

EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM  
USING A TARGET WITH KNOWN DYNAMICS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT DENİZ AYKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2008

Approval of the thesis:

**EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM  
USING A TARGET WITH KNOWN DYNAMICS**

submitted by **MURAT DENİZ AYKIN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İsmet Erkmen  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Assist. Prof. Dr. Afşar Saranlı  
Supervisor, **Electrical and Electronics Engineering Dept., METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. Mübeccel Demirekler  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assist. Prof. Dr. Afşar Saranlı  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Prof. Dr. Gözde Bozdağı Akar  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assoc. Prof. Dr. Aydın Alatan  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assist. Prof. Dr. Buğra Koku  
Mechanical Engineering Dept., METU \_\_\_\_\_

**Date:**

**27.08.2008**

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: MURAT DENİZ AYKIN

Signature :

# ABSTRACT

## EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM USING A TARGET WITH KNOWN DYNAMICS

Aykın, Murat Deniz

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assist. Prof. Dr. Afşar Saranlı

August 2008, 86 pages

Multi camera measurement systems are widely used to extract information about the 3D configuration or “state” of one or more real world objects. Camera calibration is the process of pre-determining all the remaining optical and geometric parameters of the measurement system which are either static or slowly varying. For a single camera, this consist of the internal parameters of the camera device optics and construction while for a multiple camera system, it also includes the geometric positioning of the individual cameras, namely “external” parameters. The calibration is a necessary step before any actual state measurements can be made from the system. In this thesis, such a multi-camera state measurement system and in particular the problem of procedurally effective and high performance calibration of such a system is considered.

This thesis presents a novel calibration algorithm which uses the known dynamics of a ballistically thrown target object and employs the Extended Kalman Filter (EKF) to calibrate the multi-camera system. The state-space representation of the target state is augmented with the unknown calibration parameters which are assumed to be static or slowly varying with respect

to the state. This results in a “super-state” vector. The EKF algorithm is used to recursively estimate this super-state hence resulting in the estimates of the static camera parameters. It is demonstrated by both simulation studies as well as actual experiments that when the ballistic path of the target is processed by the improved versions of the EKF algorithm, the camera calibration parameter estimates asymptotically converge to their actual values. Since the image frames of the target trajectory can be acquired first and then processed off-line, subsequent improvements of the EKF algorithm include *repeated* and *bidirectional* versions where the same calibration images are repeatedly used. Repeated EKF (R-EKF) provides convergence with a limited number of image frames when the initial target state is accurately provided while its bidirectional version (RB-EKF) improves calibration accuracy by also estimating the initial target state.

The primary contribution of the approach is that it provides a fast calibration procedure where there is no need for any standard or custom made calibration target plates covering the majority of camera field-of-view. Also, human assistance is minimized since all frame data is processed automatically and assistance is limited to making the target throws. The speed of convergence and accuracy of the results promise a field-applicable calibration procedure.

Keywords: Calibration and Identification, Visual Tracking, Dynamics.

# ÖZ

## ÇOK KAMERALI BİR ÖLÇÜM SİSTEMİNİN BİLİNER DİNAMİĞİ OLAN BİR HEDEF İLE KALİBRE EDİLMESİ

Aykın, Murat Deniz

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Afşar Saranlı

Ağustos 2008, 86 sayfa

Çok kameralı sistemler bir veya birden çok objenin 3 boyutlu konfigürasyonu ya da “durum” kestirimi ile ilgili bilgi elde etmek amacıyla sıklıkla kullanılmaktadır. Kamera kalibrasyonu statik ya da yavaş değişen optik ve geometrik parametrelerin önceden belirlenmesidir. Kalibrasyon tek bir kamera için, kameranın yapımı ve optiği ile ilgili iç parametrelerin belirlenmesine karşılık gelirken, çok kameralı bir sistemde kameraların birbirlerine göre pozisyonlarının kestirimini, yani “dış” parametre kestirimini de içermektedir. Ölçüm sistemi ile durum kestirimi yapabilmek için kalibrasyon gerekli ve önemli bir adımdır. Bu tezde çok kameralı bir ölçüm sistemi ile bu sistemin yöntemsel olarak etkin ve yüksek performanslı kalibrasyonu problemi incelenmiştir.

Bu tez kapsamında balistik olarak fırlatılan bir objenin bilinen dinamiğinin genişletilmiş Kalman Filtresi (EKF) yardımıyla çok kameralı bir sistemin kalibrasyonu için kullanılması önerilmiştir. Hedef objenin dinamiğinin durum-uzayı gösteriminin bilinmeyen statik ya da yavaş değişen kamera parametreleri ile genişletilmesiyle “süper durum vektörü” elde edilir. EKF durum kestirme algoritması süper durum vektörünün tahmin edilmesi için kullanılmakta

ve böylece bilinmeyen kamera parametreleri de hesaplanmaktadır. Yapılan simülasyon çalışmaları ve gerçek deneyler hedefin ballistik yörüngesinin önerilen iki yeni EKF algoritması ile işlenmesi sonucunda kamera kalibrasyon parametrelerinin gerçek değerlerine sonuçta kararlı bir şekilde yakınsadığını göstermiştir. Kalibrasyon için gerekli olan görüntüler elde edildikten sonra çevirim dışı bir şekilde işlenmeleri mümkün olduğundan, aynı görüntülerin defalarca kullanıldığı *özyinelemeli ve iki yönlü* EKF versiyonları önerilmiştir. Özyinelemeli EKF (R-EKF) hedef objenin ilk durum vektörü doğru olarak verildiğinde parametrelerin sınırlı bir sayıda görüntü karesi kullanılarak yakınsamasını sağlarken iki yönlü özyinelemeli EKF (RB-EKF) ise parametre kestirimini yanı sıra ilk durum vektörünün de kestirimini gerçekleştirdiğinden daha doğru sonuç elde edilmesini sağlamaktadır.

Önerilen algoritmanın en büyük katkısı önceden hazırlanmış desenli bir plakanın kameraların çoğunluğunun görüş alanını kapsayacak şekilde hizalanması gerekmeden hızlı ve kolay uygulanabilir bir kalibrasyon prosedürü sağlamasıdır. Ayrıca kalibrasyon karelerinin otomatik işlenmesi, insan desteğinin sadece hedef fırlatma ile sınırlanmasını sağlamıştır. Doğru ve hızlı bir şekilde sonuç üretilmesi algoritmanın sahada uygulanabilir bir yöntem olduğunu göstermiştir.

Anahtar Kelimeler: Kalibrasyon ve Tanımlama, Görsel İzleme, Dinamik Hareket.

*To my loving mother  
and to the memory of my beloved father*

## ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my supervisor, Professor Afşar Saranlı for his personal guidance, encouragement, constructive comments and for his outstanding support throughout this work. Special thanks to Professor Uluç Saranlı for the inspirational ideas and productive discussions during the course of this research.

I owe my most sincere gratitude to Professor Gözde Bozdağı Akar for her incredible understanding, hospitality and continuous support.

I wish to express my warm and sincere thanks to Mehmet Oğuz Bici, Anıl Aksay, Emrah Bala, Cem Vedat Işık, Berkan Solmaz and all colleagues from Multi Media Research Group, for their suggestions and technical supports and also for the very pleasant working environment.

I also wish to thank Cevahir Çıgla, Elif Vural, Engin Türetken, Eren Halıcı and Osman Serdar Gedik for their valuable friendship, entertainment and availability to help me whenever it was necessary.

During this work I have collaborated with many colleagues for whom I have great regard, and I wish to extend my warmest thanks to all those who have helped me with my work in the RHex Research Group.

Lastly I owe my loving thanks to my parents, Emel İnal and Kerim Serdar Aykın. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

# TABLE OF CONTENTS

ABSTRACT . . . . .	iv
ÖZ . . . . .	vi
DEDICATION . . . . .	viii
ACKNOWLEDGMENTS . . . . .	ix
TABLE OF CONTENTS . . . . .	x
LIST OF TABLES . . . . .	xiii
LIST OF FIGURES . . . . .	xiv
LIST OF ABBREVIATIONS . . . . .	xviii
CHAPTERS	
1 Introduction . . . . .	1
1.1 General . . . . .	1
1.2 Scope and Contribution of the Thesis . . . . .	3
1.3 Outline of the Dissertation . . . . .	4
2 Background on Camera Calibration . . . . .	5
2.1 Internal Camera Calibration . . . . .	6
2.2 External Camera Calibration . . . . .	9
2.3 Object Detection . . . . .	11
2.4 Links between Calibration and State Estimation . . . . .	12
3 The Extended Kalman Filter . . . . .	14
3.1 General . . . . .	14
3.2 Overview of the EKF Algorithm . . . . .	15
3.3 Simultaneous State and Parameter Estimation . . . . .	17
3.3.1 The SSPE Algorithm . . . . .	17

	3.3.2	An Example on SSPE . . . . .	18
	3.3.3	An Example on Simultaneous Localization and Mapping . . . . .	20
4		Simultaneous Tracking and Camera Calibration . . . . .	22
	4.1	Ballistic Motion and Camera Measurement Models . . . . .	22
	4.1.1	Ballistic Motion Model . . . . .	23
	4.1.2	The Pinhole Camera Model . . . . .	24
	4.2	Raw measurements: Detection of the Target . . . . .	25
	4.3	Definition of Super-State . . . . .	27
	4.3.1	Augmented System Representation . . . . .	28
	4.3.2	Augmented System Equations . . . . .	30
5		Estimating the Super-State . . . . .	32
	5.1	Standard EKF Algorithm . . . . .	33
	5.2	Repeated EKF Algorithm . . . . .	34
	5.3	Repeated Bidirectional EKF Algorithm . . . . .	36
6		Simulation Experiments . . . . .	38
	6.1	Experiment 1: R-EKF / Free Fall Motion . . . . .	38
	6.2	Experiment 2: RB-EKF / Free Fall Motion . . . . .	42
	6.3	Experiment 3: Comparison of R-EKF and RB-EKF . . . . .	43
	6.4	Experiment 4: RB-EKF / Ballistic Motion . . . . .	49
	6.5	Experiment 5: RB-EKF / Multiple Trajectories . . . . .	52
	6.6	Discussion on Simulation Experiments . . . . .	55
7		Physical Experiments . . . . .	57
	7.1	The Multi Camera Setup Components . . . . .	57
	7.1.1	CTR-1472 MPEG-4 Compressor . . . . .	58
	7.1.2	Measurement Cameras . . . . .	58
	7.1.3	Connection Cables, Tripods and Clips . . . . .	59
	7.2	Internal Calibration . . . . .	61
	7.3	Experiment 1: R-EKF / Free Fall Motion . . . . .	63
	7.4	Experiment 2: RB-EKF / Free Fall Motion . . . . .	66
	7.5	Experiment 3: RB-EKF / Ballistic Motion . . . . .	68

7.6	Experiment 4: Perpendicular Camera Placement . . . . .	71
7.7	Experiment 5: Arbitrary Camera Placement . . . . .	74
7.8	Experiment 6: Using Multiple Trajectories for Arbitrary Camera Placement . . . . .	76
7.9	Discussion on Physical Experiments . . . . .	77
8	Conclusions and Future Work . . . . .	81
	REFERENCES . . . . .	83

## LIST OF TABLES

### TABLES

Table 3.1	The extended Kalman filter algorithm in discrete time . . . . .	16
Table 6.1	The final re-projection, angle and center errors for R-EKF and RB-EKF . . .	46
Table 6.2	The final re-projection, angle and center errors for single and double throw cases . . . . .	53
Table 7.1	The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with R-EKF. . . . .	64
Table 7.2	The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with RB-EKF. . . . .	68

## LIST OF FIGURES

### FIGURES

Figure 4.1	The Euclidean transformation between the world and camera coordinate frames. $C$ is the camera center, $X$ is the 3D world point, $x$ is the 2D projection onto the camera image plane . . . . .	25
Figure 4.2	Detection of the target calibration object. (a) Mean image of the calibration frames. (b) Standard deviation image of the calibration frames. (c) Threshold image of a calibration frame. (d) Binary image of a calibration frame. The center of the target is also marked. . . . .	27
Figure 4.3	Two-camera system assumed for illustration purposes to be mounted on a fixture with the unknown rotation and translation between them consisting the external calibration parameters. . . . .	28
Figure 4.4	First camera shown on the <i>basis coordinate frame</i> . The relation between the basis frame and <i>Cam1 coordinate frame</i> is represented with the rotation angles $\alpha^{c1}$ and $\beta^{c1}$ around $z$ and $x$ -axes respectively. . . . .	29
Figure 5.1	The flow chart for the repeated EKF algorithm. . . . .	35
Figure 5.2	The flow chart of the repeated bidirectional EKF algorithm. . . . .	37
Figure 6.1	The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm. . . . .	39
Figure 6.2	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	40

Figure 6.3	The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	42
Figure 6.4	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	44
Figure 6.5	The synthetic images of a free falling target captured by a stereo camera system. The motion and observation noise variances are taken as 0.002 m and 0.1 pixels. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by (a) Repeated EKF; (b) Repeated Bidirectional EKF. . . . .	46
Figure 6.6	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations. Since the initial target state is not truly provided the calibration parameters are not accurately obtained (a) Rotation angles; (b) Translation parameters. . . . .	47
Figure 6.7	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations. Even though the initial target state is not truly provided the calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	48
Figure 6.8	The synthetic images of the ballistic trajectory of a thrown target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	49
Figure 6.9	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	51
Figure 6.10	The cameras are placed perpendicular to each other and at the diagonal corners of a square with 1.5 meter edge length. . . . .	52
Figure 6.11	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations when two different target trajectories are used for calibration. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . .	54

Figure 7.1	Eurotech CTR-1472 frame grabber and MPEG-4 Compressor . . . . .	58
Figure 7.2	Outland Tech. UWC-300 underwater cameras (black cylinder, top) . . . . .	59
Figure 7.3	The connection cables and connectors. (a) PC104 stack which includes CTR-1472 MPEG-4 compressor and the manufactured cable set connecting the cameras to the stack. (b) CTR-1472 connector. . . . .	59
Figure 7.4	Tripod-clamp-camera trio and the calibration pattern. (Shown for a single camera) . . . . .	60
Figure 7.5	4 pieces of BWC-300 underwater cameras and synchronous video captur- ing with these cameras. . . . .	60
Figure 7.6	The calibration view obtained from 4 cameras synchronously. This kind of chessboard pattern shown in the figure is used for internal calibration purpose. . .	62
Figure 7.7	The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm. . . . .	64
Figure 7.8	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	65
Figure 7.9	The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	66
Figure 7.10	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> it- erations, (a) Rotation angles; (b) Translation parameters. . . . .	67
Figure 7.11	The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	69
Figure 7.12	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> it- erations, (a) Rotation angles; (b) Translation parameters. . . . .	70
Figure 7.13	The cameras are placed at the diagonal corners of a square with 1 meter edge length. . . . .	71

Figure 7.14 The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	72
Figure 7.15 Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	73
Figure 7.16 The extracted and re-projected 2-D positions of a ballistically moving target captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . .	74
Figure 7.17 Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	75
Figure 7.18 The extracted and re-projected 2-D positions of three ballistically moving targets captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm for the (a) first; (b) second and (c) third trajectories. . . . .	78
Figure 7.19 Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	79

## LIST OF ABBREVIATIONS

<b>Cam</b>	Camera
<b>EKF</b>	Extended Kalman Filter
<b>fps</b>	Frame per second
<b>FOV</b>	Field of View
<b>KF</b>	Kalman Filter
<b>MDP</b>	Markov Decision Process
<b>ML</b>	Maximum Likelihood
<b>MMSE</b>	Minimum Mean Square Error
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>R-EKF</b>	Repeated EKF
<b>RB-EKF</b>	Repeated Bidirectional EKF
<b>SSPE</b>	Simultaneous State and Parameter Estimation
<b>STCC</b>	Simultaneous Tracking and Camera Calibration

# CHAPTER 1

## Introduction

### 1.1 General

The use of a multi-camera setup to extract information about both static and dynamic 3D configuration (state) of real world objects is an important problem of 3D vision [1, 2, 3]. In the particular application case of robot vision, the cameras may be part of a robotic platform (such as a mobile robot, unmanned aerial, land or underwater vehicle) and the state in question may belong to the platform in motion. Alternatively, the camera setup may be mounted externally as a measurement setup to externally measure the robot body state with high precision [4, 5]. The increase in precision is due to the possibility of off-line processing of data as well as abundance of computational hardware as compared to the case of being on a moving platform. Commercial multi-camera "motion capture" systems exist for this latter problem of precise robot body state measurement [6]. All of these systems necessarily require steps for calibration, namely the derivation of the internal parameters of each camera involved (internal calibration) as well as the parameters of the geometric arrangement of these cameras (external calibration) [7]. The collection of these calibration parameters characterize everything in the measurement setup and they can be used to extract the actual measurement variables (e.g. the 3D position of an external object).

Internal and external parameters show different characteristics with respect to rate of change or sensitivity to external disturbances. In general, camera itself is a rigidly constructed object and hence the internal parameters of the camera is more robust as compared to the external parameters. Given that the internal calibration parameters are obtained at low and medium accuracy levels temporal variations in these parameters for consumer-grade cameras are gen-

erally not significant [8]. Therefore, a procedurally efficient and fast method for the internal calibration is not considered critical since it is usually performed per camera and can remain valid for a long time. On the other hand, the external calibration, which characterizes the geometric arrangement of the individual cameras, is much less robust and has to be repeated every time the arrangement of the cameras is changed. For some applications where the cameras are mounted permanently, e.g. in a laboratory setup, these are much less prone to disturbances and a slow and manual calibration may be tolerated. For some other applications such as in robotics and for experimental measurements in the field, the situation is significantly different. A multi-camera setup in this case is often constructed on tripods and specifically to conduct one particular set of experiments. The setup is open to disturbances due to the field conditions such as loose ground, wind, accidental bumping to the tripods and similar. In these circumstances, the re-calibration of the setup may be necessary on a much more frequent basis and the re-calibration effort may constitute a large percentage of the experimental effort. This is clearly undesirable.

In our considered application problem, a multi-camera setup is deemed necessary to measure the body-state of an experimental hexapod platform: RHex [9]. Such a system is critical for good experimental practice with aim of generating ground truth data for the evaluation of the on-board state estimation, control and navigation algorithms. Since computationally more complex off-line state estimation algorithms can be used on an external system, those estimates can generally be considered as ground truth as compared to what can be achieved on-board.

RHex is a robotic platform claiming outdoor mobility on complex terrain. Advancing the state of research on this platform demands field experimentation in various environments. This is opposed to many mobile robotics platforms constrained to laboratory environment where a fixed measurement setup can be affordable. The application demands a fast and procedurally efficient calibration process (procedure and algorithm) since the cameras are to be carried to the site and suitably located for each experiment. There is high probability that a natural disturbance may move the setup and render the external calibration invalid. In each such occurrence, the external calibration has to be repeated, making ease of calibration procedure critical. If the calibration have to be performed frequently one needs an efficient calibration method that is both computationally fast but which also requires minimal human effort, e.g., by requiring minimal materials and special equipment such as specially built calibration pan-

els. The design of such a calibration method with competent calibration performance is the focus of the present thesis.

## 1.2 Scope and Contribution of the Thesis

In this thesis, a novel calibration algorithm for obtaining the external parameters of a camera setup is presented. The method uses the known dynamics of a target object for calibration. The aim is to exploit this extra knowledge about an easy to use target object to minimize the calibration effort and provide a fast and procedurally efficient technique. It is claimed that the proposed method is particularly beneficial for any application where frequent re-calibration is required such as outdoor field measurement applications using multiple-cameras.

In this thesis, we assume that a set of cameras, whose internal parameters are known apriori, are placed and oriented such that the required measurement area is within the overlapping section of the field-of-views (FOV) of all cameras. It is further assumed that their resulting external calibration parameters are to be determined. The motivation of the present work suggests that an easy to carry rigid target and simple to generate 3D motion with well known dynamics can be used to generate the calibration data set. A feasible example is a small rigid ball with well defined color and suitable mass which is ballistically thrown within the camera joint FOV. In 3D space, the motion of this ball is fully described by Newton's equations of motion. Assuming that wind and other secondary disturbances are negligible, the dynamics of this motion can therefore be modeled with good accuracy.

It is the claim and contribution of the present thesis that given enough calibration data (in the form of camera frames from all involved cameras) the repeated and bidirectional extensions of the EKF algorithm asymptotically converge to the estimates of the camera calibration parameters. We claim that standard EKF can provide usable estimates if there are large number of image frames while the proposed extensions can generate these estimates with much less number of image frames. The repeated use of the calibration data (R-EKF) provides convergence while its repeated and bi-directional use (RB-EKF) further improves the accuracy of the estimates.

The Kalman framework takes into account the process and observation noises such as wind effect and pixel errors. We also demonstrate that the RB-EKF version can tolerate significant

uncertainty in the initial state of the target (i.e. the initial position and velocity of the target). It should also be pointed out that despite our focus being on the use of the ball's motion to find the external calibration parameters of the camera setup, it is observed that an interesting complementary problem is also solved, namely that of estimating the state of a target with known dynamics with an uncalibrated camera setup.

At this point it should be stated that the proposed method is not limited to the present application scenario and can be extended to other areas: E.g., with the growing interest in 3D-TV, 3D multi-camera capture of sports events becomes important, requiring the external calibration of the camera setup. Due to the existence of a well defined ball, which follows a ballistic path, in most competition games (tennis, football, basketball, etc.), the method presented here is directly applicable.

### **1.3 Outline of the Dissertation**

The motivation and the context for this work together with the main contributions are presented in Chapter 1. Chapter 2 summarizes the state of the art relating to algorithms for internal and external calibration and also relating to basic video tracking methods. Chapter 3 focuses on the Kalman framework and the Extended Kalman Filter (EKF) and the usage in simultaneous state and parameter estimation (SSPE) problems. The definition of the considered calibration problem within an EKF framework and the augmentation of the target state with the unknown camera parameters to form the "super-state" is presented in Chapter 4. Chapter 5 presents the two approaches proposed for estimating the complete state with a limited number of measurement data, namely the R-EKF and RB-EKF algorithms. Chapter 6 and Chapter 7 presents the results of the experiments which are performed to evaluate the performance of the implemented calibration method, with both simulated and actual physical camera setups and image frame data. Finally, Chapter 8 gives the conclusions of this thesis together with possible avenues for future studies.

## CHAPTER 2

### Background on Camera Calibration

In all measurement systems, the calibration of the measurement instrument is a necessary step in order to find the transformation between the raw readings from the instrument to the actual physical measurement variables of interest. When the state of the world or the ego-state (state of the body on which the sensor(s) are mounted) is being measured through the use of cameras as sensors, the system needs calibration so that the raw camera images can be transformed into this measurement of the state. Essentially, calibration parameters can be seen as the collection of all parameters that characterize the transformation between the raw data from the instrument and the desired variables to be measured. Calibration is a compulsory initial step for any measurement system including a multi-camera 3D vision system.

For the particular case of camera calibration, we are concerned with the process of estimating the internal (intrinsic) and external (extrinsic) parameters of a camera system. The internal parameters characterize the optical properties of the lens system in the camera as well as the interaction of this lens system with the other components of the camera such as the sensor. In short, it is a collection of all parameters that determine how the image is being formed on the sensor of a single camera. External parameters on the other hand relate this camera unit (with its internal parameters) to a global reference frame, hence represent the 3D geometrical placement and orientation. If multiple cameras are involved, the external calibration parameter set represent this geometric specification for all cameras hence also describing their relative positions and orientations.

This chapter reviews the previous work on camera calibration and its historical evolution. Most of the previous work focus on internal calibration techniques for a single camera. However especially for the last two decades, the number and variety of the cameras have been

increasing rapidly while their prices have been decreasing, making multiple camera imaging systems more accessible to the research community. The necessity of external calibration methods for such multiple camera setups have encouraged recent studies on this subject.

A review of the state of the art techniques is introduced first for the internal calibration of a camera and then for the external calibration of a multiple camera network in sections 2.1 and 2.2 respectively. This is followed by an overview of object detection methods in section 2.3 which are necessary to generate *features* or *observations* for the calibration algorithms. Finally the relationship between calibration and dynamic state estimation (tracking) using Kalman Filtering is introduced in 2.4. Also in this last section, a survey of the earlier calibration methods utilizing the Extended Kalman Filter is presented.

## 2.1 Internal Camera Calibration

The image formation process can be seen as a mapping from the 3D world to the 2D image plane. The usual modeling of this mapping is by *central projective mapping*. In this mapping rays reflected from a 3D point in space is passed through a fixed point which is the *center of projection*. The rays intersect the *image plane*, which is placed at a specific distance from the center of projection, and their intersection forms the image point. Throughout this process involving a single camera, the depth information is lost due to the loss of one dimension.

