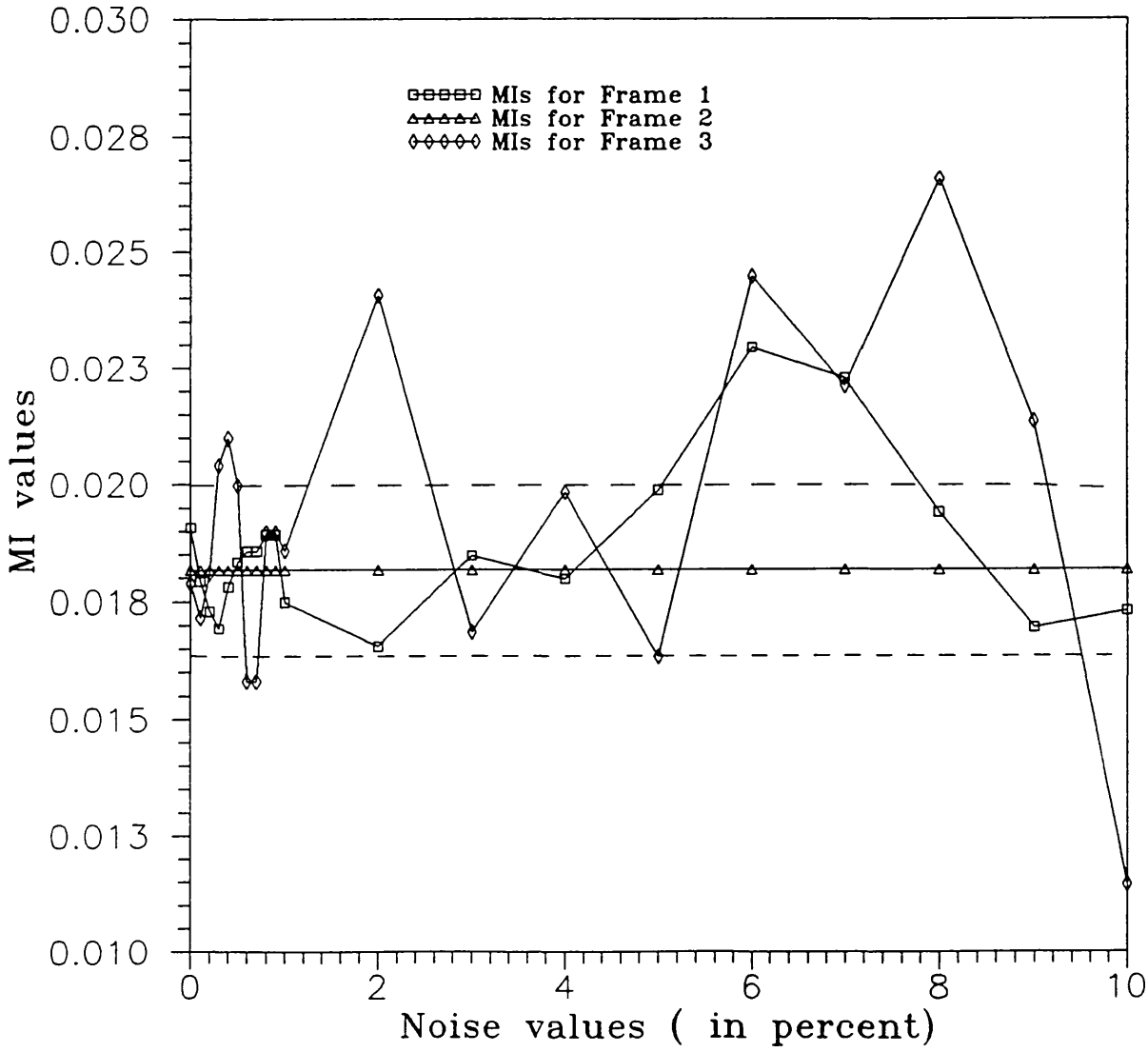


Figure 7.4
Effect of noise on BETA–5



CHAPTER EIGHT

USING MOMENTS FOR MOTION ESTIMATION

8.1 INTRODUCTION:

In chapters 6 and 7 we saw the importance of moment invariants as features for identification purposes and also saw how they can be used for matching segments when an object is in motion. In this chapter we discuss a formulation which uses moments of the matched segments for estimating the motion parameters.

8. 2 FORMULATION USING FORWARD AND BACKWARD MOTION: PURE ROTATION CASE:

In this formulation moments are used to estimate the motion parameters of an object comprising of a Rigid Planar Patch (RPP).

8.2.1 Forward motion:

Let a RPP object be rotated in 3-D space by an angle θ around an axis which is defined by the direction cosines (n_1,n_2,n_3) in the x,y and z axis respectively. Let X and Y axis of the image plane be parallel to the x and y axis of the 3-D axis. Now we consider a point p_i(x ,y,z) on the RPP. Let p_i' (x′, y′, z′) be the coordinates after rotation. (See Figure 8.1)

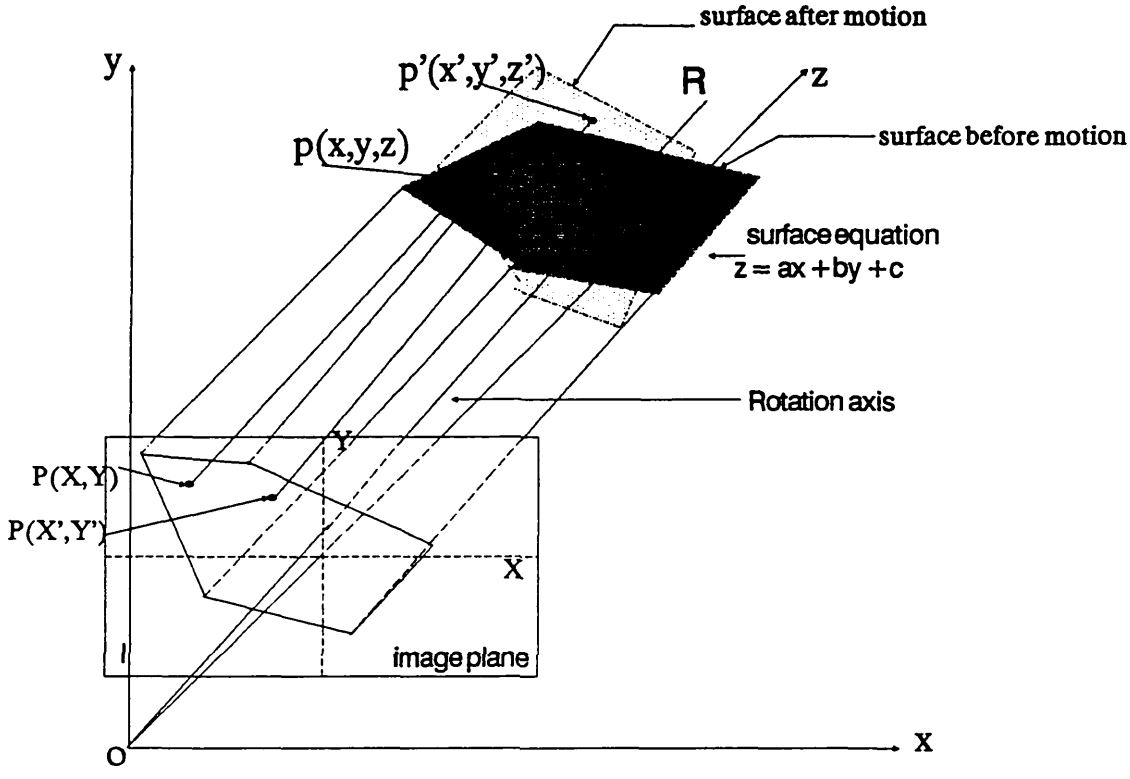


Figure 8.1

Orthographic projection of a rigid planar surface.

The points p_i and p_i' are related by the following equation

$$x' = \mathbf{R}\, x \tag{8.1}$$

where

R is the rotation matrix due to the 3-D rotational motion and is given by

$$\mathbf{R} = \begin{bmatrix} n_1^2+(1-n_1^2)\cos\theta & n_1n_2(1-\cos\theta)-n_3\sin\theta & n_1n_3(1-\cos\theta)+n_2\sin\theta \\ n_1n_2(1-\cos\theta)+n_3\sin\theta & n_2^2+(1-n_2^2)\cos\theta & n_2n_3(1-\cos\theta)-n_1\sin\theta \\ n_1n_3(1-\cos\theta)-n_2\sin\theta & n_2n_3(1-\cos\theta)+n_1\sin\theta & n_3^2+(1-n_3^2)\cos\theta \end{bmatrix}$$

where,

n_1,n_2, n_3 are the direction cosines of the rotation axis and θ being the rotation angle.

x and x' are the vectors which defines the 3-D points before and after motion respectively.

Assuming θ, the rotation angle ,to be small, **R** can be reduced to,

$$\mathbf{R}_a = \begin{bmatrix} 1 & r_{12} & r_{13} \\ -r_{12} & 1 & r_{23} \\ -r_{13} & -r_{23} & 1 \end{bmatrix} \tag{8.2}$$

where,

$\mathbf{R}_a$ = Approximate rotation matrix

$$r_{12} = -n_3\theta \; ; \; r_{13} = n_2\theta \; ; \; r_{23} = -n_1\,\theta. \tag{8.3}$$

Let the RPP be defined in the 3-D space as below,

$$z = a.x + b.y + c \tag{8.4}$$

where, { a, b, c} are known as the planar coefficients of the RPP.

If we project the space points parallel to the z-axis (Orthographic projection) into the image plane (2-D plane) and substitute for the planar equation as in equation 8.4 we get,

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} (1+a\,r_{13}) & (r_{12}+ b\,r_{13}) \\ (-r_{12}+a\,r_{23}) & (1+b\,r_{23}) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} c\,r_{13} \\ c\,r_{23} \end{bmatrix} \tag{8.5}$$

If we have many of the image points we will be able to calculate the moment invariants of the RPP before and after motion in the image plane. As already pointed out that moment invariants are the invariant features irrespective of scale, translation and rotation. Thus, ideally we expect them to be the same for the same RPP before and after motion. However, due to noise they will be slightly different. Hence , a threshold is set to detect whether they are equal before and after motion , within the threshold. If they satisfy the condition then we can say that the points are from one and the same RPP of the object. Hence as mentioned in the previous chapters the lower order moments are used to estimate the motion parameters.

The $(p,q)^{th}$ centralised moment for the RPP *after motion* is given by,

$$\mu'_{pq} = \int_{Ar} (X' - \overline{X'})^p (Y' - \overline{Y'})^q f(X', Y')\, dA \tag{8.6}$$

where, $\overline{X'}$, $\overline{Y'}$ are the centroids after motion around which the central moments are taken.

$f(X', Y')$ represents the intensity function after motion.

The integration is performed over the area of the RPP.

If the imaging characteristics do not change during the actual motion then the intensity function can be assumed the same as that of the intensity function before motion and can be taken as unity throughout the patch. The intensity is mainly used for segmenting the image. Regions having same/similar intensities are grouped together to form a region of interest. Having identified the regions , the intensity can be set to any value say unity. Hence if a pixel is within the selected region of interest then the pixel will have an intensity value of one. Otherwise the pixel is not considered at all for that particular region of interest. Representing the equation (8.6) in discrete form:

$$\mu'_{pq} = \sum (X' - \overline{X'})^p (Y' - \overline{Y'})^q \qquad (8.7)$$

where $\sum$ means the double summation taken over all the points in the same region.

For the second order central moments , we write

$\mu'_{20} = \sum (X' - \overline{X'})^2$

Substituting for X' from equation (8.5) and using A for $\overline{X'}$ we get,

$\mu'_{20} = \sum \Big((1+a\, r_{13})\, X + (r_{12} + b\, r_{13})\, Y + c\, r_{13} - A \Big)^2$

$$\mu'_{20} = (1+a\, r_{13})^2 \sum X^2 + 2\,(1+a\, r_{13})\,(r_{12}+b\, r_{13}) \sum XY + (r_{12}+ b\, r_{13})^2 \sum Y^2 +$$
$$(cr_{13} - A)^2 \sum 1 + 2(1+ar_{13})\,(cr_{13} - A) \sum X + 2(r_{12}+br_{13})\,(cr_{13} - A) \sum Y \qquad (8.8)$$

Using the moments for the summation of X, Y, X^2, XY and Y^2 and rearranging

we get,

$$\mu'_{20} - m_{20} + 2\,A\,m_{10} - A^2 m_{00} = [\,2\,a\,r_{13} + a^2\,r_{13}^2]\,m_{20} +$$
$$[2\{\,r_{12} + b\,r_{13} + a\,r_{12}\,r_{13} + ab\,r_{13}^2\}]\,m_{11} +$$
$$[r_{12}^2 + 2\,b\,r_{12}\,r_{13} + b^2\,r_{13}^2\,]\,m_{02} +$$
$$[2\{c\,r_{13} - A\,a\,r_{13} + a\,c\,r_{13}^2\}]\,m_{10} +$$
$$[2\{-A\,r_{12} - A\,b\,r_{13} + c\,r_{12}\,r_{13} + b\,c\,r_{13}^2\}]\,m_{01} +$$
$$[\,-2\,A\,c\,r_{13} + c^2\,r_{13}^2]\,m_{00}. \qquad (8.9)$$

Rearranging equation (8.9) and bringing all terms on Right Hand Side we define,

$$F_1 = r_{13}^2(a^2 m_{20} + 2abm_{11} + b^2 m_{02} + 2acm_{10} + 2bcm_{01} + c^2 m_{00}) + 2r_{12}r_{13}(am_{11} + bm_{02} + cm_{01})$$
$$2\,r_{13}\Big((am_{20} + bm_{11} + cm_{10}) - A(am_{10} + bm_{01} + cm_{00})\Big) + 2r_{12}m_{11} + r_{12}^2 m_{02} - 2Ar_{12}m_{01}$$
$$-\{\mu'_{20} - m_{20} + 2Am_{10} - A^2 m_{00}\} \qquad (8.9a)$$

Writing μ'_{11} in the same way we get,

$$\mu'_{11} = \sum\Big((1 + a\,r_{13})\,X + (\,r_{12} + b\,r_{13})\,Y + c\,r_{13} - A\,\Big)\Big((-r_{12} + a\,r_{23})\,X +$$
$$(\,1 + br_{23})\,Y + c\,r_{23} - B\,\Big). \qquad (8.10)$$

where, $\overline{Y'}$ is replaced with B

$$\mu'_{11} = (1 + a\,r_{13})\,(-r_{12} + ar_{23})\sum X^2 + [\,(1 + a\,r_{13})\,(\,1 + b\,r_{23}) +$$
$$(\,r_{12} + b\,r_{13})\,(-r_{12} + a\,r_{23})]\sum XY + (r_{12} + b\,r_{13})\,(1 + b\,r_{23})\sum Y^2 +$$
$$(cr_{13} - A)\,(\,c\,r_{23} - B)\sum 1 + [(1 + ar_{13})\,(cr_{23} - B) + (\,c\,r_{13} - A)$$
$$(\,-r_{12} + a\,r_{23})\,]\sum X + [(r_{12} + br_{13})\,(cr_{23} - B) + (\,1 + b\,r_{23})$$
$$(\,cr_{13} - A\,)]\sum Y$$

(8.11)

Substituting as above and rearranging we get,

$$\mu'_{11} - m_{11} + B\,m_{10} + A\,m_{01} - A\,B\,m_{00} = [-r_{12} + ar_{23} - ar_{12}r_{13} + a^2 r_{13}r_{23}]\ m_{20} +$$
$$[-r_{12}^2 + a\,r_{13} + b\,r_{23} + a\,r_{12}\,r_{23} - b\,r_{12}\,r_{13} + 2ab\,r_{13}\,r_{23}]\ m_{11} + [r_{12} + b\,r_{13} +$$
$$br_{12}r_{23} + b^2 r_{13}r_{23}]m_{02} + [Ar_{12} - Bar_{13} - Aar_{23} + cr_{23} + 2acr_{13}r_{23} -$$
$$cr_{12}r_{13}]\ m_{10} + [-B\,r_{12} - B\,b\,r_{13} + cr_{13} - Abr_{23} + cr_{12}r_{23} + 2bcr_{13}r_{23}]m_{01}$$
$$+ [\,-B\,c\,r_{13} - A\,c\,r_{23} + c^2\,r_{13}\,r_{23}]\ m_{00} \qquad (8.12).$$

Defining in a similar way as in (8.9a) for the above equation, we write,

$$F_2 = r_{12}(-m_{20} + m_{02} + Am_{10} - Bm_{01}) + r_{23}\big(am_{20} + bm_{11} + cm_{10} - A(am_{10} + bm_{01} + cm_{00})\big) +$$
$$r_{13}\big(am_{11} + bm_{02} + cm_{01} - B(am_{10} + bm_{01} + cm_{00})\big) - r_{12}r_{13}(am_{20} + bm_{11} + cm_{10}) +$$
$$r_{13}r_{23}(a^2 m_{20} + 2abm_{11} + b^2 m_{02} + 2acm_{10} + 2bcm_{01} + c^2 m_{00}) + r_{12}r_{23}(am_{11} + bm_{02} + cm_{01})$$
$$- r_{12}^2 m_{11} - \{\mu'_{11} - m_{11} + Bm_{10} + Am_{01} - ABm_{00}\} \qquad (8.12a)$$

Writing μ'_{02} on the same lines as the above,

$$\mu'_{02} = \Big(\sum (-r_{12} + a\,r_{23})\,X + (\,1 + b\,r_{23})\,Y + c\,r_{23} - B\,\Big)^2 \qquad (8.13)$$

$$\mu'_{02} = (-r_{12} + a\,r_{23})^2 \sum X^2 + 2\,(-r_{12} + a\,r_{23})\,(\,1 + b\,r_{23}) \sum XY + (1 + b\,r_{23})^2$$
$$\sum Y^2 + (cr_{23} - B)^2 \sum 1 + 2(-r_{12} + ar_{23})\,(cr_{23} - B) \sum X +$$
$$2(1 + br_{23})\,(\,cr_{23} - B\,) \sum Y$$

(8.14).

Substituting as in previous equation and rearranging we get,

$$\mu'_{02} - m_{02} + 2\,B\,m_{01} - B^2 m_{00} = [r_{12}^2 - 2\,a\,r_{12}\,r_{23} + a^2\,r_{23}^2]\ m_{20} +$$
$$[2\{-r_{12} + a\,r_{23} - b\,r_{12}\,r_{23} + ab\,r_{23}^2\}]\ m_{11} + [2\,b\,r_{23} + b^2\,r_{23}^2\,]\ m_{02} +$$
$$[2\{-c\,r_{12}\,r_{23} + B\,r_{12} - B\,a\,r_{23} + a\,c\,r_{23}^2\}]\ m_{10} + [2\{cr_{23} - Bbr_{23} + bcr_{23}^2\}]\ m_{01} +$$
$$[\,-2\,B\,c\,r_{23} + c^2\,r_{23}^2]\ m_{00}. \qquad (8.15)$$

Similarly we define,

$$F_3=r_{12}^2m_{20}-2r_{12}r_{23}(am_{20}+bm_{11}+cm_{10})+r_{23}^2(a^2m_{20}+2abm_{11}+b^2m_{02}+2acm_{10}+2bcm_{01}+c^2m_{00})+2r_{23}\Big((am_{11}+bm_{20}+cm_{01})-B(am_{10}+bm_{01}+cm_{00})\Big)+2r_{12}(Bm_{10}-m_{11})-\{\mu'_{02}-m_{02}+2Bm_{01}-B^2m_{00}\} \quad (8.15a)$$

In equations (8.9, 8.12, and 8.15) the unknowns are r_{12}, r_{13} , r_{23} , a, b and c. Because of orthographic projection the absolute depth information is lost; hence we can find only the relative depths, a/c and b/c values: or equivalently 'c' can be assumed unity as it is a global scaling factor [113] . We can also observe that (8.9, 8.12 and 8.15) are non-linear equations in the unknown variables mentioned above.

Let p = a/c ; q =b/c

$$R_3= c\ r_{13}\ ;\ R_{12} = r_{12} \text{ and } R_6 = c\ r_{23} \quad (8.16)$$

The above modified variables are substituted in equations (8.9, 8.12 and 8.15). For the resultant equations only the Right Hand Side of the equations are written below. It is noted from equation (8.16) that the rotation parameters r_{13} and r_{23} are magnified by the factor 'c'. If 'c' is unity then the estimated values of R_3 and R_6 will straightaway give the values of r_{13} and r_{23}. On the other hand if 'c' is different from unity the values obtained are to be scaled. However, this requires knowledge of the value of 'c'.

RHS of equation (8.9)

$$(2pR_3+p^2R_3^2)m_{20}+m_{11}[2(R_{12}+qR_3+pR_{12}R_3+pqR_3^2)]+[R_{12}^2+2qR_{12}R_3+q^2R_3^2]m_{02}+[2(R_3-ApR_3+pR_3^2)]m_{10}+[2(-AR_{12}-AqR_3+R_{12}R_3+qR_3^2)]m_{01}+[-2AR_3+R_3^2]\ m_{00} \quad (8.17)$$

RHS of equation (8.12)

$$(-R_{12}+ p\ R_6 - p\ R_{12}R_3 + p^2R_3R_6)\ m_{20} + (\ p\ R_3+qR_6 + pqR_3R_6-R_{12}^2$$
$$+ p\ R_{12}R_6- qR_{12}R_3+pqR_3R_6)m_{11}+(R_{12}+qR_3+qR_{12}R_6+q^2R_3R_6)m_{20}$$
$$+(AR_{12}-BpR_3+R_6-ApR_6-R_{12}R_3+2pR_3R_6)m_{10} + (\ -BR_{12}-BqR_3$$
$$+R_3-AqR_6+\ R_{12}R_6+2qR_3R_6)m_{01}+(-BR_3-AR_6\ +R_3R_6)m_{00} \qquad (8.18)$$

RHS of equation (8.15)

$$(\ R_{12}^2-2pR_{12}R_6+p^2R_6^2)m_{20} +[\ 2(\ -R_{12}+pR_6-qR_{12}R_6+pqR_6^2)]\ m_{11}$$
$$+(2qR_6+q^2R_6^2)m_{02}+2\ (\ BR_{12}-BpR_6-R_{12}R_6+pR_6^2)m_{10}+\ 2\ (\ R_6-BqR_6$$
$$+qR_6^2)m_{01}+\ (\ -\ 2BR_6\ +\ R_6^2)\ m_{00} \qquad (8.19)$$

From (8.17),(8.18) and (8.19) we can observe that the equations are non-linear and are in terms of the unknown parameters, R_3, R_{12}, R_6, p and q. These equations are rewritten as a sum of linear and non-linear combination of the unknown parameters. That is equation (8.17) can be written as,

$$R_3[\ 2\ m_{10}-2Am_{00}] +R_{12}[\ 2\ m_{11}-\ 2\ Am_{01}] + R_6[0]$$
$$+R_3[2pm_{20}-2Apm_{10}+2qm_{11}-2Aqm_{01}]$$
$$+R_3[\ p^2R_3m_{20}+2pqR_3m_{11}+q^2R_3m_{02}+2pR_3m_{10}+2qR_3m_{01}+\ R_3m_{00}]$$
$$+R_{12}[2pR_3m_{11}+2qR_3m_{02}+R_{12}m_{02}+2\ R_3m_{01}]+\ R_6[0] \qquad (8.20)$$

Similarly equation (8.18) is written as,

$$R_3[\ m_{01}-Bm_{00} +p(\ m_{11}-Bm_{10})+\ q(m_{02}-Bm_{01})]$$
$$+R_{12}[\ -m_{20}+A\ m_{10}+m_{02}-\ Bm_{01}] +$$
$$R_6[m_{10}-A\ m_{00}+p(m_{20}-Am_{10}) +q(\ m_{11}-A\ m_{01})]$$
$$+R_3[p^2R_6m_{20}+pq\ R_6m_{11}]$$
$$+R_{12}[-\ pR_3m_{20}-R_{12}m_{11}+qR_6m_{02}+pR_6m_{11}-R_3m_{10}-qR_3m_{11}]$$
$$+R_6[q^2R_3m_{02}+pqR_3m_{11}+2qR_3m_{01}+R_3m_{00}+2pR_3m_{10}+R_{12}m_{01}] \qquad (8.21).$$

and equation (8.19) can be written as

$$R_3[\,0] + R_{12}[\,-2\,m_{11} + 2\,B\,m_{10}] +$$
$$R_6[2m_{01} - 2Bm_{00} + p(2\,m_{11} - 2Bm_{10}) + q(2\,m_{02} - 2\,B\,m_{01})]$$
$$+R_{12}[R_{12}m_{20} - 2R_6m_{10} - 2\,p\,R_6m_{20} - 2qR_6m_{11}]$$
$$+R_6[p^2R_6m_{20} + 2\,pqR_6m_{11} + q^2R_6m_{02} + 2\,p\,R_6m_{10} + 2q\,R_6m_{01} + R_6m_{00}] \qquad (8.22)$$

From equations (8.20),(8.21) and (8.22) it can be seen that the original equations are non-linear in nature and any solution method based on these equations will be iterative. We also note that the number of unknowns are *five*. and the number of available equations are only *three*. Hence we need some more information . In the following we show that if we take *three frames* continuously we can , to some extent overcome this under-determined model of motion parameter estimation.

Now we assume three frames are available. (See Figure 8.2). The actual motion takes place at time instants t=1,2 and 3. It is possible to consider that an RPP at t=2 is undergoing a backward motion from t=2 to t=1. In this way , we can assume the initial frame is at t=2. Hence, the above formulation and the system of equations for the forward motion , namely from t=2 to t=3 will still hold good. As far as the backward motion is concerned we have to write the equations again. Since the rotation matrix is an orthonormal matrix the inverse of the rotation matrix is nothing but the transpose of the rotation matrix.