Most cameras are modeled by the central projective mapping assumption [10]. The center of the lens of a camera is assumed as the center of projection. A ray of light from a 3D point in the world passes through the lens and produces an image point on a film or an opto-electronic capturing device (i.e., imaging sensor).

Homogenous coordinates allows affine transformations to be represented by a matrix and make calculations possible in projective space [11]. The mapping of a point in  $\mathbb{R}^3$  to  $\mathbb{R}^2$  on the image plane by a camera can be represented by a projection matrix  $\mathbf{P}$  in the homogenous coordinates. This  $\mathbf{P}$  matrix can be decomposed into two matrices  $\mathbf{K}$  representing the internal parameters of a camera and  $[\mathbf{R} \mid \mathbf{t}]$  representing the rotation and translation between the camera and the world coordinate frames. The transformation is given by Eqns. (2.1) and (2.2) and

further elaborated in section 4.1.2.

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (2.1)$$

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \quad (2.2)$$

Internal calibration of a camera is therefore the task of obtaining the  $\mathbf{K}$  matrix which represents the inner characteristics of the camera. These calibration parameters are specifically

- *focal length*: the distance between the center of projection and the image plane,
- *pixel aspect ratio*: the width of a pixel divided by its height,
- *principal point offset*: the misplacement of the image origin,
- *skew*: the non-orthogonality of the camera axes.

Classic photogrammetric methods use non-linear optimization for solving all of the calibration parameters simultaneously [12, 13, 14]. The parameter values are directly computed by a non-iterative closed form algorithm. The algorithms are fast since no iterations are involved, but the lens distortion can not be incorporated into these algorithms and noisy measurements may cause quick deterioration of performance.

More recently, two-step methods which can successfully handle lens distortions have been proposed in the literature [15, 16, 17, 18, 19]. All of these techniques first estimate the calibration parameters using a closed form solution similar to earlier methods. Then, the parameters are improved iteratively through nonlinear optimization, taking the lens distortion into account.

These two-step methods make use of a special planar calibration pattern printed on a plate and placed in the camera FOV. By the use of a planar chess-board pattern, a large number of data points can be supplied and the algorithms yield reliable results by minimizing the average *re-projection error* on the collection of the points. Average re-projection error can be defined as the mean square error between the scene points visible in the actual camera and their projection in the reference camera whose characteristics are described by the employed camera model and the acquired calibration parameters [20].

As a further improvement to the aforementioned algorithms, a four-step calibration procedure is suggested by Heikkilä and Silvén [21]. The authors claim that these additional steps further

improve the compensation for distortion and provide the necessary correction. Also a linear method for solving the parameters of the inverse model is presented in this work.

The suggested algorithm given in [22] aims to generalize the camera model, and in particular, try to provide a solution for cameras which do not have a single effective viewpoint (termed as *non-central cameras* by the authors). To achieve this goal, a very general mapping between rays and pixels is used instead of a *central projection mapping*.

A recent survey summarizes the current approaches on camera calibration and present a comparison of different algorithms [8]. The efficiency of the bundle adjustment techniques are presented with experimental results. The changing profiles of the radial distortion for different color channels is also discussed.

The calibration methods which are proposed by Tsai [16] and Zhang [19] are widely accepted and popularly used for computer vision. Although these two-step methods are primarily internal calibration methods, it is suggested that they can be extended for calibrating the external parameters as well. However, these extensions for calibrating a camera network are subject to practical difficulties. This is due to the fact that successful external calibration requires that multiple different orientations of a 2D planar calibration pattern to be viewable from the joint-FOVs of all cameras in the network. Indeed, such a pattern can be seen from two (or more) different cameras only if both cameras are placed on the same side of the plane. Furthermore the pattern should be appropriately scaled in accordance with the magnification of the cameras and also in accordance with the placement and orientation of their FOVs.

The camera calibration toolbox which is part of the MATLAB package (from MathWorks Inc.) provides another algorithm used extensively for internal and partly for external calibration [23]. However the same visibility problem of the calibration plate still exists.

These popular camera calibration techniques yield reliable results by minimizing the overall re-projection error defined over the collection of data points that are provided by the planar chess-board pattern. However it is the very same planar fixed sized calibration object that causes significant visibility, scalability and procedural complexity problems when the methods are used for external calibration. It should be noted that these problems form the primary motivations for our approach where we rather provide a simple but moving calibration target avoiding these difficulties.

The basic idea behind our approach is the use of a simple to detect target which is moving with known dynamics. For this we propose a ballistically moving rigid object such as a ball with given color. The path followed by such a projectile can easily be viewed by multiple cameras looking from all possible angles, even when the cameras are facing each other. Furthermore, the calibration field can be utilized maximally by throwing the ball according to the distance between the cameras. So the present method not only solves the visibility problem but it also handles the scalability problem as well. The calibration target considered (i.e. a ball) can easily be obtained, and the calibration procedure is practical and fast. Finally, the knowledge of the global gravity vector provides additional information about the external calibration which is not present in other techniques (unless the calibration plate is not carefully aligned with the gravity vector).

## 2.2 External Camera Calibration

When a  $3D$  scene is projected onto a plane, depth information is lost during the process. Hence it is not possible to extract the complete geometry of a scene from a single view. The  $3D$  structure of a scene through image analysis can either be obtained by a multi camera system [24] or by taking multiple frames of the same scene with a single camera moving in space. For both cases a fundamental problem, that is called external calibration, is to obtain the camera positions and orientations for each image frame. Without this a priori knowledge of the acquisition system it is not possible to obtain the full  $3D$  scene structure. It should be noted that we only consider standard image sensors and we do not consider specialized measuring devices that provide depth information directly, such as laser range scanners.

Multi camera external calibration is the task of obtaining the rotation and translation of each camera with respect to each other or relative to a global world coordinate frame. Let  $[\mathbf{R} \mid \mathbf{t}]$  matrix represent the rotation and translation for a camera. The  $\mathbf{R}$  matrix of size  $3 \times 3$  is called as the *rotation matrix* and represents the rotations around the coordinate axes. The  $\mathbf{t}$  matrix of size  $3 \times 1$  is called as the *translation vector* and represents the position of the camera along the three coordinate axes. Since the transpose of a rotation matrix is equal to its inverse [7],  $\mathbf{R}$  actually consists of only 3 unknowns. If the camera coordinate system is considered, the negative of the translation vector  $\mathbf{t}$  also represents the origin of the world reference frame in camera coordinates. Therefore a total of 6 parameters are required for the external calibration

of each camera, specifically 3 being for representing the orientation (rotation) of the camera and 3 being for the position (translation) of each camera.

The inverse problem of image formation is computing the camera location and orientation from the projection matrix. It involves solution of simultaneous nonlinear equations in several variables and is considered difficult. To accomplish this inversion a simple analytical technique that works well in practice is suggested in [25].

In [26] the issues concerning the calibration of multiple camera systems are discussed. It is argued that the ease of production of the calibration object hence that of the calibration process is in a trade off with the accuracy of calibration, because the most accurate calibration process would likely require a complicated calibration object and process. In the present thesis, we demonstrate that with the use of known target dynamics, this is not necessarily the case. We aim at a very simple calibration target and a simple process while also maintaining the accuracy of calibration.

The work in [27] also makes use of multiple images of a static chessboard calibration pattern. Cross projection errors between the cameras are minimized for extending single camera calibration method to multiple cameras.

Complete multi-camera calibration procedures, which uses a laser pointer for image generation, are presented by Han et al [28], and Svoboda et al [29]. The assumption of only reasonable overlap between camera subgroups makes the procedures of the algorithms to be easily applicable. The drawback of these methods appear to be the necessity of at least three cameras for calibration, hence making the calibration of stereo cameras impossible. The procedure also requires total darkness, rendering the outdoor applicability of these methods limited. The calibration methods presented in the present thesis on the other hand can seamlessly handle stereo cameras and can also work in daylight and outdoors provided that the target is detectable over the background scene.

Other approaches are presented in [30] and [31]. Instead of tracking a single calibration point in image sequences, a reference bar with light sources on it is moved giving additional depth information which is the constant length of the bar. Note that this additional information requires the construction of a rather complicated calibration bar with embedded laser pointers on it. In our approach constant gravitational acceleration gives equivalent additional informa-

tion without the associated complication in the calibration target or procedure.

## 2.3 Object Detection

Object detection is the process of detecting and determining the position of an object within a given image. Clearly, for the methods discussed in the present thesis, one needs to “detect” the calibration object, reliably determining its center position within the image frame, possibly at sub-pixel resolution. The problem can be very simple (such as in the case of a white ball with a black background farther away) or may get gradually harder (such as trying to detect the ball over a busy urban or natural background). The problem can be helped by the use of target color selection or target illumination (e.g. with IR LEDs mounted on the target) combined with appropriate filter on the cameras.

The algorithms on this subject are mostly focused on the application areas like face and car detection in the sequential video frames. Some of these basic methods are overviewed in this section.

Object detection algorithms that are specialized to certain objects of interest are commonly based on learning the characteristics of these objects [32, 33, 34, 35, 36, 37, 38, 39]. These algorithms need too many training examples and depend on huge training databases [40].

An extremely fast object detection technique is presented by Viola et al [37, 38]. This study not only uses learning to train the classifiers, but also improves the detection time of the features by representing the image in a new format. Furthermore, by connecting the classifiers in a cascade manner the background image is discarded and hence more time is spent on object-like regions. This study is improved with the introduction of a novel set of rotated Haar-like features by Lienhart et al [39].

Two probabilistic approaches are presented in the histogram based method for 3D object detection by Schneiderman et al [41] and visual learning method based on density estimation in a high dimensional space by Moghaddam et al [32].

Throughout our study we assume that the cameras are fixed, and only one very small calibration target is moving in the common camera FOV. Therefore the background image can approximately be acquired by obtaining the mean image of the video sequence, significantly

reducing the complexity of the target detection problem. For this reason we used Svoboda’s LED detection method [29] with some modification for the automatic detection of the calibration target, instead of using a complicated learning based algorithm. In this manner we not only can avoid the computational cost of a complex method, but avoid the need for a training database.

## 2.4 Links between Calibration and State Estimation

The approach in the present thesis is based on the idea that the estimation of static parameters can be integrated into a dynamic state estimation problem. The benefit of this approach is its ability to make use of the knowledge of target dynamics in order to better observe the calibration target positions, hence attaining high precision with a relatively coarse target object.

This idea is clearly not entirely new. For example, our problem has significant similarities with the Simultaneous Localization and Mapping (SLAM) approach, where noisy optimal state estimation (Kalman filtering) formulation is used to simultaneously estimate a moving robot’s position (dynamic state) and the map of the environment (static state) [42, 43].

This approach has the property that a generalized *system* and its associated *augmented state* is defined such that the unknown locations of the features of the environment (i.e. the map) are made part of the augmented state of the system and are subjected to the optimal estimation process. Here, the known dynamics of the robot platform, namely the *model of the system* is exploited for a better estimate than is possible simply by static observations of the robot position relative to known features of the environment. Moreover, the fact that the robot itself is moving makes static observations of landmarks with respect to a global reference frame impossible

In the present study, we build on this idea in a different context and with a different objective. Indeed, there are other instances of the idea in the existing literature, even in the present context. However, we believe that the specific motivation and solution approach considered in the present study to be of significantly more general than any other existing work in the literature, hence of considerable value.

For example, a number of studies attempt to give the camera orientations [44, 45] or rectify

the ground plane by tracking moving objects [46], while solving a planar relationship using the video sequences from  $n$  fixed cameras at unknown positions. The planarity constraint is ensured by placing the cameras high enough from the ground plane. We do not have such a constraint in the present work and provide a more general calibration algorithm as a result. An extension to these works is [47], which takes the topology of the camera overlap as an unknown and estimate a planar tracking correspondence model in addition to the calibration parameters of the large camera network.

Kalman Filter is used in [48] to recursively estimate the motion, pointwise structure, and focal length from feature correspondences, and in [49] for determining the unknown transformation between a camera and an IMU. No known dynamic model of a target is exploited in either of these studies hence the Kalman Filter may just be acting as a low pass filter.

Again in [50], a calibration algorithm based on Kalman Filter is presented. The method claim to reduce human intervention during calibration but can not fully eliminate it. One  $2D-3D$  correspondence of the moving object and at least 6 other points on a rigid object are needed to initialize EKF algorithm and guarantee its convergence. The necessity of a planar ground is also another constraint and drawback of this approach. We should note that for our case in the present thesis, neither any  $2D-3D$  correspondences nor any extra information is needed for the initialization of the algorithms.

## CHAPTER 3

### The Extended Kalman Filter

This chapter aims to give the basic knowledge about Kalman filtering framework and in particular the use of Extended Kalman filter (EKF) in simultaneous state and parameter estimation (SSPE) problems. The objective is to provide the adequate background to facilitate our discussion in chapter 4, which is built upon the assumption of a general knowledge on these issues.

A review of the EKF algorithm is given in section 3.2. The SSPE method is discussed in 3.3. Finally an example that illustrates the SLAM approach is given in 3.3.2.

#### 3.1 General

Kalman filter (KF) is the *optimal* state estimator for discrete-time linear dynamic systems driven by *white noise* [51]. KF is also proved to be the best linear state estimator in the linear *non-Gaussian* case.

Starting from an initial state which is assumed to be a random variable with a certain mean and covariance, Kalman filter estimates the next states by representing the dynamics of the system as a Markov Decision Process (MDP) which assumes that the state of the system depends only on the previous state and action [52]. Hence, KF is a recursive estimator, which predicts the state only from the previous time step and current measurement instead of the history of observations as in batch estimation methods. If the initial state, the process and the measurement noises are Gaussian and *mutually independent*, KF is the minimum mean square error (MMSE) estimator [53].

Although the linear system with *zero-mean white noise* assumption seems to restrict the applicable usage of the filter, through a number of extensions, this technique can be used in numerous problems. The extensions make it possible to map the problem at hand into one that meets the above constraints. For example most non-trivial systems are non-linear and they violate the linearity assumption of the basic KF. Such non-linearities in the process or observation models can be handled by the Extended Kalman filter (EKF) which is an approximation to the optimal estimate.

Occasionally some of the continuous-valued parameters of the system may be unknown. In such cases the state can be augmented by adding the unknown parameters and the EKF can be used for estimating both the state and the system parameters. This simultaneous state and parameter estimation (SSPE) method is popularly used in robot mapping problems and specifically named as the simultaneous localization and mapping (SLAM).

### 3.2 Overview of the EKF Algorithm

The Kalman filter is the best studied state estimation technique and it was invented by Swerling (1958) and Kalman (1960). KF assumes that the observations are linear functions of the state and the next state is a linear function of the previous state. Note that a Gaussian random variable which is passed through a linear function yields another Gaussian random variable. This knowledge plays an important role in the derivation of KF.

Unfortunately neither the systems nor the measurements can always be modeled by only linear transformations. In fact the state transitions and measurements are rarely linear since real life systems lack the simplicity that would enable us to use the KF directly. The non-linearity of the process or measurement models (or both) can be associated by the extended Kalman filter (EKF).

In EKF the state transition and observation models are assumed to be nonlinear, differentiable functions of the state. Let us express the state transition and observation equations:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1} \quad (3.1)$$

$$\mathbf{z}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \quad (3.2)$$

Here  $\mathbf{x}_{k+1}$  and  $\mathbf{x}_k$  are state vectors, and  $\mathbf{u}_{k+1}$  is the control vector at time  $k + 1$ . The process

and observation uncertainties at time  $k + 1$  are modeled by the random variables  $\mathbf{w}_{k+1}$  and  $\mathbf{v}_{k+1}$  respectively. The predicted state is computed from the present estimate by the state transition function  $f$  and similarly the measurement is computed from the predicted state by the observation function  $h$ . Since  $f$  and  $h$  are not directly applicable for the covariance calculation their Jacobian is calculated at each time step by using the latest predicted states. This Jacobian computation step is the key idea underlying the EKF approximation which is called *linearization*.

The Jacobian matrices can be used in KF equations resulting in the extended Kalman filter equations given in Table 3.1.  $\mathbf{Q}_{k+1}$  and  $\mathbf{R}_{k+1}$  are the covariance matrices of the posterior state

Table 3.1: The extended Kalman filter algorithm in discrete time

<p><b>Algorithm EKF</b>( <math>\mu_k, \Sigma_k, \mathbf{u}_{k+1}, \mathbf{z}_{k+1}</math> )</p> <p><i>Prediction:</i></p> $\bar{\mu}_{k+1} = f(\mu_k, \mathbf{u}_{k+1})$ $\bar{\Sigma}_{k+1} = \mathbf{F}_{k+1} \Sigma_k \mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1}$ <p><i>Update:</i></p> $\mathbf{K}_{k+1} = \bar{\Sigma}_{k+1} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \bar{\Sigma}_{k+1} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1}$ $\mu_{k+1} = \bar{\mu}_{k+1} + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - h(\bar{\mu}_{k+1}))$ $\Sigma_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \bar{\Sigma}_{k+1}$ <p><i>Return</i>(<math>\mu_{k+1}, \Sigma_{k+1}</math>)</p>
--

and measurement noises. The state transition and measurement Jacobians are defined as

$$\mathbf{F}_{k+1} = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mu_k, \mathbf{u}_{k+1}}, \quad \mathbf{H}_{k+1} = \frac{\partial h}{\partial \mathbf{x}} \Big|_{\bar{\mu}_{k+1}} \quad (3.3)$$

Similar to the Kalman filter, the EKF represents the state at time  $k$  by the mean  $\mu_k$  and the covariance  $\Sigma_k$ . The filter updates these parameters when  $\mu_k$  and  $\Sigma_k$  are given as the input of the EKF together with the control  $\mathbf{u}_{k+1}$  and measurement  $\mathbf{z}_{k+1}$  for the next time step. The output is the estimate at time  $k + 1$ , represented by  $\mu_{k+1}$  and  $\Sigma_{k+1}$ .

In the prediction step the predicted belief  $\bar{\mu}_{k+1}$  and  $\bar{\Sigma}_{k+1}$  is calculated representing the belief for the next time step by only incorporating the control  $\mathbf{u}_{k+1}$  but not the measurement  $\mathbf{z}_{k+1}$ . In the update step the *Kalman gain*  $\mathbf{K}_{k+1}$  and *innovation*, which is the difference between the measurement  $\mathbf{z}_{k+1}$  and the expected measurement  $h(\bar{\mu}_{k+1})$ , are computed. Innovation is used to update the predicted belief into the desired one up to a degree specified by the Kalman gain. The derivation of the EKF algorithm is available in [51], [53], [54] and [55].

The major drawback of the extended Kalman filter is its non-optimality. The EKF may even quickly diverge if the modeling of the system is not correctly handled or if the initial state estimate is not close to the actual state. The unobservability of the process may also cause divergence of the EKF [55]. Yet the EKF can give reasonable results and in fact practically it is the standard estimation method for many applications such as navigation systems and GPS.

### 3.3 Simultaneous State and Parameter Estimation

The extended Kalman filter can be used for state estimation in nonlinear dynamic systems. In general the parameters of the state transition function  $f$  and measurement function  $h$  should be known for EKF employment. When there are unknown system parameters which are *continuous valued* or *slowly varying* it is still possible to obtain the state and the parameters simultaneously by means of an extended Kalman filter.

#### 3.3.1 The SSPE Algorithm

The simultaneous state and parameter estimation (SSPE) method suggests augmenting the *base state* by the unknown system parameters. Let us denote the unknown parameters by a vector  $\theta$ , the base state vector by  $\mathbf{x}$ , and the augmented state vector by  $\mathbf{y}$ . The state equation in (3.1) can be rewritten as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \theta_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1} \quad (3.4)$$

and the dynamic equation of the parameter vector is assumed to be time invariant

$$\theta_{k+1} = \theta_k \quad (3.5)$$

Then the *super-state*  $\mathbf{y}$  consist of  $\mathbf{x}$  and  $\theta$

$$\mathbf{y}_k \equiv \begin{bmatrix} \mathbf{x}_k \\ \theta \end{bmatrix} \quad (3.6)$$

Combining Eqns. (3.4), (3.5) and (3.6) we get the following expression for the super-state

$$\mathbf{y}_{k+1} = \hat{f}(\mathbf{y}_k, \mathbf{u}_{k+1}) + \hat{\mathbf{w}}_{k+1} \quad (3.7)$$

For completeness the minor update in equation (3.2) is also given as

$$\mathbf{z}_{k+1} = h(\mathbf{y}_{k+1}) + \mathbf{v}_{k+1} \quad (3.8)$$

The EKF can be used with these resulting state and observation equations (3.7) and (3.8) for the estimation of the entire super-state. Note that the measurement vector is not needed to be augmented or redefined as the state vector even if the measurement function  $h$  may also be a function of  $\theta$ .

The parameter dynamics are assumed to be constant by Eqn. (3.5). So their covariance will asymptotically converge to zero. In addition to this, if the process noise corresponding to these parameters is also assumed to be zero, then the filter gain will tend to zero which will result in wrong estimates. Thus the situation where the parameter variances are inclined to zero is undesirable. In order to overcome this problem an *artificial process noise*  $\mathbf{w}_{k+1}^\theta$  may be added to the unknown parameter vector. This correspond to replacing Eqn. (3.5) by

$$\theta_{k+1} = \theta_k + \mathbf{w}_{k+1}^\theta \quad (3.9)$$

The artificial noise  $\mathbf{w}_{k+1}^\theta$  prevents the variances of the parameter estimates from converging to zero, so that the calculated variances can be balanced with the estimation errors. This *pseudo-noise* assumption also enables the filter to estimate slowly varying parameters. The standard deviation of the process noise is generally chosen as a few percent of the guessed value of the parameter. Further *tuning of the filter* can be done as explained in [53].

### 3.3.2 An Example on SSPE

Let us now give an example about the use of EKF in a simultaneous state and parameter estimation problem. For simplicity assume that the base state is a one dimensional vector and the state equation be given as

$$x_{k+1} = a_{k+1} x_k + b_{k+1} u_{k+1} + w_{k+1}^{base} \quad (3.10)$$

where  $a_{k+1}$  and  $b_{k+1}$  are the unknown system parameters which may slowly vary in time, and  $w_{k+1}^{base}$  is the base state process noise at time  $k + 1$ . Let us also define the observation as a one dimensional vector and the related equation as

$$z_{k+1} = c_{k+1} x_{k+1} + v_{k+1} \quad (3.11)$$

where  $c_{k+1}$  is again an unknown measurement parameter.

As defined in section 3.3.1, the augmentation of the base state with the unknown parameters gives us a four-dimensional state vector  $\mathbf{y}_k$  and the corresponding process noise  $\mathbf{w}_k$ .

$$\mathbf{y}_k \equiv \begin{bmatrix} x_k \\ a_k \\ b_k \\ c_k \end{bmatrix} \quad \text{and} \quad \mathbf{w}_k \equiv \begin{bmatrix} w_k^{base} \\ w_k^a \\ w_k^b \\ w_k^c \end{bmatrix} \quad (3.12)$$

Note that  $w_k^a$ ,  $w_k^b$  and  $w_k^c$  are the pseudo process noises. The augmented state equation is then

$$\begin{aligned} \mathbf{y}_{k+1} &= \begin{bmatrix} y_k^1 \\ y_k^2 \\ y_k^3 \\ y_k^4 \end{bmatrix} = \begin{bmatrix} y_k^1 & y_k^2 + y_k^3 u_k \\ y_k^2 \\ y_k^3 \\ y_k^4 \end{bmatrix} + \begin{bmatrix} w_k^{base} \\ w_k^a \\ w_k^b \\ w_k^c \end{bmatrix} \\ &= f(\mathbf{y}_k, u_k) + \mathbf{w}_k \end{aligned} \quad (3.13)$$

which exactly has the same form as given in equation (3.7). The observation equation

$$\begin{aligned} z_{k+1} &= y_{k+1}^4 y_{k+1}^1 + v_{k+1} \\ &= h(\mathbf{y}_{k+1}) + v_{k+1} \end{aligned} \quad (3.14)$$

also has the form given in equation (3.8).