8.2.2 Backward motion:

Let p(x,y,z) and $p''(x'',y'',z'')$ be the points at t=2 and t=1 respectively. The space points p and p'' are related by

$$\mathbf{x} = \mathbf{R}\,\mathbf{x}'' \qquad (8.23)$$

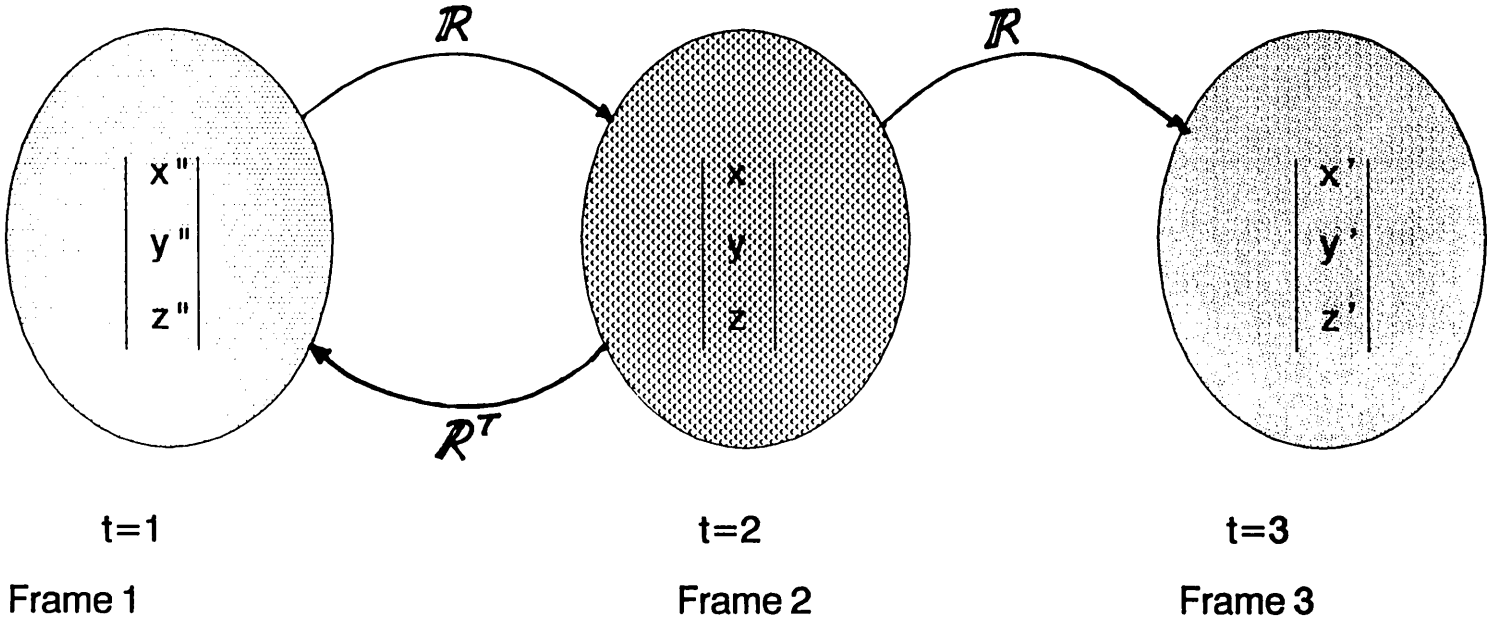


Figure 8.2

Arrangement of image sequences.

therefore,

$$\mathbf{x''} = \mathbf{R}^{-1}\,\mathbf{x} \tag{8.24}$$

As in the forward motion case the rotation angle is assumed small and hence the approximated rotation matrix is used. The same planar equation can be used in the backward motion as well. Substituting the planar equation for the z term in the above equation and projecting the resulting space points orthographically on to the image plane we get,

$$\begin{bmatrix} X'' \\ Y'' \end{bmatrix} = \begin{bmatrix} (1-a\,r_{13}) & (-r_{12}-b\,r_{13}) \\ (r_{12}-a\,r_{23}) & (1-b\,r_{23}) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} -c\,r_{13} \\ -c\,r_{23} \end{bmatrix} \tag{8.25}$$

Following on similar lines as in the forward motion it is possible to get *three* equations in terms of centralised moments and the motion parameters.

Let $D = \frac{m''_{10}}{m''_{00}}$; $E = \frac{m''_{01}}{m''_{00}}$ be the centroids of the RPP at t=1.

The centralised moments of order (p+q) for the RPP at t=1 is given by,

$$\mu''_{pq} = \int (X''-\overline{X''})^p \,.\, (Y''-\overline{Y''})^q \, f_1(X'',Y'')\, dA \tag{8.26}$$

We relate the centralised moments of order 2 to the motion parameters. It may be mentioned that the image characteristic does not change in this instance, that is to say

$$f_1(X'',Y'') = f_2(X, Y)^1$$

$$\mu''_{20} = \sum (X''- D)^2$$

Substituting for X'' we get ,

$$\mu''_{20} = \Big(\sum (1-a\,r_{13})\,X + (-r_{12}-b\,r_{13})\,Y - c\,r_{13} - D\Big)^2$$

$$\mu''_{20} = (1-ar_{13})^2\sum X^2 - 2(1-ar_{13})(r_{12}+b\,r_{13})\sum XY + (-r_{12}-br_{13})^2\sum Y^2 +$$

$$(cr_{13}+D)^2\sum 1 - 2(1-ar_{13})(cr_{13}+D)\sum X + 2(r_{12}+br_{13})(cr_{13}+D)\sum Y \tag{8.27}$$

---

[1] $f_i(.\,,.)$ represents the i^{th} frame intensity function

Using the moments for the summation of X, Y, X^2, XY and Y^2 and rearranging we get,

$$\mu''_{20} - m_{20} + 2\,D\,m_{10} - D^2 m_{00} = [-2\,a\,r_{13} + a^2 r_{13}^2]\,m_{20} -$$
$$[2\{r_{12} + b\,r_{13} - a\,r_{12}\,r_{13} - ab\,r_{13}^2\}]\,m_{11} +$$
$$[r_{12}^2 + 2\,b\,r_{12}\,r_{13} + b^2 r_{13}^2]\,m_{02} -$$
$$[2\{c\,r_{13} - D\,a\,r_{13} - a\,c\,r_{13}^2\}]\,m_{10} +$$
$$[2\{D\,r_{12} + D\,b\,r_{13} + c\,r_{12}\,r_{13} + b\,c\,r_{13}^2\}]\,m_{01} +$$
$$[+2\,D\,c\,r_{13} + c^2 r_{13}^2]\,m_{00}. \qquad (8.28)$$

Writing μ''_{11} in the same way we get,

$$\mu''_{11} = \sum\{(1 - a\,r_{13})\,X - (r_{12} + b\,r_{13})\,Y - (c\,r_{13} + D)\}\{(r_{12} - a\,r_{23})\,X + (1 - b r_{23})\,Y - (c\,r_{23} + E)\}$$

$$\mu''_{11} = (1 - a r_{13})(r_{12} - a r_{23})\sum X^2 + [(1 - a r_{13})(1 - b r_{23}) - (r_{12} + b r_{13})(r_{12} - a r_{23})]\sum XY$$
$$- (r_{12} + b r_{13})(1 - b r_{23})\sum Y^2 + (c r_{13} + D)(c r_{23} + E)\sum 1 - [(1 - a r_{13})(c r_{23} + E) + (c r_{13} + D)$$
$$(r_{12} - a r_{23})]\sum X - [-(r_{12} + b r_{13})(c r_{23} + E) + (1 - b r_{23})(c r_{13} + D)]\sum Y \qquad (8.29)$$

Substituting as above and rearranging we get,

$$\mu''_{11} - m_{11} + E\,m_{10} + D\,m_{01} - D\,E\,m_{00} = [r_{12} - a\,r_{23} - a\,r_{12}\,r_{13} +$$
$$a^2 r_{13}\,r_{23}]\,m_{20} + [-r_{12}^2 - a\,r_{13} - b\,r_{23} + a\,r_{12}\,r_{23}$$
$$- b\,r_{12}\,r_{13} + 2ab\,r_{13}\,r_{23}]\,m_{11} + [-r_{12} - b\,r_{13} + b\,r_{12}\,r_{23}$$
$$+ b^2 r_{13}\,r_{23}]\,m_{02} + [-D\,r_{12} + E\,a r_{13} + D\,a\,r_{23} - c\,r_{23}$$
$$+ 2\,a\,c\,r_{13}\,r_{23} - c\,r_{12}\,r_{13}]\,m_{10} + [E\,r_{12} + E\,b\,r_{13}$$
$$- c\,r_{13} + D\,b_{23} + c\,r_{12}\,r_{23} + 2\,bc\,r_{13}\,r_{23}]\,m_{01}$$
$$+ [E\,c\,r_{13} + D\,c\,r_{23} + c^2 r_{13}\,r_{23}]\,m_{00}$$

(8.30).

Writing μ''_{02} on the same lines as the above,

$$\mu''_{02} = \Big\{ \sum (r_{12} - a\, r_{23})\, X + (1 - b\, r_{23})\, Y - (c\, r_{23} + E) \Big\}^2$$

$$\mu''_{02} = (r_{12} - a r_{23})^2 \sum X^2 + 2(r_{12} - a r_{23})(1 - b r_{23}) \sum XY +$$

$$(1 - b r_{23})^2 \sum Y^2 + (c r_{23} + E)^2 \sum 1 - 2(r_{12} - a r_{23})(c r_{23} + E) \sum X -$$

$$2(1 - b r_{23})(c r_{23} + E) \sum Y \qquad 8.31)$$

Substituting as in equation and rearranging we get,

$$\begin{aligned} \mu''_{02} - m_{02} + 2\, E\, m_{01} - E^2 m_{00} = {} & [r_{12}^2 - 2\, a\, r_{12}\, r_{23} + a^2\, r_{23}^2]\, m_{20} + \\ & [2\{ r_{12} - a\, r_{23} - b\, r_{12}\, r_{23} + ab\, r_{23}^2 \}]\, m_{11} + \\ & [-2\, b\, r_{23} + b^2\, r_{23}^2]\, m_{20} - \\ & [2\{ c\, r_{12}\, r_{23} + E\, r_{12} - E\, a\, r_{23} - a\, c\, r_{23}^2 \}]\, m_{10} - \\ & [2\{ c\, r_{23} - E\, b\, r_{23} - b\, c\, r_{23}^2 \}]\, m_{01} + \\ & [2\, E\, c\, r_{23} + c^2\, r_{23}^2]\, m_{00} \end{aligned} \qquad (8.32)$$

Using equations (8.28),8.30)and (8.32) we can define F'_1, F'_2 and F'_3 similar to F_1, $F_2$and F_3 as in equations (8.9a),(8.12a) and (8.15a) These functions will be used in scheme-2 of the motion parameter estimation algorithm.

We rewrite equations (8.28),(8.30) and (8.32) in terms of p,q, R_3, R_{12} and R_6 , which are defined already in the forward motion formulation and rearranging the terms in such a way that the linear terms in p and q , and the non-linear terms in p and q are separated . Hence from the RHS of (8.28) we get,

$$\begin{aligned} & p[\{ -2\, R_3\, m_{20} + 2\, R_3\, R_{12}\, m_{11} + 2\, R_3^2\, m_{10} + 2\, D\, R_3\, m_{10} \}] \\ & + p^2 R_3^2\, m_{20} + q^2\, R_3^2\, m_{02} + 2\, p\, q\, R_3^2\, m_{11} \\ & + q[-2\, R_3\, m_{11} + 2\, R_3^2\, m_{01} + 2\, D\, R_3\, m_{01} + 2\, R_3\, R_{12}\, m_{02}] \\ & + R_{12}^2\, m_{02} - 2\, R_{12}\, m_{11} - 2\, R_3\, m_{10} + 2\, R_3\, R_{12}\, m_{01} + 2\, D\, R_{12}\, m_{01} \\ & + (R_3^2 + 2\, D\, R_3)\, m_{00} \end{aligned} \qquad (8.33)$$

RHS of equation (8.30),

$$p\,[\,-R_6\,m_{20}-\,R_3\,R_{12}\,m_{20}\,-\,R_3\,m_{11}+\,R_{12}\,R_6\,m_{11}\,+\,D\,R_6\,m_{10}+\,E\,R_3\,m_{10}$$
$$+2\,R_3\,R_6\,m_{10}]\,+\,p^2\,R_3\,R_6\,m_{20}+\,2\,p\,q\,R_3\,R_6\,m_{11}\,+\,q^2\,R_3\,R_6\,m_{02}+\,q\,[\,-\,R_6\,m_{11}\,-$$
$$R_3\,R_{12}\,m_{11}\,+\,R_{12}\,R_6\,m_{02}-R_3\,m_{02}\,+\,D\,R_6\,m_{01}\,+\,E\,R_3m_{01}+\,2\,R_3\,R_6\,m_{01}]\,+\,R_{12}$$
$$m_{20}\,-\,R_{12}^2\,m_{11}\,-\,R_{12}\,m_{02}\,-\,R_6\,m_{10}-D\,R_{12}\,m_{10}-\,R_{12}\,R_3\,m_{10}\,-\,R_3\,m_{01}\,+\,E\,R_{12}$$
$$m_{02}\,+\,R_{12}\,R_6\,m_{01}+(\,E\,R_3\,+\,D\,R_6+\,R_3\,R_6)\,m_{00} \qquad (8.34)$$

RHS of equation (8.32) ,

$$p[\,-\,2\,R_{12}\,R_6\,m_{20}\,-\,2\,R_6\,m_{11}\,+\,2\,R_6^2\,m_{10}\,+\,2\,E\,R_6\,m_{10}]$$
$$+\,p^2\,R_6^2\,m_{20}\,+\,2\,p\,q\,R_6^2\,m_{11}\,+\,q^2\,R_6^2\,m_{02}$$
$$+\,q\,[\,-2\,R_{12}\,R_6\,m_{11}\,-\,2\,R_6\,m_{02}\,+\,2\,R_6^2\,m_{01}+\,2\,E\,R_6\,m_{01}]+$$
$$R_{12}^2\,m_{20}\,+\,2\,R_{12}\,m_{11}\,-\,2\,R_{12}\,R_6\,m_{10}\,-\,2\,E\,R_{12}\,m_{10}\,-\,2\,R_6\,m_{01}$$
$$+(\,R_6^2\,+\,2\,E\,R_6)\,m_{00} \qquad (8.35)$$

From equations (8.33,8.34 and 8.35) we can see that they are non-linear in p and q. Hence it is not possible to have a simple solution.

In the following we propose an *iterative solution method* for solving the above non-linear equations.

8.2.3 Scheme–1

We first separate the non-linear equations into two sets of equations; one involving the terms containing the unknowns R_3, R_{12} and R_6 (rotation parameters) and the second set in terms of the unknowns p and q (planar coefficients). In fact, we separate the forward motion and the backward motion into the above two sets, respectively.

Initially the approximate equations are used in which we use the coefficient of the linear terms in the unknowns (R_3, R_{12} and R_6) alone. These equations are solved assuming some initial values for p and q, known as the

"Guess" solution. This results in a first approximation to the values R_3, R_{12} and R_6. The solutions will be approximate only as we used approximate equations. Subsequently the accurate equations are used. That is, the second order terms in R_3, R_{12} and R_6 whose values can be calculated from the initial solution are added to the approximate equations . This results in improving the solution of R_3 , R_{12} and R_6. The correction is repeated for a required number of iterations or until the solution settles. Thus we get a solution for the unknowns R_3, R_{12} and R_6 for the given initial guess values of p and q.

Similarly, the second set of equations are functions of p and q whose coefficients are functions of R_3, R_{12} and R_6 the values of which have been found above. Thus the equations can be solved for p and q. The second set of equations are three in number and the required unknowns are two; hence, the least square solution is sought. As in the first set of equations, initially the approximate solution for p and q are found by using only the linear coefficient of p and q. Using the initial solution for p and q, the accurate solution can be found by including the second order terms into the coefficients of p and q. The whole process can be iterated for a required number of solutions or until a stable solution is obtained. The algorithm is shown in Figure 8.3.

8.2.4 Theorems for the existence of the planar coefficients:

Before discussing the solutions for the simulations which we have performed , let us analyse some of the issues in the above formulation. We are interested in finding the feasibility of the above method and its validity. We are interested to find under what conditions the above formulation fails. Based on our rigorous analysis we propose the following Theorems.

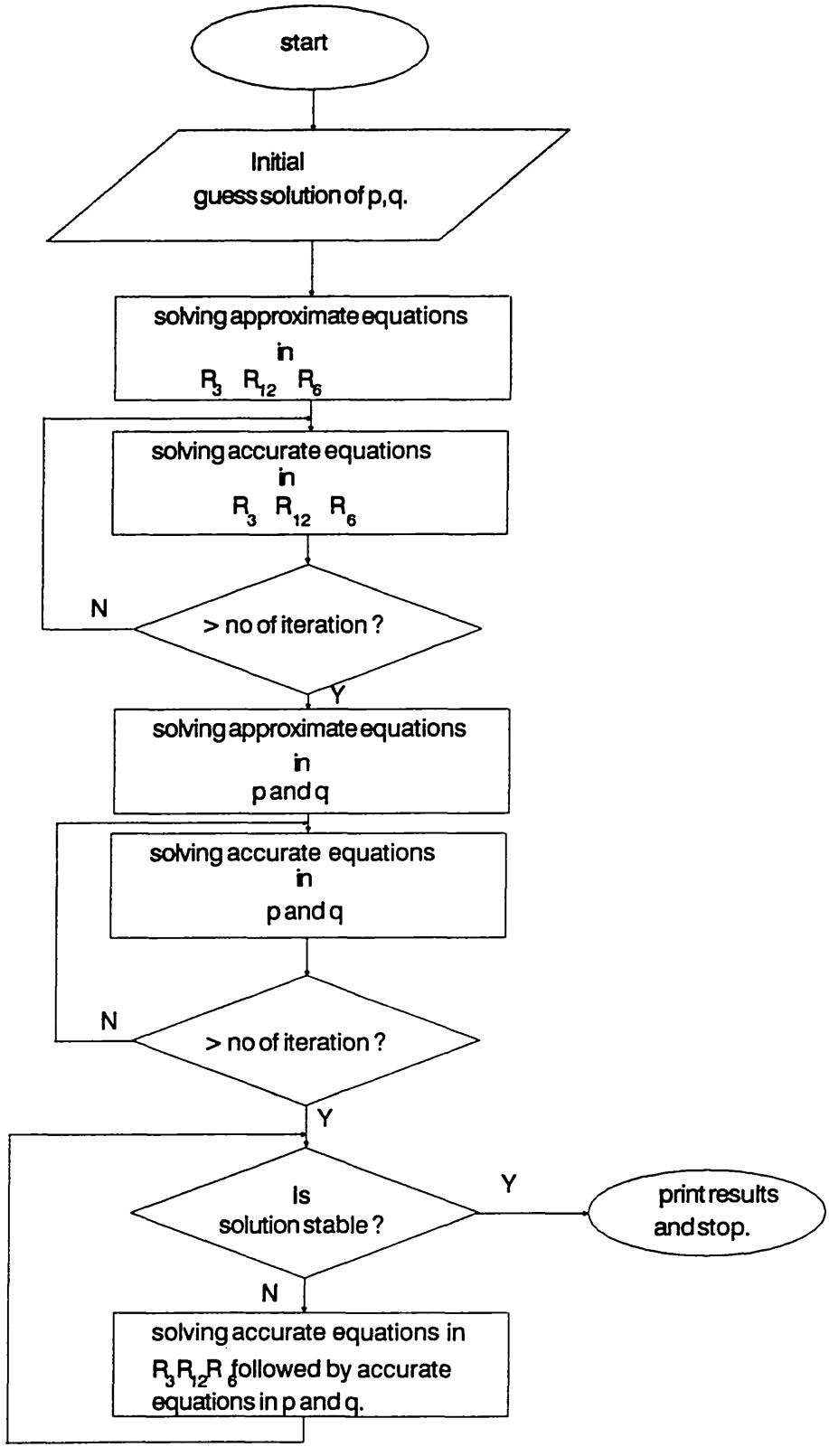


Figure 8.3 Flow chart for scheme 1.

THEOREM 1

If both the direction cosines of the rotation axes in the x and y directions are zero then the planar coefficients are indeterminate.

THEOREM 2

At least one of the direction cosines of the rotation axis in the x and y directions should be non-zero so as to get a real solution for the planar coefficient.

proof: (theorem 1 and theorem 2)

In the backward motion formulation set we have three equations. However,we have to find only two unknowns, p and q. One way is to use a least squares procedure. Even in this case the rank of the coefficient matrix should be 2. In other words the determinant of one of the sub matrices of order 2 should have a non-zero value. We use this fact in proving the theorems 1 and 2.

Using equations (8.33,8.34 and 8.35) with only the linear terms in R_3, R_{12} and R_6, the following B matrix is formed.

$$B=\begin{bmatrix} 2R_3(-m_{20}+R_{12}m_{11}+R_3m_{10}+Dm_{10}) & 2R_3(-m_{11}+R_3m_{01}+R_{12}m_{02}+Dm_{01}) \\ \{R_3(-m_{11}+E\,m_{10})+R_6(-m_{20}+Dm_{10}) & \{R_3(-m_{02}+Em_{01})+R_6(-m_{11}+D\,m_{01}) \\ \quad -R_3R_{12}m_{20}+R_{12}R_6m_{11}+2R_3R_6m_{10}\} & \quad -R_3R_{12}m_{11}+R_{12}R_6m_{02}+2R_3R_6m_{01}\} \\ 2R_6(-R_{12}m_{20}-m_{11}+R_6m_{10}+E\,m_{10}) & 2R_6(-R_{12}m_{11}-m_{02}+R_6m_{01}+Em_{01}) \end{bmatrix}$$

Now, taking the first and third row of B matrix and forming the coefficient matrix we get,

$$B1 = \begin{bmatrix} 2R_3(-m_{20}+R_{12}m_{11}+R_3m_{10}+Dm_{10}) & 2R_3(-m_{11}+R_3m_{01}+R_{12}m_{02}+Dm_{01}) \\ 2R_6(-R_{12}m_{20}-m_{11}+R_6m_{10}+E\,m_{10}) & 2R_6(-R_{12}m_{11}-m_{02}+R_6m_{01}+Em_{01}) \end{bmatrix}$$

Taking the determinant of B1 we write,

$$|B1| = (2R_3)(2R_6)\Big\{\Big(-m_{20}+R_{12}m_{11}+R_3m_{10}+Dm_{10}\Big)\Big(-R_{12}m_{11}-m_{02}+R_6m_{01}+Em_{01}\Big)-\Big(-R_{12}m_{20}-m_{11}+R_6m_{10}+Em_{10}\Big)\Big(-m_{11}+R_3m_{01}+Dm_{01}+R_{12}m_{02}\Big)\Big\}$$

from which we get,

$$|B1| = 4R_3R_6\Big\{R_{12}^2(m_{20}m_{02}-m_{11}^2)+R_6R_{12}(m_{11}m_{01}-m_{10}m_{02})+R_3R_{12}(m_{20}m_{01}-m_{10}m_{11})+R_{12}(Em_{01}m_{11}-Dm_{11}m_{01}+Dm_{20}m_{01}-Em_{10}m_{02})+R_6(-m_{20}m_{01}+m_{11}m_{10})+R_3(m_{02}m_{10}-m_{11}m_{01})+(m_{20}m_{02}-Em_{20}m_{01}-Dm_{02}m_{10}-m_{11}^2+Dm_{11}m_{01}+E\,m_{10}m_{11})\Big\}$$

Case i: The determinant value will be zero if

(a) $R_3 = 0$

or (b) $R_6 = 0$

or (c) both R_3 and R_6 are zero.

Case ii: If $R_3 \neq 0$ and $R_6 \neq 0$ and $R_{12} = 0$ then the determinant value is

$$|B1| = 4R_3R_6\Big\{R_6(-m_{20}m_{01}+m_{11}m_{10})+R_3(m_{02}m_{10}-m_{11}m_{01})+(m_{20}m_{02}-Em_{20}m_{01}-Dm_{02}m_{10}-m_{11}^2+Dm_{11}m_{01}+E\,m_{10}m_{11})\Big\}$$

When all the image plane points are diagonal then $m_{10}=m_{01}$; D=E; and $m_{20}=m_{11}=m_{02}$ then again the determinant value will be zero. Intuitively, we see this condition means that the plane in the 3-D space is perpendicular to the image plane and diagonal to the x,y plane, in which case we will see only a line in the image plane.(See Figure 8.4)

Taking rows 1 and 2 of the B matrix we get,

B2=

$$\begin{bmatrix} 2\,R_3(\,-m_{20}+R_{12}m_{11}+R_3m_{10}+Dm_{10}) & 2R_3(-m_{11}+R_3m_{01}+R_{12}m_{02}+Dm_{01}) \\ \{R_3(-m_{11}+\,E\,m_{10})+\,R_6(-m_{20}+Dm_{10}) & \{R_3(-m_{02}+Em_{01})+R_6(-m_{11}+D\,m_{01}) \\ \quad -R_3R_{12}m_{20}+R_{12}R_6m_{11}+2R_3R_6m_{10}\} & \quad -R_3R_{12}m_{11}+R_{12}R_6m_{02}+2R_3R_6m_{01}\} \end{bmatrix}$$

The determinant of B2 is

$$|B2\,| =(2\,R_3)\,\Big\{\Big(R_3m_{20}+R_{12}m_{11}+Dm_{10}-m_{20}\Big)\Big(R_3\{-m_{20}+Em_{01}\}+R_6\{-m_{11}+Dm_{01}\}$$
$$-R_3R_{12}m_{11}+R_{12}R_6m_{02}+2R_3R_6m_{01}\Big)-\Big(-m_{11}+R_3m_{01}+Dm_{01}R_{12}m_{02}\Big)\Big(R_3\{-m_{11}+$$
$$Em_{10}\}+R_6\{-m_{20}+Dm_{01}\}-R_3R_{12}m_{11}+R_{12}R_6m_{02}+2R_3R_6m_{01}\Big)\Big\}$$

$$|B2| \;=\; 2R_3^2\Big\{R_3\Big(-m_{02}m_{10}+m_{11}m_{01}\Big)+R_6\Big(m_{11}m_{10}+m_{20}m_{01}\Big)+R_3R_{12}\Big(m_{20}m_{01}$$
$$-m_{10}m_{11}\Big)\;+\;R_6R_{12}\Big(m_{11}m_{01}-m_{02}m_{10}\Big)+R_{12}\Big(Em_{11}m_{01}-Dm_{11}m_{10}-E\;m_{02}m_{10}$$
$$+Dm_{20}m_{01}\Big)+R_{12}^2\Big(-m_{11}^2+m_{20}m_{02}\Big)+\Big(m_{20}m_{02}-Dm_{11}m_{02}-E\;m_{20}m_{01}-m_{11}^2+Em_{11}m_{10}$$
$$+Dm_{11}m_{01}\Big)\Big\}$$

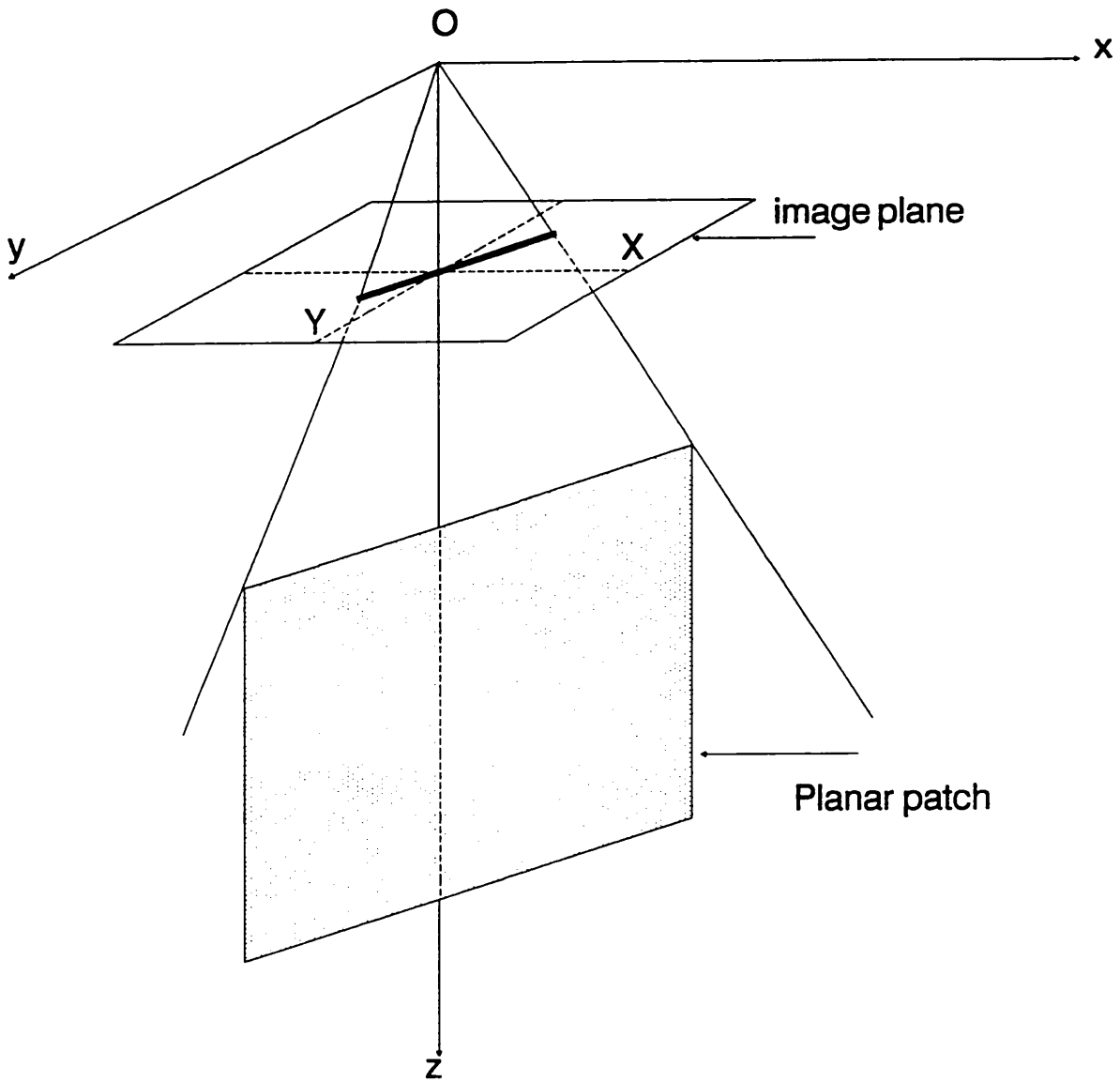


Figure 8.4

Indeterminate configuration of a rigid planar patch.