Let us define the process noise and measurement noise covariances.

$$\mathbf{Q} = \text{diag}(q_1, q_2, q_3, q_4) \quad \text{and} \quad \mathbf{R} = r_1 \quad (3.15)$$

For EKF implementation the Jacobian of the functions  $f$  and  $h$  should also be obtained. In our example these matrices are easily calculated as

$$\mathbf{F}_k = \begin{bmatrix} y_k^2 & y_k^1 & u_k & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_k = \begin{bmatrix} y_{k+1}^4 & 0 & 0 & y_{k+1}^1 \end{bmatrix} \quad (3.16)$$

The super-state  $\mathbf{y}$  can now be estimated by using the EKF algorithm given in Table 3.1. With this simple example we have shown the steps of augmenting the state vector by the unknown system parameters and computing the necessary matrices to be able to use EKF for the parameter estimation. However note that the real life problems may be much more complicated than

our simple example. There may be too many unknown parameters and after the augmentation process the state vector dimension may become quite large. This will be shown by a robot mapping example.

### 3.3.3 An Example on Simultaneous Localization and Mapping

Simultaneous localization and mapping is one of the most fundamental problems in robotics. An autonomous vehicle or robot, which is moving in an unknown environment, should not only be able to estimate its state but also the unknown map parameters of its surroundings. This concurrent mapping and localization problem can in fact be treated as an SSPE problem.

Assume that a robot is moving on a planar surface with the ability to rotate to a desired direction. The base state of such a robot is composed of three elements  $p_k^x$ ,  $p_k^y$  and  $p_k^\theta$  which denote the robot's coordinates at time  $k$ . The 2D location of the robot is denoted by  $p_k^x$ ,  $p_k^y$  and the orientation is denoted by  $p_k^\theta$ . When the robot state is augmented with the unknown map parameters the combined vector is given by

$$\begin{aligned} \mathbf{y}_k &= \begin{pmatrix} x_k \\ \theta \end{pmatrix} \\ &= (p_k^x \ p_k^y \ p_k^\theta \ m_{1,x} \ m_{1,y} \ s_1 \ m_{2,x} \ m_{2,y} \ s_2 \ \dots \ m_{N,x} \ m_{N,y} \ s_N)^T \end{aligned} \quad (3.17)$$

where  $m_{i,x}$ ,  $m_{i,y}$  are the coordinates of the  $i$ -th landmark, for  $i = 1, \dots, N$  and  $s_i$  is its signature. The size of the base state is increased by  $3N$  where  $N$  denotes the number of landmarks in the map hence the dimension of the complete state is  $3N + 3$ .

In this example the number of landmarks are assumed to be predefined and known by the robot. A more general approach also assumes that the correspondences are not known and an incremental *maximum likelihood* (ML) estimator is used to determine these correspondences. In fact there are various SLAM algorithms that are produced to reply different requirements. Still, all these methods are built upon the basic SSPE algorithm presented in section 3.3.1.

There may be more than 1000 point landmarks which will result in a huge super-state vector even in the known correspondence case. The robot pose is defined by only three variables while the complete state dimension may reach up to thousands or even more. This clearly demonstrates that we may need to process a much larger super-state vector than the base state

that we want to estimate. Usually at each different orientation of the robot just a few landmarks are observable, so only a small part of the complete state is updated at each time step  $k$ . Instead of using the complete state with the basic EKF algorithm, the SLAM algorithms check whether the landmark has ever been seen before or not and updates the state by using the observed features only. In SLAM observing a landmark does not only improve the position estimate of that landmark but it also improves the robot pose estimate as well as the position estimates of the other landmarks. Since SLAM is not the main subject of this thesis it deemed not necessary to provide the complete SLAM algorithm. However the interested readers can find a complete study on SLAM in [54].

## CHAPTER 4

### Simultaneous Tracking and Camera Calibration

In this chapter the concept of simultaneous state and parameter estimation is applied to the problem of simultaneous tracking and camera calibration (STCC). The approach presented in this thesis suggests that an easy to carry rigid target and simple to generate 3D motion with well known dynamics can be used to generate the calibration data set. Newton's equations of motion fully describe the ballistic trajectory followed by a thrown target in 3D space, and they form the state transition equations for the base state. On the other hand external camera parameters can be taken as the static or slowly varying unknowns of the system and hence they can be made part of the state by augmenting the base state as described in section 3.3.1.

Dynamic motion model for a ballistically moving target and the model of an optical measurement device are given in section 4.1. The detection of the target from the camera views is explained in 4.2. Finally the problem is transformed into an SSPE form by the augmentation of the base state in section 4.3.

#### 4.1 Ballistic Motion and Camera Measurement Models

The state and observation equations of the system are needed for the use of an extended Kalman filter. The state equations of a ballistically thrown object and the observation equations of a camera under the assumption of central projection mapping is derived in sections 4.1.1 and 4.1.2 respectively.

### 4.1.1 Ballistic Motion Model

Objects under the influence of gravitational attraction follow a ballistic trajectory when thrown with an initial state (initial position and velocity vectors) if other disturbance effects like the air friction can be ignored. Under normal weather conditions (no wind, rain, or snow) and for short motion trajectories, the air-friction can be reasonably ignored for a dense object. Assume that  $y$ -axis coincide with the negative direction of gravity, hence placing the  $x$  and  $z$ -axes on the ground plane. Then the equations of motion for the trajectory of a ballistically moving object can be derived by using Newton's 2nd law. If the velocity of a thrown object is decomposed into its  $x$ ,  $y$  and  $z$  components the velocity update equation can be written as

$$\mathbf{V}_{k+1} = \begin{bmatrix} V_{k+1}^x \\ V_{k+1}^y \\ V_{k+1}^z \end{bmatrix} = \begin{bmatrix} V_k^x \\ V_k^y - g\Delta T \\ V_k^z \end{bmatrix} = \mathbf{V}_k + \begin{bmatrix} 0 \\ -g\Delta T \\ 0 \end{bmatrix} \quad (4.1)$$

The components which are parallel to the ground  $V_k^x$  and  $V_k^z$  do not change with respect to time since there are no force acting in these directions during the ballistic flight. Hence we can drop  $k$  from these components and use  $V^x$  and  $V^z$  hereafter. Gravitational force acts towards earth and therefore the velocity component in  $y$  direction decreases by an amount determined by the gravitational acceleration  $g$  and the time difference  $\Delta T$  between each time step. Integration of the velocity update equation w.r.t time yields the position update equation as

$$\begin{aligned} \mathbf{X}_{k+1} &= \begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ Z_{k+1} \end{bmatrix} = \begin{bmatrix} X_k + V^x \Delta T \\ Y_k + V_k^y \Delta T - \frac{g\Delta T^2}{2} \\ Z_k + V^z \Delta T \end{bmatrix} \\ &= \mathbf{X}_k + \mathbf{V}_k \Delta T + \begin{bmatrix} 0 \\ -\frac{g\Delta T^2}{2} \\ 0 \end{bmatrix} \end{aligned} \quad (4.2)$$

The complete state equation in discrete-time can then be given by

$$\mathbf{s}_{k+1}^{dyn} = \mathbf{A} \mathbf{s}_k^{dyn} + \mathbf{u} \quad (4.3)$$

where  $\mathbf{s}_k^{dyn}$  stands for the dynamic system state at time  $k$ ,  $\mathbf{A}$  represents the state transition matrix and  $\mathbf{u}$  corresponds to the input as described in Eqns. (4.4), (4.5) and (4.6).

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$$\mathbf{s}_k^{dyn} = \begin{bmatrix} X & Y & Z & V^x & V^y & V^z \end{bmatrix}_k^T \quad (4.5)$$

$$\mathbf{u} = \begin{bmatrix} 0 & -\frac{g\Delta T^2}{2} & 0 & 0 & -g\Delta T & 0 \end{bmatrix}^T \quad (4.6)$$

$[X \ Y \ Z]^T$ ,  $[V^X \ V^Y \ V^Z]^T$  are the position and velocity vectors of the ballistic target object and  $g$  represents the gravitational acceleration along the *negative*  $y$ -axis direction.  $\Delta T$  is the sampling period.

#### 4.1.2 The Pinhole Camera Model

In this section we give the pinhole camera model since it consist a fundamental part of the overall problem. A pinhole camera can be represented with a central projection mapping from the 3D world to a 2D image [7]. Using the homogenous coordinate system this mapping can be represented by a  $3 \times 4$  matrix  $\mathbf{P}$ , which can be decomposed into  $3 \times 3$   $\mathbf{K}$  and  $3 \times 4$   $[\mathbf{R} \ | \ \mathbf{t}]$  matrices. Here,  $\mathbf{K}$  represents the inner parameters of the camera such as focal length, aspect ratio, principal point offset, and skew, while  $\mathbf{R}$  and  $\mathbf{t}$  represent the external parameters such as camera orientation and displacement with respect to a world coordinate frame as illustrated in Figure 4.1. Here, we have vectors  $\mathbf{X}$  and  $\mathbf{x}$  as the homogenous points in 3D world and 2D image coordinates respectively. The defining equations of this projection are given in Eqns. (4.7), (4.8), and (4.9). Note that  $\mathbf{P}$  has 11 degrees of freedom: 5 for  $\mathbf{K}$  ( $\alpha_x$ ,  $\alpha_y$ ,  $x_0$ ,  $y_0$ ,  $s$ ), 3

for  $\mathbf{R}$  and 3 for  $\mathbf{t}$ .

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (4.7)$$

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \quad (4.8)$$

$$\mathbf{K} = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.9)$$

Projection matrix,  $\mathbf{P}$ , can be divided into submatrices  $\mathbf{K}$  and  $[\mathbf{R} \mid \mathbf{t}]$  as shown in Eqn. (4.8). Note that the world and the camera coordinate frames are related due to a rotation and a translation.  $\mathbf{X}$ , is first multiplied with  $[\mathbf{R} \mid \mathbf{t}]$ , which corresponds to representing the 3D point in the camera coordinate frame. Then the resultant point passes through internal parameter matrix,  $\mathbf{K}$ , providing the 2D projection,  $\mathbf{x}$ . In Eqn. (4.9)  $\alpha_x$  and  $\alpha_y$  represent the focal length of the camera in terms of pixel dimensions towards  $x$  and  $y$  directions respectively;  $[x_0, y_0]^T$  is the principal point in terms of pixel dimensions and  $s$  is the skew parameter. Detailed information about camera models and multiple view geometry can be found in [7].

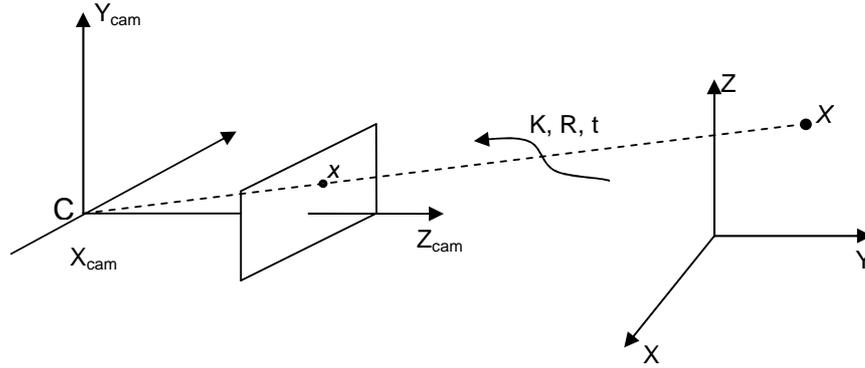


Figure 4.1: The Euclidean transformation between the world and camera coordinate frames.  $C$  is the camera center,  $X$  is the 3D world point,  $x$  is the 2D projection onto the camera image plane

## 4.2 Raw measurements: Detection of the Target

The EKF based algorithms discussed in this work clearly need the pixel coordinates of the calibration target in the image frames, possibly at sub-pixel resolution. In a simulation the target can be selected as a point mass with a predefined 3D trajectory, hence its projection, which absolutely corresponds to a single point on the image frame, can directly be computed

via pinhole camera model. However in reality, there is no exact knowledge about the actual target trajectory. Additionally, the calibration target may be chosen as a small ball which will definitely cover many pixels in the image frames. Therefore to provide an easy and robust calibration method, detection of the target and extraction of its center pixel location should be performed automatically for each calibration frame.

Using fixed cameras with a single moving target in the calibration frames reduces the complexity of the object detection problem. Hence the detection and localization steps at each calibration frame can be performed in an easy and reliable manner by the following algorithm:

- For each camera the mean of the calibration frames are obtained. Each mean image approximately provides the static background scene for the corresponding camera.
- The background image is subtracted from each calibration frame to obtain the target position at that frame.
- This difference image is converted to binary scale. Dilation and erosion operations are applied consecutively. Then,
  - if only a single blob exists, the pixels within the blob are assumed to correspond to the target projection. The center of the target (in pixel coordinates) is calculated by averaging the pixel coordinates.
  - if there are more than one blob, the operator is asked to select the target by clicking on the image that is being processed, so that the blob corresponding to the target can be separated from the others and averaging the pixels on that blob yields the target center.
  - else if there are no blobs left, the localization can not be achieved.

This algorithm can be used to distinguish any moving target on a constant background with subpixel accuracy. The examples of a mean, standard deviation, difference and binary images are shown in 4.2.

Note that if colored cameras are used, then the target color can particularly be chosen different from the background for easier detection. For example if the dominant surrounding color is green then the target can be chosen as red colored ball, and the given algorithm can exactly

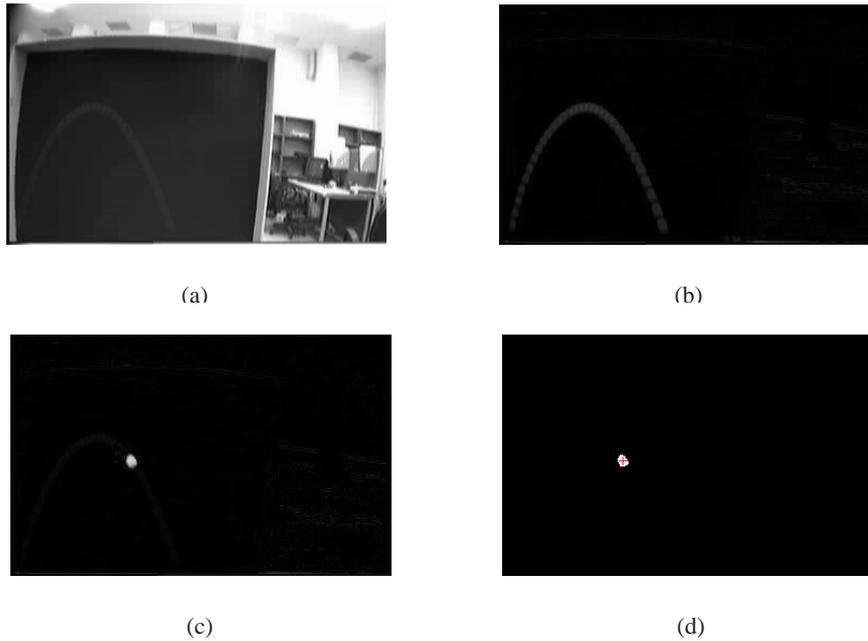


Figure 4.2: Detection of the target calibration object. (a) Mean image of the calibration frames. (b) Standard deviation image of the calibration frames. (c) Threshold image of a calibration frame. (d) Binary image of a calibration frame. The center of the target is also marked.

be applied in the red band. Such an algorithm is used by Svoboda et al in [29] for automatic detection of LED projections.

When the localization of the target is not achieved no measurement is available for the corresponding frame. Hence the measurement update equations in the EKF algorithm can not be executed at that time step. But the state can still be updated and the filter can progress to the next time step. So missing of the target position in a few frames is not a big deal since omission of the measurement update once in a while does not practically prevent the filter to converge.

### 4.3 Definition of Super-State

The motion model, the camera model and the detection of the target in the calibration frames are explained in the previous parts. From now on the transformation of the problem into the SSPE framework is demonstrated. In section 4.3.1 the external camera parameters are defined and the base state is augmented with these unknown parameters, and in section 4.3.2 the state

and measurement equations are extended for the overall state.

### 4.3.1 Augmented System Representation

In the following discussion, the system is described and the augmented state representation is defined for the two camera case. The results can quite easily be generalized to a multi camera problem but this generalization is considered to be beyond the scope of this section.

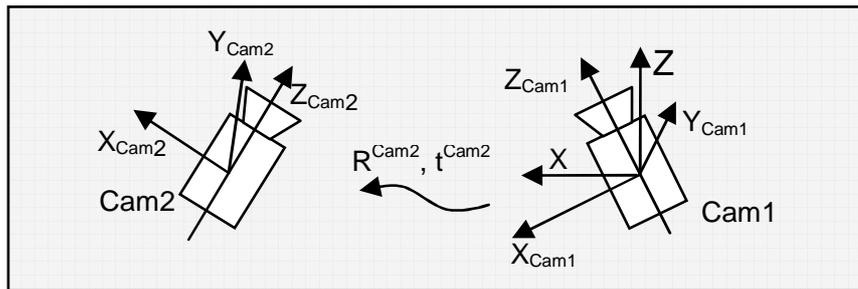


Figure 4.3: Two-camera system assumed for illustration purposes to be mounted on a fixture with the unknown rotation and translation between them consisting the external calibration parameters.

The system is composed of two arbitrarily placed cameras whose internal calibration parameters are predetermined. For illustration of the basic ideas, suppose the two cameras are rigidly mounted on a fixture as shown in 4.3. Assume the cameras are labeled as Cam1 and Cam2. Our aim is to find the unknown but constant rotation and translation of the cameras with respect to the basis frame.

Let  $x$ ,  $y$  and  $z$ -axes form a right handed reference frame that we call the *basis coordinate frame*. Without losing generality centers of the basis frame and the *Cam1 coordinate frame* are assumed to be coincident as shown in 4.4. Actually the necessity of a separate basis frame instead of using Cam1-frame as the reference frame arises from the fact that the gravitational pull is towards the center of the *earth*. Therefore  $y$ -axis should always coincide with the negative gravity vector direction in order to use Eqn. (4.3).

Note that as long as the center of the calibration target object thrown into the camera FOV is the only processed feature in the image, the rotation of the fixture around  $y$ -axis is undetectable. To visualize this, assume that the mounting fixture is rotated around  $y$ -axis. The position of the trajectory in the image plane will change. However, since the global loca-

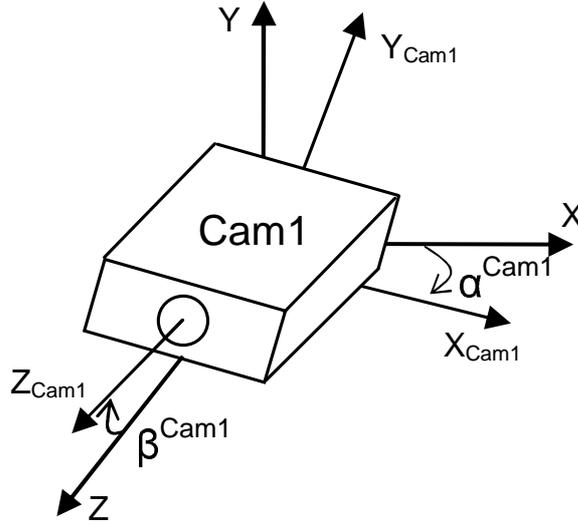


Figure 4.4: First camera shown on the *basis coordinate frame*. The relation between the basis frame and *Cam1 coordinate frame* is represented with the rotation angles  $\alpha^{c1}$  and  $\beta^{c1}$  around  $z$  and  $x$ -axes respectively.

tion of this trajectory is also unknown (except for the direction of the gravity vector which is observable from the trajectory) we cannot deduce this global rotation angle. Therefore, we conclude that the measurements taken from the target are invariant of the rotation around  $y$ -axis.

On the other hand, a rotation of Cam1 around  $z$  and  $x$ -axes will cause the object to accelerate at a different direction in the image plane measurements, indicating the global direction of gravity. Therefore these rotations of the fixture,  $\alpha^{c1}$  and  $\beta^{c1}$ , are measurable and are calibration unknowns of the process which should be determined. Rotation and translation of Cam2, on the other hand, is completely independent from the basis frame, and therefore all these 6 parameters (3 rotations and 3 translations) are calibration parameters that are also to be determined.

The EKF algorithm can be used in order to track the state of the calibration object and simultaneously estimate the static unknown parameters of the camera system. Dynamic object state,  $\mathbf{s}^{dyn}$ , can be augmented with the static camera parameters to form a super-state, which will be used as the state vector in the EKF algorithm formulation.

The calibration object state,  $\mathbf{s}^{dyn}$ , is already defined in Eqn. (4.5). Now, let  $\mathbf{s}^{sta}$  represent the static calibration parameters. Then the augmented super-state is the concatenation of  $\mathbf{s}^{dyn}$  with  $\mathbf{s}^{sta}$  as given in Eqn. (4.10). This augmentation procedure enables us to use such a powerful

noisy state estimation tool, EKF, for static parameter estimation.

$$\mathbf{s} = \begin{pmatrix} \mathbf{s}^{dyn} \\ \mathbf{s}^{sta} \end{pmatrix}, \quad (4.10)$$

$$\mathbf{s}^{sta} = (\alpha^{c1} \beta^{c1} \alpha^{c2} \beta^{c2} \Gamma^{c2} C_x^{c2} C_y^{c2} C_z^{c2})^T \quad (4.11)$$

where  $\alpha^{c1}$  and  $\beta^{c1}$  are the rotation angles of Cam1 around  $z$  and  $x$ -axes respectively;  $\alpha^{c2}$ ,  $\beta^{c2}$  and  $\Gamma^{c2}$  are the rotation angles of Cam2 around  $z$ - $x$ - $z$ -axes respectively, and finally  $C_x^{c2}$ ,  $C_y^{c2}$  and  $C_z^{c2}$  represent the center location of Cam2 with respect to the basis frame.

### 4.3.2 Augmented System Equations

We will now define the system state and measurement equations in a state space representation which are necessary for the EKF implementation. Since the camera parameters are invariant in time, their update matrix can be represented with an  $8 \times 8$  identity matrix and the overall state update equation can be written as

$$\mathbf{s}_{k+1} = \hat{\mathbf{A}}\mathbf{s}_k + \hat{\mathbf{u}} \quad (4.12)$$

where

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & \mathbf{0}_{6 \times 8} \\ \mathbf{0}_{8 \times 6} & \mathbf{I}_{8 \times 8} \end{pmatrix}, \quad \hat{\mathbf{u}} = \begin{pmatrix} \mathbf{u} \\ \mathbf{0}_{8 \times 1} \end{pmatrix} \quad (4.13)$$

The measurement equations (4.14) and (4.15) are directly written by using the pinhole camera model which is explained in section 4.1.2. The tracked point (i.e. the center of the calibration object) is projected into the camera planes; hence our measurements are the pixel locations of the projection. 3D target position  $(X \ Y \ Z)^T$  is projected to give us corresponding 2D measurements  $(u^{c1} \ v^{c1})^T$  and  $(u^{c2} \ v^{c2})^T$ .

$$\lambda \begin{pmatrix} u^{c1} \\ v^{c1} \\ 1 \end{pmatrix}_k = \mathbf{K}_1 \left[ \mathbf{R}^{c1}(\alpha, \beta) \ \mathbf{0}_{3 \times 1} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_k \quad (4.14)$$

$$\lambda \begin{pmatrix} u^{c2} \\ v^{c2} \\ 1 \end{pmatrix}_k = \mathbf{K}_2 \left[ \mathbf{R}^{c2}(\alpha, \beta, \Gamma) \ \hat{\mathbf{t}}^{c2} \right] \begin{pmatrix} Z \\ Y \\ Z \\ 1 \end{pmatrix}_k \quad (4.15)$$

$$\hat{\mathbf{t}}^{c2} = -\mathbf{R}^{c2}(\alpha, \beta, \Gamma) \begin{pmatrix} C_x^{c2} \\ C_y^{c2} \\ C_z^{c2} \end{pmatrix} \quad (4.16)$$

$\mathbf{K}_1$  and  $\mathbf{K}_2$  are the internal calibration matrices for Cam1 and Cam2.  $\mathbf{R}^{c1}$  and  $\mathbf{R}^{c2}$  stands for the rotation matrices of Cam1 and Cam2 with respect to the *basis frame*. Finally  $\hat{\mathbf{t}}^{c2}$  is the translation of Cam2 in the world coordinate frame and it can be formulated in terms of the state variables as in Eqn. (4.16).

## CHAPTER 5

### Estimating the Super-State

In order to use the Kalman filter, the observations should be linear functions of the state and the next state should be a linear function of the previous state. In our problem although the state transition equation is a linear one, the measurement equations are highly nonlinear which removes the possibility of direct KF employment at its simplest form. Instead, an extended Kalman filter, which can handle the non-linearities of the observation equation, should be used.

Let the measurements  $(u^{c1} \ v^{c1})^T$  and  $(u^{c2} \ v^{c2})^T$  be concatenated to form the complete measurement vector,  $\mathbf{z}$ , as

$$\mathbf{z} = (u^{c1} \ v^{c1} \ u^{c2} \ v^{c2})^T \quad (5.1)$$

Then the disturbances affecting the process and observations should be modeled by adding random white Gaussian noises  $\mathbf{v}$  and  $\mathbf{w}$  to obtain the usual EKF relations

$$\mathbf{s}_{k+1} = \hat{\mathbf{A}}\mathbf{s}_k + \hat{\mathbf{u}} + \mathbf{w}_{k+1} \quad (5.2)$$

$$\mathbf{z}_{k+1} = h(\mathbf{s}_{k+1}, \mathbf{u}_{k+1}) + \mathbf{v}_{k+1}. \quad (5.3)$$

where  $h$  is obtained from (4.14), and (4.15). The process noise  $\mathbf{w}$  models the disturbances affecting the target motion such as air friction while the observation noise  $\mathbf{v}$  models the disturbances affecting the cameras such as possible vibrations during filming or other pixel errors. The process and observation noise covariances are given as

$$\mathbf{Q} = \text{diag}(q_1, q_2, \dots, q_{13}, q_{14}) \quad \text{and} \quad \mathbf{R} = \text{diag}(r_1, r_2, r_3, r_4). \quad (5.4)$$

For EKF implementation the Jacobian of  $h$  is computed by MATLAB 7.1. Note that together with the state transition matrix  $\hat{\mathbf{A}}$ , this Jacobian matrix  $\mathbf{H}$  determines how well internal states

of our system can be inferred by knowledge of its outputs, in other words the *observability* of the system. For that purpose numerical matrix entries can be regularly checked during the EKF iterations. The Jacobian calculation is a simple procedure, however since our observation equations are highly nonlinear the resultant  $\mathbf{H}$  matrix is extremely complex to give in the content of this thesis.

At this point all the necessary equations and matrices are obtained to reduce the problem to an EKF framework.

## 5.1 Standard EKF Algorithm

In most applications, EKF is used for tracking a platform in real time. Sensor measurements are continuously processed by the EKF together with the system dynamics and a suboptimal state estimation is performed. Even if the initial state of the system is not known exactly, the filter converges to the true state in time.

In our problem we have checked the convergence of the EKF by implementing the filter for the calibration of a two camera system in a simulation environment. Even though the super-state is set to a wrong initial value, EKF can yield accurate estimates of the static calibration parameters as well as the final target state. However, for the filter to converge to the true state, about 1000 filter iterations are performed and hence that many data points of the flying target is used during the iterations. Although in a simulation environment data points can be supplied as much as desired by increasing the fps rates of the cameras, for real camera setups that can not be done. For a standard 30 fps camera setup, it takes more than half a minute to capture 1000 frames. That is too long time for a ballistically moving target to stay in the joint-FOV of the cameras. For reasonable camera poses, when the calibration object is thrown into the joint-FOV it may stay there for about one second and only 30 measurements can be obtained by a 30 fps camera setup. How can we satisfy the EKF to converge with so few measurements?

In order to solve this convergence problem of EKF with limited number of calibration data, 2 different algorithms are proposed: Repeated EKF and Repeated Bidirectional EKF. As understood from their names the essence of these methods depends on using the same measurements repeatedly.

## 5.2 Repeated EKF Algorithm

Let us first divide the state,  $\mathbf{s}$  into its sub-parts:

$$\mathbf{s} = \begin{pmatrix} \mathbf{X} & \mathbf{V} & \mathbf{C} \end{pmatrix}^T \quad (5.5)$$

where  $\mathbf{X}$  corresponds to the position,  $\mathbf{V}$  corresponds to the velocity of the calibration object, and  $\mathbf{C}$  stands for the calibration parameters. Let us continue with the assumption that a target is thrown into the joint FOV of a two camera system, and the ballistic trajectory of the target is simultaneously captured by each camera yielding 30 frames to be used for calibration. Starting from a wrong initial state, the calibration parameters would not be obtained accurately after only 30 filter iterations.

The very first idea to solve this problem is to repeatedly use the measurements at hand. Let us define the *filter* and *throw* iterations:

- *Filter Iteration*: The extraction of the next state from the present state; in other words a single iteration of the EKF
- *Throw Iteration*: All filter iterations for one set of measurement data (i.e., 30 iterations of the filter).

After the completion of one throw iteration the state will not probably converge, but it will be estimated to some extent. We suggest keeping the calibration parameters at their final values and starting the extended Kalman filter again and again (Figure 5.1). So, even though the same observations are used, the state can be estimated better and better. Since the calibration is performed off-line, there is enough time for the repetition of the iterations.

The idea here is to make many throw iterations and hope for the calibration parameters to converge to their real values in a reasonable time. The calibration parameters are kept and provided to the next throw iteration while the target state is reset to its initial value, so the path followed by the ball is estimated better and better since the calibration parameters approach towards their real values and vice versa. This “*keep the static parameters, reset the dynamic state*” idea forms the essence of the **repeated EKF** (R-EKF) algorithm.

It should be reminded that simple KF yields the optimal solution with a single use of the measurements. So repeated use of the same measurements can not improve the results even

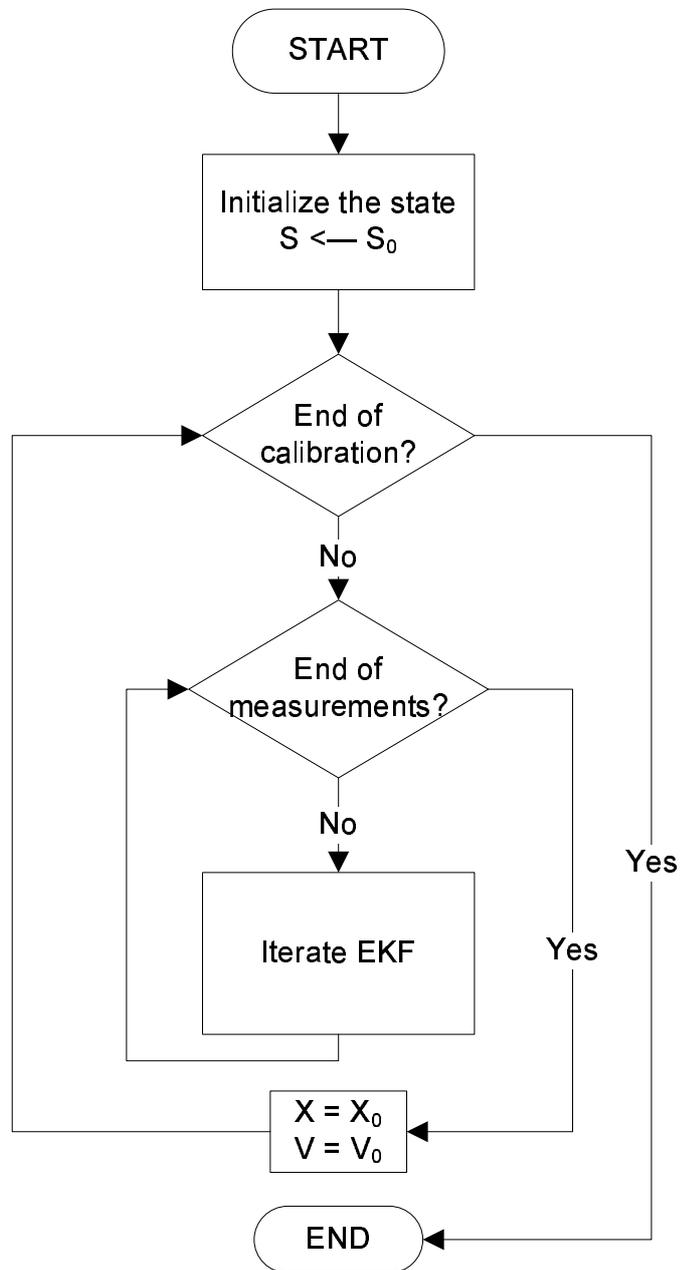


Figure 5.1: The flow chart for the repeated EKF algorithm.

after smoothing the state. On the other hand the EKF linearizes the non-linear functions at the predicted state, hence linearization performance can be improved by smoothing the state. Therefore even by using the same measurements, better estimates can be obtained by an EKF due to this improvement in the linearization.

The drawback of the R-EKF algorithm is that the results are highly dependent on the initial uncertainty of the dynamic state. Actually not only the camera parameters but also the target position and velocity are not available at the beginning, because unfortunately the target is thrown at an arbitrary position with an unknown velocity. The initial state of the calibration target is not estimated during the iterations, but always the originally given  $[\mathbf{X}_0 \mathbf{V}_0]^T$  vector is used. So the estimate for this initial state vector should be provided via external measurements as in [50]. Human factor and measurement errors which may badly affect the calibration results is inevitable with this approach.

### 5.3 Repeated Bidirectional EKF Algorithm

The second idea is to implement EKF in both forward and backward directions. The calibration object state and the camera parameters are again estimated repeatedly, but this time in two directions. In other words the target motion is first tracked in the forward direction and then in the reverse direction. This algorithm remedies the main problem of R-EKF since the initial target state is predicted when the reverse motion estimation is performed.

The flow sequence of the **repeated bidirectional EKF** (RB-EKF) algorithm is shown by the flowchart in Figure 5.2. The augmented state is set to  $\mathbf{s}_0$  initially and *forward throw* estimation is performed till all the frames are swept one time. Then the target trajectory is estimated from backwards by using the measurements in reverse order. Note that the velocity vector should be negated at the end of a *forward* and *backward throw* iteration while the position vector and camera parameters remain same.

RB-EKF yields both the dynamic and static states with negligible error in a short time. As stated beforehand the accuracy of the estimations depends on the initial state uncertainty, hence being able to estimate the initial state reduces this dependency. Removing the necessity of measuring the initial target state and hence avoiding the human related errors, this method not only provides an easily applicable calibration, but it also reduces the final error.

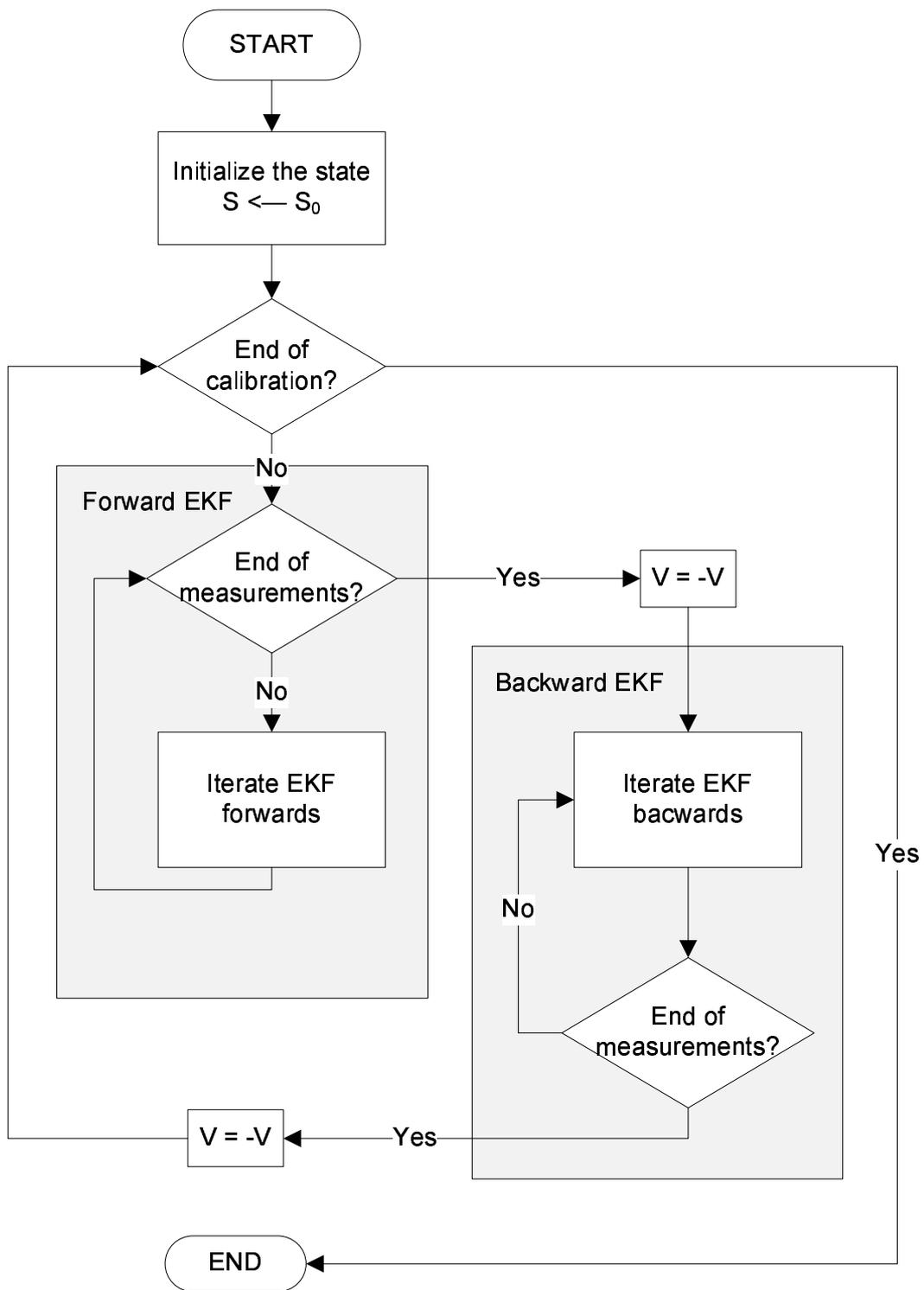


Figure 5.2: The flow chart of the repeated bidirectional EKF algorithm.

## CHAPTER 6

### Simulation Experiments

In this chapter the experiments which are performed on a computer simulation environment are presented. The proposed calibration method is tested on a simulation platform prepared in MATLAB7.1. For data generation, instead of the real camera images of a dynamically moving target, the dynamic motion model and pinhole camera model are used to generate simulated data points.

The experiments are chosen to identify the scenarios that can be encountered in real life problems. First a target in free fall is used for calibration with R-EKF and RB-EKF algorithms in sections 6.1 and 6.2 respectively. Section 6.3 focuses on the comparison of R-EKF with RB-EKF when the initial target state is not known. White Gaussian random process and measurement noises are also added on the measurement data for a more realistic evaluation. In section 6.4 the performance of RB-EKF is investigated for a ballistic trajectory and then multiple trajectories are used for calibration in section 6.5. Finally a general discussion of the simulation experiments is given in section 6.6.

#### 6.1 Experiment 1: R-EKF / Free Fall Motion

In our first simulation experiment we try to initially show that R-EKF method can yield camera poses under some facilitative but restrictive assumptions. As discussed in Chapter 5, the performance of the R-EKF algorithm is highly dependent on the initial uncertainty of the dynamic state. In order to use R-EKF efficiently, the initial state of the calibration target should be measured and provided to initialize the EKF, which makes R-EKF an impractical algorithm to use on the field. Since this experiment is a preliminary testing of the concept, we assume

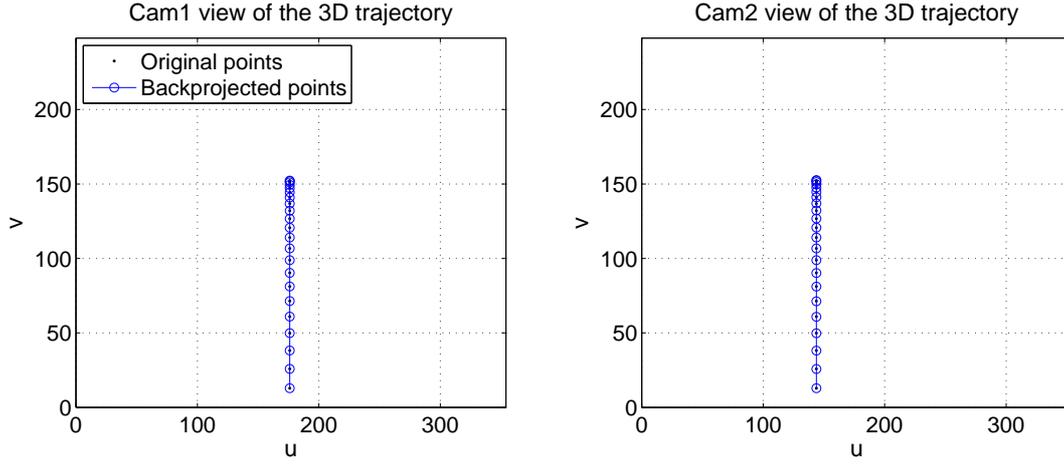


Figure 6.1: The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm.

that the initial target state is known even though such an assumption is not realistic.

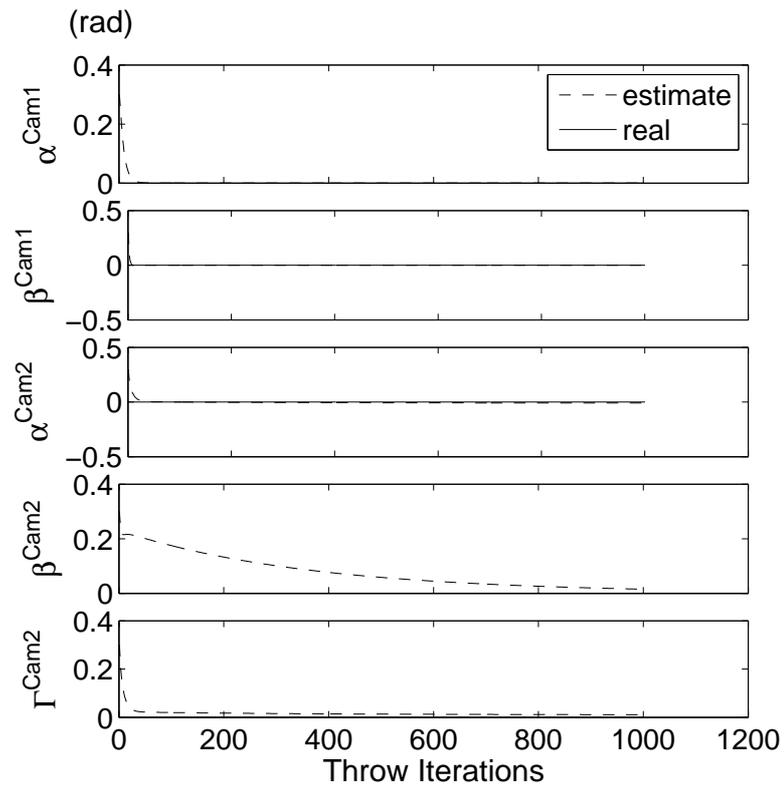
For a real case the initial position of the calibration target can approximately be measured by means of a ruler although that will not be practical. However to obtain its initial velocity is even harder. Yet a *free falling* target may reduce the unknowns since the initial velocity of a free falling object is a well known  $\vec{0}$  m/s vector. So, a target can be dropped into the joint FOV of a multi camera system and the captured image sequence can be used for the external calibration. In this experiment we try to inspect the convergence performance of the R-EKF algorithm by using the synthetic data of a free falling target.

In our simulation setup two cameras are placed in parallel with 0.2 m distance between them and zero rotation with respect to the basis frame. Such a configuration is similar to a human's vision system, so for easy understanding one can assume that a stereo camera is mounted on a planar platform which is placed parallel to the ground. When a point mass falls freely in front of the camera setup, the image sequences obtained by these two cameras are shown in Figure 6.1.

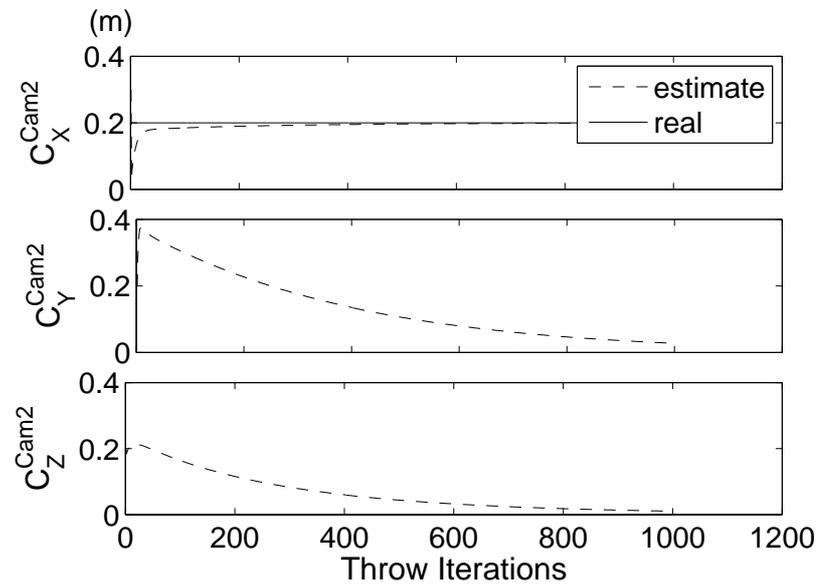
The target is dropped from 0.25 m above, 1.8 m in front of Cam1. So its initial condition vector is equal to

$$\mathbf{X}_0 = [0 \ 0.25 \ 1.8]^T, \quad \mathbf{V}_0 = [0 \ 0 \ 0]^T.$$

The camera parameters are all equal to zero except the  $x$  position of Cam2 center, due to the



(a)



(b)

Figure 6.2: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

0.2 m gap between the cameras in  $x$  direction. Hence the original parameter vector is equal to

$$\mathbf{C} = [0 \ 0 \ 0 \ 0 \ 0 \ 0.2 \ 0 \ 0]^T.$$

The actual initial target state,  $[\mathbf{X}_0 \ \mathbf{V}_0]$ , is exactly used for initializing the R-EKF, so

$$\hat{\mathbf{X}}_0 = [0 \ 0.25 \ 1.8]^T, \quad \hat{\mathbf{V}}_0 = [0 \ 0 \ 0]^T$$

and the initial estimate vector for the camera parameters is assumed to be

$$\hat{\mathbf{C}}_0 = [0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.2 \ 0.2]^T.$$

As illustrated in Figure 5.1 during the throw iterations the initial state of the target  $[\mathbf{X}_0^T \ \mathbf{V}_0^T]^T$  is repeatedly used while the calibration parameter estimates  $\hat{\mathbf{C}}$  converge towards their actual values  $\mathbf{C}$ . The convergence curves of the rotation and translation parameters are shown in Figure 6.2

The evaluation of 1000 throw iterations by R-EKF algorithm has taken for about 70.9 seconds with a hyper-threaded Pentium4, 3.0GHz CPU. The camera parameters are estimated as

$$\hat{\mathbf{C}} = [0.0000 \ 0.0000 \ -0.0092 \ 0.0151 \ 0.0109 \ 0.1992 \ 0.0271 \ 0.0101]^T$$

which is almost same as the original parameter vector  $\mathbf{C}$ . The re-projection pixel error is computed as  $6.5712 \times 10^{-4}$  pixels for Cam1 and 0.1056 pixels for Cam2. Note that this is the error in the image plane. In a simulation since the real parameters are known we can obtain the norm of the difference between the real camera parameters and the estimated ones. The average of the rotation and translation errors for both cameras are obtained as 0.0208 radians and 0.0290 meters respectively.

In this experiment we have shown that if the initial state of the calibration object is measurable, then R-EKF algorithm can yield accurate results. However, since the initial state of the target is not estimated during iterations, the results are highly dependent on the initial state uncertainty. This is the major drawback of the R-EKF method. Also note that throughout this experiment we assume there are no disturbances which would cause noisy measurements. But unfortunately in real life unwanted disturbances do exist. That's why without noise analysis the actual performance of the R-EKF algorithm can not be fully understood. In fact due these disturbances we even may not be able to obtain the initial state of the calibration target accurately which may tragically diminish the success of the R-EKF method.