Case i: The determinant value will be zero if

(a) $R_3 = 0$ (irrespective of R_6 and R_{12})

Case ii: If $R_3 \neq 0$ and $R_6 \neq 0$ and $R_{12}=0$ then the determinant value is

$$|B2| = 2R_3^2\{R_3(-m_{02}m_{10}+m_{11}m_{01})+R_6(m_{11}m_{10}+m_{20}m_{01})$$
$$(m_{20}m_{02}-Dm_{11}m_{02}-Em_{20}m_{01}-m_{11}^2+ Em_{11}m_{10}+Dm_{11}m_{01})\}$$

Again the determinant value will be zero when the object is perpendicular to the image plane and diagonal to the x and y axis of the 3-D space as explained above.

When taking rows 2 and 3 of the B matrix we get,

B3=

$$\begin{bmatrix} \{R_3(-m_{11}+ E\, m_{10})+ R_6(-m_{20}+Dm_{10}) & \{R_3(-m_{02}+Em_{01})+R_6(-m_{11}+D\, m_{01}) \\ -R_3R_{12}m_{20}+R_{12}R_6m_{11}+2R_3R_6m_{10}\} & -R_3R_{12}m_{11}+R_{12}R_6m_{02}+2R_3R_6m_{01}\} \\ \\ 2\, R_6(\, -R_{12}m_{20}-m_{11}+R_6m_{10}+Em_{10}) & 2R_6(-R_{12}m_{11}+R_6m_{01}-m_{02}+Em_{01}) \end{bmatrix}$$

and its determinant is

$$|\, B3\, | = (2\, R_6)\Big\{\{R_3(-m_{11}+ E\, m_{10})+ R_6(-m_{20}+Dm_{10})-R_3R_{12}m_{20}+R_{12}R_6m_{11}$$
$$+2R_3R_6m_{10}\}\{(-R_{12}m_{11}+R_6m_{01}-m_{02}+Em_{01})\}-\{(\, -R_{12}m_{20}-m_{11}$$
$$+R_6m_{10}+Em_{10})\}\{R_3(-m_{02}+Em_{01})+R_6(-m_{11}+D\, m_{01})-R_3R_{12}m_{11}+R_{12}R_6m_{02}$$
$$+2R_3R_6m_{01}\}\Big\}$$

$$|B3| = 2R_6^2\Big\{R_{12}\big(Dm_{20}m_{01}-Dm_{10}m_{11}-Em_{11}m_{01}-Em_{02}m_{10}\big)+R_{12}^2\big(m_{20}m_{02} -m_{11}^2\big)+R_3R_{12}\big(m_{20}m_{01}-m_{11}m_{10}\big)+R_6\big(-m_{20}m_{01}+m_{11}m_{10}\big)+R_3\big(m_{11}m_{01}- m_{10}m_{02}\big)+R_6R_{12}\big(m_{01}m_{11}-m_{02}m_{10}\big)+\big(-Em_{01}m_{20}+m_{20}m_{02}-Dm_{10}m_{02}-m_{11}^2 +Em_{10}m_{11}+Dm_{11}m_{01}\big)\Big\}$$

Case i: When $R_6 = 0$ the determinant of B_3 becomes zero hence the values of p and q cannot be determined. This is irrespective of the values of R_3 and R_{12}.

Case ii: When $R_3 \neq 0$ and $R_6 \neq 0$ and $R_{12}=0$ we get,

$$|B3| = 2R_6^2\Big\{R_6\big(-m_{20}m_{01}+m_{11}m_{10}\big)+R_3\big(m_{11}m_{01}-m_{10}m_{02}\big)+\big(-Em_{01}m_{20} +m_{20}m_{02}-Dm_{10}m_{02}-m_{11}^2+Em_{10}m_{11}+Dm_{11}m_{01}\big)\Big\}$$

The determinant will be zero when the object is perpendicular to the x and y plane and is also diagonal the to the x and y axis as in the previous case.

Q.E.D□

8.2.4 Simulation and results:

Data generation: In our simulation we arbitrarily generated N (N=1000) synthetic 3-D points. These points are then made to undergo a rotational motion in 3 frames for a known rotation angle through a 3-D rotation axis. The generated points are orthographically projected and the image points are taken. From the image points the moment invariants are found for the 3 frames. As they are the same within a limit, the lower order moments are used for finding the motion parameters as explained in the previous sections. These lower order moments are obtained as by- products while finding the MIs.

We conducted a few simulation experiments on the above iterative scheme. Though the iterative scheme is straight forward, we faced a lot of problems while actually performing the experiments. Most of the times, the algorithm converged

to a wrong solution. After a large number of unsuccessful trials, we obtained a few presentable results. They are as shown in Tables T.8.1. and Table T.8.2. For the first experiment we fixed the distance z of the plane to be a constant value of 5 units from the lens center. We have varied the angle of rotation from 1° to 6°. In the second experiment we varied this depth value in the range of 5 units to 40 units while keeping the rotation angle at 5°.

The failure to converge to the correct solution is mainly due to the cross iterative nature of the algorithm. Any error in the approximate solution will be further enhanced in the accurate solution. As we use approximate equations in the first iterative loop we expect an error in the approximate solution. We noted that if the initial guess solutions for p and q are not properly chosen , the algorithm converges to a wrong solution.

The relative error in the parameters varies randomly. In some cases, for example R_6, the relative error decreases as we increase the angle of rotation, only up to a stage and thereafter it increases. However, this is not a common feature for other parameters. In addition the error in the planar coefficients steadily increases as we increase the rotation angle.

When the depth of the plane is increased in the range of 5units, we observe, that generally the error performance improves. However the relative error in case of p and q is extremely large when the depth value is small. This is not completely unexpected as the orthographic projection model is valid for objects that are farther from the camera.

In the above experiments we did not add noise while calculating the moment values. Nevertheless, while assuming an initial guess solution for p and q at least about ±5% deviation is added to the correct solution. Also there will be

TABLE – T.8.1.

Effect of Rotation angle on Motion parameter estimation(Scheme-1)

ROTATION AXIS: (75°, 80°, 18.16°)

PLANE ORIENTATION: (0.02, 0.04, 5.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.0152 0.01847 21.51	−0.0166 −0.01641 −1.145	−0.0226 −0.02554 13.01	0.004 0.003 −25	0.008 0.0068 −15	5.0
2	actual estimated % error	0.0303 0.0269 −11.87	−0.0332 −0.02335 −29.67	−0.0452 −0.0372 −17.699	0.004 0.0028 −29.5	0.008 0.0095 19.5	5.0
3	actual estimated % error	0.0455 0.0701 54.04	−0.0498 −0.03977 −20.14	−0.0678 −0.07536 11.15	0.004 0.00880 120.00	0.008 0.00938 17.25	5.0
4	actual estimated % error	0.0606 0.0750 23.78	−0.06634 −0.06564 −1.055	−0.0903 −0.07823 −13.36	0.004 0.00168 −58	0.008 0.00927 15.875	5.0
5	actual estimated % error	0.0758 0.09292 22.59	−0.08292 −0.04392 -47.03	−0.1129 −0.0236 -79.10	0.004 0.1255 *[1]	0.008 0.08459 *	5.0
6	actual estimated % error	0.0909 0.0588 −35.31	−0.09950 −0.04531 −54.46	−0.1355 −0.00635 −95.31	0.004 0.2618 *	0.008 0.1628 *	5.0

[1] * represents that atleast one of the parameter does not converge to the actual value after 30 iterations. However, it converges to a wrong value.Hence, the relative error is not calculated.

TABLE – T.8.2.

Effect of depth values on Motion parameter estimation (scheme-1)

ROTATION AXIS: (75°, 80°, 18.16°)
PLANE ORIENTATION: (0.02, 0.04)
ANGLE OF ROTATION: 5°

No	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.0758 0.08837 16.583	−0.0829 −0.04295 −48.19	−0.1129 −0.02029 −82.02	0.004 0.12817 *	0.008 0.09557 *	5.0
2	actual estimated % error	0.1515 0.02821 −81.38	−0.0829 −0.02463 −70.289	−0.2259 −0.00447 −98.02	0.002 0.75122 *	0.004 0.7483 *	10.0
3	actual estimated % error	0.2273 0.00146 −99.36	−0.0829 −0.07304 −11.89	−0.3388 −0.01007 −97.0	0.00133 1.6995 *	0.00267 0.32939 *	15.0
4	actual estimated % error	0.3031 0.08119 −73.21	−0.0829 −0.01122 −86.47	−0.4517 −0.03901 −91.36	0.001 0.13151 *	0.002 0.15455 *	20.0
5	actual estimated % error	0.3788 0.40275 6.323	−0.0829 −0.08262 −0.338	−0.5647 −0.54741 −3.06	0.0008 0.00086 7.5	0.0016 0.00168 5.0	25.0
6	actual estimated % error	0.4546 0.45902 0.972	−0.0829 −0.08304 0.1689	−0.6776 −0.63209 −6.72	0.00067 0.00067 0.0	0.00013 0.00103 *	30.0
7	actual estimated % error	0.6061 0.57492 −5.144	−0.0829 −0.08314 0.290	−0.9036 −0.85279 −5.60	0.0005 0.00075 50	0.001 0.00072 −28.0	40.0

noise in the moment values due to discrete calculation. We believe this will give a performance equivalent to that of adding noise in the data generation routine.

The above iterative scheme has the only advantage of requiring a small number of initial values for the parameters. In this case all we need is a guess solution for the plane. The above scheme yields an unpredictable result. The other scheme which is to be presented in section 8.3, however, needs , five parameters to be initially given.

8.3. SCHEME-2: LEVINBERG–MARQUARDT METHOD:

In most of the non-linear minimization problems it is found successful to use the Levinberg-Marquardt algorithm(LeM) [114]. As it is a powerful global minimization procedure we are interested in analysing the performance of the above algorithm using the Levinberg-Marquardt method. Besides, the relative error percentage in the motion parameters is large in the iterative method suggested above.

LeM is an improved Gauss-Newton method. The main advantage is in loosening the condition on the initial guess so that it can converge in a larger region to the global minimum. The basic equation for the LeM is given below,

$$W^{(k+1)} = W^{(k)} - [\,\mu_k I + J(W^{(k)})^{T}.J(W^{(k)})\,]^{-1} J(W^{(k)})^{T}.F(W^{(k)}) \tag{8.36}$$

where

$F = (F_1, F_2, F_3, F_1', F_2', F_3')$ and primed quantities refer to backward

motion. Also J(.) is the Jacobian of the function F which is given by the following matrix

$$J_{ij}= \begin{bmatrix} \frac{\partial F_1}{\partial R_3} & \frac{\partial F_1}{\partial R_{12}} & \frac{\partial F_1}{\partial R_6} & \frac{\partial F_1}{\partial p} & \frac{\partial F_1}{\partial q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial F'_3}{\partial R_3} & \frac{\partial F'_3}{\partial R_{12}} & \frac{\partial F'_3}{\partial R_6} & \frac{\partial F'_3}{\partial p} & \frac{\partial F'_3}{\partial q} \end{bmatrix}$$

$$\mu_k = C \cdot \| F\ (\ W^{(k)}\ ) \|_1^2$$

$$C = 10 \qquad 10 \leq \| F\ (W^{(k)}) \|_1$$

$$= 1 \qquad 1 \leq \| F\ (W^{(k)}) \|_1 < 10$$

$$= 0.01 \qquad \| F\ (W^{(k)}) \|_1 < 1$$

It is observed from our simulation that when the error between the actual values of the equations and the estimated values of the equations is more a larger value of C is taken. The higher the value of C the larger is the correction and hence the system tries to converge to the minimum more quickly.

The actual equations for the partial derivatives of the functions are given in Appendix A–2

8.4 SIMULATION AND RESULTS:

The data generation is as explained in section 8.2.4. The values of moments, moment invariants, and the centralised moments for the three frame sequence are fed to the LeM algorithm. An initial guess solution is also given. The

algorithm estimates the unknown parameters iteratively. In the following we present some of the results.

The algorithm converges after 10 iterations for the first experiment, as tabulated in Table T.8.3. The error is small even for an erroneous guess solution. However, the initial guess solution should be carefully considered, otherwise oscillations will occur and the system will never converge. Sometimes the algorithm converges to a wrong solution.

Figure 8.5 shows one example for the performance of the algorithm. From the graph it is clearly seen that the algorithm converges to a stable solution after about eight iterations. In Table T.8.3 the error performance is given for the data(md2r1).

We have conducted several experiments on the algorithm. In particular we have varied the depth of the plane in 3-D, from 5 units to 50 units. The results for a few depth values are given in Tables T.8.4 – T.8.7. We used two sets of initial guess solutions. As the depth value increases the error in the initial guess will be more. Even for such erroneous guess solutions the system converges, of course with varying number of iterations. When the angle of rotation is small the convergence is faster, however, the error in this case is large. As the angle of rotation is increased the error percentage reduces. (Tables T.8.4–T.8.7). These results are for the noiseless case. Here the number of points in the given regions is assumed to be 1000 pixels. If we assume an image resolution of about 512 x 512, this regional area comes to around 0.381% of the total area of the image. Also when the depth is larger than 30 units, the failure to convergence rate is more. This is mainly due to a large error in the initial guess solution. If the initial guess solution in such cases is modified we get a better performance. We have also tried

TABLE– T.8.3

Convergence of Levinberg-Marquart algorithm (MD2R1.DAT)

INPUT PARAMETERS:

ROTATION AXIS: (75°, 80°, 18.16°)

ANGLE OF ROTATION: 4°

PLANE ORIENTATION : (0.05, 0.02, 10)

ACTUAL VALUES: (0.02950, −0.06064, −0.01807, 0.05, 0.02, 10.0)

INITIAL GUESS SOLUTION: (0.01, 0.01, 0.01, 0.01, 0.01, 1.0)

ITE. NO	R_3	R_{12}	R_6	p	q
1	1.02720E-01	-1.22854E-02	-5.16846E-02	1.19908E-01	-2.77701E-02
2	8.43677E-02	-9.52544E-03	-2.14971E-02	4.56572E-02	5.29685E-03
3	1.43442E-02	-2.23886E-02	-6.58062E-02	-5.75344E-03	3.78039E-02
4	1.93031E-01	-2.90474E-02	-9.20951E-02	1.62114E-02	7.46622E-04
5	3.01157E-01	-4.99028E-02	-1.88935E-01	-4.87202E-04	7.18851E-03
6	3.31279E-01	-6.01392E-02	2.14357E-01	4.95132E-03	1.16393E-03
7	3.15151E-01	-6.07107E-02	-1.98302E-01	4.67814E-03	1.81489E-03
8	3.05232E-01	-6.06572E-02	-1.88160E-01	4.82438E-03	1.92670E-03
9	3.00116E-01	-6.06327E-02	-1.83899E-01	4.91440E-03	1.96647E-03
10	2.97463E-01	-6.06297E-02	-1.82172E-01	4.95985E-03	1.98483E-03
11	2.96041E-01	-6.06295E-02	-1.81283E-01	4.98403E-03	1.99453E-03

Figure 8.5
Performance of Levinberg–Marquardt method.

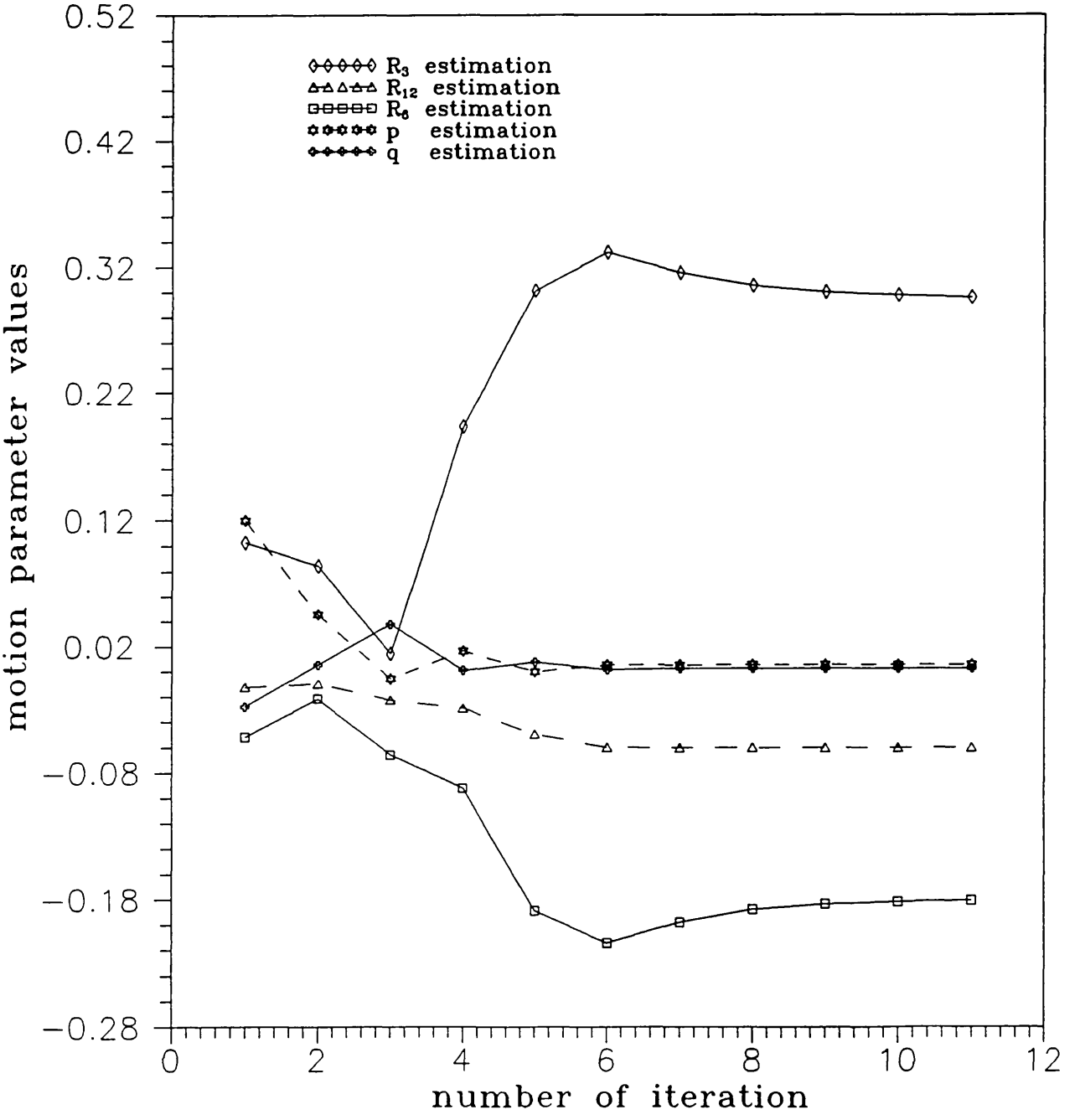


TABLE –T 8.4

Motion parameter estimation (scheme-2): depth = 10 units.

ROTATION AXIS: (75°, 80°, 18.16°)

PLANE ORIENTATION: (0.02, 0.04, 10.0)

INITIAL GUESS: (0.002, −0.02 , −0.003, 0.01, 0.03, 4.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.0303 0.02866 −5.403	−0.01658 −0.01652 −0.362	−0.0452 −0.04199 −7.102	0.002 0.00215 7.5	0.004 0.00433 8.25	10.0
2	actual estimated % error	0.0606 0.06266 3.398	−0.03317 −0.03328 0.332	−0.0903 −0.09373 3.798	0.002 0.00194 −3.0	0.004 0.00384 −4..0	10.0
3	actual estimated % error	0.0909 0.09185 1.0451	−0.04975 −0.04980 0.1005	−0.1355 −0.13931 2.812	0.002 0.00195 −2.5	0.004 0.00388 −3.0	10.0
4	actual estimated % error	0.1212 0.12085 −0.2888	−0.06634 −0.066358 −0.027	−0.1807 −0.17975 −0.526	0.002 0.001981 −0.95	0.004 0.004004 0.1	10.0
5	actual estimated % error	0.1515 0.15212 0.4092	−0.08292 −0.08291 −0.01206	−0.2259 −0.22706 0.514	0.002 0.001991 −0.45	0.004 0.003978 −0.55	10.0
6	actual estimated % error	0.1818 0.18241 0.336	−0.09950 −0.099509 −0.0093	−0.2710 −0.27186 0.317	0.002 0.001995 −0..25	0.004 0.00499 24.75	10.0

TABLE – T.8.5.

Motion parameter estimation(Scheme-2): depth = 25 units.

ROTATION AXIS: (75°, 80°, 18.16°)
PLANE ORIENTATION: (0.02, 0.04, 25.0)
INITIAL GUESS: (0.002, −0.02 , −0.003, 0.01, 0.03, 4.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.07575 0.074860 −1.1749	−0.01658 −0.01653 −0.3016	−0.113 −0.1097 −2.920	0.0008 0.000824 3.0	0.0016 0.001657 3.5625	25.0
2	actual estimated % error	0.1515 0.15578 2.8251	−0.03317 −0.03327 0.3015	−0.22575 −0.23326 3.3267	0.0008 0.00078 −2.5	0.0016 0.001552 −3.0	25.0
3	actual estimated % error	0.22725 0.22547 −0.783	−0.04975 −0.04971 −0.0804	−0.33875 −0.330817 −2.342	0.0008 0.000818 2.25	0.0016 0.00164 2.5	25.0
4	actual estimated % error	0.3030 0.30229 −0.0033	−0.06634 −0.066356 0.0241	−0.45175 −0.450716 −0.229	0.0008 0.000791 −1.125	0.0016 0.001598 −0.125	25.0
5	actual estimated % error	0.37875 0.37787 −0.2323	−0.08292 −0.082917 −0.00362	−0.56475 −0.563697 −0.186	0.0008 0.000801 0.125	0.0016 0.001602 0.125	25.0
6	actual estimated % error	0.4545 0.45541 0.2002	−0.09950 −0.099509 0.00904	−0.6775 −0.678615 0.1646	0.0008 0.000799 −0.125	0.0016 0.001602 0.125	25.0

TABLE – T.8.6

Motion parameter estimation(scheme-2): depth = 30 units.

ROTATION AXIS: (75°, 80°, 18.16°)
PLANE ORIENTATION: (0.02, 0.04, 30.0)
INITIAL GUESS: (0.002, −0.02 , −0.003, 0.01, 0.03, 15.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.0909 0.091293 0.4323	−0.01658 −0.016525 −0.3317	−0.1356 −0.132167 −2.5317	0.00067 0.0006784 2.0895	0.00133 0.001376 3.4586	30.0
2	actual estimated % error	0.1818 0.181557 -0.1337	−0.03317 −0.033278 0.3255	−0.2709 −0.27141 0.1883	0.00067 0.000673 0.4478	0.00133 0.001334 0.3008	30.0
3	actual estimated % error	0.2727 0.27168 −0.3740	−0.04975 −0.049705 −0.0905	−0.4065 −0.39826 −2.0271	0.00067 0.0006798 1.4627	0.00133 0.001328 2.5564	30.0
4	actual estimated % error	0.3636 0.36454 −0.2580	−0.06634 −0.066358 0.027	−0.5421 −0.54220 0.0185	0.00067 0.000657 −1.9403	0.00133 0.001328 −0.1504	30.0
5	actual estimated % error	0.4545 0.01290 *	−0.08292 −0.037758 *	−0.6777 −0.24407 *	0.00067 0.03249 *	0.00133 0.002204 *	30.0
6	actual estimated % error	0.5454 0.54394 −0.2677	−0.09950 −0.099509 0.00904	−0.8130 −0.81066 −0.2878	0.00067 0.0006691 −0.1343	0.00133 0.001338 0.6015	30.0

TABLE –T.8.7

Motion parameter estimation(Scheme-2): depth = 40 units.

ROTATION AXIS: (75°, 80°, 18.16°)
PLANE ORIENTATION: (0.02, 0.04, 40.0)
INITIAL GUESS: (0.002, −0.02 , −0.003, 0.01, 0.03, 15.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.1212 0.120817 −0.2723	−0.01658 −0.016532 −0.2895	−0.1808 −0.17747 −1.842	0.0005 0.0005096 1.92	0.0010 0.001024 2.40	40.0
2	actual estimated % error	0.2424 0.23993 −1.019	−0.03317 −0.03327 0.30148	−0.3612 −0.35918 −0.5592	0.0005 0.000508 1.6	0.0010 0.001009 0.9	40.0
3	actual estimated % error	0.3636 −0.0553 *	−0.04975 −0.0027 *	−0.5420 −0.1570 *	0.0005 −0.0377 *	0.0010 0.01559 *	40.0
4	actual estimated % error	0.4848 0.48562 0.169	−0.06634 −0.066357 0.0256	−0.7228 −0.72215 −0.0899	0.0005 0.000494 −1.2	0.0010 0.000997 −0.3	40.0
5	actual estimated % error	0.6060 −0.00623 *	−0.08292 −0.01064 *	−0.9036 −0.26497 *	0.0005 0.04498 *	0.0010 0.00104 *	40.0
6	actual estimated % error	0.7272 −.00623 *	−0.09950 −0.02640 *	−0.9036 −0.26497 *	0.0005 0.04998 *	0.0010 −.0382 *	40.0

normalising the 3-D data when the depth values are more than 30 units. In general normalising improves the convergence to a correct solution. However, in such circumstances the error is also slightly more.(See Tables T.8.8 and T.8.9).