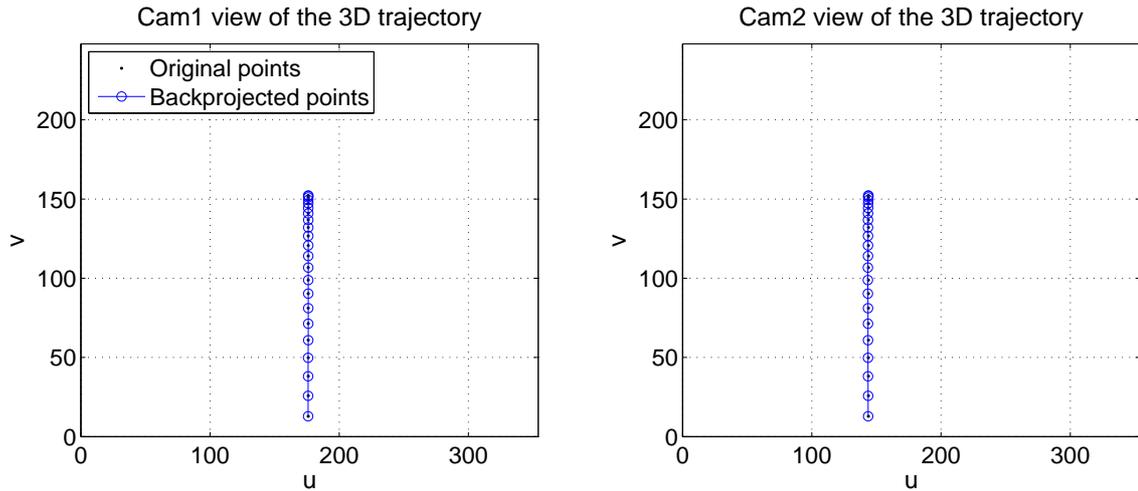


Figure 6.3: The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

## 6.2 Experiment 2: RB-EKF / Free Fall Motion

In this section the first experiment is repeated for the RB-EKF algorithm. We try to show that RB-EKF converges under facilitating assumptions. The same orientation of the camera setup is used again to capture the same free falling motion of the target. Hence  $\mathbf{X}_0$ ,  $\mathbf{V}_0$  and  $\mathbf{C}$  vectors have the same values as in section 6.1. The initial estimate of the target state and the camera parameter vectors  $\hat{\mathbf{X}}_0$ ,  $\hat{\mathbf{V}}_0$ ,  $\hat{\mathbf{C}}_0$  are also provided to the RB-EKF algorithm as in section 6.1.

The original and re-projected points are shown in Figure 6.3 and the convergence curves of the RB-EKF algorithm are plot in Figure 6.4. Note that RB-EKF algorithm estimates not only the camera parameters but also the initial target state. However in this experiment the initial target state is exactly provided and since trying to estimate an already given parameter may lead to some error, the resultant calibration error of the RB-EKF is slightly more than that of the R-EKF.

It takes 136.7 seconds to complete 1000 throw iterations for RB-EKF algorithm. This is nearly twice as much as it takes with the R-EKF method, because one throw iteration of RB-EKF includes both the forward and backward iteration of the target movement. The calibration

vector is estimated as

$$\hat{\mathbf{C}} = [0.0004 \ 0.0450 \ -0.0120 \ 0.0454 \ 0.0102 \ 0.2048 \ 0.0003 \ 0.0006]^T$$

which is very close to the original parameter vector  $\mathbf{C}$ . The re-projection error is computed as 0.0227 pixels for Cam1 and 0.1613 pixels for Cam2. The rotation and translation errors are obtained as 0.0658 radians and 0.0048 meters respectively.

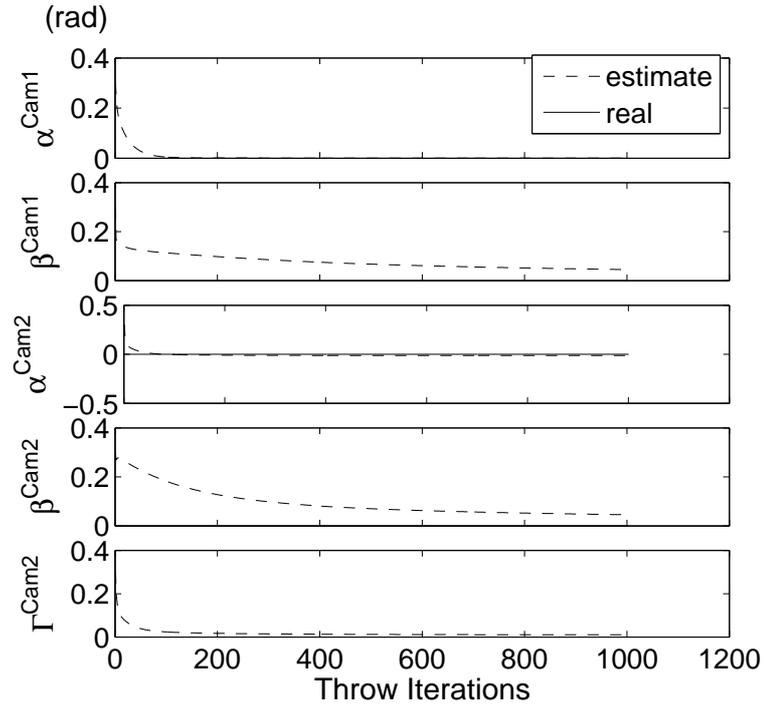
In this experiment we have shown that if the initial state of the calibration object can be measured accurately, RB-EKF algorithm can yield the parameter estimates accurately. Note that RB-EKF can also handle initial uncertainty since the initial target state is also estimated. That makes RB-EKF a practical method that may be used on the field applications because very accurate initial state measurement is quite difficult to obtain in real cases. Even when one tries to measure the position of the target by means of a ruler, the uncertainty of the initial state is unavoidable. Therefore we should further test RB-EKF method under more realistic assumptions.

### 6.3 Experiment 3: Comparison of R-EKF and RB-EKF

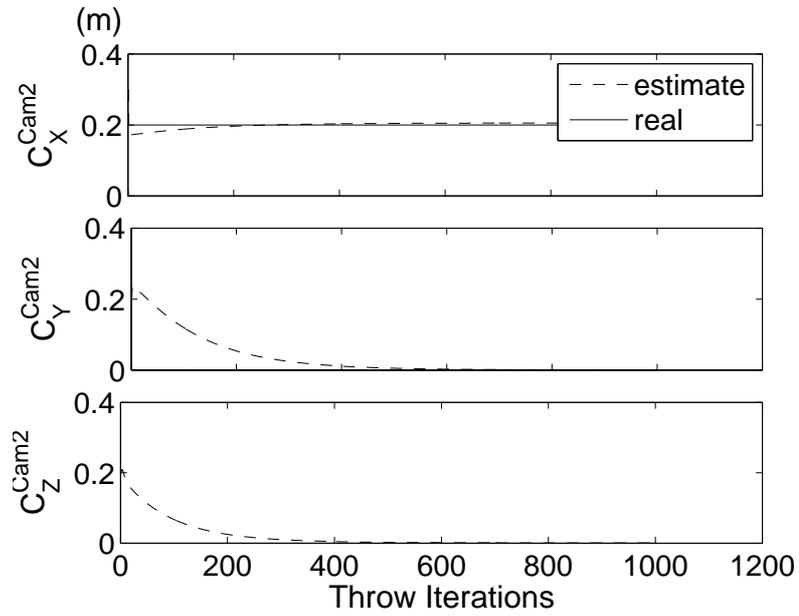
In the previous experiments we have mentioned that the initial state may not be accurately measurable and also other disturbances may affect the dynamic motion of the target object as well as its projections on to the image plane. The external disturbances may cause the actual target motion deviate from the dynamic target motion model and also the actual camera measurements deviate from the ones that would be obtained by the pinhole camera model. In this part the R-EKF and RB-EKF algorithms are compared under reasonable initial uncertainty and noise effects.

Introduction of an additional measurement system to obtain an accurate initial estimate of the target state would result in a complicated and pointless calibration procedure. Because if one already has the necessary equipment to obtain the target pose in an accurate manner, then that may also be used to obtain the camera poses. So a practical and efficient calibration method, which uses the EKF for parameter estimation, should be able to tolerate the uncertainties of the initial target state.

In this part the camera poses and the target trajectory are used as in section 6.1, but this time



(a)



(b)

Figure 6.4: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

the initial target state and the camera poses are assumed to be guessed by the operator by only inspecting the system. Therefore the initial position uncertainty must be tolerated by at least half a meter, because that much mistake is fairly reasonable for human prediction. The initial velocity of the free falling target is assumed to be  $\vec{0}$  m/s. Hence

$$\hat{\mathbf{X}}_0 = [0 \ 0 \ 1.5]^T, \quad \hat{\mathbf{V}}_0 = [0 \ 0 \ 0]^T$$

vectors may be accepted as reasonable initial estimates for the target state. Then the initial super-state estimate is given as

$$\mathbf{s}_0 = [\hat{\mathbf{X}}_0^T \ \hat{\mathbf{V}}_0^T \ \hat{\mathbf{C}}_0^T]^T \quad (6.1)$$

where the initial camera parameter vector  $\hat{\mathbf{C}}_0$  is taken same as the first experiment.

In order to model the disturbances, process and observation noises are added while obtaining the camera views of the falling object. The target position and the camera measurements are disturbed with White Gaussian noises. The process and observation noise variances are set to 0.002 m and 0.1 pixels respectively.

The measurements and the re-projected points obtained after 1000 throw iterations with R-EKF and RB-EKF methods are shown in Figure 6.5. The convergence curves of the R-EKF and RB-EKF algorithms are illustrated in Figure 6.6 and Figure 6.7. The resultant errors for both algorithms are presented in Table 6.1.

The final estimation vector of the camera parameters are obtained as

$$\hat{\mathbf{C}}^{R-EKF} = [0.0015 \ 0.1916 \ -0.0125 \ 0.1263 \ 0.0318 \ 0.1852 \ -0.0683 \ -0.1170]^T$$

$$\hat{\mathbf{C}}^{RB-EKF} = [0.0193 \ 0.0828 \ 0.0194 \ 0.0824 \ -0.0223 \ 0.2345 \ -0.0021 \ 0.0053]^T.$$

The initial target state is also estimated by RB-EKF as

$$\hat{\mathbf{X}}_0 = [-0.0045 \ 0.1173 \ 2.1923]^T, \quad \hat{\mathbf{V}}_0 = [0.0379 \ -0.1758 \ -0.7339]^T.$$

RB-EKF has clearly better performance in this more realistic example. The final camera parameters obtained by RB-EKF turned up to be closer to their actual values compared to R-EKF. The final re-projection, angle and translation errors for RB-EKF are obtained to be quite smaller than those for R-EKF as seen in Table 6.1. Also when the convergence curves in Figure 6.6 and Figure 6.7 are compared, RB-EKF seems to be the superior method between the two. Therefore it is concluded that RB-EKF should be preferred over R-EKF when the initial target state is not known accurately.

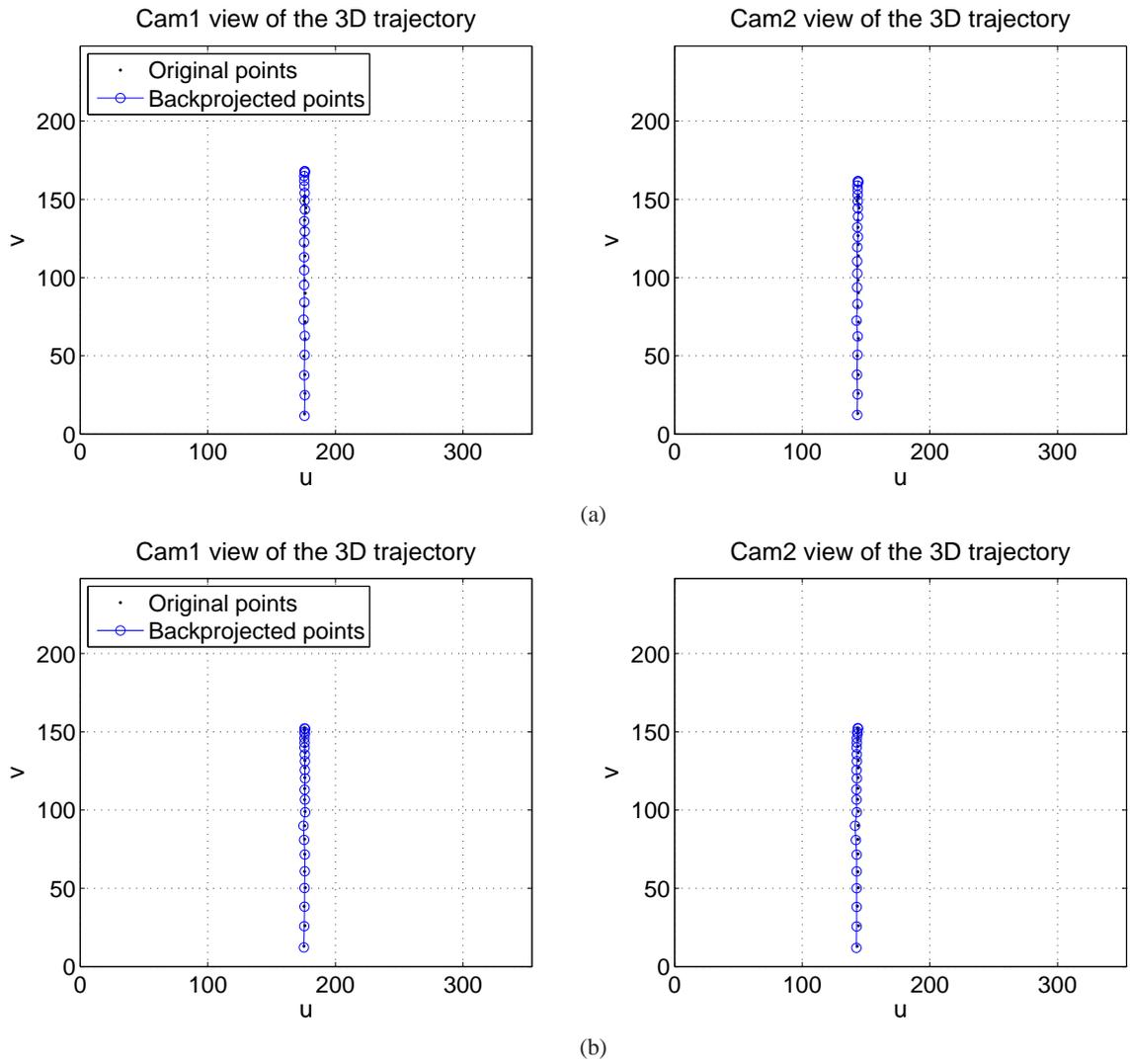
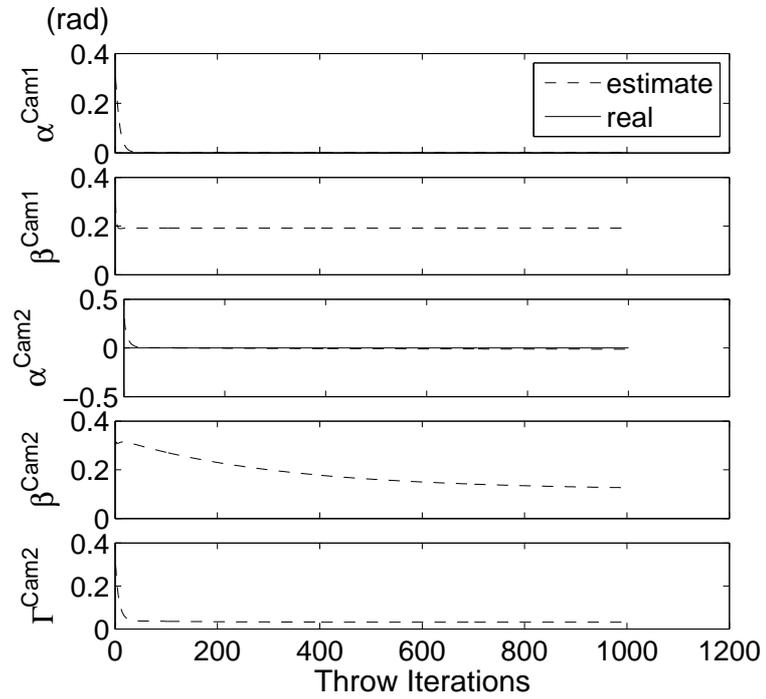


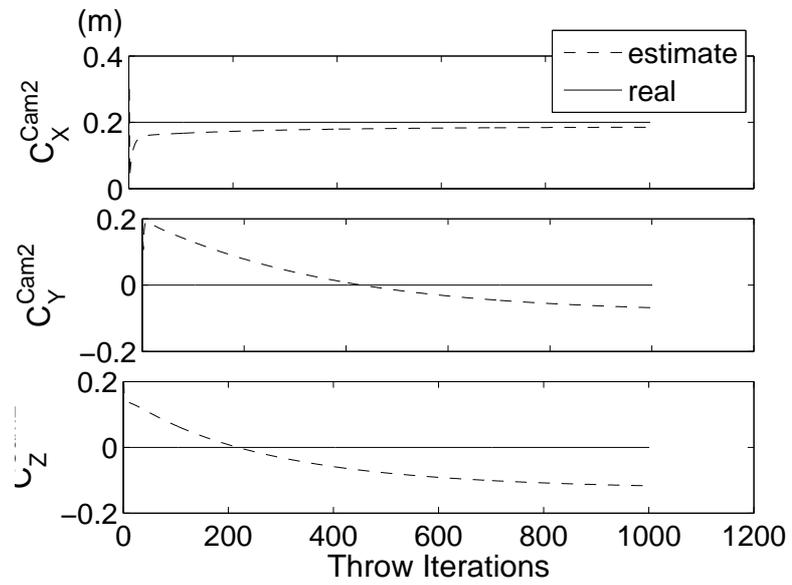
Figure 6.5: The synthetic images of a free falling target captured by a stereo camera system. The motion and observation noise variances are taken as 0.002 m and 0.1 pixels. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by (a) Repeated EKF; (b) Repeated Bidirectional EKF.

Table 6.1: The final re-projection, angle and center errors for R-EKF and RB-EKF

	$Cam1_{Reproj\_Error}$	$Cam2_{Reproj\_Error}$	Angle Error	Center Error
R-EKF	5.8282	3.5549	0.2320	0.1363
RB-EKF	0.4822	0.7799	0.1220	0.0350

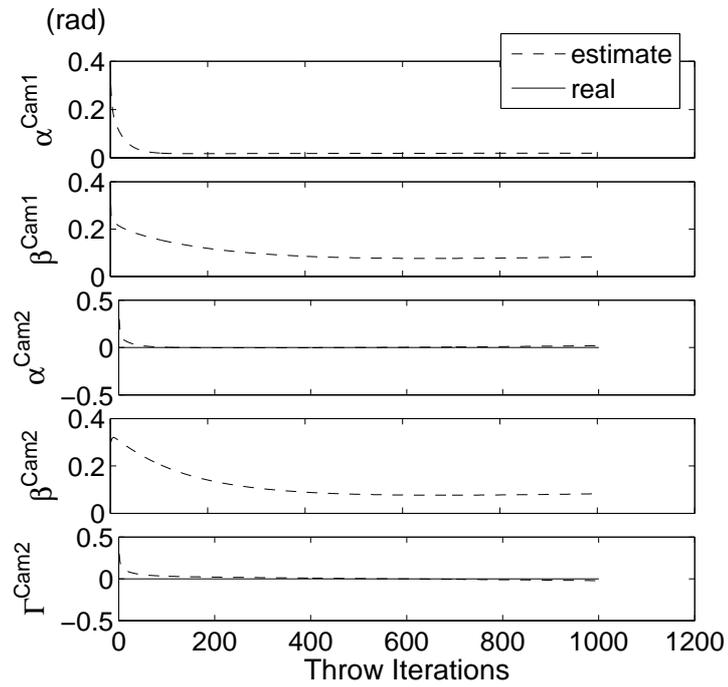


(a)

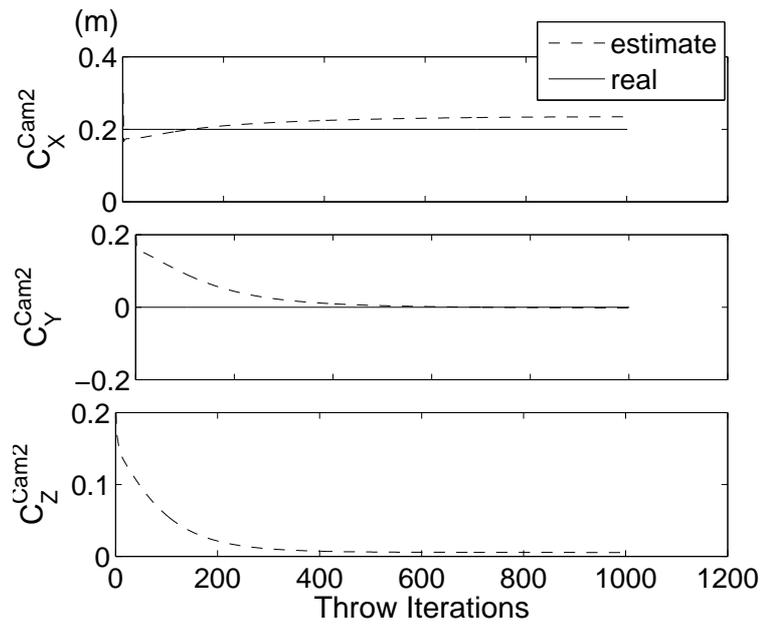


(b)

Figure 6.6: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations. Since the initial target state is not truly provided the calibration parameters are not accurately obtained (a) Rotation angles; (b) Translation parameters.



(a)



(b)

Figure 6.7: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations. Even though the initial target state is not truly provided the calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

## 6.4 Experiment 4: RB-EKF / Ballistic Motion

Until now we have focused on a free falling target scenario for calibration since the initial velocity uncertainty is minimized in this manner. However such one dimensional motion of the target may cause singularities in the estimation process. For example different orientations of the cameras may yield the same calibration frames. Assume two cameras are placed on a circle which is parallel to the ground and let principle axes of the cameras be congruent with the diameter of the circle. When the target is dropped towards the center of this circle all the cameras on the circle yields the same image sequence. So there may be more than one possible orientation for the cameras when a free falling target is used for camera pose estimation.

Another drawback of the free fall case is that only a small part of the camera image planes are covered by the vertical motion of the target. But the image planes should be covered as much as possible by the calibration object in order to estimate the camera characteristics better. To overcome these problems we move into a more information gathering motion now that RB-EKF can handle arbitrary initial conditions. So instead of dropping the calibration object, we throw it into the joint camera FOV with an initial velocity, hence the object follows a 2D ballistic trajectory. Therefore not only the singularities will be prevented but also greater regions on the image planes will be covered.

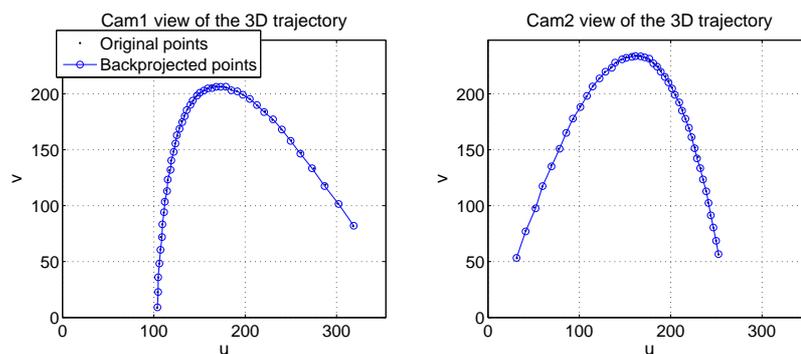


Figure 6.8: The synthetic images of the ballistic trajectory of a thrown target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

Since the initial velocity of the target can not be measured easily, it should be estimated

together with the calibration parameters. RB-EKF can estimate the initial target state and as shown in section 6.3, it yields more accurate results than R-EKF when the initial target state is not accurately known. So for the *ballistic throw* cases only the results obtained by RB-EKF are presented.

Suppose that two cameras are placed at the diagonal corners of a square whose edge length is equal to 1.5 m as shown in Figure 6.10. Also assume that Cam2 is placed 0.2 m above Cam1 level. Let the orientations of the cameras be arranged to make their principal axes nearly perpendicular to each other. Then assume that the operator throws a target into the joint camera FOV. An example initial target state for this setup and a calibration parameter vector can be given as

$$\mathbf{X}_0 = [0.5 \ 0 \ 1]^T, \quad \mathbf{V}_0 = [-1.2 \ 4 \ 1.2]^T,$$

$$\mathbf{C} = [-0.2 \ -0.2 \ -\pi/2 \ \pi/2 \ \pi/2 \ 1.5 \ 0.2 \ 1.5]^T.$$

and reasonable initialization vectors may be selected as

$$\hat{\mathbf{X}}_0 = [0.7 \ 0.2 \ 0.7]^T, \quad \hat{\mathbf{V}}_0 = [-1 \ 3 \ 1]^T,$$

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ -1.5 \ 1.5 \ 1.5 \ 1 \ 0 \ 1]^T.$$

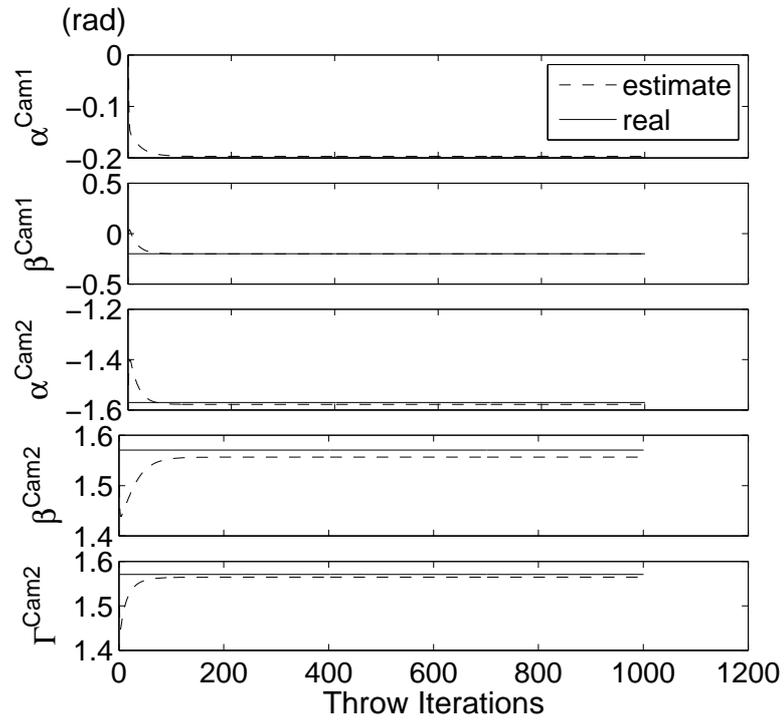
We disturb the target position and the camera measurements by White Gaussian noises with variances set to 0.002 m and 0.1 pixels. Starting from the super-state,  $\mathbf{s}_0 = [\hat{\mathbf{X}}_0^T \ \hat{\mathbf{V}}_0^T \ \hat{\mathbf{C}}_0^T]^T$ , after 1000 throw iterations RB-EKF yields the initial target state and the camera parameters as

$$\hat{\mathbf{X}}_0 = [0.4960 \ 0.0002 \ 0.9931]^T, \quad \hat{\mathbf{V}}_0 = [-1.1917 \ 4.0040 \ 1.2121],$$

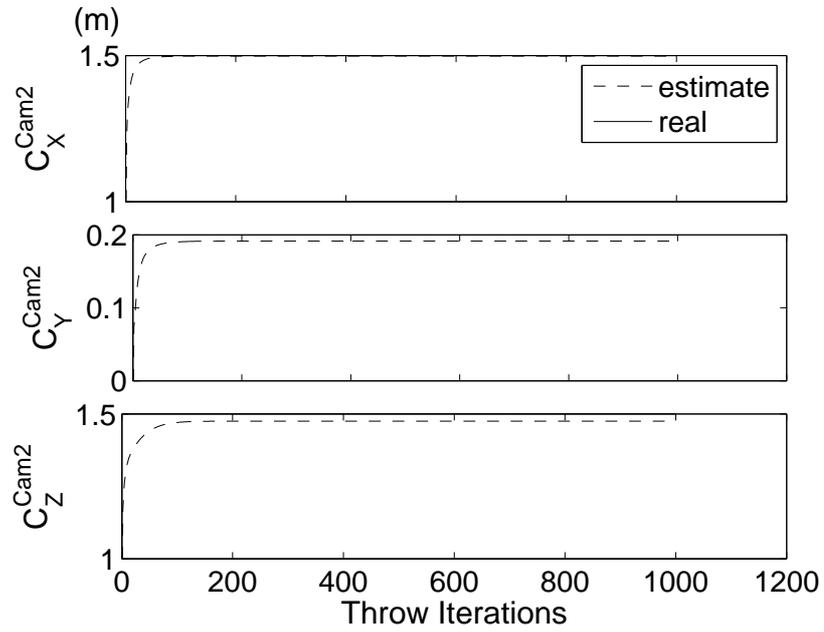
$$\hat{\mathbf{C}} = [-0.1972 \ -0.2024 \ 1.5784 \ 1.5566 \ 1.5645 \ 1.4982 \ 0.1913 \ 1.4750].$$

All the re-projected points are close to their original measured locations as shown in Figure 6.8. Also note that the convergence is actually achieved with less than 200 throw iterations as illustrated in Figure 6.9.

The re-projection error is computed as 0.5177 pixels for Cam1 and 0.4998 pixels for Cam2. The rotation and translation errors are obtained as 0.0177 radians and 0.0265 meters respectively.



(a)



(b)

Figure 6.9: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

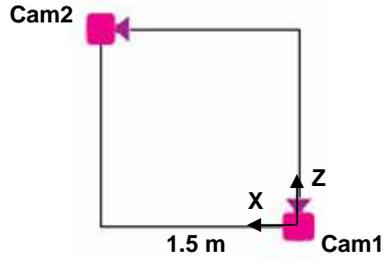


Figure 6.10: The cameras are placed perpendicular to each other and at the diagonal corners of a square with 1.5 meter edge length.

In this example we have demonstrated that RB-EKF can give successful results for both the camera pose estimates and the target trajectory even when the initial target state is not measurable. So, a ballistic motion can be used for calibration by RB-EKF method. Hence the calibration results will be improved since the singularities in the free fall case are prevented and the image plane is better covered by the target.

## 6.5 Experiment 5: RB-EKF / Multiple Trajectories

For pose estimation of a two camera setup we have defined 8 parameters to be estimated. With the additional 6 variables corresponding to the target state, a total of 14 parameters should be predicted. By using a single target trajectory however, only a small number of measurements can be obtained. Although repeated evaluation of EKF provides convergence with a limited number of measurements, more calibration data would probably improve the estimations.

Another drawback of a *single throw* scenario is that, the EKF may get stuck with a local minima, and hence the parameters may not be precisely predicted. To avoid such cases, multiple trajectories can be used instead of a single one. In other words the calibration object can be thrown a few times and various trajectories of the target can be captured. This surely provides additional calibration data that may improve the estimates. In this experiment the effect of using *multiple trajectories* during the calibration procedure is investigated.

Suppose that for the same camera orientation and target motion described in 6.4, the calibration object is thrown for a second time with the initial conditions

$$\mathbf{X}_0 = [-0.5 \ 0.25 \ 1.3]^T, \quad \mathbf{V}_0 = [1.2 \ 3.5 \ 1.2]^T.$$

Table 6.2: The final re-projection, angle and center errors for single and double throw cases

	Cam1 <sub>Reproj.Error</sub>	Cam2 <sub>Reproj.Error</sub>	Angle Error	Center Error
First Throw	0.5177	0.4998	0.0177	0.0265
Second Throw	0.5757	0.5296	0.0150	0.0350
Both Throws	0.5006	0.6152	0.0054	0.0122

Let their initial estimates be taken as

$$\hat{\mathbf{X}}_0 = [0 \ 0 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

We disturb the target position and the camera measurements by White Gaussian noises with variances set to 0.002 m and 0.1 pixels. When the calibration is performed by using the second trajectory only, after 1000 throw iterations RB-EKF yields

$$\hat{\mathbf{X}}_0 = [-0.5002 \ 0.2690 \ 1.3062]^T, \quad \hat{\mathbf{V}}_0 = [1.2140 \ 3.5089 \ 1.2027],$$

$$\hat{\mathbf{C}} = [-0.2044 \ -0.2116 \ 1.5650 \ 1.5647 \ 1.5707 \ 1.5171 \ 0.2302 \ 1.4952].$$

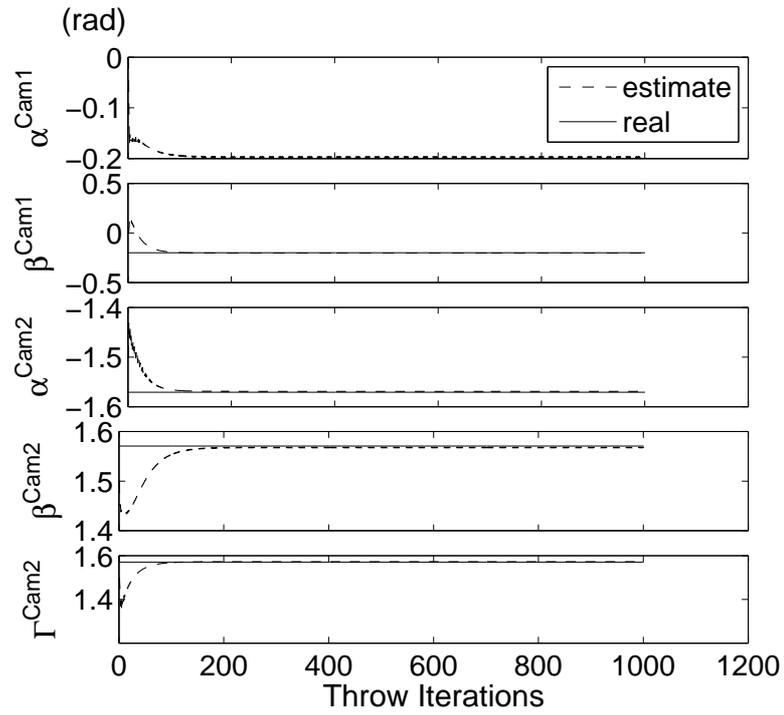
The re-projection error is computed as 0.5757 pixels for Cam1 and 0.5296 pixels for Cam2. The rotation and translation errors are obtained as 0.0150 radians and 0.0350 meters respectively.

When two of the trajectories are successively used to estimate the camera poses, after 1000 throw iterations RB-EKF yields

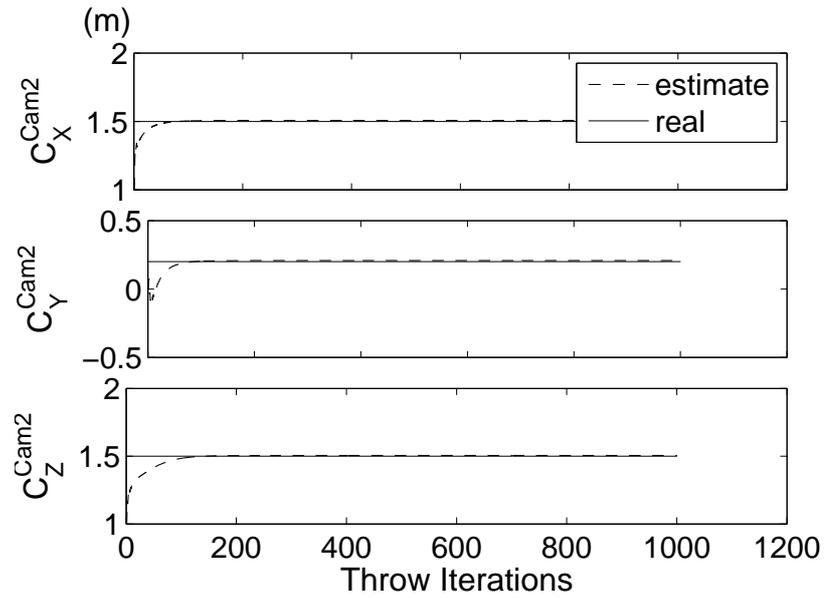
$$\hat{\mathbf{C}} = [-0.1972 \ -0.2029 \ 1.5692 \ 1.5679 \ 1.5720 \ 1.5075 \ 0.2080 \ 1.5053].$$

The re-projection error is computed as 0.5006 pixels for Cam1 and 0.6152 pixels for Cam2. The rotation and translation errors are obtained as 0.0054 radians and 0.0122 meters respectively. The convergence plots are given in Figure 6.11.

The errors corresponding to the single and double throw scenarios are compared in Table 6.2. When the pixel errors are observed, using both trajectories seem not to have any significant advancement compared to using either of the first or the second trajectory alone. However, although pixel error is useful to evaluate the real life performance, it is computed on the image plane only. In the simulation we have the actual values of the calibration parameters and hence the real rotation and translation errors. When they are compared the performance



(a)



(b)

Figure 6.11: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations when two different target trajectories are used for calibration. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

of the multiple throw case turns out to be significantly better. Therefore the calibration data can be increased by throwing the target several times, and even when the pixel error, which is the fundamental performance measure for a real life setup, does not improve, the estimated parameters will probably be closer to their actual values. So, we are especially encouraged to try calibration by multiple throws to increase the accuracy.

## 6.6 Discussion on Simulation Experiments

In this chapter simulation experiments are performed to understand the workings and evaluate the convergence performances of the proposed methods. Experiments are realized to compare the performances of the R-EKF and RB-EKF methods. The effects of using a free falling target or a ballistically moving target and the effects of using a single or multiple trajectories for calibration are also investigated.

In the early experiments a free falling target trajectory is used since the initial velocity of such a motion is a well defined 0 m/s vector. R-EKF has the best performance when the initial target state is exactly provided for EKF initialization. However such an exact knowledge of the initial state is not easy to obtain in real life. Further experiments show that unlike R-EKF, RB-EKF can tolerate significant initial state uncertainties. Hence RB-EKF is a much more suitable method in practical applications, especially for the field applications where the ease of the calibration method is desired most, but the uncertainty is unavoidable.

A free fall motion reduces the unknowns of the initial target state and opens the way to use R-EKF, however such a motion is not sufficiently informative since only a small part of the camera image planes are covered. Additionally different poses of the cameras may yield the same calibration frames for a free falling target. This means that there may be more than one possible solution for the same set of calibration frames in a free fall scenario, and the filter may converge to one of these solutions according to its initialization. In order to prevent such situations, using a ballistic trajectory is suggested. Although that would increase the initial state uncertainty and R-EKF can not be utilized anymore, RB-EKF can still be reliably employed as shown in section 6.4.

The accuracy also depends on the sufficiency of the available calibration data. Increasing the calibration data would probably result in better pose estimates. One way of increasing the

number of calibration points is to use more than one target trajectory. Though each additional trajectory brings 6 additional unknowns corresponding to the initial target state more accurate parameter estimates are obtained as shown in section 6.5. Although the average reprojection pixel errors stay at the same level as in a single trajectory case, the parameter estimates are much closer to their actual values when multiple target trajectories are used.

To sum up, the simulation experiments demonstrate that for practical applications the RB-EKF method is superior to R-EKF method. Furthermore using a ballistic motion and multiple trajectories are superior to using a free fall motion and a single trajectory. Therefore the camera parameters can be estimated best by using RB-EKF method together with multiple ballistic trajectories.

## CHAPTER 7

### Physical Experiments

Simulation experiments help us to understand the workings of the presented algorithms. However, although in simulations something works excellent, in real life we have to deal with several unexpected situations and difficulties. These difficulties and real life performances of our methods are discussed in this chapter.

First, the components of the multi camera setup that is used throughout the experiments are described in section 7.1. Next, the internal parameters of the cameras are obtained by MATLAB calibration toolbox in section 7.2. Then the physical experiments to test the proposed R-EKF and RB-EKF methods are presented. A free falling target is used for calibration with R-EKF and RB-EKF algorithms in sections 7.3 and 7.4. Then RB-EKF is used with a single ballistic target trajectory to obtain the camera poses for parallel, perpendicular and arbitrary placement of the cameras in sections 7.5 and 7.6 and 7.7 respectively. Section 7.8 presents the RB-EKF performance when multiple target trajectories are used for pose estimation of arbitrarily placed cameras and finally a general discussion of the physical experiments is given in section 7.9.

#### 7.1 The Multi Camera Setup Components

In this part the multi camera setup, which is prepared for evaluating the performance of our algorithm, is described. The utilized video capture card and the analog cameras are presented in sections 7.1.1 and 7.1.2 respectively. The connection cables tripods and clamps are displayed in section 7.1.3.



Figure 7.1: Eurotech CTR-1472 frame grabber and MPEG-4 Compressor

### 7.1.1 CTR-1472 MPEG-4 Compressor

The CTR-1472 is a high performance four channel MPEG-4 Video Compressor that supports real-time video encoding. The CTR-1472 can be used to capture analog video streams (from PAL/NTSC cameras, VCR and other video sources) up to four sources at the same time. CTR-1472 can encode frames in MPEG-4 format and send them to a host via PCI BUS. This PC/104-Plus compliant CTR-1472 MPEG-4 compressor, which is developed by Eurotech Company is shown in Figure 7.1.

### 7.1.2 Measurement Cameras

The UWC-300 camera, the product of Outland Technologies Inc., is an underwater camera designed for deep sea operation. It is illustrated in Figure 7.2. UWC-300 can serve as a general purpose camera for our experiments since its image quality in the air is also acceptable.

The camera sensor is sensitive to low light conditions (down to 0.0003 lux) and the sensor provides a 600 horizontal lines of resolution. The UWC-300 camera comes with a 3.6 mm focal length integrated lens. It gives analog video output and therefore necessitate a video frame grabber. Due to the large amount of data from multiple cameras, the captured board requires a video compressor. The cameras are therefore used together with CTR-1472 MPEG-4 compressor.



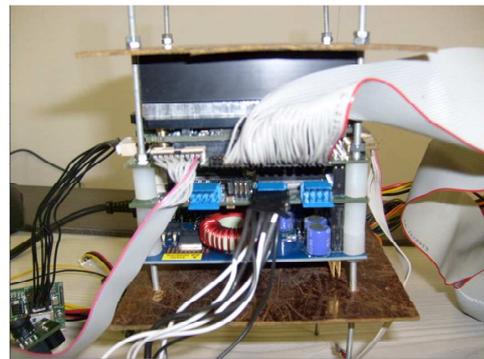
Figure 7.2: Outland Tech. UWC-300 underwater cameras (black cylinder, top)

### 7.1.3 Connection Cables, Tripods and Clips

UWC-300 and other underwater cameras have their standardized underwater connectors which are not commonly available. To work in the laboratory, connectors are manufactured by us in the laboratory environment. After some trials, these connectors are produced to minimize noise and provide data with high quality. The PC104 stack which is set in the laboratory and the attached connectors are shown in Figure 7.3(a). The connection cables are coaxial CCTV cables which also include a number of control signal lines. Another connector which was needed in the laboratory is the one that connects the video cables to the CTR-1472 MPEG-4 compressor as shown in Figure 7.3(b).



(a)



(b)

Figure 7.3: The connection cables and connectors. (a) PC104 stack which includes CTR-1472 MPEG-4 compressor and the manufactured cable set connecting the cameras to the stack. (b) CTR-1472 connector.



Figure 7.4: Tripod-clamp-camera trio and the calibration pattern. (Shown for a single camera)

To hold the cameras in a secure manner during the internal and external calibration suitable set of tripods (SLIK-ABLE 300DX) and clamps(MONFROTTO SUPER CLAMP 035) are acquired. This setup is shown in Figure 7.4 with a single camera and Figure 7.5 shows the synchronous video capture from the set of four cameras and the associated preliminary software interface.



Figure 7.5: 4 pieces of BWC-300 underwater cameras and synchronous video capturing with these cameras.

## 7.2 Internal Calibration

After all the connections are made in order to capture video synchronously from 4 cameras as a first step, the software which was provided by Eurotech is updated and used so that the video frames could be captured, compressed and saved. When the camera hardware became ready, the captured frames could be compressed and saved in the MPEG-4 format with the help of the software provided by Eurotech, the manufacturer of the frame grabber/compressor card. When MPEG-4 compressor captures in PAL format with 100 fps rate the frames are obtained in  $354 \times 248$  pixel resolution and if the frame rate is 25 fps then the resolution is doubled.

While the work on video capturing was in progress, we contacted the Eurotech Company and obtained the source code of the video capturing software. The source code is valuable as it provides a large number of examples on how the drivers of the card can be configured and used. This allows us to incorporate the functionality into our own software and modify and extend what is provided by the standard test interface of the manufacturer. As an example, in order to perform the internal calibration of the cameras, a single frame of a static scene is needed instead of a long video sequence. This problem is solved by adding a *SaveOne-Frame* function to the source code so that single frame capturing and saving in JPEG format is achieved. This allowed the capture of the necessary images which are used to obtain the internal calibration parameters of the cameras with the help of the MATLAB Calibration Toolbox [23] since the calibration software uses the photographs (or a single video frame) of a chess pattern Figure 7.6 from different angles.

Although MATLAB Calibration Toolbox inspired the main initialization step from Zhang's calibration algorithm [19], the closed form estimation of the internal parameters is slightly different. For example the distortion parameters are not estimated at the initialization step, and intrinsic camera model is inspired from Heikkilä and Silvén's work [21] which includes two extra distortion coefficients corresponding tangential distortion. Radial and tangential distortion model is taken directly from the Brown's work [56] which is one of the first introductions of the camera model.

In the simulations two cameras have been used for testing our external calibration algorithms. Although the internal parameters were perfectly known in the simulations, for the physical

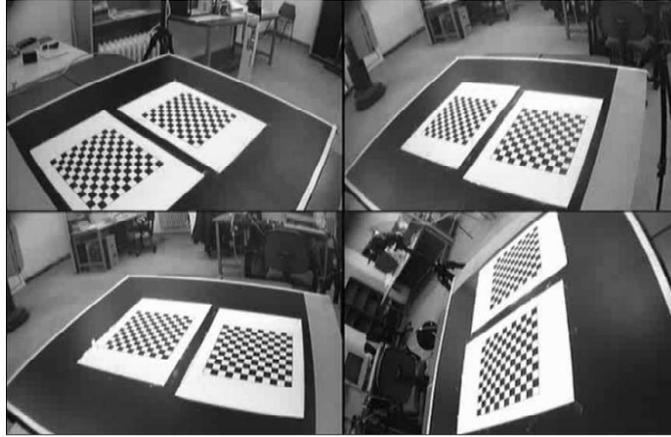


Figure 7.6: The calibration view obtained from 4 cameras synchronously. This kind of chess-board pattern shown in the figure is used for internal calibration purpose.

experiments, first of all, the internal parameters of the cameras should be obtained. We have performed the internal calibration by MATLAB calibration toolbox in order to calibrate two of the UWC-300 cameras. For the first camera focal length, principal point, skew, and distortion coefficients are obtained as

$$\begin{aligned}
 \text{Focal length :} \quad \mathbf{fc} &= [277.44915 \ 298.82608] \pm [1.19135 \ 1.27202] \\
 \text{Principal point :} \quad \mathbf{cc} &= [183.64332 \ 117.42620] \pm [1.59237 \ 1.64054] \\
 \text{Skew :} \quad \mathbf{alpha\_c} &= [-0.00084] \pm [0.00093] \\
 \text{Distortion :} \quad \mathbf{kc} &= [-0.42093 \ 0.20117 \ -0.00270] \\
 &\pm [0.01084 \ 0.02578 \ 0.00095] \\
 \text{Pixel error :} \quad \mathbf{err} &= [0.28266 \ 0.26950]
 \end{aligned}$$

and for the second camera the same parameters are obtained as

$$\begin{aligned}
 \text{Focal length :} \quad \mathbf{fc} &= [270.39002 \ 291.26888] \pm [1.42153 \ 1.55879] \\
 \text{Principal point :} \quad \mathbf{cc} &= [180.25090 \ 118.13432] \pm [1.61322 \ 1.68951] \\
 \text{Skew :} \quad \mathbf{alpha\_c} &= [-0.00026] \pm [0.00099] \\
 \text{Distortion :} \quad \mathbf{kc} &= [-0.39303 \ 0.14915 \ -0.00353] \\
 &\pm [0.01171 \ 0.02362 \ 0.00099] \\
 \text{Pixel error :} \quad \mathbf{err} &= [0.30324 \ 0.30778]
 \end{aligned}$$

In a real image formation process the lens distortion may cause great fluctuations from the pinhole camera model. Apart from the focal length, principal point and skew parameters, MATLAB calibration toolbox yields the distortion coefficients too. Luckily an additional

utility to undistort the images is also included in the toolbox. We integrated that function in our external calibration method, so that undistorted images of a calibration target can be acquired and directly used by our calibration algorithms. Otherwise the distorted images would be used in the expense of low calibration accuracy. In fact we previously tried to include the distortion model in the EKF equations to be able to use the original distorted images acquired from the cameras, but the evaluation time for the Jacobian matrix was increased too much due to the higher complexity.

Finally note that the internal calibration error is in the order of 0.3 pixel for both cameras. In the simulations no such error corresponding to the internal calibration results was taken into account. Anyway this additional disturbance can be accepted as part of the observation noise in the EKF equations.