The performance of R_3, R_{12}, R_6, p and q for varying the depth values from 5 units to 30 units are given in Figures 8.6–8.10. From the graphs it is clear that the relative error percent improves as the angle of rotation is increased. Also when the object is close to the camera the error is seen to be more. This can be expected, as the formulation is valid for orthographic projection. As the depth of the RPP increases, the orthographic assumption is more closely approximated.

From Figure 8.7 it can be seen that the relative performance in the case of R_{12} is much less compared to the relative error performance of R_3 and R_6. This is due to the independent nature of R_{12} with 'c', the depth of the plane.

We have also conducted several simulation experiments to study the effect of varying the orientation of the plane while keeping the rotation axis fixed. In all cases (See Tables T.8.10 and T.8.11) we see an improved performance when the orientation of the plane is increased. We have included the noise effect also in these simulations. The noise is kept at 0.5%. As can be expected, the relative errors in noisy cases are larger compared to those without noise.

8.5 PURE TRANSLATION ESTIMATION USING MOMENTS:

In previous sections we discussed a method that utilised central moments to estimate the rotation parameters and the orientation of the rigid planar patch. In the following we see how the moments are used to estimate the translational velocities. Again we consider only the orthographic projection case. Due to the

TABLE – T.8.8.

Effect of normalising the data:depth=30 units.

ROTATION AXIS: (75°, 80°, 18.16°)
PLANE ORIENTATION: (0.02, 0.04, 30.0)
INITIAL GUESS: (0.002, −0.02 , −0.003, 0.01, 0.03, 20.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated error	0.00303 0.002932 −3.2343	−0.01658 −0.016331 −1.5018	−0.00452 −0.004204 −6.9912	0.0200 0.0193 −3.5	0.0400 0.1423 5.75	1.0
2	actual estimated % error	0.00606 −0.00602 −0.6271	−0.03317 −0.03308 −0.2713	−0.00903 −0.009012 −0.199	0.0200 −0.020167 0.835	0.0400 0.040211 0.5275	1.0
3	actual estimated % error	0.00909 0.008815 −3.0253	−0.04975 −0.049576 −0.3497	−0.01355 −0.013235 −2.3247	0.0200 0.02086 4.3	0.0400 0.041033 2.5825	1.0
4	actual estimated % error	0.01212 0.012217 0.7987	−0.06634 −0.066291 −0.0739	−0.01807 −0.018192 0.6751	0.0200 0.019798 −1.01	0.0400 0.03968 −0.8	1.0
5	actual estimated % error	0.01515 0.014921 −1.5102	−0.08292 −0.08286 −0.0759	−0.02259 −0.022402 −0.8322	0.0200 0.020279 1.395	0.0400 0.040421 1.0525	1.0
6	actual estimated % error	0.01818 −0.00953 *	−0.09950 −0.05617 *	−0.02710 −0.0029 *	0.0200 0.4029 *	0.0400 −0.947 *	1.0

TABLE –T.8.9.

Effect of normalising the data: depth=40 units.

ROTATION AXIS: (75°, 80°, 18.16°)
PLANE ORIENTATION: (0.02, 0.04, 40.0)
INITIAL GUESS: (0.002, −0.02 , −0.003, 0.01, 0.03, 20.0)

θ°	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.00303 0.00300 −0.944	−0.01658 −0.01653 −0.3197	−0.00452 −0.00439 −2.92	0.02 0.02001 0.05	0.04 0.040928 2.32	1.0
2	actual estimated % error	0.00606 0.0060586 −0.0231	−0.03317 −0.03318 0.0338	−0.00903 −0.00889 −1.5947	0.0200 0.020241 1.205	0.0400 0.04055 1.375	1.0
3	actual estimated % error	0.00909 0.009123 0.363	−0.04975 −0.04979 0.0884	−0.01355 −0.01338 0.2148	0.0200 0.020123 0.6125	0.0400 0.040496 1.24	1.0
4	actual estimated % error	0.01212 0.012333 1.757	−0.06634 −0.06624 −0.1538	−0.01807 −0.01814 0.3874	0.0200 0.01974 −1.3	0.0400 0.03969 −0.775	1.0
5	actual estimated % error	0.01515 0.014923 −1.4983	−0.08292 −0.08283 −0.1085	−0.02259 −0.02240 −0.828	0.0200 0.020342 −1.71	0.0400 0.0404500 1.125	1.0
6	actual estimated % error	0.01818 0.01813 −0.2745	−0.09950 −0.09955 0.0546	−0.02710 −0.02972 0.2140	0.0200 0.02003 0.150	0.0400 0.039956 −0.11	1.0

TABLE –T.8.10.

Effect of varying the planar orientation (no noise) .

ROTATION AXIS: (70°, 80°,)
ANGLE OF ROTATION 5°
INITIAL GUESS: (0.014, −0.08 , −0.28, 0.02, 0.03, 10.0)

NO	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.01515 0.015734 3.8376	−0.08059 −0.08053 −0.0757	−0.02985 −0.03045 2.0168	0.01 0.009882 −1.18	0.02 0.019629 −1.855	1.0
2	actual estimated % error	0.01515 0.015590 2.9043	−0.08059 −0.08054 −0.0633	−0.02985 −0.03042 1.9112	0.015 0.014781 −1.46	0.025 0.0219 −1.7344	1.0
3	actual estimated % error	0.01515 0.01551 2.3762	−0.08059 −0.08055 −0.0558	−0.02985 −0.03036 1.7219	0.02 0.01972 −1.3845	0.03 0.029524 −1.5867	1.0
4	actual estimated % error	0.01515 0.015475 2.1452	−0.08059 −0.08046 −0.0521	−0.02985 −0.03036 1.6918	0.025 0.02464 1.44	0.035 0.03444 −1.6	1.0
5	actual estimated % error	0.01515 0.015434 1.8739	−0.08059 −0.08055 −0.0707	−0.02985 −0.03033 1.6044	0.03 0.02957 −1.4333	0.04 0.03941 −1.48	1.0

TABLE –T.8.11.

Effect of varying the planar orientation (with noise) .

ROTATION AXIS: (70°, 80°,)
ANGLE OF ROTATION 5°
INITIAL GUESS: (0.014, −0.08 , −0.28, 0.02, 0.03, 10.0)
Noise included: 0.5%

NO	TYPE	R_3	R_{12}	R_6	p	q	Z
1	actual estimated % error	0.01515 0.004239 −72.021	−0.08059 −0.05974 −25.8717	−0.02985 −0.02284 −23.484	0.01 0.006799 −32.01	0.02 0.02919 45.95	1.0
2	actual estimated % error	0.01515 0.004502 −70.2838	−0.08059 −0.05984 −25.7476	−0.02985 −0.02276 −23.7521	0.015 0.013122 −12.52	0.025 0.035901 43.604	1.0
3	actual estimated % error	0.01515 0.00487 −67.8482	−0.08059 −0.05997 −25.5863	−0.02985 −0.022652 −24.1139	0.02 0.019868 −0.66	0.03 0.042728 42.4267	1.0
4	actual estimated % error	0.01515 0.005283 −65.1287	−0.08059 −0.061123 −24.1556	−0.02985 −0.022497 −24.6332	0.025 0.026989 7.956	0.035 0.04977 42.1857	1.0
5	actual estimated % error	0.01515 0.00569 −62.4422	−0.08059 −0.06027 −25.2140	−0.02985 −0.022302 −25.2864	0.03 0.03439 14.6333	0.04 0.056989 42.4725	1.0

Figure 8.6
Relative error performance of R_3
with depth values.

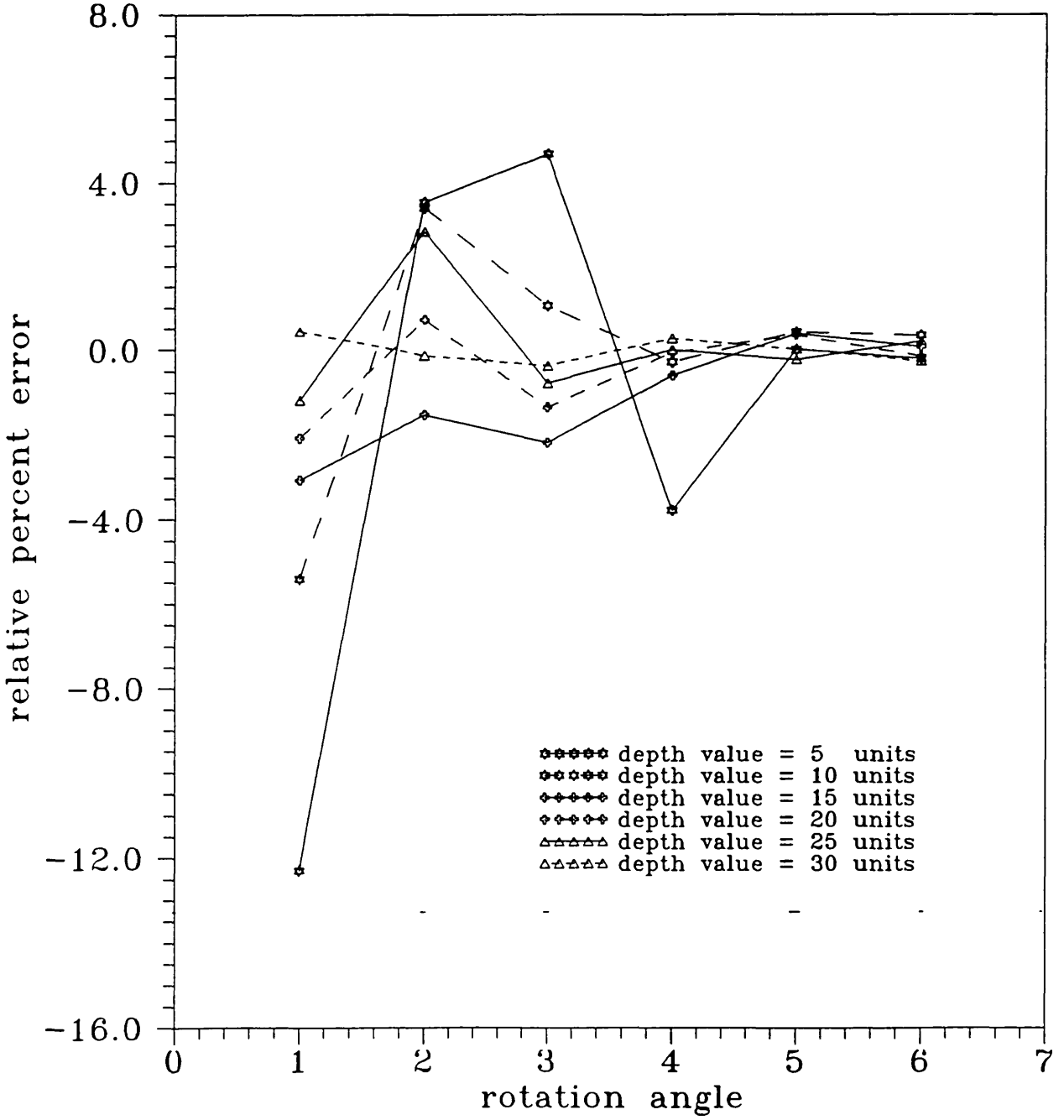


Figure 8.7
Relative error performance of R_{12}
with depth values.

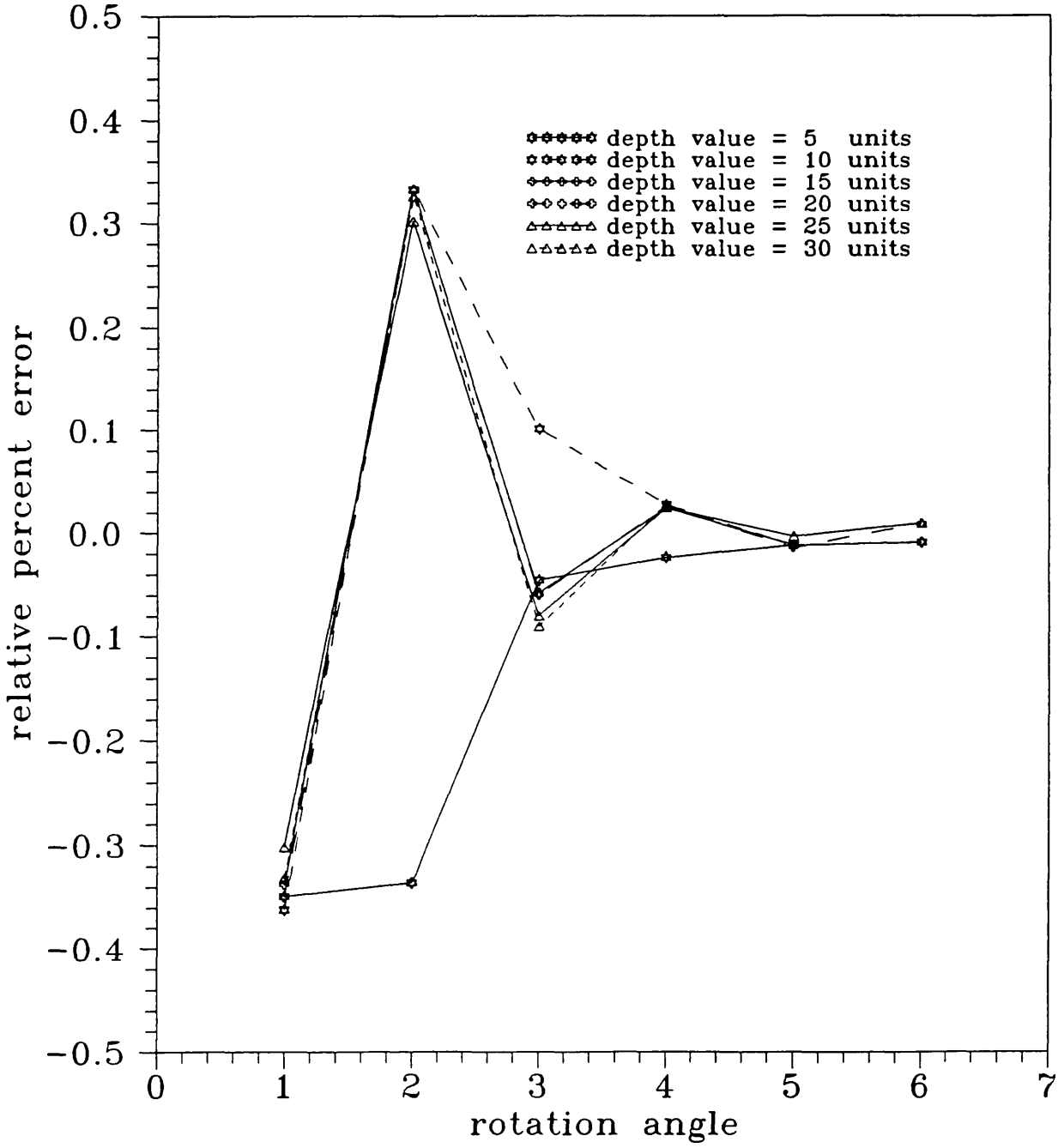


Figure 8.8
Relative error performance of R_6
with depth values.

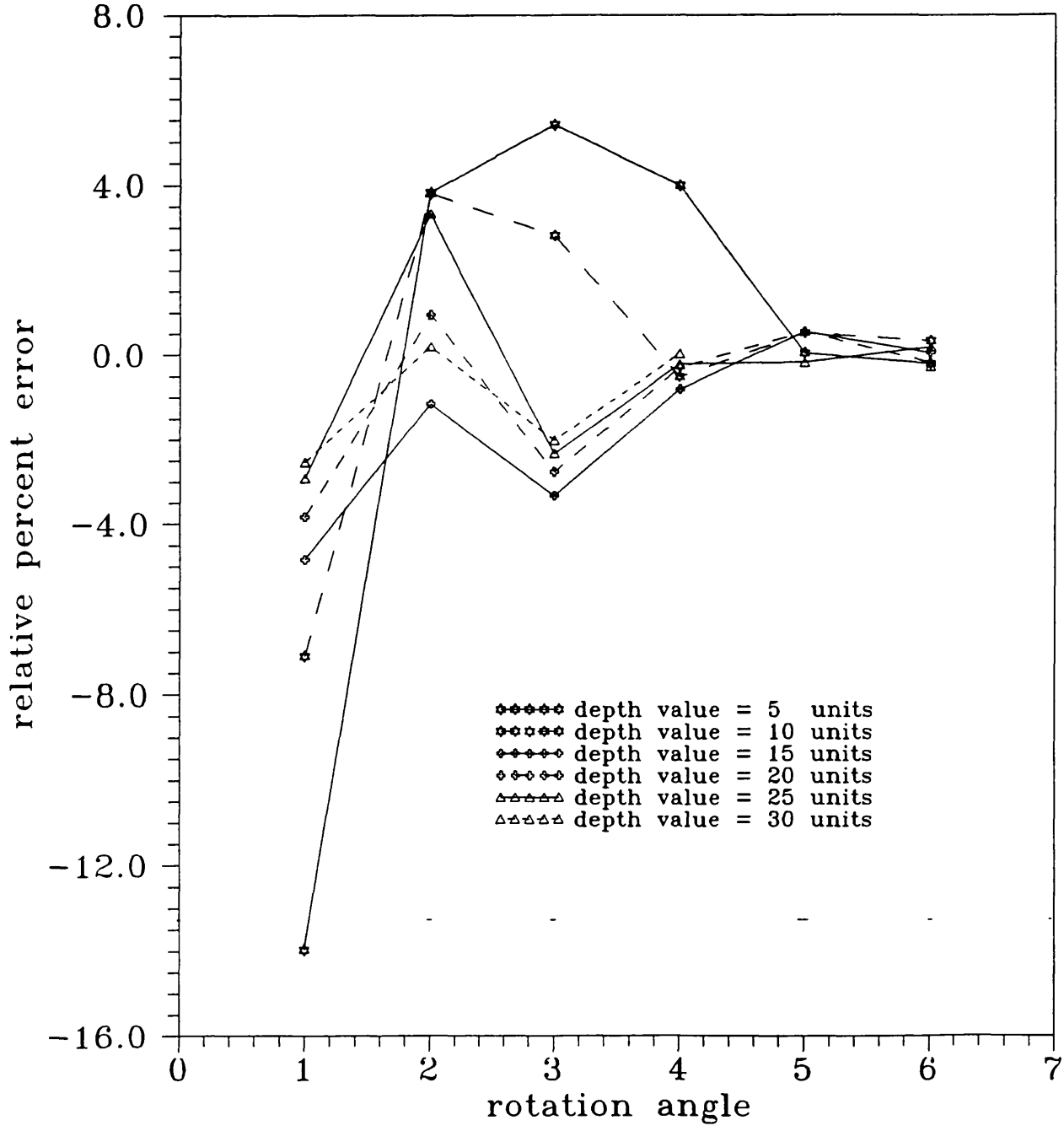


Figure 8.9
Relative error performance of p
with depth values.

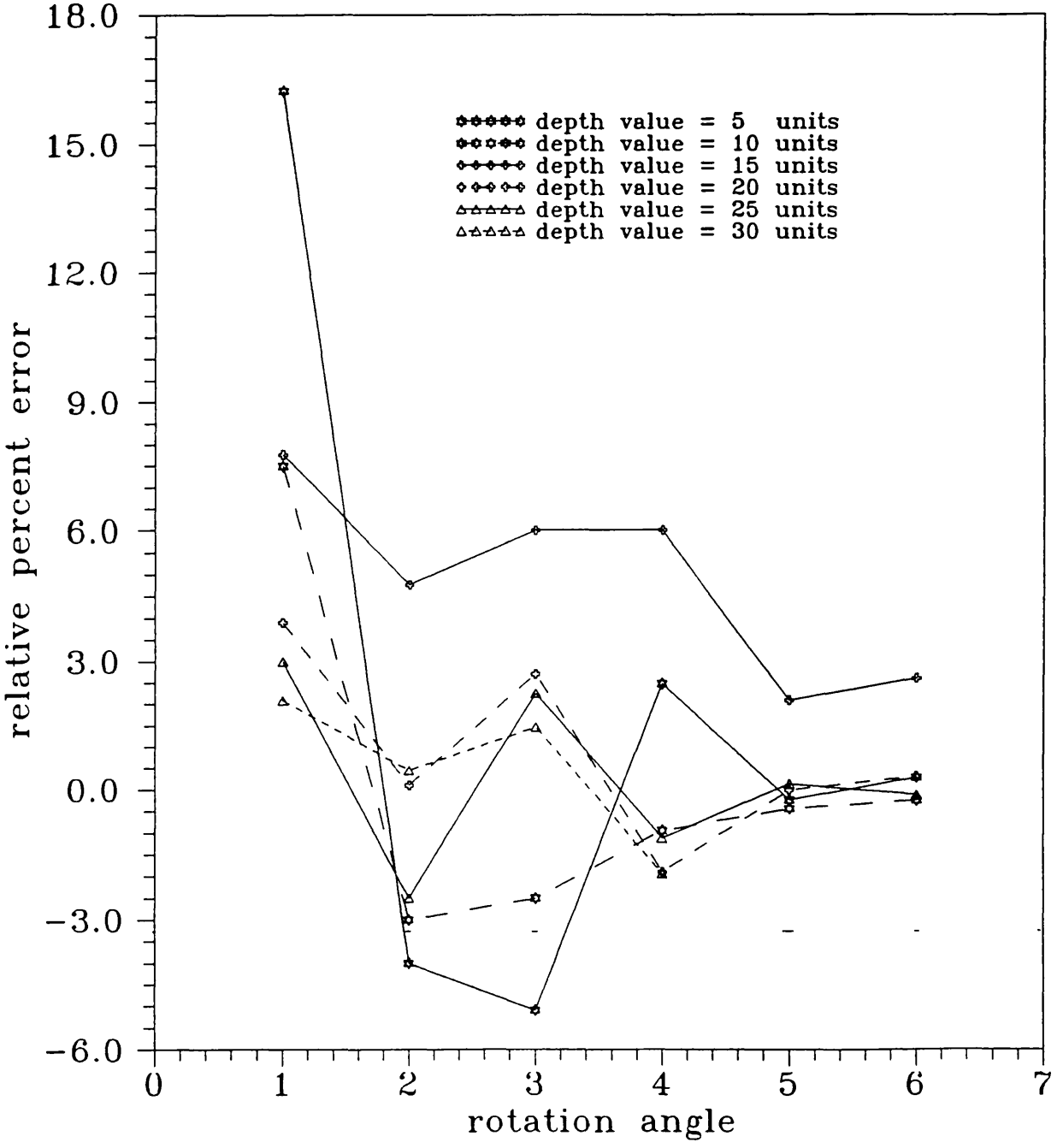
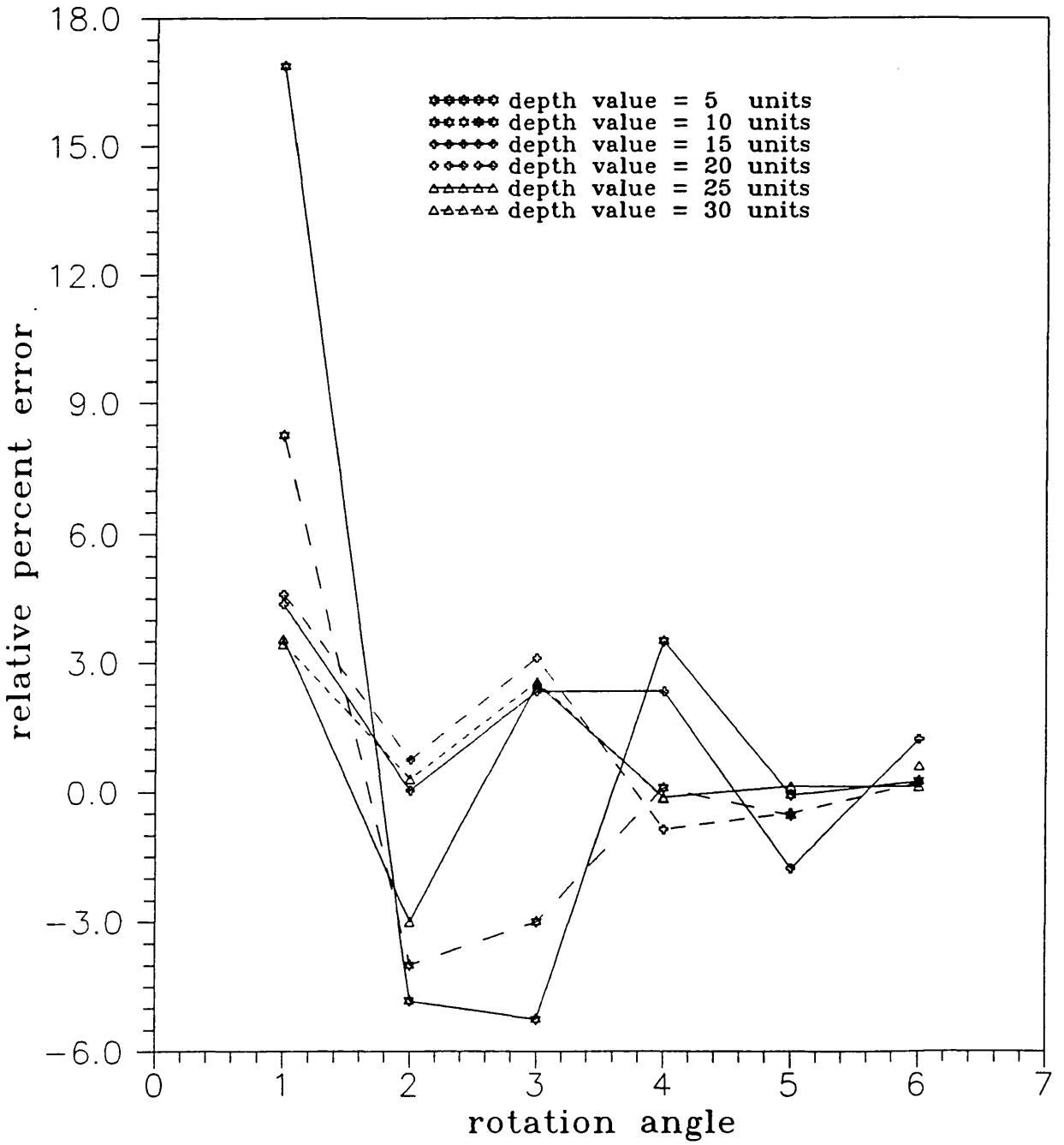


Figure 8.10
Relative error performance of q
with depth values.



nature of the projection , we lose the translation in the z direction and hence we can only determine the translational velocities in the x and y directions. Consider a set of points that undergo a 3-D translation motion. Let p(x,y,z) and $p'(x',y',z')$ be one such point before and after motion. It should be mentioned here that there is no restriction on the 3-D configuration of the point. However, we used points on a planar patch as in the previous case. The motion is represented as

$$x' = x + T \tag{8.37}$$

where,

x and x' are 3-D point vectors before and after motion,

T is the translation vector and is given by $(t_x, t_y, t_z)^T$

For orthographic projection, the Image points before and after motion are given by,

$$X = x \; ; \quad Y = y \tag{8.38a}$$

$$X' = x' ; \quad Y' = y' \tag{8.38b}$$

Now taking the first order moments as given in equation (6.1) for the image point after motion, we get,

$$m'_{pq} = \int (X')^p (Y')^q \, f(X',Y') \, dA \tag{8.39}$$

Writing the above equation as a discrete sum and using unity for $f(X',Y')$ we get,

$$m'_{pq} = \sum X'^p \, Y'^q \tag{8.40}$$

Back projecting the image points in the 3-D space we get,

$$m'_{pq} = \sum x'^p \, y'^q \tag{8.41}$$

Using the translation motion given in (8.37) we get,

$$m'_{pq} = \sum (x - t_x)^p \, (y - t_y)^q \tag{8.42}$$

Substituting equations (8.38 a and 8.38 b) we get,

$$m'_{pq} = \sum (X - t_x)^p \, (Y - t_y)^q \tag{8.43}$$

Taking the first order moments we get,

$$m'_{10} = \sum (X - t_x)^1 = m_{10} + t_x\, m_{00} \tag{8.44a}$$

$$m'_{01} = \sum (Y - t_y)^1 = m_{01} + t_y\, m_{00} \tag{8.44b}$$

Hence ,

$$t_x = \frac{m'_{10}}{m'_{00}} - \frac{m_{10}}{m_{00}} \tag{8.45}$$

that is , $t_x = \overline{X}_{\text{after motion}} - \overline{X}_{\text{before motion}}$

Similarly ,

$$t_y = \frac{m'_{01}}{m'_{00}} - \frac{m_{01}}{m_{00}} \tag{8.46}$$

that is , $t_y = \overline{Y}_{\text{after motion}} - \overline{Y}_{\text{before motion}}$

where ($\overline{X}$, $\overline{Y}$) and ($\overline{X'}$, $\overline{Y'}$) represent the centroids of the patch before and after motion respectively.