### 7.3 Experiment 1: R-EKF / Free Fall Motion

In this experiment the simulation experiment in section 6.1 is repeated with real data. Two cameras are tried to be placed in parallel with zero rotations. The main difference from the simulation experiment is the distance between the cameras towards  $x$ -axis. Instead of a 0.2 m gap the cameras are separated by half a meter distance. A free falling target is used to calibrate the two camera system with R-EKF algorithm. The initial target state is measured as

$$\hat{\mathbf{X}}_0 = [0.25 \ 0.35 \ 1.10]^T, \quad \hat{\mathbf{V}}_0 = [0 \ 0 \ 0]^T,$$

and the camera parameter vector is initially set as

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0.5 \ 0 \ 0]^T.$$

After 1000 throw iterations, R-EKF algorithm yields the camera parameters as

$$\hat{\mathbf{C}} = [-0.0191 \ 0.0160 \ 0.0106 \ 0.0145 \ 0.0083 \ 0.4916 \ -0.0054 \ -0.0162]^T.$$

As discussed in section 6.1 the target position is measured by a ruler and the initial velocity is taken as  $\vec{0}$  m/s since the calibration object falls freely. However, the acquired initial position value is not exact due to uncontrollable measurement errors. The initial velocity is not also absolutely equal to  $\vec{0}$  m/s, because the moment that the target is dropped can not be synchronized with the camera capturing instant. Furthermore while releasing the calibration object,

Table 7.1: The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with R-EKF.

	$\alpha^{c1}$	$\beta^{c1}$	$\alpha^{c2}$	$\beta^{c2}$	$\Gamma^{c2}$	$C_x^{c2}$	$C_y^{c2}$	$C_z^{c2}$	$err^{c1}$	$err^{c2}$
$ex^1$	-0.02	0.02	0.01	0.01	0.01	0.50	0.00	-0.02	5.16	4.97
$ex^2$	-0.01	-0.05	0.02	0.01	0.02	0.48	0.08	0.00	7.92	7.73
$ex^3$	-0.01	0.07	0.03	0.08	0.18	0.53	0.04	-0.07	8.12	7.02
$ex^4$	-0.03	0.08	0.02	0.05	0.01	0.51	-0.05	-0.08	8.76	7.14
$ex^5$	0.02	-0.05	0.00	0.05	0.00	0.49	0.13	0.04	7.60	5.38

undesired movements in  $x$  and  $z$  directions may also occur. Still taking the initial velocity of a free falling target as  $\vec{0}$  m/s is a reasonable assumption.

The final re-projection pixel errors for Cam1 and Cam2 are 5.1617 and 4.9735 pixels respectively. Although our algorithm is not an optimization method that minimizes the pixel error, this re-projection error is a measure that reflects the accuracy of our estimation. Figure 7.7 illustrates the original and re-projected points for both cameras. The convergence graphs for the calibration parameters are shown in Figure 7.8.

In four additional examples the initial position of the calibration object is changed and the same experiment is repeated. In other words the calibration target is freely dropped from different locations but the cameras are not moved. R-EKF results for 1000 throw iterations are presented in Table 7.1. Although all of the parameters are consistent with our expectations the pixel errors are changing from 5 to 9 pixels. This much error is mainly caused by the inaccurate measurement of the initial target state.

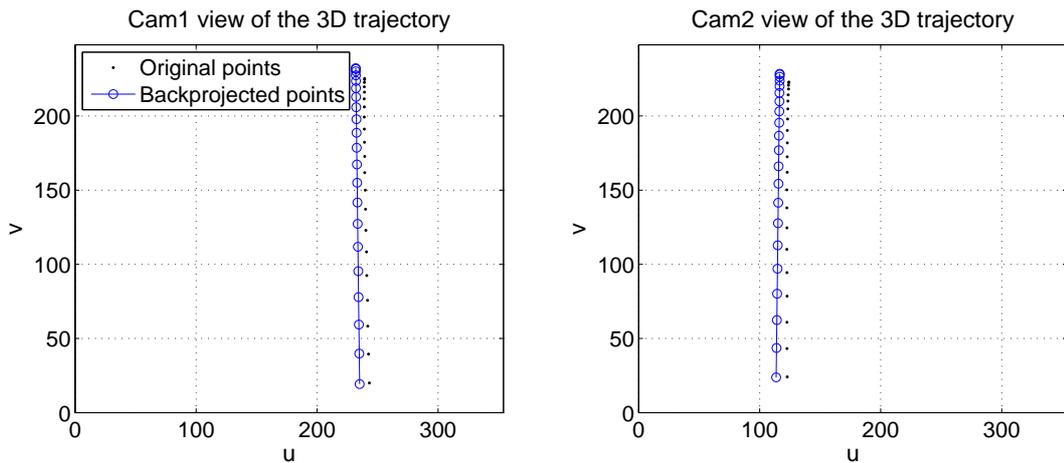
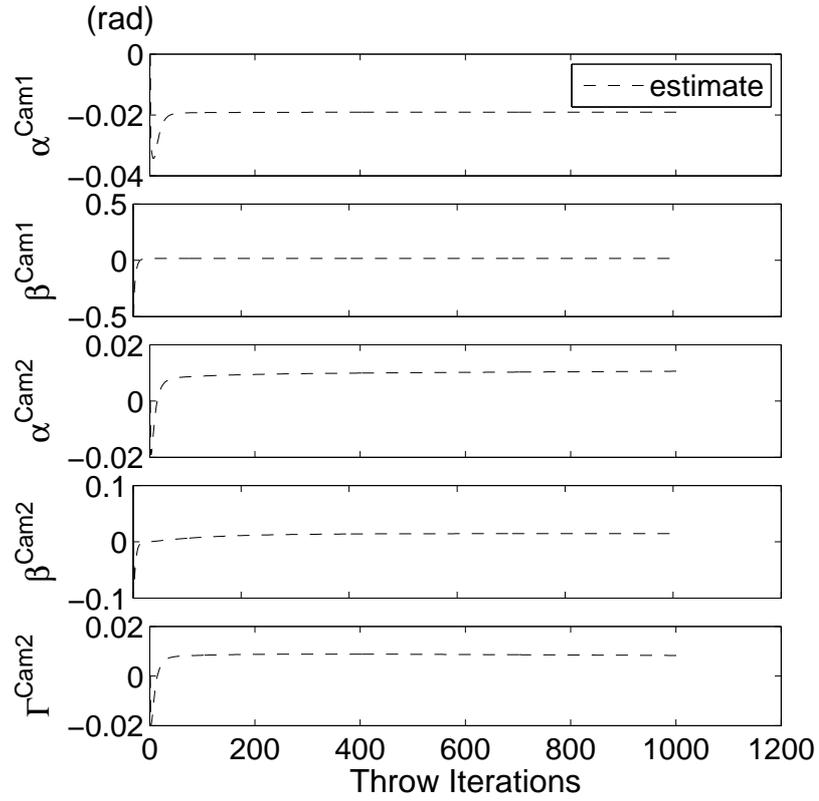
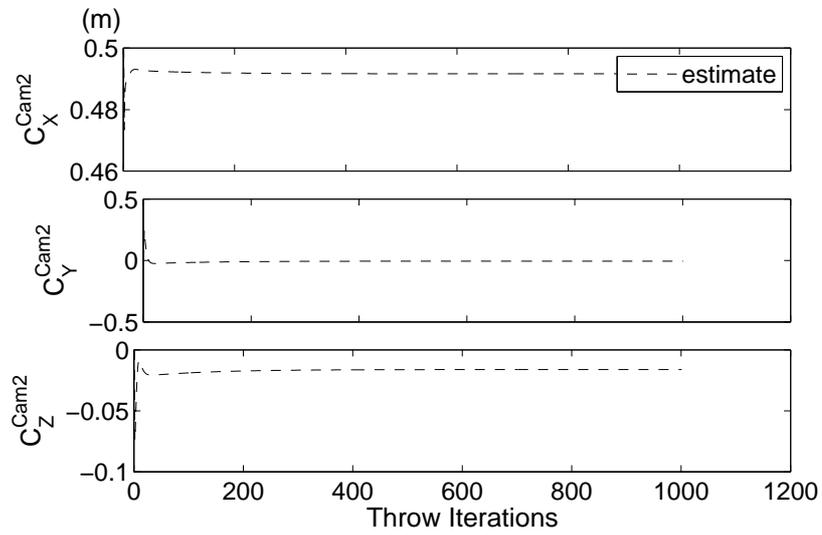


Figure 7.7: The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm.



(a)



(b)

Figure 7.8: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

## 7.4 Experiment 2: RB-EKF / Free Fall Motion

In this section the first experiment is repeated for the RB-EKF algorithm. The initial target state and the initial camera parameter vector are provided to the RB-EKF algorithm as in section 7.3 since the same observation data (i.e. captured videos of the free falling target) is used. At the end of 1000 throw iterations RB-EKF yields

$$\hat{\mathbf{C}} = [-0.0459 \ 0.0265 \ -0.0015 \ 0.0220 \ -0.0105 \ 0.5142 \ -0.0266 \ -0.0071]^T$$

and also the initial target state is also estimated as

$$\hat{\mathbf{X}} = [0.3040 \ 0.3238 \ 1.1714]^T, \quad \hat{\mathbf{V}} = [-0.0446 \ -0.0441 \ -0.0233]^T$$

with only 0.1448 and 0.2693 re-projection pixel errors for Cam1 and Cam2 respectively. The camera parameters are similar to the ones obtained by the R-EKF, however the final re-projection errors are much more smaller. The main reason for such an error reduction with the RB-EKF is that the initial target state is also estimated in addition to the camera parameters. Remember that in the simulations since the initial target state was exactly known and provided, R-EKF had a better performance. Unfortunately in real life such an exact measurement seems not possible to be acquired, so in practice the RB-EKF algorithm works much more efficiently.

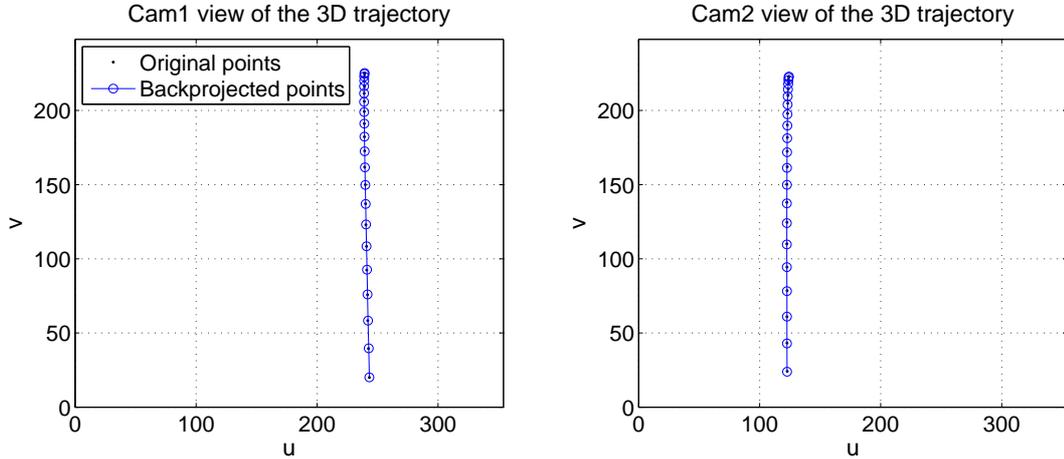
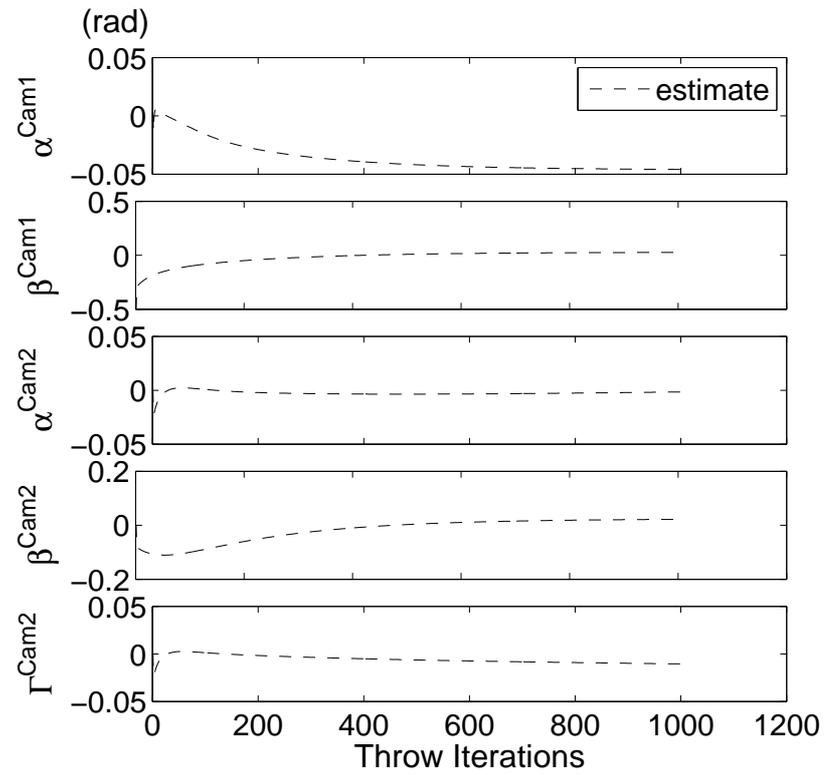
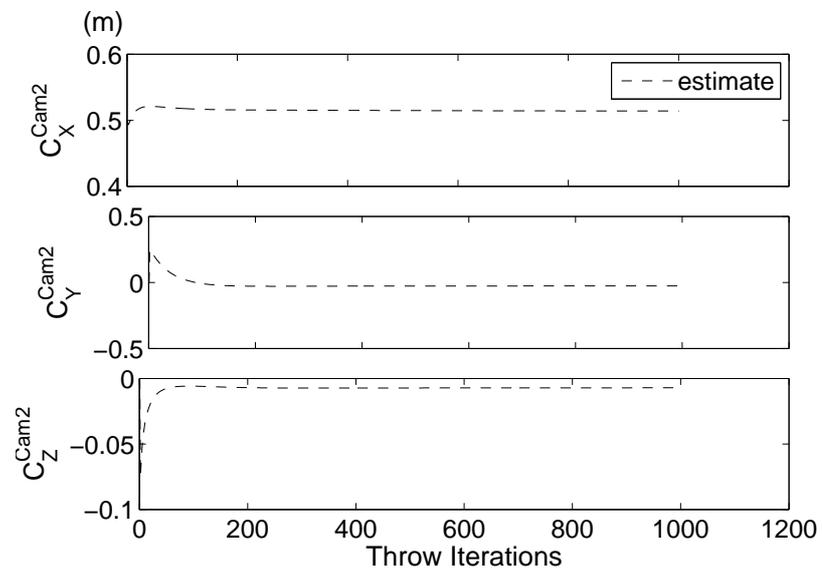


Figure 7.9: The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.



(a)



(b)

Figure 7.10: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

Table 7.2: The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with RB-EKF.

	$\alpha^{c1}$	$\beta^{c1}$	$\alpha^{c2}$	$\beta^{c2}$	$\Gamma^{c2}$	$C_x^{c2}$	$C_y^{c2}$	$C_z^{c2}$	$err^{c1}$	$err^{c2}$
$ex^1$	-0.05	0.03	0.00	0.02	-0.01	0.51	-0.03	-0.01	0.14	0.27
$ex^2$	-0.02	0.05	0.02	0.07	0.02	0.49	0.03	-0.02	0.15	0.31
$ex^3$	-0.02	0.00	0.03	0.05	-0.01	0.52	0.08	-0.02	0.18	0.46
$ex^4$	-0.03	0.02	0.01	0.04	0.00	0.52	0.00	-0.02	0.21	0.42
$ex^5$	-0.02	-0.02	0.00	-0.01	-0.02	0.53	0.00	-0.02	0.37	0.67

The original and re-projected points are shown in Figure 7.9 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.10. For the additional four examples in which the target is initiated from different places in the joint-FOV of the cameras while the cameras are kept fixed, the RB-EKF results after 1000 throw iterations are summarized in Table 7.2. The re-projection pixel errors are on the order of 0.2 – 0.5 pixels and it should be noted that part of it is caused by the internal calibration error. The calibration parameters are obtained as expected and the parameter estimations in different examples are nearly equal to each other.

### 7.5 Experiment 3: RB-EKF / Ballistic Motion

In this experiment a ballistic trajectory is used for calibrating the two camera system instead of a free fall motion. The cameras are kept fixed at their parallel poses as described in section 7.3. Since the calibration target is thrown into the joint camera FOV, the initial position and velocity of the target are not known approximately as in the previous cases, but they can only be roughly guessed by the operator.

Since the cameras are placed in a parallel position with half a meter distance between them, the calibration parameters are approximately known in our example. However when the cameras are placed in an obscure manner with rotation and translation in all directions, the operator may not be able to provide the initial estimation for the camera poses as accurate as we have done until now. In this experiment we observe the performance of the RB-EKF algorithm when the initial estimates for the target state and camera parameters are not provided precisely.

The initial estimate for the camera orientations and target state is assumed to be provided by the operator by only looking at the camera poses and calibration frames. We have estimated

the initial target state roughly by looking at the calibration frames as

$$\hat{\mathbf{X}}_0 = [-0.25 \ 0 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

It is a reasonable assumption that human eye may predict the rotation and translation parameters with less than 0.5 rad, and 0.5 m errors. So the camera parameters are distorted by that amount and the initial parameter estimate vector is provided as

$$\hat{\mathbf{C}}_0 = [0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0 \ 0.5 \ 0.5]^T.$$

After 1000 throw iterations with RB-EKF method the complete state is obtained as

$$\begin{aligned} \hat{\mathbf{X}} &= [0.0355 \ 0.0733 \ 0.8864]^T, \quad \hat{\mathbf{V}} = [1.1149 \ 1.8869 \ 0.7419]^T, \\ \hat{\mathbf{C}} &= [-0.0091 \ 0.0120 \ -0.1212 \ 0.0503 \ 0.1260 \ 0.5158 \ 0.0408 \ -0.0126]^T \end{aligned}$$

with 0.4009 and 1.3721 re-projection pixel errors for Cam1 and Cam2. Note that all of the camera parameters are obtained as expected. One can argue that  $-0.1212$  and  $0.1260$  values are too high because their expected value is approximately zero. However since  $z$ - $x$ - $z$  rotation is used,  $\alpha^{c2}$  and  $\Gamma^{c2}$  terms both correspond to the rotations around  $z$ -axis. Since  $\beta^{c2}$  is also approximately zero, these  $-0.1212$  and  $0.1260$  terms nearly cancel each other validating the parallelism of the cameras.

The original and re-projected points are shown in Figure 7.11 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.12.

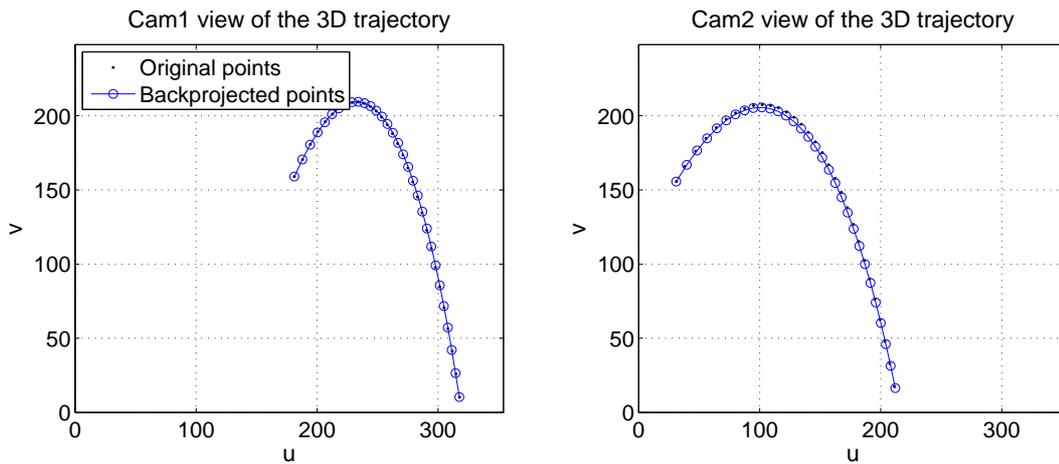
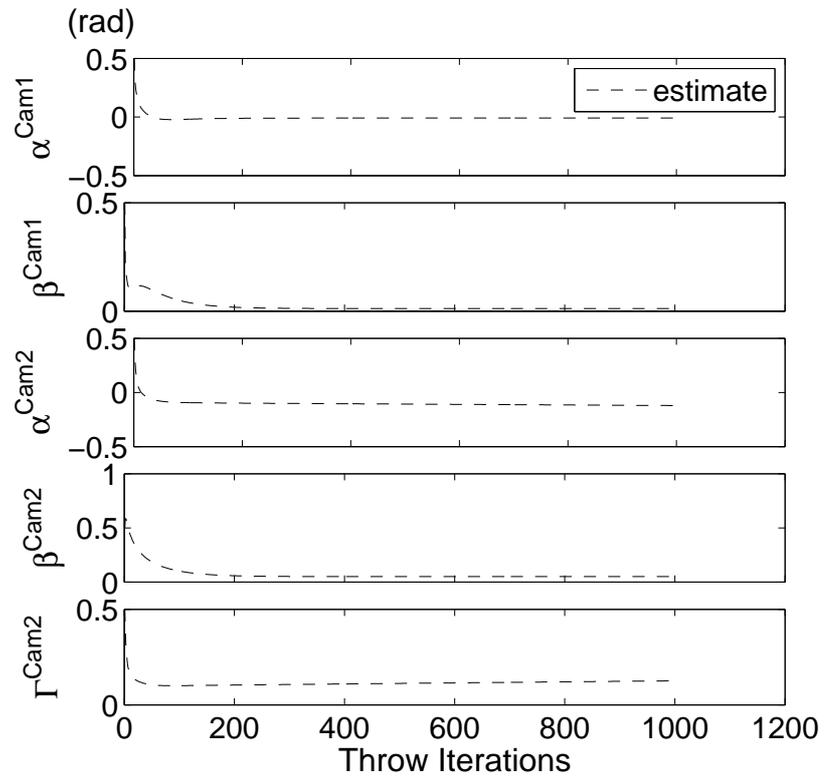
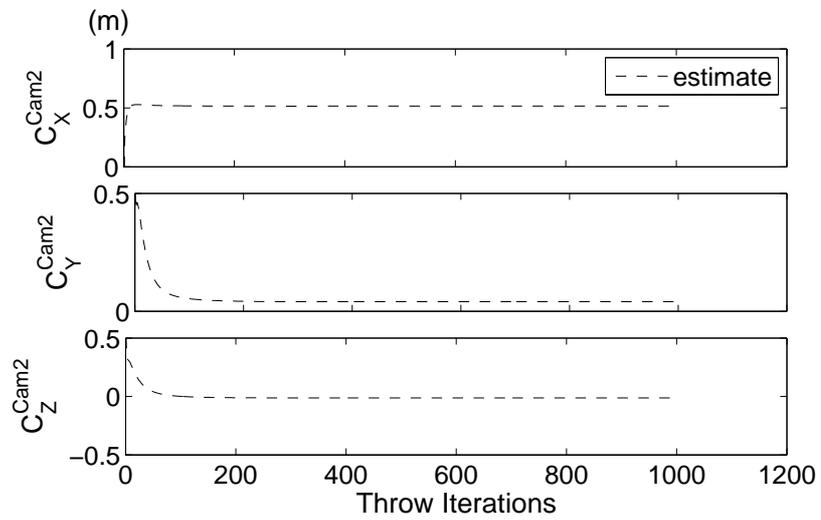


Figure 7.11: The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.



(a)



(b)

Figure 7.12: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

## 7.6 Experiment 4: Perpendicular Camera Placement

Up to now the experiments are performed to calibrate two cameras which are placed in a parallel configuration. Such a placement is not only quite useful for stereo imaging, but it also helps us to verify the trustiness of our algorithm since the actual camera poses are approximately known. However the proposed method is not limited to calibrate the cameras that are placed in parallel orientation. In this part we calibrate a two camera setup when the cameras are placed on the diagonal corners of a square with 1 meter edge length and Cam1 is rotated 90 degrees around y-axis with respect to the basis coordinate frame as shown in Figure 7.13.

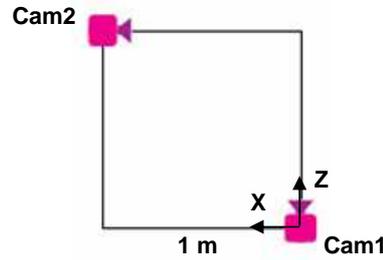


Figure 7.13: The cameras are placed at the diagonal corners of a square with 1 meter edge length.