Using the above we conducted simulation tests by incorporating noise. The noise added to the data is similar to that discussed in the Pure rotation case. From Table T.8.12 , it is seen that the noise in the pure translation case does not have much effect. This is because of the large number of pixels used for estimating the translation parameter. In fact with two points the translation parameters can be estimated.

TABLE T.8.12.

Motion estimation: pure translation.

Actual translation:
(t_x, t_y, t_z) = (12.0 , 6.0, 2.0)
Number of points: 1000

Noise %	estimated t_x	estimated t_y	error(%) t_x	error(%) t_y
0	12.0000	6.0000	0	0
0.1	12.0004	5.9979	0.0033	-0.036
0.2	11.9998	6.0018	−0.00167	0.03
0.3	11.9992	6.0020	−0.0071	0.033
0.4	11.9996	6.0002	−0.0033	0.0033
0.5	11.9989	5.9981	−0.00917	−0.0317
0.6	11.9972	5.9996	−0.0233	−0.0025
0.7	12.0017	5.9989	0.0142	−0.0192
0.8	12.0038	6.0003	0.0321	0.005
0.9	12.0017	6.0012	0.0135	0.0383
1.0	12.0023	6.0038	0.02375	0.0633
2.0	12.0033	5.9974	0.0542	0.0558
3.0	11.9854	5.9959	−0.1221	0.0683
4.0	11.9906	6.0079	−0.0783	0.1317
5.0	11.9919	6.0066	−0.0675	0.1092

CHAPTER NINE

USING STEREO IMAGE SEQUENCES FOR MOTION ESTIMATION

9.1 INTRODUCTION:

In the previous chapter we have seen a method which used moments of regions in three consecutive image frames for motion estimation. The method can be modified for a stereo based motion estimation . In this chapter an attempt is made to use two stereo image sequences for motion estimation. This stereo based algorithm also uses moments as features for motion estimation. The basic principle involved is in identifying one planar patch in a pair of stereo images and using this patch for motion estimation.

9.2 STEREO BASED FORMULATION :

In the following we present a novel formulation for motion estimation using moments as features. Let two cameras C_1 and C_2 be arranged so that they are on the same base line. They are separated by a small known distance 'd'. The optical axes of the two cameras are arranged so that they are both perpendicular to the same x-y plane. Let p(x, y, z) be a point in the rigid planar patch (RPP) moving in the 3-D space. We consider a pure rotation case only at the moment. The rotation motion is achieved by rotating the object with respect to a rotation axis which is defined by the direction cosines of the three axes n_1, n_2 and n_3 with respect to C_1. $p'(x', y', z')$ is the point after motion . The same point with respect to the second camera is defined as $p_c(x_c, y_c, z_c)$ and $p'_c(x'_c, y'_c, z'_c)$ before and after

motion respectively. The basic geometry of the two cameras is as shown in figure 9.1.

Let us assume that all the points are from the same patch whose equation is defined as z=ax+by+c with respect to C_1. When the object moves, we have two situations : one with respect to camera C_1 and the other with respect to camera C_2. It can be easily seen that the motion of the RPP with respect to C_1 is identical to that of the forward motion formulation discussed in the previous chapter. Equations (8.9, 8.12 and 8.15) relate this motion. The equations are written as below

$$\begin{aligned}\mu'_{20} - m_{20} + 2\,A\,m_{10} - A^2 m_{00} = {} & [\,2\,a\,r_{13} + a^2\,r_{13}^2]\,m_{20} + \\ & [2\{\,r_{12} + b\,r_{13} + a\,r_{12}\,r_{13} + ab\,r_{13}^2\}]\,m_{11} + \\ & [r_{12}^2 + 2\,b\,r_{12}\,r_{13} + b^2\,r_{13}^2\,]\,m_{02} + \\ & [2\{c\,r_{13} - A\,a\,r_{13} + a\,c\,r_{13}^2\}]\,m_{10} + \\ & [2\{-A\,r_{12} - A\,b\,r_{13} + c\,r_{12}\,r_{13} + b\,c\,r_{13}^2\}]\,m_{01} + \\ & [\,-2\,A\,c\,r_{13} + c^2\,r_{13}^2]\,m_{00}. \end{aligned} \tag{9.1}$$

$$\begin{aligned}\mu'_{11} - m_{11} + B\,m_{10} + A\,m_{01} - A\,B\,m_{00} = {} & [-r_{12} + ar_{23} - ar_{12}r_{13} + a^2 r_{13}r_{23}]\,m_{20} + \\ & [-r_{12}^2 + a\,r_{13} + b\,r_{23} + a\,r_{12}\,r_{23} - b\,r_{12}\,r_{13} + 2ab\,r_{13}\,r_{23}]\,m_{11} + [r_{12} + b\,r_{13} + \\ & br_{12}r_{23} + b^2 r_{13}r_{23}]m_{02} + [Ar_{12} - Bar_{13} - Aar_{23} + cr_{23} + 2acr_{13}r_{23} - \\ & cr_{12}r_{13}]\,m_{10} + [-B\,r_{12} - B\,b\,r_{13} + cr_{13} - Abr_{23} + cr_{12}r_{23} + 2bcr_{13}r_{23}]m_{01} \\ & + [\,-B\,c\,r_{13} - A\,c\,r_{23} + c^2\,r_{13}\,r_{23}]\,m_{00} \end{aligned} \tag{9.2}.$$

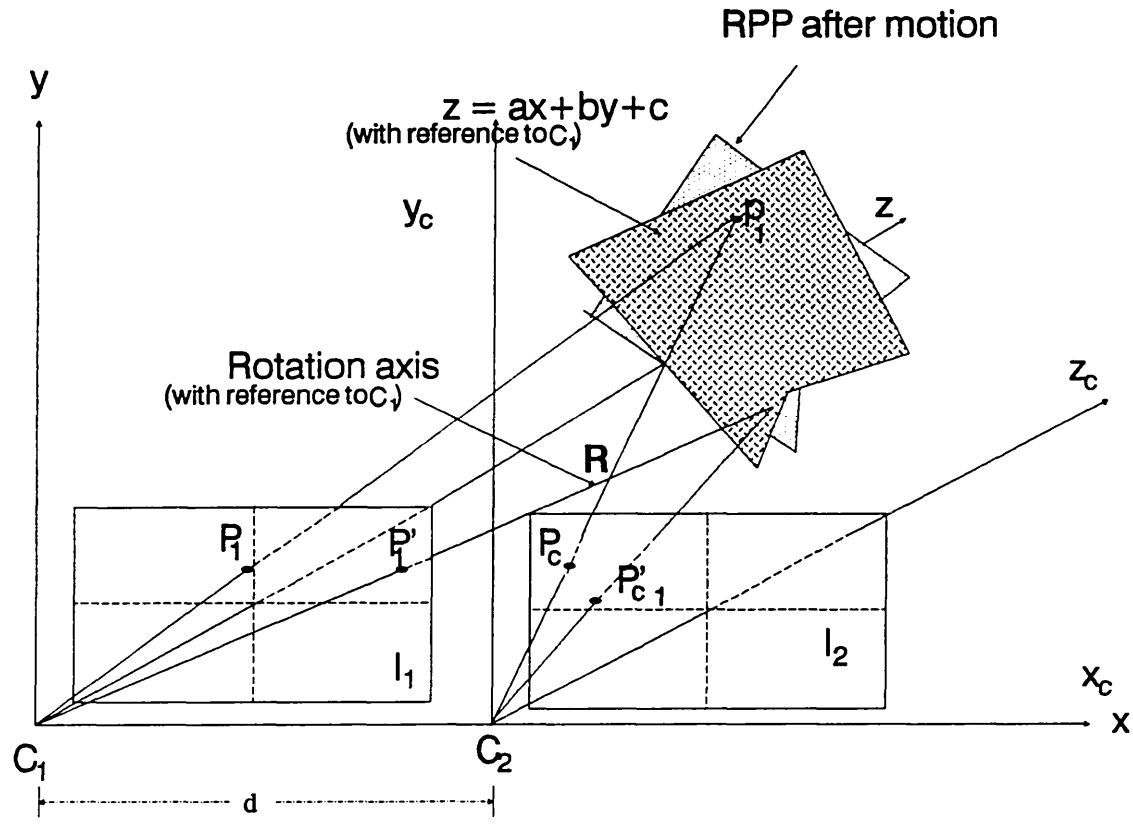


Figure 9.1

Geometry for stereo based motion estimation.

$$\mu'_{02} - m_{02} + 2\,B\,m_{01} - B^2 m_{00} = [r_{12}^2 - 2\,a\,r_{12}\,r_{23} + a^2\,r_{23}^2]\,m_{20} +$$
$$[2\{-r_{12} + a\,r_{23} - b\,r_{12}\,r_{23} + ab\,r_{23}^2\}]\,m_{11} + [2\,b\,r_{23} + b^2\,r_{23}^2]\,m_{20} +$$
$$[2\{-c\,r_{12}\,r_{23} + B\,r_{12} - B\,a\,r_{23} + a\,c\,r_{23}^2\}]\,m_{10} + [2\{cr_{23} - Bbr_{23} + bcr_{23}^2\}]\,m_{01} +$$
$$[-2\,B\,c\,r_{23} + c^2\,r_{23}^2]\,m_{00}. \qquad (9.3)$$

where A and B are the centroids of the patch after motion with respect to C_1 and are given by $A = \frac{m'_{10}}{m'_{00}}$ and $B = \frac{m'_{01}}{m'_{00}}$

From two images of the RPP with respect to C_1 it is possible to calculate μ'_{20}, μ'_{11}, μ'_{02} , $m_{ij}(i,j\epsilon\ 0,1,2\ )$, A and B. The unknowns are r_{13}, r_{12}, r_{23}(rotation parameters) , a,b and c(planar coefficients).The above equations are written in terms of the above unknown variables as follows:

From equation (9.1) we define ,

$$F_1 = r_{13}^2\{a^2m_{20} + 2abm_{11} + b^2m_{02} + c^2m_{00} + 2acm_{10} + 2bcm_{01}\} + r_{12}^2 m_{02}$$
$$+2\,r_{13}(aAC_3 + bAC_2 + cAC_1) + 2r_{12}r_{13}(am_{11} + bm_{02} + cm_{01}) + 2\,r_{12}AC_2$$
$$+\{\mu'_{20} - m_{20} + 2\,A\,m_{10} - A^2 m_{00}\} \qquad (9.4)$$

From equation (9.2) we define ,

$$F_2 = r_{13}r_{23}\{a^2m_{20} + 2abm_{11} + b^2m_{02} + c^2m_{00} + 2acm_{10} + 2bcm_{01}\} - r_{12}^2 m_{11}$$
$$-r_{12}r_{13}(am_{20} + bm_{11} + cm_{10}) + r_{13}(aAC_5 + bAC_6 + cAC_4)$$
$$+r_{23}(aAC_3 + bAC_2 + cAC_1) + r_{12}r_{23}(bm_{02} + am_{11} + cm_{01})$$
$$+r_{12}[AC_6 - AC_3] + \{\mu'_{11} - m_{11} + Bm_{10} + Am_{01} - ABm_{00}\} \qquad (9.5)$$

Similarly from equation (9.3) we define,

$$F_3 = r_{23}^2\{a^2m_{20} + 2abm_{11} + b^2m_{02} + c^2m_{00} + 2acm_{10} + 2bcm_{01}\} + r_{12}^2 m_{20}$$
$$+2r_{23}(aAC_5 + bAC_6 + cAC_4) - 2r_{12}r_{23}(bm_{11} + am_{20} + cm_{10})$$
$$-2r_{12}AC_5 - \{\mu'_{02} - m_{02} + 2Bm_{01} - B^2 m_{00}\} \qquad (9.6)$$

The constants AC_1, AC_2, AC_3, AC_4, AC_5 and AC_6 are as given in the appendix A.2.

We see that equations (9.4),(9.5)and (9.6) are non-linear in the unknown variables r_{12}, r_{13}, r_{23}, a , b and c.

Now let us see the motion of the RPP with respect to camera C_2. In the following, we derive the basic equations for the motion that takes place with respect to the camera C_2. We use orthographic projection as before. The features are again the first and second order moments.

Due to the arrangement of camera C_2, the planar patch equation will be different for this orientation of the camera. It is given by,

$$z_c = a'.\, x_c + b'.\, y_c + c' \tag{9.7}$$

where,

$$a = a' \tag{9.8a}$$

$$b = b' \tag{9.8b}$$

$$c = c' + d.a \tag{9.8c}$$

As we know the distance 'd' between the cameras it is possible to calculate c'. The points before and after motion with respect to camera C_2 is given by,

$$x'_c = \mathbf{R}\, x_c \tag{9.9}$$

As before we are interested in using first and second order centralised moments of the patch.

Projecting the space points (x_c,y_c,z_c) orthographically onto the image plane I_2 and substituting the planar equation 9.7 we get,

$$\begin{bmatrix} X'_c \\ Y'_c \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_4 & A_5 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \end{bmatrix} + \begin{bmatrix} A_3 \\ A_6 \end{bmatrix} \tag{9.10}$$

where,

$$A_1 = \left(1 + a\, r_{13}\right) \tag{9.11a}$$

$$A_2 = \left(r_{12} + b\, r_{13}\right) \tag{9.11b}$$

$$A_3 = c'\, r_{13} \tag{9.11c}$$

$$A_4 = \left(-r_{12} + a\, r_{23}\right) \tag{9.11d}$$

$$A_5 = \left(1 + b\, r_{23}\right) \tag{9.11e}$$

$$A_6 = c'\, r_{23} \tag{9.11g}$$

$$D = \overline{X_c'}\ ;\ E = \overline{Y_c'}$$

As before the centralised second order moments after the motion of the plane are now used to get three more equations. The centralised second order moments in this case are denoted by $\overline{{}_c\mu_{20}'}$, $\overline{{}_c\mu_{11}'}$ and $\overline{{}_c\mu_{02}'}$.

$$\overline{{}_c\mu_{20}'} = \sum\ (X_c' - \overline{X_c'})^2 \tag{9.12}$$

where,

$\overline{X_c'}$ = centroid of the patch after motion with respect to C_2

Expanding equation (9.12) we get,

$$\overline{{}_c\mu_{20}'} = \sum\Big\{A_1^2\, X_c^2 + A_2^2\, Y_c^2 + (A_3 - D)^2 + 2\, X_c Y_c\, A_1 A_2 + 2\, X_c (A_1 A_3 - A_1 D) + 2\, Y_c (A_2 A_3 - A_2 D)\Big\} \tag{9.13}$$

The above equation can be expanded and substituted for the general moments in place of X_c, Y_c, X_c^2, $X_c Y_c$ and Y_c^2 which can be written in terms of r_{13}, r_{12}, r_{23}, a, b and c. The complicated equation has linear and non-linear terms in the above unknowns. Expanding (9.13) and rearranging all terms on the Right Hand side we define

$$F_4 = r_{13}^2\left(a^2\, {}_cm_{20} + 2\, ab\, {}_cm_{11} + b^2 {}_cm_{02} + 2bc'\, {}_cm_{01} + 2ac'\, {}_cm_{10} + c'^2\, {}_cm_{00}\right) + r_{12}^2\, {}_cm_{02} + 2r_{12}r_{13}\left(a\, {}_cm_{11} + b\, {}_cm_{02} + c'\, {}_cm_{01}\right) + 2r_{13}\left(a\, {}_cm_{20} + b\, {}_cm_{11} + c'\, {}_cm_{10} - D(a\, {}_cm_{10} + b\, {}_cm_{01} + c'\, {}_cm_{00})\right) - 2Dr_{12}\, {}_cm_{01} + 2r_{12}\, {}_cm_{11} + {}_cm_{20} - 2D\, {}_cm_{10} + D^2\, {}_cm_{00} - \overline{{}_c\mu_{20}'} \tag{9.14}$$

where the moments ${}_cm_{00}$, ${}_cm_{10}$, ${}_cm_{01}$, ${}_cm_{20}$, ${}_cm_{11}$ and ${}_cm_{02}$ are with respect to C_2.

The values of ${}_cm_{ij}$ can be found from the image patches.

Writing $\overline{{}_c\mu_{11}'}$ in the same way as that of $\overline{{}_c\mu_{20}'}$ we get,

$$\overline{{}_c\mu_{11}'} = \sum \left(X_c' - \overline{X_c'} \right) \left(Y_c' - \overline{Y_c'} \right) \tag{9.15}$$

$$\overline{{}_c\mu_{11}'} = \sum \Big\{ A_1A_4X_c^2 + X_cY_c(A_1A_5 + A_2A_4) + A_2A_5Y_c^2 + X_c\Big(A_1(A_6-E) + A_4(A_3-D)\Big) + Y_c\Big(A_2(A_6-E) + A_5(A_3-D)\Big) + [(A_6-E)(A_3-D)]\Big\} \tag{9.16}$$

Similarly we define,

$$\begin{aligned} F_5 = {} & r_{12}\Big({}_cm_{02} - {}_cm_{20} + D\,{}_cm_{01} - DE\,{}_cm_{00}\Big) + r_{23}\Big(a\,{}_cm_{20} + b\,{}_cm_{11} + c'\,{}_cm_{10} - D(a\,{}_cm_{10} + b\,{}_cm_{01} \\ & + c'\,{}_cm_{00})\Big) + r_{13}\Big(a\,{}_cm_{11} + b\,{}_cm_{02} + c'\,{}_cm_{01} - E(a\,{}_cm_{10} + b\,{}_cm_{01} + c'\,{}_cm_{00})\Big) - \\ & r_{12}r_{13}(a\,{}_cm_{20} + b\,{}_cm_{11} + c'\,{}_cm_{10}) + r_{12}r_{23}\Big(a^2\,{}_cm_{20} + 2ab\,{}_cm_{11} + b^2\,{}_cm_{02} + 2bc'\,{}_cm_{01} + \\ & 2ac'\,{}_cm_{10} + c'^2\,{}_cm_{00}\Big) + r_{12}r_{23}(a\,{}_cm_{11} + b\,{}_cm_{02} + c'\,{}_cm_{01}) - r_{12}^2\,{}_cm_{11} \\ & - \Big(\overline{{}_c\mu_{11}'} - {}_cm_{11} + E\,{}_cm_{10} + D\,{}_cm_{01} - DE\,{}_cm_{00}\Big) \end{aligned} \tag{9.17}$$

D and E are the centroids of the patch with respect to C_2.

Writing $\overline{{}_c\mu_{02}'}$ as above we get,

$$\overline{{}_c\mu_{02}'} = \sum \left(Y_c' - \overline{Y_c'} \right)^2 \tag{9.18}$$

Expanding,

$$\begin{aligned} \overline{{}_c\mu_{02}'} = \sum \Big\{ & A_4^2\,X_c^2 + A_5^2\,Y_c^2 + (A_6-E)^2 + 2\,X_cY_c\,A_4A_5 + 2\,X_c(A_4A_6 - A_4E) + \\ & 2\,Y_c(A_5A_6 - A_5E)\Big\} \end{aligned} \tag{9.19}$$

Similarly we define

$$\begin{aligned} F_6 = {} & +r_{12}^2\,{}_cm_{02} - 2r_{12}r_{23}(a\,{}_cm_{20} + b\,{}_cm_{11} + c'\,{}_cm_{10}) + \\ & r_{23}^2\Big(a^2\,{}_cm_{20} + 2\,ab\,{}_cm_{11} + b^2\,{}_cm_{02} + 2bc'\,{}_cm_{01} + 2ac'\,{}_cm_{10} + c'^2\,{}_cm_{00}\Big) + \\ & 2r_{12}(-{}_cm_{11} + E\,{}_cm_{10}) + 2\,r_{23}(a\,{}_cm_{11} + b\,{}_cm_{02} + c'\,{}_cm_{01} - E(a\,{}_cm_{10} + b\,{}_cm_{01} + c'\,{}_cm_{00})\Big) \\ & - \overline{{}_c\mu_{02}'} + {}_cm_{02} - 2E\,{}_cm_{01} + E^2\,{}_cm_{00} \end{aligned} \tag{9.20}$$

Thus we get two sets of three equations in which the unknowns are in non-linear form. As in the previous chapter we can solve this system of equations using the LeM method.The basic equation for the LeM is given below,

$$W^{(k+1)} = W^{(k)} - [\mu_k I + J(W^{(k)})^T . J(W^{(k)})]^{-1} J(W^{(k)})^T . F(W^{(k)}) \tag{9.21}$$

where $F(.)=\{F_1, F_2, F_3, F_4, F_5, F_6\}$ and $J(.,.)$ is the Jacobian of the functions F_1, F_2 ... F_6 which is given by the following matrix

$$J_{ij} = \begin{bmatrix} \frac{\partial F_1}{\partial r_{13}} & \frac{\partial F_1}{\partial r_{12}} & \frac{\partial F_1}{\partial r_{23}} & \frac{\partial F_1}{\partial a} & \frac{\partial F_1}{\partial b} & \frac{\partial F_1}{\partial c} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial F_6}{\partial r_{13}} & \frac{\partial F_6}{\partial r_{12}} & \frac{\partial F_6}{\partial r_{23}} & \frac{\partial F_6}{\partial a} & \frac{\partial F_6}{\partial b} & \frac{\partial F_6}{\partial c} \end{bmatrix}$$

$$\mu_k = C \cdot \|F(W^{(k)})\|_1^2$$

$$C = 10 \qquad 10 \leq \|F(W^{(k)})\|_1$$

$$= 1 \qquad 1 \leq \|F(W^{(k)})\|_1 < 10$$

$$= 0.01 \qquad \|F(W^{(k)})\|_1 < 1$$

The actual elements of the Jacobian matrix are given in Appendix A–3. It should be noted here , that the Jacobian matrix in the above LeM equation is a 6x6 matrix unlike in the previous formulation. Now we have six unknowns and all the six unknowns can be estimated as we have stereo vision. In addition, the

images are identified using the moment invariant functions as described already in the previous chapter.

9.3 SIMULATION AND RESULTS:

All the results to be presented here are for synthetic images. The data generation is similar to that discussed in the previous chapter, except for the two camera arrangement. We generated N (N=1000 in most cases) points which lie on the RPP given by equation z=ax+by+c with respect to C_1. The data points after motion are calculated as explained in the forward motion formulation case of Chapter 8. The generated 3-D space points before and after motion are projected orthographically to get the image points before and after motion. These image points correspond to the camera C_1.

The 3-D space points before and after motion with respect to camera C_2 are obtained from the 3-D space points already generated for camera C_1 and using equations (9.7),(9.8) and (9.9). As before , the image points are derived using an orthographic projection.

Having derived the image points before and after motion with respect to camera C_1 and C_2 we calculate the moment invariants . The moment invariants are within a pre-selected threshold and hence the lower order moments are used for motion estimation.

The stereo based algorithm is also LeM based. In addition to the moments, the moment invariants and the centralised moments with respect to C_1 and C_2, we give an initial guess solution for r_{13}, r_{12}, r_{23}, p, q and r to the LeM algorithm. The solutions are obtained by successive iterations. We obtain the solution after it converges to a constant value or after a fixed number, say 50 , of iterations.

We performed several simulation experiments based on the above formulation. In particular, we studied the effect of varying the number of pixels in the RPP, varying the rotation angle, and varying the RPP distance from the origin with respect to camera C_1. The results are as shown in figures 9.2–9.7. In these simulation experiments we gave two different initial guess solutions. In the first case, we gave the actual values of parameters r_{13}, r_{12}, r_{23}, p , q and r as the initial guess. In the second case we distorted each of the parameters by 1% from the actual value, and gave that as the initial guess solution. The results for both these initial conditions are presented in figures 9.2–9.7.

From the results we observe that increasing the number of pixels improves the error performance of various parameters. The relative error percentage for the rotation parameters, was a minimum value at 500 pixels and increases a little at 750 pixels and stays at a constant value for further increases in the number of pixels. However, this minimum error occurs at value of 750 pixels for the planar coefficients estimation. As the size of the patch increases, we expect an improvement in the error performance.

In addition increasing the rotation angle , in general , improves the error performance. However, it should be mentioned that increasing the rotation angle more than 10° would invalidate our initial approximation made for the rotation matrix. The error performance for the planar coefficients is, however, very random.

Increasing the RPP distance in general , should improve the performance of the parameters, as the orthographic projection assumption truly holds good at larger distances from the camera. This seems to be true as far as r_{13} and r_{23} are concerned , except at 45 units for r_{23}. However, the error rate increases

Figure 9.2
Effect of varying number of pixels on rotation parameters.

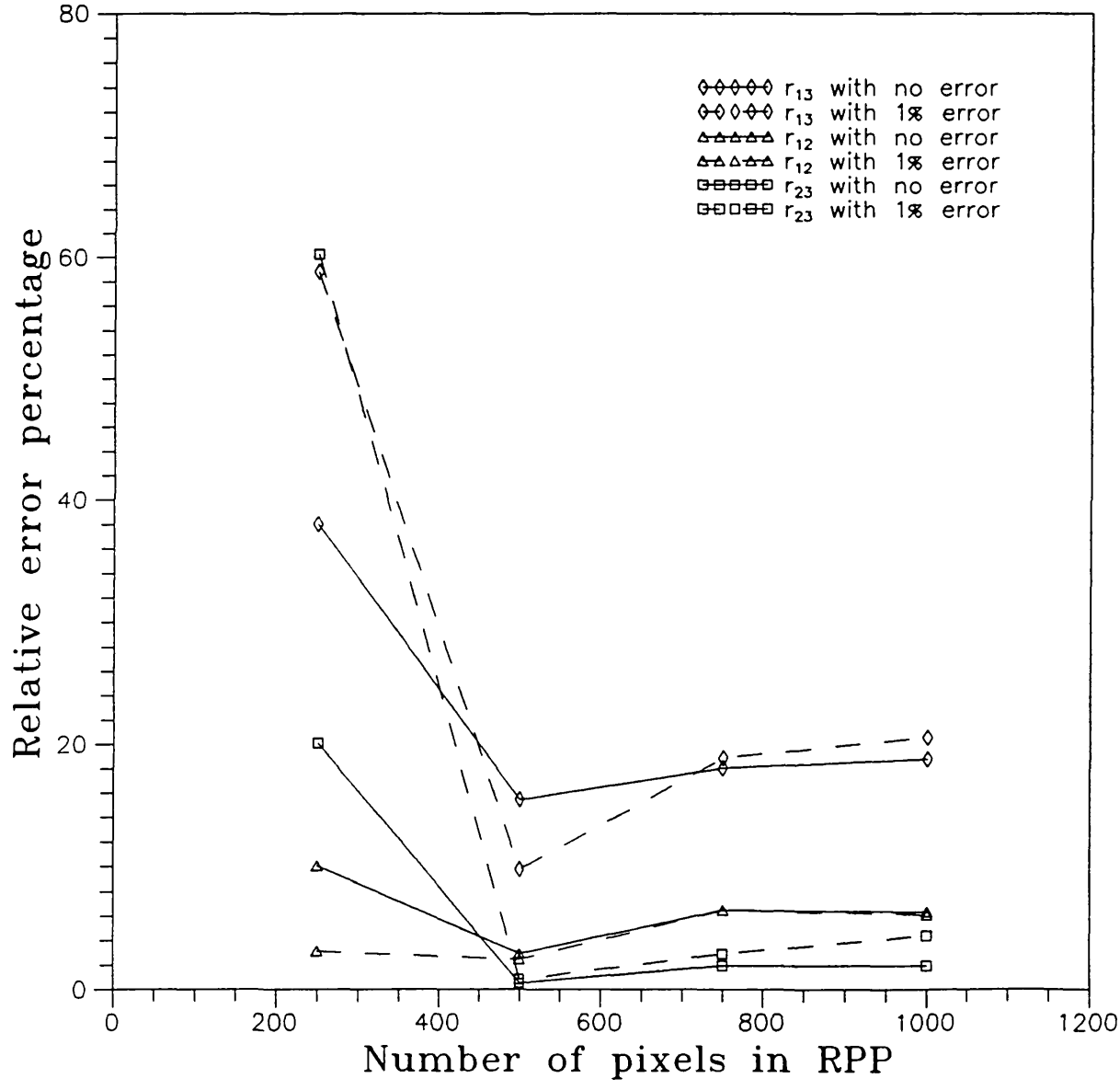


Figure 9.3
Effect of varying number of points on Planar coefficients.

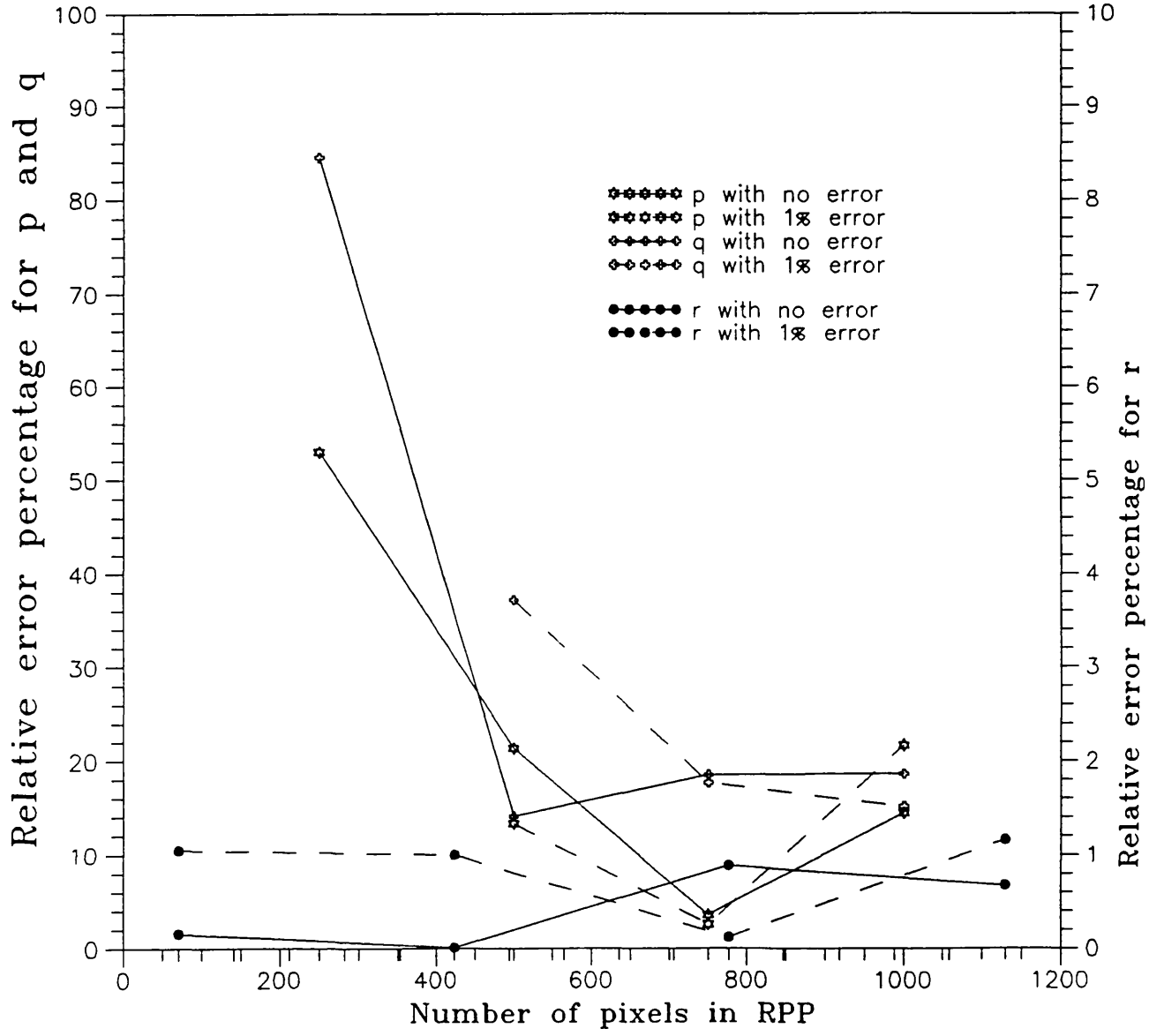


Figure 9.4
Effect of varying rotation angle on rotation parameters.

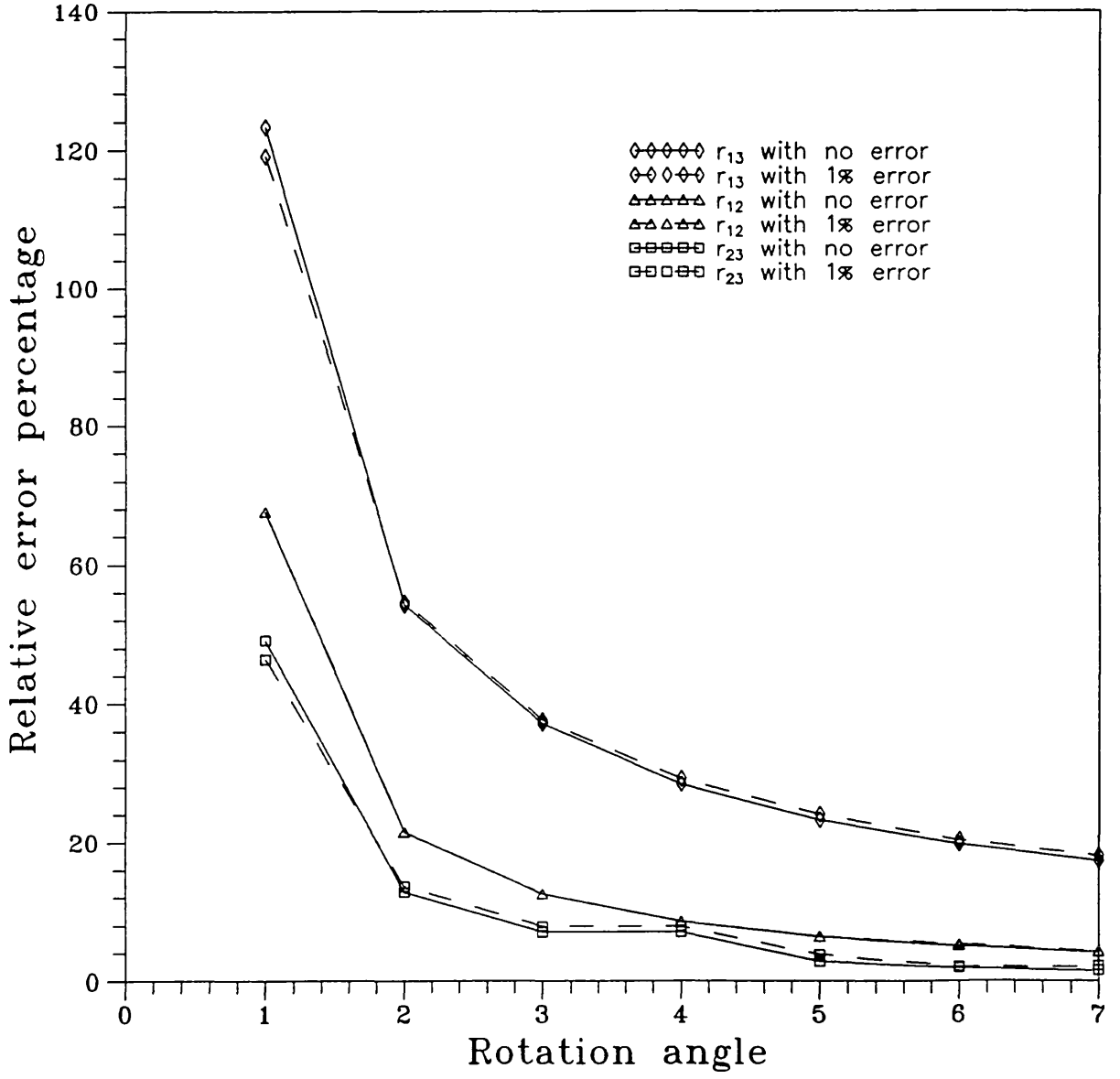


Figure 9.5
Effect of varying rotation angle on
Planar coefficients.

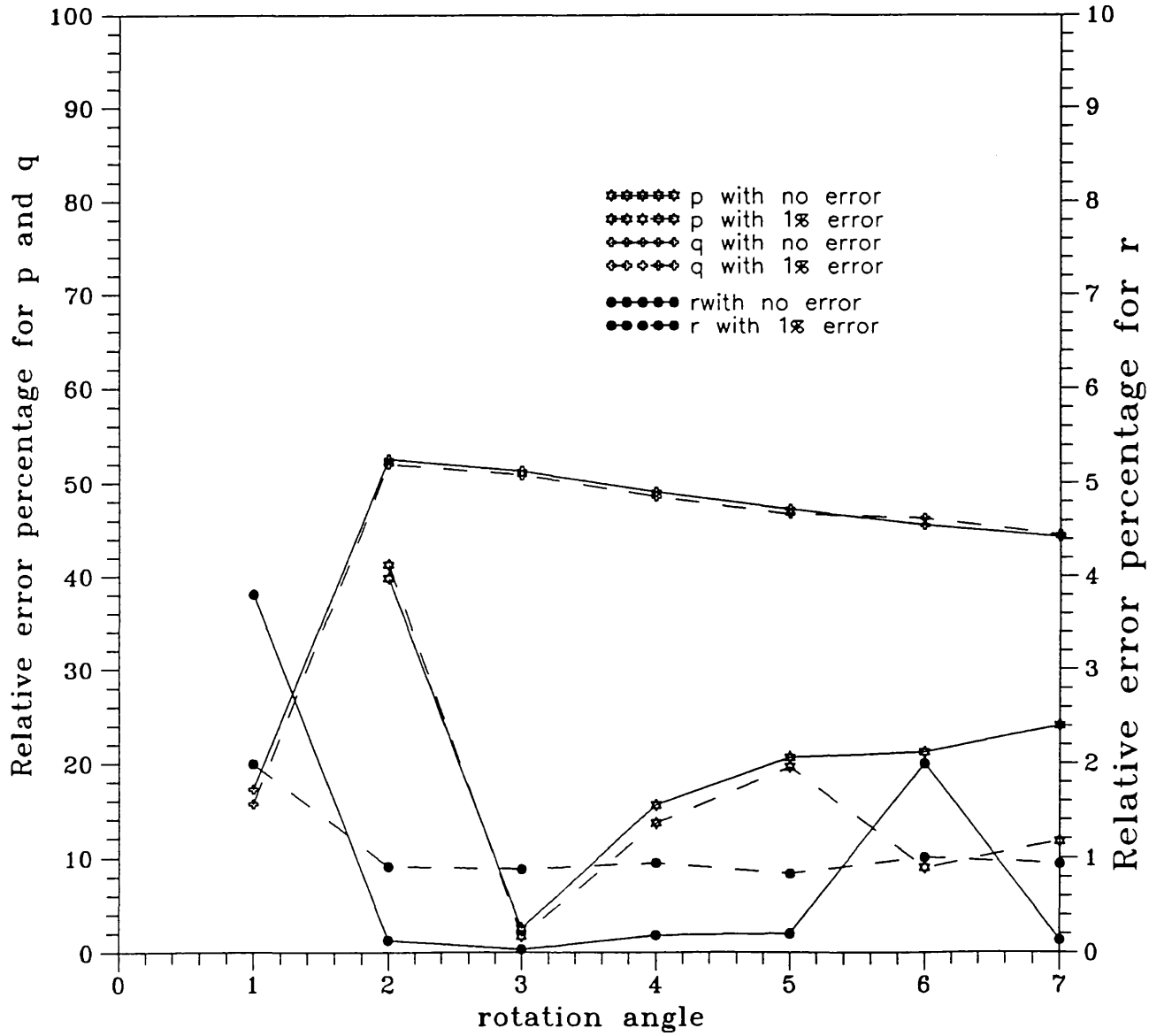


Figure 9.6
Effect of varying RPP distance on rotation parameters.

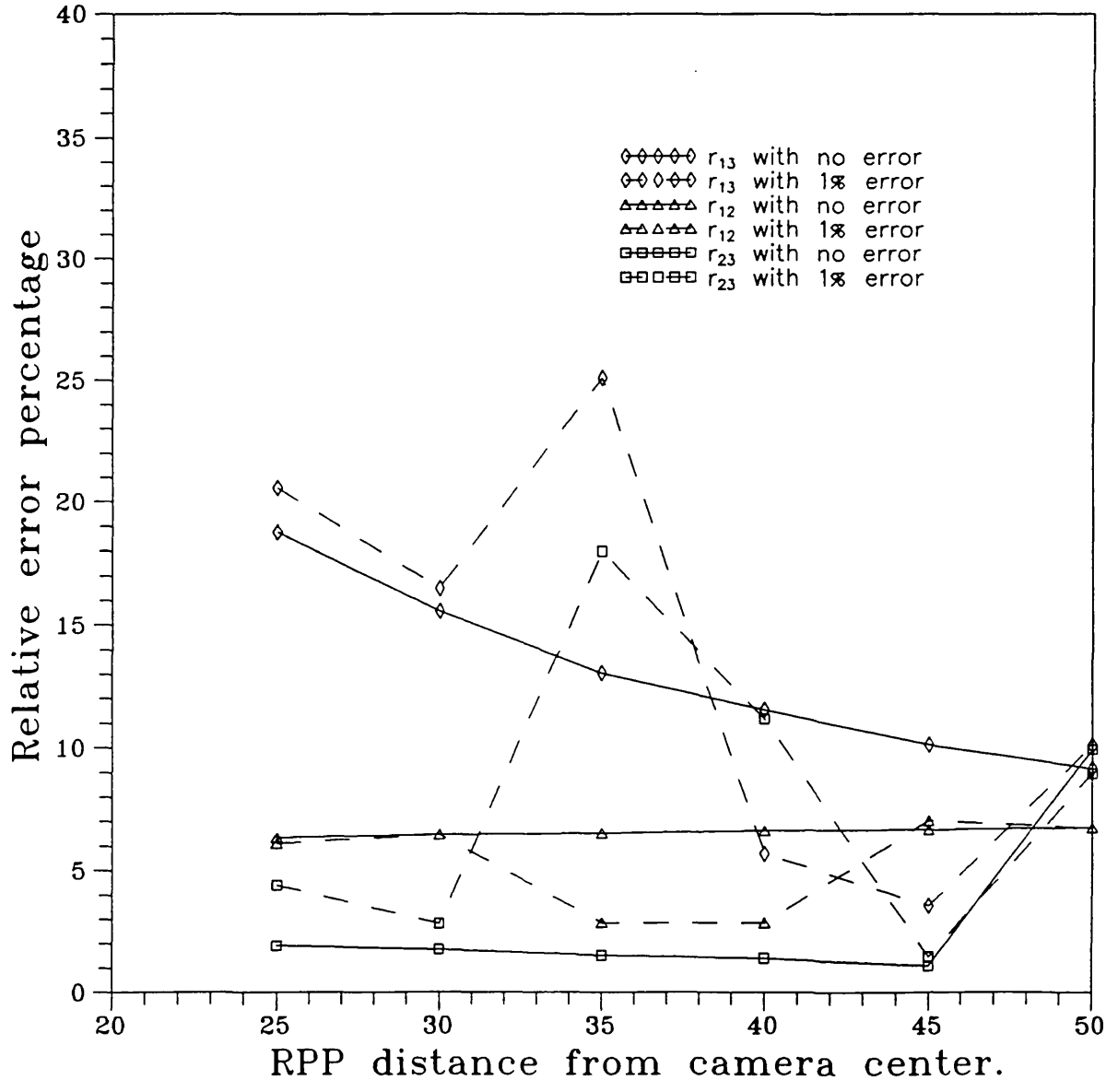
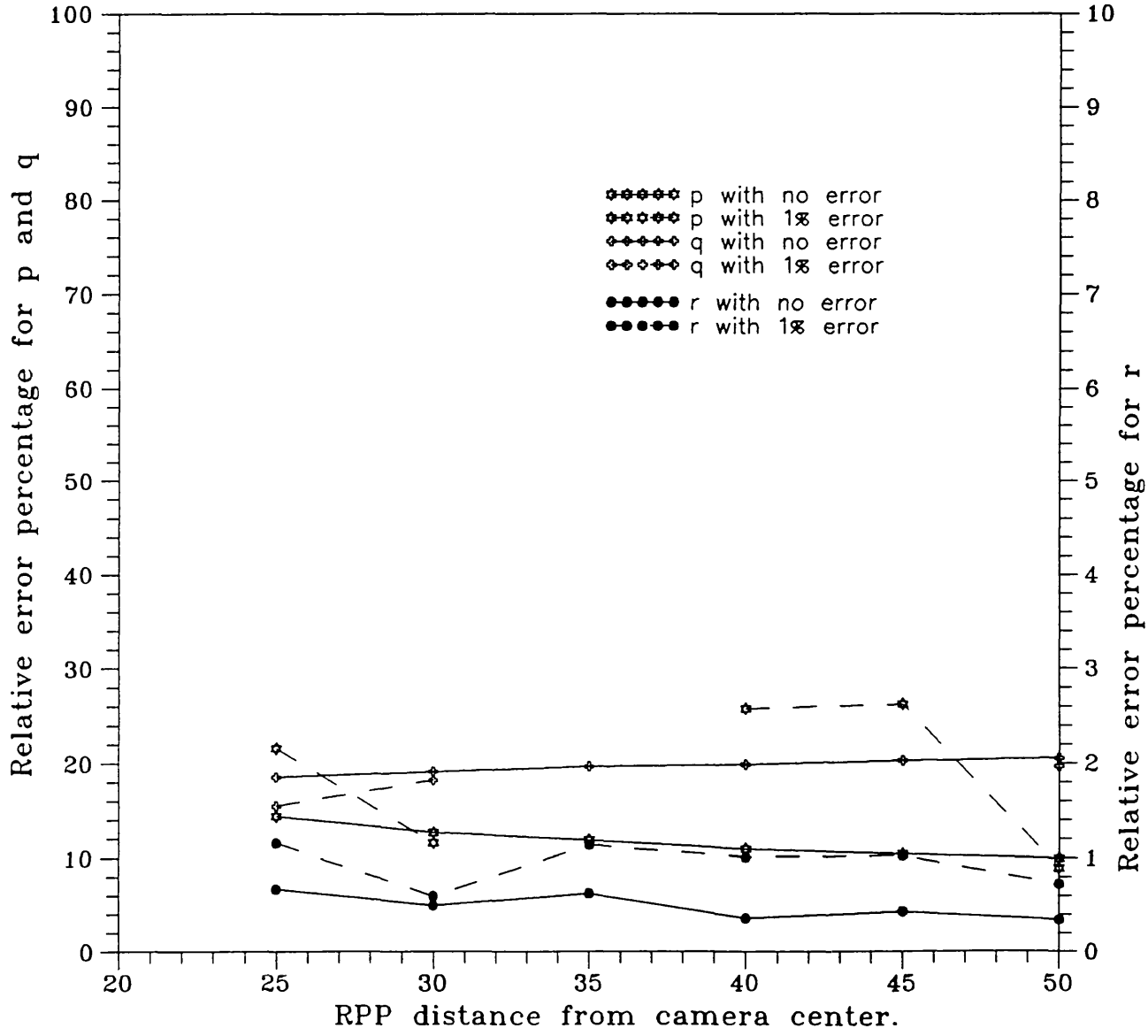


Figure 9.7
Effect of varying RPP distance on
Planar coefficients.



for r_{12}, p , q and r as we increase the distance of the plane from the camera center.

Also when there is an error in the initial condition the error performance of the algorithm deteriorates. However in some cases we observe a better performance; such instances are only a few.

The tables T9.1–T9.3 show the results for the above simulation experiments. We have also included the noise performance in these experiments. Addition of noise is similar to that explained in chapter-7. The results are shown in table T-9.4. As expected , addition of noise increases the error percentages.

In all the experiments we noted that the error performance of the planar coefficients, namely, p,q and r is relatively inferior to that of the rotation parameters. This may be due to the stereo arrangement. When the plane is perpendicular to z-axis of camera C_1, it may not be truly perpendicular to the z-axis of the camera C_2. Hence the orthographic projection assumption for camera C_2 is not perfectly satisfied. This may be one of the reasons for such behaviour.

In conclusion, we say that further work is needed in stereo vision system.

TABLE –T 9.1

Effect of varying the number of points.

INPUT:

ROTATION AXIS: (65°, 70°, 29.707°)

PLANE ORIENTATION: (0.04, 0.05, 25.0)

ANGLE OF ROTATION: 5°

INITIAL GUESS: (0.02259,−0.07580,−0.03688,0.04,0 05,25.0)

NO POINTS	TYPE	r_{13}	r_{12}	r_{23}	p	q	r
250	0 devia[1]	0.01399	-0.06818	-0.04429	0.01881	0.00777	25.040
	error[2]	-38.04	-10.05	20.09	-52.97	-84.47	0.16
	1% devia	*0.03588*	*-0.07342*	*-0.05911*	*-0.65546*	*1.41514*	*25.262*
	error	*58.83*	*-3.13*	*60.27*	*-1738.64*	*-2730.27*	*1.05*
500	0 devia	0.19095	-0.07358	-0.03669	0.03145	0.04293	25.002
	error	-15.47	-2.93	-0.52	-21.38	-14.13	0.009
	1% devia	*0.02479*	*-0.07393*	*-0.03718*	*0.03466*	*0.03141*	*25.251*
	error	*9.77*	*-2.47*	*0.81*	*-13.36*	*-37.19*	*1.002*
750	0 devia	0.18518	-0.07087	-0.03617	0.03858	0.04704	24.777
	error	-18.03	-6.50	-1.93	-3.56	-18.53	0.89
	1% devia	*0.01833*	*-0.07087*	*-0.03580*	*0.03898*	*0.04115*	*25.03*
	error	*-18.86*	*-6.50*	*-2.92*	*-2.56*	*-17.70*	*0.12*
1000	0 devia	0.18351	-0.07099	-0.03617	0.03422	0.04071	24.832
	error	-18.76	-6.35	-1.93	-14.44	-18.58	0.674
	1% devia	*0.01795*	*-0.07117*	*-0.03525*	*0.03135*	*0.04244*	*25.289*
	error	*-20.56*	*-6.11*	*-4.41*	*-21.63*	*-15.12*	*1.16*

[1] 0 devia in this table and the following tables means that the initial guess is the actual values and 1% deviation means that the initial guess is 1% away from the actual values.

[2] error here means the relative percentage error

TABLE –T 9.2

Effect of varying the rotation angle.

INPUT:

ROTATION AXIS: (70°, 80°, 22.555°) NUMBER OF POINTS: 1000

PLANE ORIENTATION: (0.02, 0.04, 30.0)

ROT. ANGLE	TYPE	r_{13}	r_{12}	r_{23}	p	q	r
1	0 devia	0.010676	-0.02702	-0.00891	-0.04386	0.04138	28.856
	error	123.23	67.59	49.09	-319.32	-17.24	-3.81
	1% devia	*0.00664*	*-0.02702*	*-0.00874*	*-0.04469*	*0.04216*	*29.401*
	error	*119.08*	*-67.59*	*46.33*	*-323.46*	*-15.68*	*-1.99*
2	0 devia	0.00278	-0.02530	-0.01042	0.02796	0.02372	29.961
	error	-54.175	-21.51	-12.747	39.81	-52.556	-0.130
	1% devia	*0.00275*	*-0.02580*	*-0.01031*	*0.02825*	*0.02397*	*30.273*
	error	*-54.653*	*-21.514*	*-13.64*	*41.265*	*-52.062*	*0.909*
3	0 devia	0.00572	-0.04228	-0.01665	0.01949	0.02435	30.009
	error	-37.118	-12.552	-7.058	-2.55	-51.3	0.031
	1% devia	*0.00567*	*-0.04228*	*-0.01651*	*0.01966*	*0.02456*	*30.265*
	error	*-37.646*	*-12.552*	*-7.839*	*-1.71*	*-50.886*	*0.882*
4	0 devia	0.00867	-0.05894	-0.02286	0.01689	0.02546	29.944
	error	-28.441	-8.585	-4.255	-15.565	-49.072	-0.184
	1% devia	*0.00858*	*-0.05893*	*-0.02262*	*0.01727*	*0.02572*	*30.284*
	error	*-29.233*	*-5.289*	*-13.655*	*-21.63*	*-48.572*	*0.947*
5	0 devia	0.01163	-0.07546	-0.02902	0.01588	0.02640	29.940
	error	-23.207	-6.364	-2.791	-20.62	-47.20	-0.199
	1% devia	*0.01152*	*-0.07546*	*-0.02872*	*0.01608*	*0.02667*	*30.249*
	error	*-23.933*	*-6.367*	*-3.367*	*-19.60*	*-46.67*	*0.832*
6	0 devia	0.01459	-0.09184	-0.03514	0.01576	0.02735	29.940
	error	-19.747	-5.033	1.904	-21.20	-45.486	-1.998
	1% devia	*0.01448*	*-0.09159*	*-0.03508*	*0.02179*	*0.02686*	*30.302*
	error	*-20.341*	*-5.292*	*-2.066*	*8.96*	*-46.23*	*1.005*
7	0 devia	0.01755	-0.10823	-0.04119	0.01520	0.02790	29.960
	error	-17.286	-4.077	-1.424	-24.00	-44.22	-0.133
	1% devia	*0.01740*	*-0.108129*	*-0.04092*	*0.01766*	*0.02778*	*30.281*
	error	*-18.007*	*-4.166*	*-2.082*	*-11.705*	*-44.432*	*0.936*

TABLE –T 9.3

Effect of varying the depth values.

INPUT:

ROTATION AXIS: (65°, 75°, 29.707°)

ANGLE OF ROTATION: 5° NUMBER OF POINTS: 1000

PLANE ORIENTATION: (0.04, 0.05, 20.0)

DEPTH VALUES	TYPE	r_{13}	r_{12}	r_{23}	p	q	r
25	0 devia	0.01835	-0.07099	-0.03617	-0.03422	0.04071	24.832
	error	-18.76	-6.35	-1.93	-14.44	-18.58	-0.67
	1% devia	*0.01795*	*-0.07117*	*-0.03525*	*0.03135*	*0.04244*	*25.290*
	error	*-20.56*	*-6.11*	*-4.41*	*-21.63*	*-15.52*	*1.16*
30	0 devia	0.01907	-0.07088	-0.03623	0.03491	0.04042	29.851
	error	-15.57	-6.49	-1.76	-12.73	-19.17	-0.496
	1% devia	*0.01866*	*-0.07088*	*-0.03584*	*0.03534*	*0.04085*	*30.180*
	error	*-16.49*	*-6.49*	*-2.83*	*-11.65*	*-18.3*	*0.599*
35	0 devia	0.01964	-0.07085	-0.03638	0.03524	0.04015	34.782
	error	-13.05	-6.53	-1.53	-11.91	-19.69	-0.624
	1% devia	*0.01692*	*-0.07363*	*-0.03024*	*-0.10905*	*0.19840*	*35.398*
	error	*-25.07*	*-2.87*	*-17.99*	*-372.63*	*296.79*	*1.14*
40	0 devia	0.01999	-0.07079	-0.03636	0.03562	0.04005	39.857
	error	-11.53	-6.61	-1.41	-10.95	-19.89	-0.357
	1% devia	*0.02387*	*-0.07796*	*-0.04099*	*0.02970*	*0.12256*	*40.404*
	error	*5.68*	*2.85*	*11.16*	*-25.745*	*145.11*	*1.01*
45	0 devia	0.02031	-0.07074	-0.03645	0.03583	0.03985	44.806
	error	-10.12	-6.67	-1.10	-10.43	-20.30	-0.431
	1% devia	*0.02339*	*-0.08114*	*-0.03634*	*0.02951*	*-0.05481*	*45.462*
	error	*3.56*	*7.04*	*-1.46*	*-26.23*	*-209.61*	*1.026*
50	0 devia	0.02053	-0.07069	-0.03647	0.03603	0.03975	49.830
	error	-9.13	-6.73	-1.11	-9.92	-20.51	-0.341
	1% devia	*0.02031*	*-0.07070*	*-0.03609*	*0.03642*	*0.04017*	*50.360*
	error	*-10.08*	*-6.73*	*-2.16*	*-8.94*	*-19.66*	*0.721*

TABLE —T 9.4.

Effect of noise on motion parameters.

INPUT:

ROTATION AXIS: (70°, 80°, 22.555°) NUMBER OF POINTS: 1000

ANGLE OF ROTATION=4°

PLANE ORIENTATION: (0.02,0.05,30.0)

Noise	TYPE	r_{13}	r_{12}	r_{23}	p	q	r
0	0 devia	0.00868	-0.05901	-0.02289	0.01532	0.03606	29.945
	error	-28.35	-8.46	-4.15	-23.4	-27.89	-0.183
	1% devia	*0.00859*	*-0.05901*	*-0.02265*	*0.01550*	*0.03644*	*30.265*
	error	*-29.109*	*-8.46*	*-5.163*	*-22.495*	*-27.128*	*0.883*
0.1	0 devia	0.00049	-0.02855	-0.01451	-0.01015	0.02297	30.036
	error	-95.899	-55.71	-39.24	-49.27	-54.07	0.122
	1% devia	*0.00049*	*-0.02855*	*-0.01438*	*-0.01024*	*0.02317*	*30.300*
	error	*-95.941*	*-55.71*	*-39.765*	*-48.825*	*-53.664*	*1.001*
0.4	0 devia	0.00176	0.00057	-0.00691	0.61385	0.21182	29.996
	error	-85.528	-100.088	-71.072	2969.24	323.636	-0.045
	1% devia	*0.00174*	*0.00057*	*-0.00684*	*0.61987*	*0.21389*	*30.291*
	error	*-85.668*	*-100.088*	*-71.353*	*2999.36*	*327.794*	*0.969*
0.9	0 devia	-0.00174	0.01901	-0.00315	0.58720	1.33270	30.121
	error	-108.869	-105.814	-83.748	2887.53	2550.32	0.271
	1% devia	*-0.00199*	*-0.00925*	*-0.00326*	*0.29178*	*1.56463*	*30.375*
	error	*-116.452*	*-114.348*	*-86.34*	*1358.89*	*3029.26*	*1.252*
2.0	0 devia	-0.00156	0.02887	-0.00195	*[3]	*	29.713
	error	-112.904	-144.788	-91.847			-0.956
	1% devia	*-0.00127*	*0.00083*	*-0.00258*	*	*	*30.006*
	error	*-110.479*	*-101.287*	*-89.196*			*0.018*
6.0	0 devia	-0.01069	-0.01418	-0.00133	*	*	29.287
	error	-188.177	-78.011	-94.447			2.375
	1% devia	*-0.00228*	*0.03661*	*-0.00447*	*	*	*29.439*
	error	*-118.804*	*-156.785*	*-81.298*			*1.871*
10.0	0 devia	-0.00186	0.09558	-0.01093	*	*	28.829
	error	-115.338	-248.253	-54.242			3.90
	1% devia	*0.00182*	*-0.03431*	*-0.00982*	*	*	*29.089*
	error	*-144.538*	*-44.775*	*-58.89*			*3.036*

[3] represents that the parameters diverge from the actual values.

CHAPTER TEN

CONCLUSION

In this thesis , the problem of three dimensional motion parameter estimation of rigid objects from image sequences is investigated. The investigation is performed using a feature based approach. The work was organised in two parts. In the first part the results of a point correspondence based motion estimation are given. Results of a region based motion estimation are given in part 2.

10.1 CONTRIBUTIONS OF THIS RESEARCH:

(1) We have improved the translation parameter estimation of a well known linear algorithm, which performed very poorly in presence of noise. In particular the modified method gives very good results in the case of a medium translation.

(2) A fast algorithm is developed for detecting the feature points. A new shuffling scheme is suggested to improve the performance of feature detection. However, it has a disadvantage of missing some of the closely adjacent corners.

(3) With regard to the correspondence analysis we used a ' relaxation based approach. Although this method can be speeded up by parallel processing methods, it has a draw back of saturating to a fixed value after a few iterations.

(4) A novel moments based estimation is proposed. Two iterative schemes are developed. It is found that one scheme based on Levinberg-Marquardt method performs better than the other scheme. Two theorems have been suggested and proved for degenerate configurations of a rigid planar patch in motion.

(5) Extension to moments based motion estimation in the case of stereo

images is tried. The main advantage is that we can estimate all the six parameters (3 for rotation parameters and 3 for planar coefficients) with this stereo formulation. This method is again iterative and based on LeM. However, the method is found to be very sensitive to initial guess solution.

Some of the findings are discussed as below.

A critical analysis made on THLAM algorithm, revealed the limitations of the method in presence of noise. The linear method gave very poor results for the motion parameter estimation, in particular to the translation motion parameter estimation. The modification suggested in Chapter-3 improved the performance of this algorithm. The modification is based on a rotation compensation method. An extensive noise analysis is conducted for both the methods. Three different real world cases are analysed, namely, (a) small translation (b) medium translation and (c) large translation. It is found that the modified method out-performs that of the linear method in all cases. In particular, it is seen that the performance of the modified method is excellent in the case of a medium translation case.

The performance of the linear and the modified methods are compared for a real world image sequence. In the course of the work we have compared various corner detectors and finally used the Zuniga and Haralick corner detector for detecting the feature points. A fast algorithm is developed for the identification of corner points. Due to the noise and discontinuity a large number of corner points are detected. To reduce the number of corner points a novel shuffling scheme is developed(Chapter 5). Some of the properties of the corner points like sharpness, contrast, time-varying cornerness etc., are derived from the detected corner points. These characteristics are used for the correspondence analysis. The correspondence analysis is based on a ' relaxation ' method.

As already mentioned, in part 2 of the thesis, we use a region based motion estimation. Here moment invariants are used for the identification of a region of interest. Once the region of interest is identified the lower order moments of the regions are used for motion estimation. The pure rotation case with orthographic projection is fully explored. We considered the motion of a planar patch in 3-D. With respect to this, we formulated the problem and derived the basic equations. During the course of work we found some of the configurations do not give unique results. To verify the feasibility of this method we proposed two theorems which also point out the degenerate configurations. We provide a simple proof for the theorems.

Two iterative schemes were developed . Extensive simulation tests were performed on these iterative schemes . Our motion estimation algorithm thus gives the estimation of the rotation parameters and the planar coefficients of the plane. One of the schemes , namely, the Levinberg - Marquardt algorithm performs better compared to the other straight forward iterative scheme. The error performance is found to improve as the angle of rotation is increased. In addition, the relative error percentage decreases as the depth of the plane is increased for a fixed rotation angle. If the data points are from a normalised set of points then the convergence rate is faster. However, in such cases, the relative error seems to be greater when compared to that of the relative percentage error without normalising.

We also considered the estimation of a pure translation motion using moments. Motion estimation in presence of noise is considered. The translation motion estimation is found to be very robust even in presence of a large noise in the data.

A novel stereo based motion estimation is modelled. This method also uses

moments as features for motion estimation. The equations are derived for the proposed method. As in the single camera method , increasing the rotation angle improves the performance. Also when the RPP size is increased we see a better performance of the method. Our simulation results showed that this method is very sensitive to the initial guess solution, in particular for the planar coefficients p,q and r.

10.2 SUGGESTIONS FOR FURTHER RESEARCH:

The rotation compensation which we have discussed in Chapter 3 is based on THLAM. As this linear method is sensitive to noise, other methods can be tried so as to get a better estimation for rotation parameters and there-by improve the rotation compensation which will yield a better estimation of translation parameters.

In Chapter 5 we described a method of motion estimation using real image sequences. However, we have used the various thresholds required by the algorithm for feature detection and correspondence analysis by trial and error method. It will be useful if a systematic study is performed on the use of various thresholds and their effect individually and collectively on the motion parameter estimation. Also, the effect of occlusion and the scale of the object / objects while they are in motion can be investigated. The relaxation based approach can be improved using some more information regarding the scene.

With regard to region based method, further work can be done for a general motion in the orthographic projection case. Also perspective projection of the RPP motion will be extremely useful. The main difficulty which arises in this case is the perspective distortion. The depth normalising experiments discussed in Chapter 8 may provide some clues in this direction. When the object is not very

much inclined with respect to the image plane, the perspective distortion will be less and hence the orthographic projection analysis can be extended. Also the study of moment invariant functions under perspective imagery is needed when dealing with perspective projection. Further work is needed in using region based methods for real image sequences where image segmentation is to be performed.

Another important extension may be tried with a combination of features namely, points lines and regions for a better motion estimation algorithm.

Motion estimation from images is an ill-posed inverse problem. For such ill-posed problems a ' regularization method ' is suggested by the mathematicians. A recent trend in motion analysis is in using regularization theory [115] for a better estimation. However, the understanding and usage of this theory to computer vision field is very little and limited. Hence further work can be directed towards modifying and using this approach appropriately.

APPENDIX A –1

FINDING THE ROTATIONAL PARAMETERS

In this Appendix we see how the rotational parameters (n_1, n_2 and θ) are found from the elements of the rotation matrix **R**. The elements r_{11}, r_{12}, r_{13} ... r_{33}. of the rotation matrix are found from THLAM discussed in Chapter-3. There are two methods available for achieving this. They are discussed as follows:

METHOD– 1 [8]

R can be decomposed into the sum of a symmetric and a skew-symmeteric matrix.

Hence,

$$\mathbf{R} = \mathbf{S} + \mathbf{K} \qquad \text{(A1 .1)}$$

where

$$\mathbf{S}=\begin{bmatrix} n_1^2+(1-n_1^2)\cos\theta & n_1n_2(1-\cos\theta) & n_1n_3(1-\cos\theta) \\ n_1n_2(1-\cos\theta) & n_2^2+(1-n_2^2)\cos\theta & n_2n_3(1-\cos\theta) \\ n_1n_3(1-\cos\theta) & n_2n_3(1-\cos\theta) & n_3^2+(1-n_3^2)\cos\theta \end{bmatrix}$$

$$\mathbf{K}=\begin{bmatrix} 0 & -n_3.\sin\theta & n_2\sin\theta \\ n_3\sin\theta & 0 & -n_1\sin\theta \\ -n_2\sin\theta & n_1\sin\theta & 0 \end{bmatrix}$$

Using r_{11}, r_{12},... r_{33} we can write the **K** matrix as

$$\mathbf{K} = \frac{1}{2}\begin{bmatrix} 0 & r_{12}-r_{21} & r_{13}-r_{31} \\ r_{21}-r_{12} & 0 & r_{23}-r_{32} \\ r_{31}-r_{13} & r_{32}-r_{23} & 0 \end{bmatrix}$$

or

$$n_1.\sin\theta = (r_{23}- r_{32})/2 \qquad \text{(A1.2 a)}$$

$$n_2.\sin\theta = (r_{13}- r_{31})/2 \qquad \text{(A1.2 b)}$$

$$n_3.\sin\theta = (r_{21}- r_{12})/2 \qquad \text{(A1.2 c)}$$

therefore,

$$n_1 = \frac{(r_{23}-r_{32})}{2.\sin\theta} \qquad \text{(A1.3 a)}$$

$$n_2 = \frac{(r_{13}- r_{31})}{2.\sin\theta} \qquad \text{(A1.3 b)}$$

$$n_3 = \frac{(r_{21}- r_{12})}{2\ \sin\theta} \qquad \text{(A1.3 c)}$$

squaring and adding we get,

$$\sin^2\theta\ (\ n_1^2 + n_2^2 + n_3^2\ ) = [\ (r_{23}- r_{32})^2 + (r_{13}- r_{31})^2 + (r_{21}-r_{12})^2]\ /4$$

Let

$$d^2 = [\ (r_{23}- r_{32})^2 + (r_{13}- r_{31})^2 + (r_{21}-r_{12})^2]$$

therefore, since $n_1^2 + n_2^2 + n_3^2 = 1$

$$\sin\theta = \pm\ d/2 \qquad \text{(A1.4)}$$

Hence θ can be found to within a sign change.

$$r_{11} = n_1^2 + (1-n_1^2)\cos\theta$$

from which we write ,

$$\cos\theta = \frac{(r_{11} - n_1^2)}{1-n_1^2}$$

Substituting for n_1^2 we get,

$$\cos\theta = \frac{r_{11}.d^2 - (r_{23} - r_{32})^2}{d^2 - (r_{23} - r_{32})^2} \qquad \text{(A1.5)}$$

Thus θ can be uniquely found.

Having found θ, n_1 and n_2 can easily be found.

METHOD 2 [116]

$$r_{11} + r_{22} + r_{33} = (1 + 2.\cos\theta) \qquad \text{(A1.6)}$$

therefore,

$$\cos\theta = (r_{11} + r_{22} + r_{33} - 1)/2.0 \qquad \text{(A1.7)}$$

Also

$$r_{11} = n_1^2(1-\cos\theta) + \cos\theta \qquad \text{(A1.8)}$$

from which we get,

$$n_1 = \sqrt{\frac{(r_{11} - \cos\theta)}{(1-\cos\theta)}} \qquad \text{(A1.9a)}$$

$$n_2 = \frac{r_{12} + r_{21}}{2.n_1(1-\cos\theta)} \qquad \text{(A1.9b)}$$

$$n_3^2 = 1-(n_1^2 + n_2^2) \qquad \text{(A1.9c)}$$

$$\sin\theta = \frac{r_{23} - r_{32}}{2.n_1} \qquad \text{(A1.10)}$$

APPENDIX A –2

JACOBIAN MATRIX ELEMENTS OF EQUATION 8.36.

The Jacobian matrix is formed from the equations (8.20,8.21,8.22, 8.33,8.34 and 8.35). For simplicity of notation we define,

$$AC_1 = (m_{10} - A\, m_{00}) \quad \text{(A2 .1a)}$$

$$AC_2 = (m_{11} - A\, m_{01}) \quad \text{(A2 .1b)}$$

$$AC_3 = (m_{20} - A\, m_{10}) \quad \text{(A2 .1c)}$$

$$AC_4 = (m_{01} - B\, m_{00}) \quad \text{(A2 .1d)}$$

$$AC_5 = (m_{11} - B\, m_{10}) \quad \text{(A2 .1e)}$$

$$AC_6 = (m_{02} - B\, m_{01}) \quad \text{(A2 .1f)}$$

$$G_1 = p^2\, m_{20} + 2.p.q.m_{11} + q^2 m_{02} + 2.p.m_{10} + 2.q.m_{01} + m_{00} \quad \text{(A2 .2a)}$$

$$G_2 = p.m_{11} + q.m_{02} + m_{01} \quad \text{(A2 .2b)}$$

$$G_3 = p.m_{20} + q.m_{11} + m_{10} \quad \text{(A2 .2c)}$$

The following partial derivatives for the equation(8.20):

$$\frac{\partial F_1}{\partial R_3} = 2.AC_1 + 2.R_3.G_1 + 2 (-Apm_{10} + qm_{11} - Aqm_{01} + p\, m_{20}) + 2.R_{12}.G_2 \quad \text{(A2 .3a)}$$

$$\frac{\partial F_1}{\partial R_{12}} = 2.AC_2 + 2.R_3.G_2 + 2.R_{12}.m_{02} \quad \text{(A2 .3b)}$$

$$\frac{\partial F_1}{\partial R_6} = 0 \quad \text{(A2 .3c)}$$

$$\frac{\partial F_1}{\partial p} = 2.R_3AC_3 + 2.R_3^2.G_3 + 2.R_3.R_{12}.m_{11} \quad \text{(A2 .3d)}$$

$$\frac{\partial F_1}{\partial q} = 2.R_3AC_2 + 2.R_3^2.G_2 + 2.R_3.R_{12}.m_{02} \quad \text{(A2 .3e)}$$

The following are for equation (8.21)

$$\frac{\partial F_2}{\partial R_3} = AC_4 + p.AC_5 + q.AC_6 + R_6.G_1 - R_{12}.G_3 \quad \text{(A2 .4a)}$$

$$\frac{\partial F_2}{\partial R_{12}} = -AC_3 + AC_6 - R_3.G_3 + R_6.G_2 \quad \text{(A2 .4b)}$$

$$\frac{\partial F_2}{\partial R_6} = AC_1 + p.AC_3 + q.AC_2 + R_3.G_1 + R_{12}.G_2 \quad \text{(A2 .4c)}$$

$$\frac{\partial F_2}{\partial p} = R_3.AC_5 + R_6.AC_3 + 2.R_3.R_6G_3 + R_6.R_{12}.m_{11} - R_3.R_{12}.m_{20} \quad \text{(A2 .4d)}$$

$$\frac{\partial F_2}{\partial q} = R_3.AC_6 + R_6.AC_5 + 2.R_3.R_6G_2 + R_6.R_{12}.m_{02} - R_3.R_{12}.m_{11} \quad \text{(A2 .4e)}$$

The following are for equation (8.22)

$$\frac{\partial F_3}{\partial R_3} = 0 \quad \text{(A2 .5a)}$$

$$\frac{\partial F_3}{\partial R_{12}} = 2.AC_5 - 2.R_6.G_3 + 2.R_{12}.m_{02} \quad \text{(A2 .5b)}$$

$$\frac{\partial F_3}{\partial R_6} = 2.AC_4 + 2.R_6.G_1 + 2\,(p.AC_5 + q.AC_6) - 2.R_{12}.G_2 \quad \text{(A2 .5c)}$$

$$\frac{\partial F_3}{\partial p} = 2.R_6AC_5 + 2.R_6^2.G_3 - 2.R_6.R_{12}.m_{20} \quad \text{(A2 .5d)}$$

$$\frac{\partial F_3}{\partial q} = 2.R_6AC_6 + 2.R_6^2.G_2 - 2.R_6.R_{12}.m_{11} \quad \text{(A2 .5e)}$$

Similarly the partial derivatives for the other three equations (8.33, 8.34 and 8.35) can be derived. These partial derivatives form the matrix elements of the Jacobian matrix. In our case the Jacobian is a 6 X 5 matrix and is given below.

$$J_{ij}= \begin{bmatrix} \frac{\partial F_1}{\partial R_3} & \frac{\partial F_1}{\partial R_{12}} & \frac{\partial F_1}{\partial R_6} & \frac{\partial F_1}{\partial p} & \frac{\partial F_1}{\partial q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial F_6}{\partial R_3} & \frac{\partial F_6}{\partial R_{12}} & \frac{\partial F_6}{\partial R_6} & \frac{\partial F_6}{\partial p} & \frac{\partial F_6}{\partial q} \end{bmatrix} \qquad \text{(A2 .6)}$$

APPENDIX A –3

JACOBIAN MATRIX ELEMENTS FOR THE LeM EQUATION 9.21.

The expressions for the constants AC_1... AC_6 are the same as in Appendix A–2.

$$BC_1=a^2m_{20}+2abm_{11}+b^2m_{02}+2acm_{10}+2bcm_{01}+c^2m_{00} \quad \text{(A3.1a)}$$

$$BC_2=am_{11}+bm_{02}+cm_{01} \quad \text{(A3.1a)}$$

$$BC_3=am_{20}+bm_{11}+cm_{10} \quad \text{(A3.1a)}$$

$$BC_4=am_{10}+bm_{01}+cm_{00} \quad \text{(A3.1a)}$$

For the equation (9.4)

$$\frac{\partial F_1}{\partial r_{13}}= 2\, r_{13}BC_1+2(AC_3\, a +AC_2 b +AC_1 c)+2\, r_{12}BC_2 \quad \text{(A3.2a)}$$

$$\frac{\partial F_1}{\partial r_{12}}= 2\, r_{12}m_{02}+2\ \ r_{13}BC_2+2AC_2 \quad \text{(A3.2a)}$$

$$\frac{\partial F_1}{\partial r_{23}}= 0 \quad \text{(A3.2a)}$$

$$\frac{\partial F_1}{\partial a}= 2\, r_{13}^2BC_3+2AC_3\, r_{13}+2r_{12}r_{13}m_{11} \quad \text{(A3.2a)}$$

$$\frac{\partial F_1}{\partial b}= 2\, r_{13}^2BC_2+2AC_2\, r_{13}+2r_{12}r_{13}m_{02} \quad \text{(A3.2a)}$$

$$\frac{\partial F_1}{\partial c}= 2AC_1\, r_{13}+ 2\, r_{13}^2BC_4+2r_{12}r_{13}m_{01} \quad \text{(A3.2a)}$$

For equation (9.5)

$$\frac{\partial F_2}{\partial r_{13}} = r_{23}BC_1 - r_{12}BC_3 + AC_5a + AC_6b + AC_4c \qquad \text{(A3.3a)}$$

$$\frac{\partial F_2}{\partial r_{12}} = r_{23}BC_2 - r_{13}BC_3 + (AC_6 - AC_3) - 2\, r_{12}m_{11} \qquad \text{(A3.3a)}$$

$$\frac{\partial F_2}{\partial r_{23}} = 2r_{13}BC_1 + 2(AC_3a + AC_2b + AC_1c) + 2r_{12}BC_2 \qquad \text{(A3.3a)}$$

$$\frac{\partial F_2}{\partial a} = 2\, r_{13}r_{23}BC_3 + AC_5r_{13} + AC_3\, r_{23} - r_{12}r_{13}m_{20} + r_{12}r_{23}m_{11} \qquad \text{(A3.3a)}$$

$$\frac{\partial F_2}{\partial b} = 2\, r_{13}r_{23}BC_2 - r_{12}r_{13}m_{11} + AC_6\, r_{13} + AC_2r_{23} + r_{12}r_{23}m_{02} \qquad \text{(A3.3a)}$$

$$\frac{\partial F_2}{\partial c} = AC_4r_{13} + AC_1r_{23} + 2\, r_{13}r_{23}BC_4 - r_{12}r_{13}m_{10} + r_{12}r_{23}m_{01} \qquad \text{(A3.3a)}$$

for equation (9.6)

$$\frac{\partial F_3}{\partial r_{13}} = 0 \qquad \text{(A3.4a)}$$

$$\frac{\partial F_3}{\partial r_{12}} = 2\, r_{12}m_{20} - 2\, r_{23}BC_3 - 2AC_5 \qquad \text{(A3.4a)}$$

$$\frac{\partial F_3}{\partial r_{23}} = 2\, r_{23}BC_1 + 2AC_5\, a + 2AC_6b - 2\, r_{12}BC_3 \qquad \text{(A3.4a)}$$

$$\frac{\partial F_3}{\partial a} = 2\, r_{23}^2BC_3 + 2AC_5\, r_{23} - 2r_{12}r_{23}m_{20} \qquad \text{(A3.4a)}$$

$$\frac{\partial F_3}{\partial b} = 2\, r_{23}^2BC_2 + 2AC_6\, r_{23} - 2r_{12}r_{23}m_{11} \qquad \text{(A3.4a)}$$

$$\frac{\partial F_3}{\partial c} = 2\, r_{23}^2 BC_4 - 2\, r_{12} r_{23} m_{10} + 2\, r_{23} AC_4 \qquad \text{(A3.4a)}$$

$c' = (c - d.a)$

$$G_{11} = a^2 {}_c m_{20} + 2ab {}_c m_{11} + b^2 {}_c m_{02} + 2ac' {}_c m_{10} + 2G {}_c m_{01} + c'^2 {}_c m_{00} \qquad \text{(A3.5a)}$$

$$G_{12} = a {}_c m_{11} + b {}_c m_{02} + c' {}_c m_{01} \qquad \text{(A3.5a)}$$

$$G_{13} = a {}_c m_{20} + b {}_c m_{11} + c' {}_c m_{10} \qquad \text{(A3.5a)}$$

$$G_{14} = a {}_c m_{10} + b {}_c m_{01} + c' {}_c m_{00} \qquad \text{(A3.5a)}$$

for equation (9.15)

$$\frac{\partial F_4}{\partial r_{13}} = 2\, r_{13} G_{11} + 2(G_{13} - DG_{14}) + 2\, r_{12} G_{12} \qquad \text{(A3.6a)}$$

$$\frac{\partial F_4}{\partial r_{12}} = 2\, r_{12}\, {}_c m_{02} + 2({}_c m_{11} - D\, {}_c m_{01}) + 2\, r_{13} G_{12} \qquad \text{(A3.6a)}$$

$$\frac{\partial F_4}{\partial r_{23}} = 0 \qquad \text{(A3.6a)}$$

$$\frac{\partial F_4}{\partial a} = 2r_{13}^2 \left\{ a {}_c m_{20} + b {}_c m_{11} + (c - 2d.a) {}_c m_{10} - d(b\, {}_c m_{01} + c')\, {}_c m_{00} \right\} +$$
$$2\, r_{13}({}_c m_{20} - d\, {}_c m_{10} - D({}_c m_{10} - d\, {}_c m_{00})) + 2\, r_{12} r_{13}({}_c m_{11} - d\, {}_c m_{01}) \qquad \text{(A3.6a)}$$

$$\frac{\partial F_4}{\partial b} = 2r_{13}^2 \left\{ a {}_c m_{11} + b {}_c m_{02} + c' {}_c m_{01} \right\} + 2\, r_{13}({}_c m_{11} - D {}_c m_{01}) + 2 r_{12} r_{13} {}_c m_{02} \qquad \text{(A3.6a)}$$

$$\frac{\partial F_4}{\partial c} = 2r_{13}^2 \left\{ a {}_c m_{10} + b {}_c m_{01} + c' {}_c m_{00} \right\} + 2\, r_{13}({}_c m_{10} - D {}_c m_{00}) + 2 r_{12} r_{13} {}_c m_{01} \qquad \text{(A3.6a)}$$

for equation (9.18)

$$\frac{\partial F_5}{\partial r_{13}}=(G_{12}-EG_{14})-r_{12}G_{13}+r_{23}G_{11} \tag{A3.7a}$$

$$\frac{\partial F_5}{\partial r_{12}}=(-{}_cm_{20}+{}_cm_{02}+D{}_cm_{10}-E{}_cm_{01})-r_{13}G_{13}+r_{23}G_{12} \tag{A3.7a}$$

$$\frac{\partial F_5}{\partial r_{23}}=(G_{13}-DG_{14})+r_{12}G_{12}+r_{13}G_{11} \tag{A3.7a}$$

$$\begin{aligned}\frac{\partial F_5}{\partial a}=&r_{13}({}_cm_{11}-E{}_cm_{10}-d{}_cm_{01}+Ed{}_cm_{00})+r_{23}({}_cm_{20}-d{}_cm_{10}-D{}_cm_{10}+Dd{}_cm_{00})\\&+r_{12}r_{13}(-{}_cm_{20}+d{}_cm_{10})+2r_{13}r_{23}\Big(a{}_cm_{20}+b{}_cm_{11}\\&+(c-2ad){}_cm_{10}-d\{b{}_cm_{01}+c'{}_cm_{00}\}\Big)\end{aligned} \tag{A3.7a}$$

$$\frac{\partial F_5}{\partial b}=r_{13}({}_cm_{02}-E{}_cm_{01})+2r_{12}r_{23}G_{12}+r_{23}({}_cm_{11}-D{}_cm_{01})+r_{13}r_{12}(-{}_cm_{11})$$
$$+r_{12}r_{23}(b{}_cm_{02}) \tag{A3.7a}$$

$$\begin{aligned}\frac{\partial F_5}{\partial c}=&r_{23}({}_cm_{10}-D{}_cm_{00})+r_{13}({}_cm_{01}-E{}_cm_{00})+r_{12}r_{13}(-{}_cm_{10})+r_{13}r_{23}G_{14}+\\&r_{12}r_{23}{}_cm_{01}\end{aligned} \tag{A3.7a}$$

for equation (9.21)

$$\frac{\partial F_6}{\partial r_{13}}= 0 \tag{A3.8a}$$

$$\frac{\partial F_6}{\partial r_{12}}=2\ r_{12}{}_cm_{20}-2\ r_{23}G_{13}-2({}_cm_{11}-E{}_cm_{10}) \tag{A3.8a}$$

$$\frac{\partial F_6}{\partial r_{23}}=-2\ r_{12}G_{13}+2r_{23}G_{11}+2(G_{12}-EG_{14}) \tag{A3.8a}$$

$$\frac{\partial F_6}{\partial a}=2r_{23}({}_cm_{11}-E_cm_{10}-d_cm_{01}+Ed_cm_{00})-2r_{12}r_{23}({}_cm_{20}-d_cm_{10})$$
$$+2r_{23}^2\left\{a_cm_{20}+b_cm_{11}+(c-2ad)_cm_{10}-d(b_cm_{01}+c'_cm_{00})\right\} \qquad \text{(A3.8a)}$$

$$\frac{\partial F_6}{\partial b}=2r_{12}r_{23}(-{}_cm_{11})+2r_{23}^2\ G_{12}+2r_{23}({}_cm_{02}-E_cm_{01}) \qquad \text{(A3.8a)}$$

$$\frac{\partial F_6}{\partial c}=2r_{12}r_{23}(-{}_cm_{10})+2r_{23}^2G_{12}+2r_{23}({}_cm_{02}-E_cm_{01}) \qquad \text{(A3.8a)}$$

REFERENCES

[1] S.W.ZUCKER, " Vision, Early ", in *Encyclopaedia of Artificial Intelligence,* Vol.2, STUART.C.SHAPIRO Ed., John Wiley & sons,Inc. Newyork,1987.

[2] I.OVERINGTON, " Vision and Acquisition: Fundamentals of human visual performance , environmental influences and applications in instrumental optics", Pentech press, London,1976.

[3] M.M.TRIVEDI and A.ROSENFELD," On making computers "SEE" ", IEEE Transactions on Systems , Man and Cybernetics, Vol. 19, No.6, pp. 1333– 1335, Nov./Dec. 1989.

[4] T.S.HUANG," *Image sequence analysis* ", Springer verlag,Berlin, 1981.

[5] T.S.HUANG, " *Image sequence processing and Dynamic scene analysis*", Springer Verlag, Berlin, 1983.

[6] H.H.NAGEL, " Image sequences–Ten(octal) years–From Phenomenology towards a theoretical foundation ", in 8^{th} International Conference on Pattern Recognition,(Paris, FRANCE), Oct. 27–31, 1986, pp. 1174–1185.

[7] J.K.AGGARWAL and N.NANDHAKUMAR, " On the computation of motion from sequences of images – A review ", Proceedings of the IEEE, Vol.76, No.8, pp. 917–935, Aug. 1988.

[8] R.Y.TSAI and T.S.HUANG," Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surface", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–6, No.1 , pp. 13–26, Jan. 1984.

[9] J.K.AGGARWAL and R.O.DUDA ," Computer analysis of moving polygonal images ", IEEE Transactions on Computer Vol. C–20, No.10, pp. 966–976, Oct. 1975.

[10] J.W.ROACH and J.K.AGGARWAL ," Computer tracking of objects moving in space ", IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI–1, No.2, pp. 127–135, Mar. 1979.

[11] _____," Determining the movement of objects from a sequence of images ", IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI–2, No.6, pp. 554–562, Nov. 1980.

[12] H.H. NAGEL," Representation of moving rigid objects based on visual observations ", IEEE Computer, Vol. 14, No. 8, pp. 29–39, Aug. 1981.

[13] R.Y.TSAI and T.S.HUANG, " Motion parameter estimation from rigid planar patch ", IEEE Transactions on Acoustics, Speech and Signal Processing,Vol. ASSP–29, No.6, pp.1147–1152, Jun.1981.

[14] J.Q. FANG and T.S.HUANG, " Solving three dimensional small –rotation motion equations ", in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (Washington,DC),June 19–23,1983, pp. 253–256.

[15] _____ , " Solving three dimensional small rotation motion equations : uniqueness , algorithms and numerical results, Computer Vision, Graphics,Image Processing,Vol.26, pp. 183–206, 1984.

[16] T.S.HUANG, " Three dimensional motion analysis by direct matching", Journal of Optical Society of America –A Vol.3, No.9, pp.1501–1503, Sept. 1986.

[17] H.C.LONGUET–HIGGINS, " A computer algorithm for reconstructing a scene from two projections ", Nature, Vol. 293, pp. 133– 155, 1981.

[18] H.C.LONGUET–HIGGINS, " The reconstruction of a scene from two projections: Configurations that defeat the 8–point algorithm ", in Proceedings of the First Conference on Artificial Intelligence Applications, (Denver, CO), Dec. 5–7, 1984, pp. 395–397.

[19] R.Y.TSAI , T.S.HUANG, and W.L.ZHU, " Estimating three – dimensional motion parameters of a rigid planar patch II: Singular Value Decomposition ",IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP–30, No.4 , pp. 525–534, Apr. 1982.

[20] R.Y.TSAI , and T.S.HUANG , " Correction to " Estimating three – dimensional motion parameters of a rigid planar patch II: Singular Value Decomposition ", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP–31, No.2 , pp. 514, Feb. 1983.

[21] _____ , " Estimating three dimensional motion parameters of a rigid planar patch, III: Finite point correspondences and three –view problem ", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP–32, No.2, pp. 213–219, Feb. 1984.

[22] R.Y.TSAI, " Estimating 3–D motion parameters and object surface structures from the image motion of curved edges", in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (Washington,DC),June 19–23,1983, pp. 259–266.

[23] J.Q.FANG and T.S.HUANG, " Some experiments on estimating the 3–D motion parameters of a rigid body from two consecutive image frames ", IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI–6,No.5, pp.545–553, Sept. 1984.

[24] X.ZHUANG, R.M.HARALICK and T.S.HUANG, " Two–view motion analysis : A unified approach ", Journal of Optical Society of America –A Vol.3, No.9, pp.1492–1500, Sept. 1986.

[25] X.ZHUANG and R.M.HARALICK, " Two view motion analysis ", in Proceedings of IEEE Computer Vision and Pattern Recognition (San Francisco, CA), June 19–23, 1985, pp. 686 – 690.

[26] O.D.FAUGERAS , F.LUSTMAN and G.TOSCONI, " Motion and structure from point and line matches" , First International conference on Computer Vision (London, U.K.,), June 8–11, 1987, pp. 25–34.

[27] G.TOSCANI and O.D.FAUGERAS," Structure from Motion using the Reconstruction & Reprojection technique on noisy and incorrect point matches ", Proceedings of the International Symposium on Signal Processing Applications, (Brisbane,AUSTRALIA) Aug. 24–28, 1987. (Pre copy)

[28] B.L.YEN and T.SHUANG," Determining 3–D motion/structure of a rigid body over 3 frames using straight line correspondences ", in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (Washington,DC),June 19–23,1983, pp. 267–272.

[29] A.MITICHE, S.SEIDA and J.K.AGGARWAL, " Determining position and displacement in space from images", in Proceedings of IEEE Computer Vision and Pattern Recognition (San Francisco, CA), June 19–23, 1985, pp. 504–509.

[30] A.MITICHE, S.SEIDA and J.K.AGGARWAL , " Using constancy of distance to estimate position and orientation in space ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–10 , No.4 pp. 594–599, Jul. 1988.

[31] _____," Line based computation of structure and motion using angular invariance ", in IEEE Computer Society Workshop on Motion representation and analysis, (South Carolina, Washington), May 7–9,1986, pp. 175–180.

[32] Y.LIU and T.S.HUANG, " Estimation of rigid body motion from line correspondences ", in IEEE Computer Society Workshop on Motion representation and analysis, (South Carolina, Washington), May 7–9,1986, pp. 47–52 .

[33] _____, " Estimation of rigid body motion using straight line correspondences ", Computer Vision,Graphics, and Image Processing, Vol.43, pp. 37–52,1988.

[34] _____, " Estimation of rigid body motion using line correspondences ", Computer Vision,Graphics, and Image Processing, Vol.44, pp. 35–57,1988.

[35] M.E.SPETSAKIS and J.Y. ALOIMONOS, " Closed form solution to the structure and motion problem from line correspondences ", in Proceedings of the Sixth AAAI National conference on Artificial Intelligence Vol.2, (Seattle, Washington), July 13–17, 1987, pp. 738–743.

[36] K.E.PRICE and R.REDDY, " Change detection and analysis in multi spectral images ", Fifth International Joint conference on Artificial Intelligence,Vol.2, (MIT, Massachusetts,U.S.A),Aug. 22–25,1977, pp. 619–625.

[37] T.Y.YOUNG and Y.L.WANG, " Analysis of three dimensional rotation and linear shape changes " Pattern recognition letters, Vol. PR–2, pp. 239–242, 1984.

[38] G.R.LEGTERS and T.Y.YOUNG, " A mathematical model for image tracking ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–4, No.6, pp. 583–594, Nov. 1982.

[39] S.GUNASEKARAN and T.Y.YOUNG, " A Region correspondence approach to the recovery of 3–Dimensional motion and structure in Dynamic scenes", a Chapter in Image Understanding in Unstructured Environment, S.CHEN Ed., World Science Series, 1987.
(Authors pre copy.)

[40] T.J. BROIDA and R.CHELLAPPA, " Estimation of object motion parameters from noisy images ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–8, No.1, pp. 90–99, Jan. 1986.

[41] _____ , " Kinematics and structure of a rigid object from a sequence of noisy images", in IEEE Computer Society Workshop on Motion representation and analysis, (South Carolina, Washington),May 7–9,1986, pp. 95–100.

[42] H.SHARIAT and K.E.PRICE," Motion estimation: the proper formulation when 3 or 4 frames are available", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (Miami Beach, U.S.A), June 22–26, 1986, pp. 396 – 399.

[43] H.SHARIAT," The Motion problem: How to use more than two frames", Ph.D Dissertation , University of Southern California, Department of Electrical Engineering, Los Angeles,CA,1986.

[44] G.TSENG and A.K.SOOD, " Analysis of long sequence for structure and motion estimation ", IEEE Transactions on System , Man and Cybernetics, Vol. SMC–19, No.6, pp. 1511–1526, Nov./Dec. 1989.

[45] B.K.P.HORN and B.G.SCHUNK, " Determining Optical flow ", Artificial Intelligence, Vol. 17, pp. 185–203, 1981.

[46] G.ADIV, " Determining three – dimensional motion and structure from optical flow generated from several moving objects ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–7 , No.4, pp.384–401, Jul. 1985.

[47] _____, " Inherent ambiguities in recovering 3–D motion and structure from noisy flow field", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–11 , No.5, pp.477–489, May 1989.

[48] A.M.WAXMAN and K.WOHN, " Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion ", International Journal of Robotics research, Vol.3, pp 95–108, 1985.

[49] A.MITICHE, " Computation of optical flow and rigid motion ", Proceedings of the workshop on Computer Vision Representation and Control,(Annapolis, Maryland, USA) April 30–May 2, 1984, pp. 63–71.

[50] _____ , " Three dimensional space from optical flow correspondence ", Computer Vision,Graphics,Image Processing, Vol.42, pp. 306–317,1988.

[51] K.KANATANI,“ Structure and motion from Optical flow under orthographic projection ”, Computer Vision,Graphics,Image Processing, Vol.35, pp. 181–199,1986.

[52] _____ ,“ Structure and motion from Optical flow under perspective projection ”, Computer Vision,Graphics,Image Processing, Vol.38, pp. 122–146,1987.

[53] B.K.P.HORN, “ *Robot Vision* ”,MIT Cambridge, Massachusetts,1986.

[54] T.S.HUANG and Y.S.SHIM,“ Linear algorithm for motion estimation : How to handle degenerate cases.”, 4th International conference on Pattern Recognition,(Oxford,UK), Mar. 28–30, 1988, pp.439–447.

[55] T.S.HUANG and O.D.FAUGERAS,“ Some properties of the E matrix in Two–view motion estimation”, IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. PAMI–11, No.12, pp. 1310–1312, Dec.1989.

[56] H.P.TRIVEDI,“ Estimation of stereo and motion parameters using a variational principle”, Image and Vision Computing, Vol. 5, No.2, pp. 181–183, May 1987.

[57] YOSHIO YASUMOTO and GERAD MEDIONI,“ Robust estimation of 3–D motion parameters from a sequence of image frames using regularization”, IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. PAMI–8, No.4, pp. 464–471, Jul. 1986.

[58] “ *Reflex Robot, AR–BASIC Programming manual* ” Rediffusion , Robot systems, American Robot corporation, 1985.

[59] G.A.HORN and T.M. HORN, “ *Mathematical Handbook for scientists and engineers* ”, Mc.Graw–Hill book company, Newyork,1968.

[60] J.K AGGARWAL and A.MITICHE, " Structure and Motion from images: fact and fiction ", Proceedings of the third Workshop on Computer Vision: Representation and Control, (Bellair,MI) Oct. 13–16, 1985, pp.127–128.

[61] H.P.MORAVEC, " Towards automatic visual obstacle avoidance ", Fifth International Joint conference on Artificial Intelligence,Vol.2, (MIT, MA., USA), Aug. 22–25,1977,pp. 584.

[62] _____ , " Visual mapping by robot rover ", Sixth International Joint conference on Artificial Intelligence,(Tokyo, JAPAN), Aug. 20–23,1979, pp. 598–600.

[63] L.DRESCHLER and H.H.NAGEL, " Volumetric Model and 3–D trajectory of a moving car derived from monocular TV frame sequences of a street scene", Computer Vision,Graphics,Image Processing,Vol.20, pp. 199–228,1982.

[64] J.H.Mc INTOSCH and K.M.MUTCH," Matching straight lines ", Computer Vision, Graphics, Image Processing,Vol.43, pp. 386–408,1988.

[65] K.E.PRICE," Matching closed contours ", IEEE Workshop on Computer Vision Representation and Control, (Annapolis, Maryland, USA) April 30–May 2, 1984,pp. 130–134.

[66] P.R. BEAUDET," Rotationally invariant image operators ",International Joint conference on Pattern Recognition, (Kyoto, Japan), Nov. 7–10,1978, pp. 579–583.

[67] R.M. HARALICK," The digital edge ", IEEE Computer Society Conference on Pattern Recognition and Image Processing,(Dallas,Texas,U.S.A), Aug. 3–5, 1981, pp. 285–291.

[68] O.ZUNIGA and R.M HARALICK , " Integrated directional derivative gradient operator ", IEEE transactions on Systems , Man and Cybernetics, Vol. SMC–17, No.3 , pp. 508–517, May/June 1987.

[69] M.A.SHAH, and R.JAIN, " Detecting time–varying corners ", IEEE International Conference on Pattern Recognition, (Montreal CANADA) Jul. 30–Aug. 2, 1984,Vol.1, pp.2–5.

[70] L.KITCHEN and A.ROSENFELD, " Gray level corner detection ", TR 887, Technical Report, University of Maryland, 1980.

[71] J.Q.FANG and T.S.HUANG, " A corner finding algorithm for image analysis and registration" Proceedings of National Conference on Artificial Intelligence, AAAI–82, (Pittsburgh,Pennsylvania,U.S.A), Aug. 18-20, 1982, pp. 46-49.

[72] O.ZUNIGA and R.M HARALICK ," Corner detection using the facet model ", Proceedings IEEE conference on Computer Vision and Pattern Recognition,(Washington,DC),June 19–23,1983, pp. 30-37.

[73] S.RAGUPATHI and R.A.KING, " Time – varying corners: a comparison and performance study ", Signal Processing IV :Theories and applications. Ed. J.L.Lacoume et al., pp. 1453–1456, EURASIP –88.

[74] R.C.GONZALEZ and P. WINTZ, *" Digital Image processing "*, Addison Wesley, Massachusetts, U.S.A, 1977.

[75] J.F.CANNY, " A computational approach to edge detection ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–8, No.6, pp. 679–698, Nov. 1986.

[76] G.MEDIONIand YASUMOTO ," Corner detection and curve representation using cubic B–splines ", Computer Vision, Graphics,Image Processing, Vol.39, pp. 267–278,1987.

[77] Z.Q.WU and A.ROSENFELD, " Filtered projections as an aid in corner detection ", Pattern Recognition , Vol. 16, No.1, pp. 31–38, 1983.

[78] K.PALER, J.FOGLEIN, J.KITTLER et al., " Local ordered gray levels as an aid to corner detection ", Pattern Recognition , Vol. 17, No.5, pp. 535–543, 1984.

[79] K.RANGARAJAN, M.SHAH and D.V.BRACKLE, " Optimal corner detector", Computer Vision, Graphics,Image Processing,Vol.20, pp. 199–228, 1988.

[80] J.R.ULLMAN , *" Interpretation of visual motion "*, A doctoral thesis, MIT,1979.

[81] J.K.AGGARWAL , L.S.DAVIS and W.N. MARTIN ," Correspondence processes in dynamic scene analysis ", Proceedings of the IEEE, Vol. 69, No.5, pp. 562–572, May 1981.

[82] A.ROSENFELD and A.C.KAC , *" Digital Picture Processing, Vol. 2 "*, Academic Press, Newyork, 1982.

[83] F.GLAZER, G.REYNOLDS and P.ANANDAN, " Scene matching by hierarchical correlation", IEEE conference on Computer Vision and Pattern Recognition,(Washington,DC),June 19–23,1983, pp. 432–441.

[84] A.ROSENFELD, R.HUMMEL and S.W.ZUCKER," Scene labelling by relaxation operations ", IEEE Transaction Systems Man Cybernetics, Vol. SMC–6, No.6, pp. 420–433, June 1976.

[85] A.ROSENFELD," Relaxation methods in Image processing and analysis ", Proceedings of the 4th International Joint Conference on Pattern Recognition, (Kyoto, JAPAN), Nov. 7–10, 1978, pp. 181–185.

[86] E.R.HANCOCK and J.KITTLER, " Edge-Labeling using dictionary- based relaxation" , IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI–12, No.2 pp. 165–181, Feb.1990.

[87] S.T. BARNARD and W.B.THOMSON, " Disparity Analysis of Images ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–2, No.4, pp. 333–340, July 1980.

[88] S.RANADE and A.ROSENFELD, " Point pattern matching ", Pattern Recognition , Vol. 12, No.5, pp. 269–275,1980.

[89] M.C.IBISON and L.ZAPALOWSKI ," On the use of relaxation labelling in the correspondence problem ", Pattern Recognition Letters , Vol. 4, pp. 103–109,1986.

[90] I.K.SETHI and R.C.JAIN, " Finding trajectories of Feature points in a monocular Image sequence ",IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–9, No.1, pp. 56–73,Jan. 1987.

[91] C.Y.WANG and H.SUN et al " Some experiments in relaxation matching using corner features ", Pattern Recognition , Vol. 16, No.2, pp. 167 –182, 1983.

[92] D.B.GOLDOF, H.LEE and T.S.HUANG, " Motion analysis of non–rigid surfaces ", IEEE conference on Computer Vision and Pattern Recognition,(Ann Arbor, MI) June 5–8 ,1988, pp. 375–380.

[93] T.POGGIO and S.EDELMAN, " A network that learns to recognize three–dimensional objects ", Nature, Vol. 343, pp. 263–266, 18 th Jan. 1990,

[94] M.K. HU , " Visual pattern recognition by moment invariants ", IRE Transactions on Information Theory, Vol. IT–8, No.2, pp. 179–187, Feb. 1962.

[95] E.B.ELLIOT, *"Algebra of quantics"*, Oxford University. Press, Newyork,Ny 2nd Ed.,1913.

[96] S.A. DUDANI, K.J.BREEDING and R.B.McGHEE," Aircraft identification by Moment invariants ", IEEE Transactions on Computers, Vol. C–26, No. 1, pp. 39–45, Jan. 1977.

[97] R.Y.WONG and E.L.HALL, " Scene matching with Invariant moments", Computer Vision,Graphics,Image Processing,Vol.8, pp. 16–24 ,1978.

[98] S. MAITRA, " Moment Invariants", Proceedings IEEE, Vol. 67, No. 4, pp. 697–699, Apr. 1979.

[99] M.R.TEAGUE," Image analysis via the general theory of moments", Journal of Optical Society of America, Vol. 70, No. 8, pp. 920–930, Aug.1980.

[100] S.S. REDDI ," Radial and Angular Moment Invariants for image identification ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–3, No. 2, pp. 240–242, Mar. 1981.

[101] A.P.REEVES and A.ROSTAMPOUR, " Shape analysis of segmented objects using moments ", IEEE conference on Pattern Recognition and Image Processing,(Dallas,Texas,U.S.A), Aug. 3–5, 1981, pp. 171–174.

[102] A.P.REEVES, R.J. PROKOP et al., " Three dimensional shape analysis using moments and Fourier descriptors ", Proceedings of the 7^{th} International conference on Pattern Recognition ,(Montreal, CANADA), Jul. 30–Aug. 2, 1984, pp. 447–450.

[103] ABU–MOSTAFA and PSALTIS," Recognitive aspects of moment invariants ", IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI–6, No.6, pp. 698–706, Nov. 1984.

[104] P.O.OGUNBONA, *" Estimation of motion for movement compensated image processing "*, Ph.D Thesis, Imperial College, University of London,1987.

[105] D.CYGANSKI and J.A.ORR," Applications of Tensor Theory to Object Recognition and orientation determination ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–7, No.6, pp. 662–673, Nov. 1985.

[106] B.BAMIEH and R.J.P.DE FIGUEIREDO, " A General Moment Invariants/ Attributed –Graph for three dimensional object recognition from a single image", IEEE Journal of Robotics and Automation, Vol. RA–2, No.1, pp. 31–41, Jan. 1986.

[107] C.H.TEH and R.T.CHIN ," On digital approximation of moment invariants ", Computer Vision, Graphics, Image Processing, Vol.33, pp. 318– 326, 1986.

[108] _____ , " Image analysis by methods of moments ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–10, No. 4 , pp. 496 – 513, Jul. 1988.

[109] F.A.SADJADI and E.L. HALL, " Three –Dimensional Moment invariants ",IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–2, No.2, pp. 127–136, Mar. 1980.

[110] C.H.LO and H.S. DON, " 3–D moment forms : their construction and application to object identification and positioning ",IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–11, No.10, pp. 1053–1064, Oct. 1989.

[111] T.L.FABER and E.M.STOKELY, " Orientation of 3–D structures in Medical Images ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI–10, No.5, pp. 626–633, Nov. 1988.

[112] F.ZAKARIA , L.J.VROOMEN et al, " Fast algorithm for the computation of moment invariants ", Pattern Recognition, Vol. PR–20, No.6 , pp. 639–643, 1987.

[113] T.S.HUANG, " Motion analysis ", in *Encyclopaedia of Artificial Intelligence,* STUART.C.SHAPIRO Ed.,Vol.2, John Wiley & sons,Inc. Newyork,1987.

[114] K.M.BROWN and J.E.DENNIS, " Derivative free analogues of Levinberg–Marquardt and Gaussian algorithms for non–linear least squares ", Numer. Math, Vol.18, pp. 289–297,1972 .

[115] M.BERTERO, T.A. POGGIO and V.TORRE," Ill–posed problems in early vision ", Proceedings of the IEEE, Vol.76, No.8, pp. 869–889, Aug. 1988.

[116] A. MITICHE and P.BOUTHEMY , " Tracking modelled objects using binocular images", Computer Vision, Graphics, Image Processing, Vol.32, pp. 384– 396, 1985.