We have estimated the initial target state roughly as

$$\hat{\mathbf{X}}_0 = [-1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

Since  $z$ - $x$ - $z$  rotation is used, Cam2 rotation parameters should initially be set as  $\alpha^{c2} = -\pi/2$ ,  $\beta^{c2} = \pi/2$ ,  $\Gamma^{c2} = \pi/2$ . So the initial parameter estimate vector is provided as

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ -\frac{\pi}{2} \ \frac{\pi}{2} \ \frac{\pi}{2} \ 1 \ 0 \ 1]^T.$$

After 1000 throw iterations with RB-EKF method the complete state is obtained as

$$\begin{aligned} \hat{\mathbf{X}} &= [-0.3618 \ -0.1101 \ 0.8088]^T, \quad \hat{\mathbf{V}} = [1.4191 \ 2.5723 \ 0.5942]^T, \\ \hat{\mathbf{C}} &= [-0.0115 \ -0.0010 \ -1.4995 \ 1.6111 \ 1.5774 \ 1.0100 \ 0.0662 \ 1.0005]^T \end{aligned}$$

with 0.3181 and 0.5495 re-projection pixel errors for Cam1 and Cam2 respectively. The first two elements of the  $\hat{\mathbf{C}}$  vector,  $\alpha^{c2}$  and  $\beta^{c2}$ , are nearly zero since Cam1 is placed approximately parallel to the ground. The orientation of Cam2 is also obtained as expected. Only the 0.0662

value for  $C_y^{c2}$  seems to be a little bit away from its expected value of zero, because the cameras are placed nearly at the same height. Still under the disturbance effects such as air friction, lens

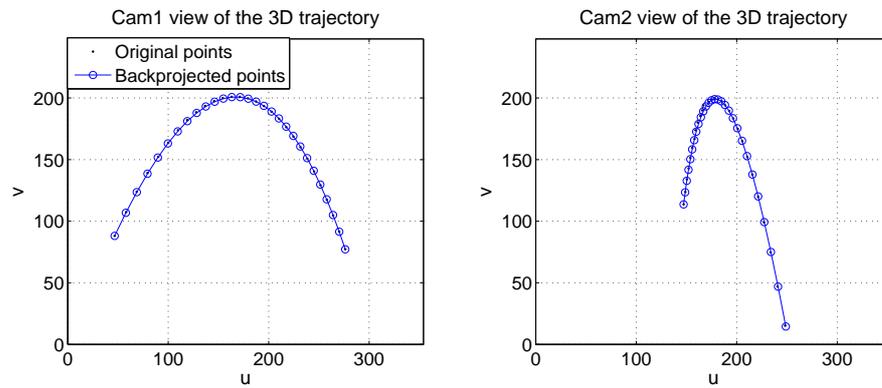
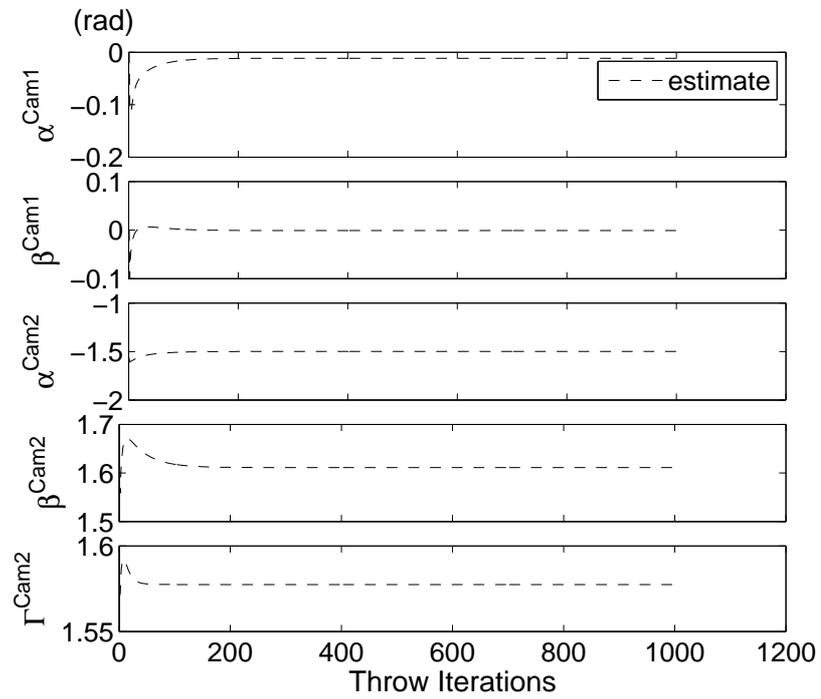


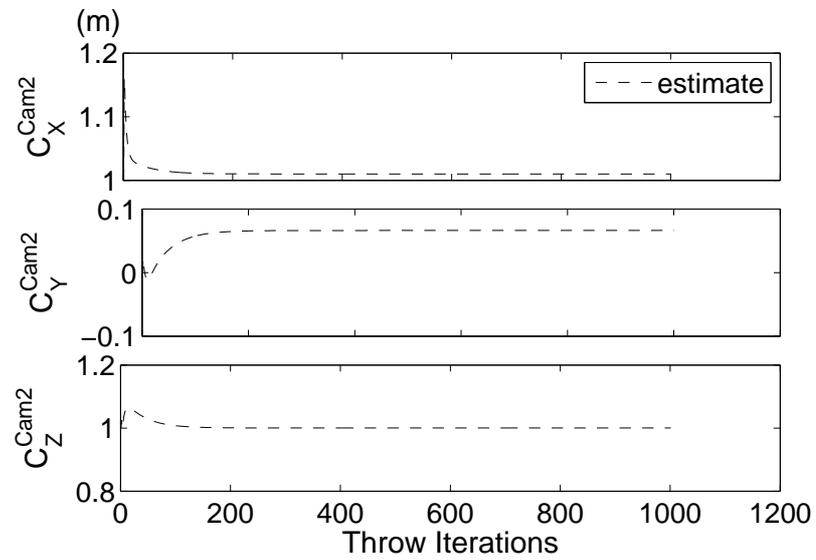
Figure 7.14: The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

distortion and pixel errors the obtained results are successful especially in the re-projection pixel error sense.

The original and re-projected points are shown in Figure 7.14 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.15.



(a)



(b)

Figure 7.15: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

## 7.7 Experiment 5: Arbitrary Camera Placement

In many practical applications the cameras may be placed arbitrarily and it may be quite difficult to provide the initializing parameters of the EKF accurately. Even though the initial state is not known it should be predicted by the operator to prevent the RB-EKF to converge to a local minimum solution which may be a wrong result. If the initial estimates for the target state and camera parameters are given in a wrong manner, the filter may even diverge. Therefore at least a rough estimate of the super-state should be provided initially.

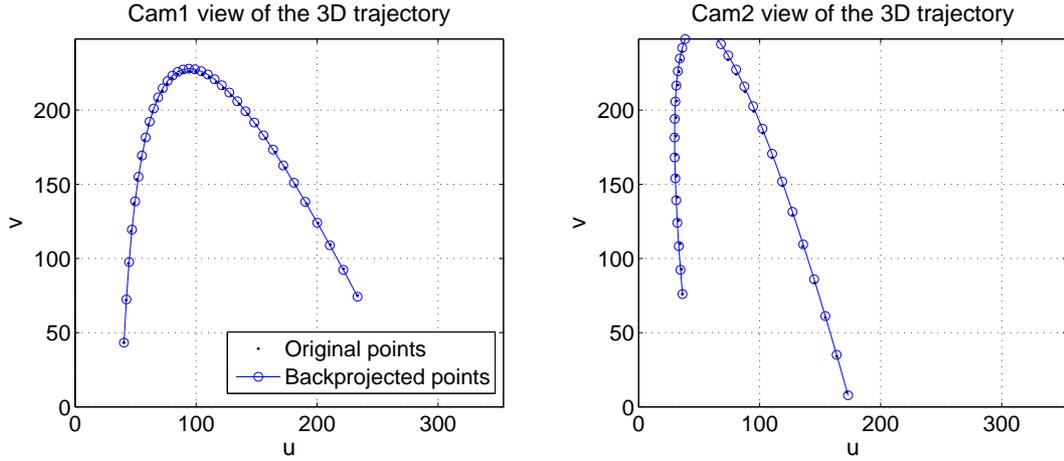


Figure 7.16: The extracted and re-projected 2-D positions of a ballistically moving target captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

In this part we place the cameras arbitrarily and give the initializing state estimate roughly to the RB-EKF. No measuring device is used to obtain the initial state estimate. Just by observing the thrown target and the camera system, the initial state estimate is predicted as

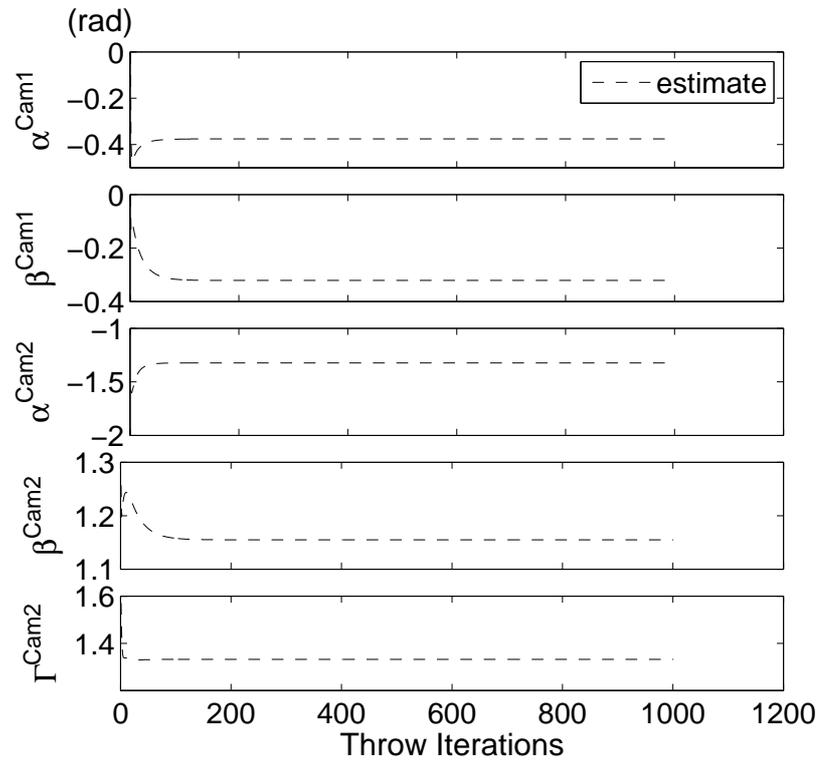
$$\hat{\mathbf{X}}_0 = [-1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ -\frac{\pi}{2} \ \frac{\pi}{2} \ \frac{\pi}{2} \ 1 \ 0 \ 1]^T.$$

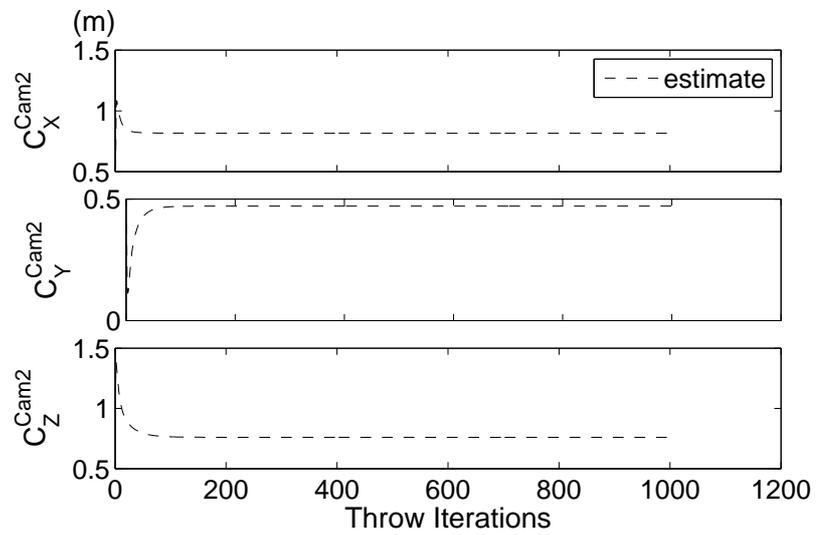
After 1000 throw iterations the RB-EKF algorithm yields the initial target state and the calibration parameter estimates as

$$\hat{\mathbf{X}} = [-0.2721 \ 0.1341 \ 0.6719]^T, \quad \hat{\mathbf{V}} = [0.7714 \ 3.2010 \ 0.4417]^T,$$

$$\hat{\mathbf{C}} = [-0.3754 \ -0.3214 \ -1.3230 \ 1.1549 \ 1.3321 \ 0.8160 \ 0.4710 \ 0.7604]^T.$$



(a)



(b)

Figure 7.17: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

The re-projection pixel errors are 1.0963 and 1.2239 for Cam1 and Cam2 respectively. The original and re-projected points are shown in Figure 7.16 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.17. Note that the target is thrown with more thrust than expected, so at the highest position it stays out of the FOV of Cam2. Hence the trajectory is not fully captured by Cam2. Anyway, those unobserved frames are handled by only executing the *prediction* steps but not the *update* steps of the EKF algorithm given in Table 3.1. So even when the calibration target is unobservable in some of the frames, the proposed method can still be effectively used.

In this experiment we have shown that when the initial target state and the camera orientation vector for an arbitrary placement of the cameras are initialized by only observing the system (without using any measurement device), RB-EKF method can yield the calibration parameters as well as the initial target state. Additionally even when the calibration object is unobservable in some of the frames, our method can still be used for the estimation of the parameters by only executing the *prediction* step of the EKF algorithm for those unobserved frames.

## 7.8 Experiment 6: Using Multiple Trajectories for Arbitrary Camera Placement

The physical experiments with a multi camera setup are finally concluded with this final calibration performance of the RB-EKF by using multiple target trajectories. The same camera configuration as in section 7.7 is kept and two additional target trajectories are captured for calibration. The initial estimate for the first target and the camera parameter vector are set as in section 7.7. For the initial states of the additional two trajectories, the following estimates are used to initialize the EKF:

$$\begin{aligned}\hat{\mathbf{X}}_0^{t_2} &= [1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0^{t_2} = [-1 \ 2 \ -1]^T, \\ \hat{\mathbf{X}}_0^{t_3} &= [-1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0^{t_3} = [1 \ 2 \ 1]^T.\end{aligned}$$

The target trajectories are alternately used for calibration and after 300 throw iterations for each trajectory (i.e. a total of 900 throw iterations) the RB-EKF algorithm yields the camera parameters as

$$\hat{\mathbf{C}} = [-0.3795 \ -0.3144 \ -1.3250 \ 1.1665 \ 1.3336 \ 0.8276 \ 0.4712 \ 0.7828]^T$$

and also the initial target state estimates for each throw as

$$\begin{aligned}\hat{\mathbf{X}}^{t_1} &= [-0.2755 \ 0.1320 \ 0.6776]^T, \quad \hat{\mathbf{V}}^{t_1} = [0.7798 \ 3.1949 \ 0.4577]^T, \\ \hat{\mathbf{X}}^{t_2} &= [0.4228 \ 0.1714 \ 0.9674]^T, \quad \hat{\mathbf{V}}^{t_2} = [-1.2062 \ 3.1798 \ -0.2982]^T, \\ \hat{\mathbf{X}}^{t_3} &= [-0.0309 \ 0.5113 \ 0.7266]^T, \quad \hat{\mathbf{V}}^{t_3} = [0.5964 \ 0.8585 \ 1.1182]^T.\end{aligned}$$

The average re-projection pixel errors are 0.6943 and 1.5296 for Cam1 and Cam2 respectively. The original and re-projected points for all three trajectories are shown in Figure 7.18 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.19. When the re-projection pixel errors are compared with the ones in section 7.7 there seems to be no improvement over using a single trajectory. However as shown in section 6.5 even if the pixel errors are not reduced by using multiple trajectories, the obtained estimates for the camera parameters can be much more accurate than a single throw case, because much more calibration points are used for pose estimation.

As a final comment, note that only 9 observation points are provided by the second trajectory as shown in Figure 7.18(b). Such less data alone can normally not be sufficient for accurate parameter estimation. However many other data points are also provided when multiple trajectories are used for calibration, hence the trajectories for which very few data points are acquired may also be used to increase accuracy.

## 7.9 Discussion on Physical Experiments

In this chapter after the components of our multi-camera setup are described and internal parameters of the cameras are obtained by MATLAB Calibration Toolbox, physical experiments are performed to demonstrate the practical usability of the proposed algorithms. After performances of the R-EKF and RB-EKF are compared by using a free falling target, RB-EKF method is tested for various positions of the cameras by either using a single or multiple ballistic trajectories.

First two experiments show the superiority of RB-EKF over R-EKF in the sense of re-projection pixel errors. R-EKF performance relies on the accurate initialization of the target state. Unfortunately the initial target state can not be measured exactly even for a free falling target,

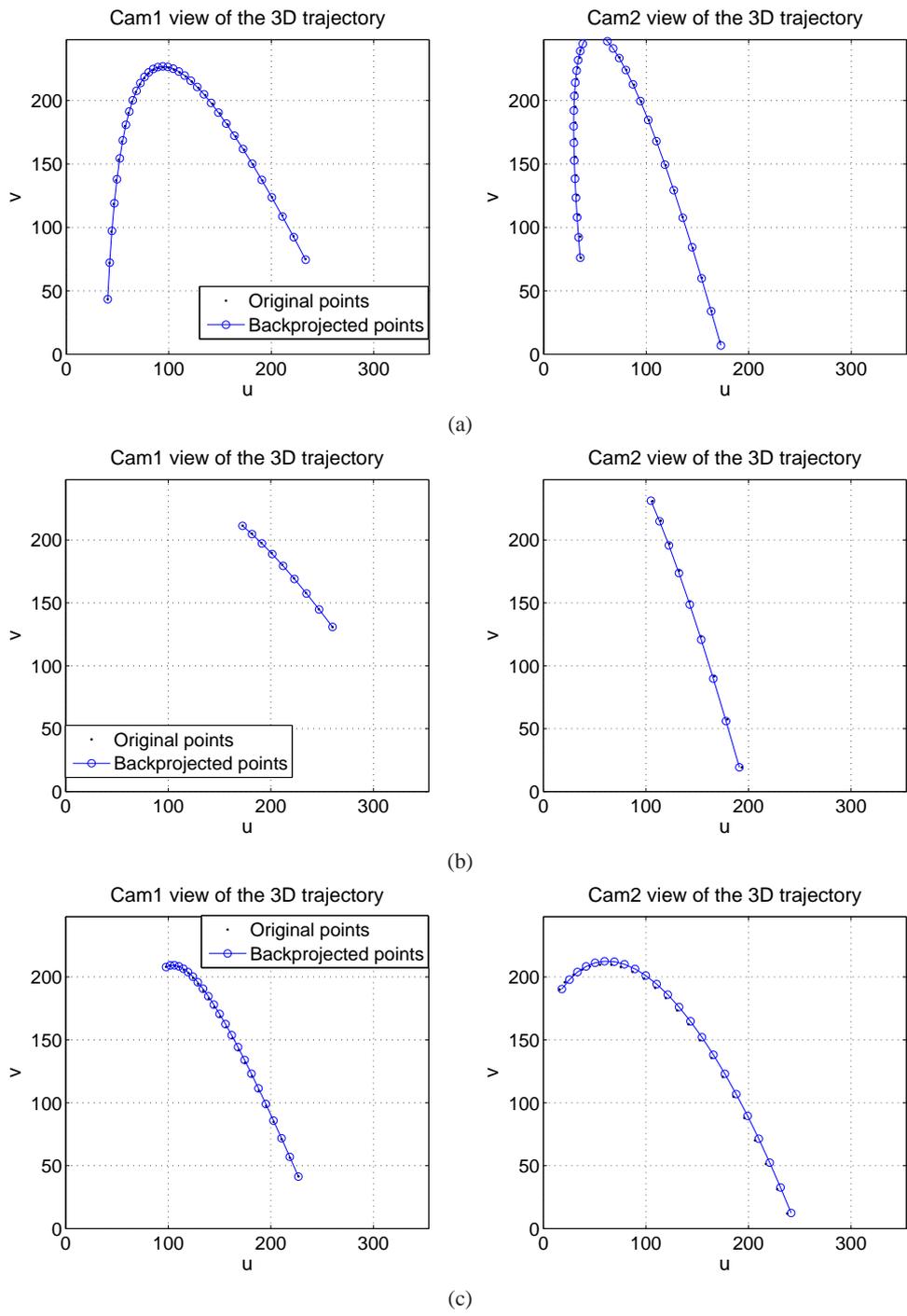
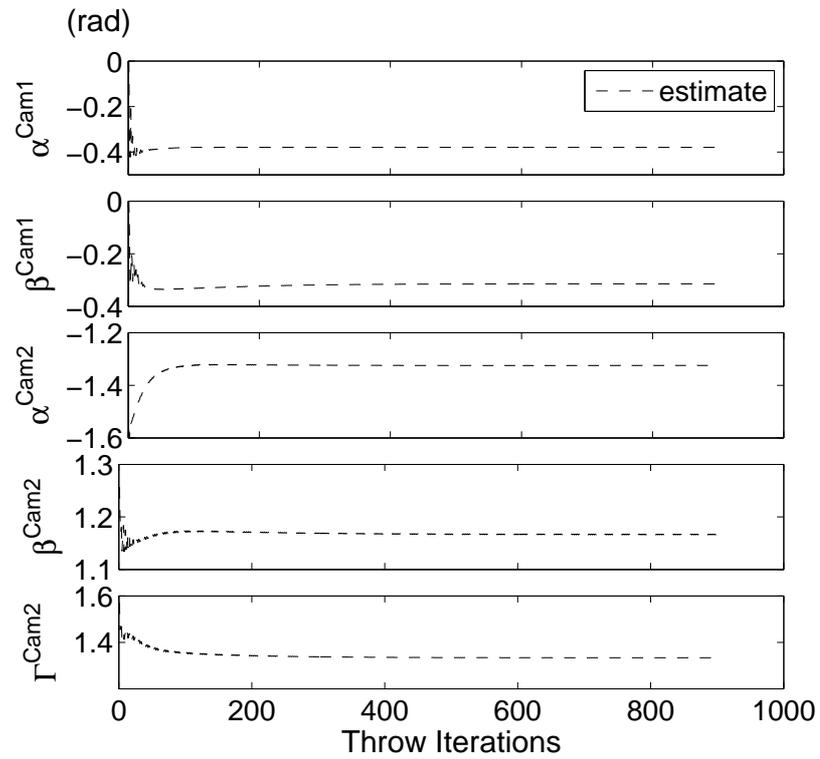
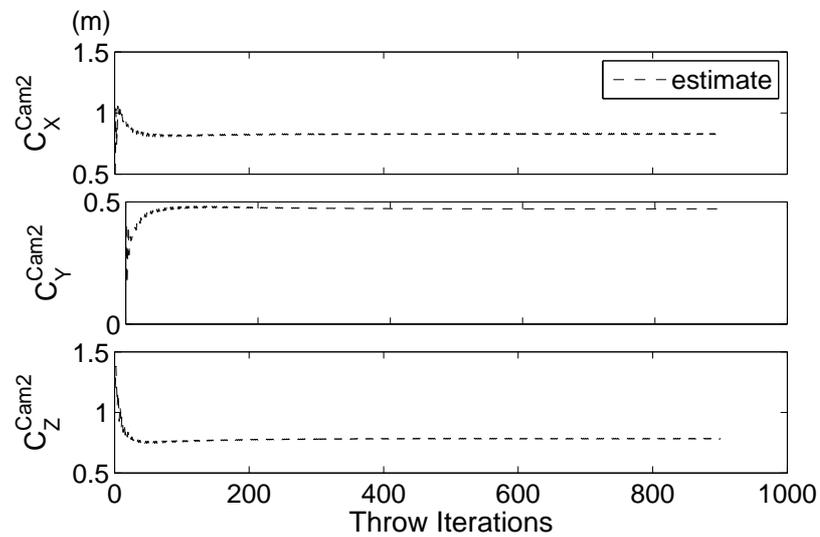


Figure 7.18: The extracted and re-projected 2-D positions of three ballistically moving targets captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm for the (a) first; (b) second and (c) third trajectories.



(a)



(b)

Figure 7.19: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

because the camera capturing and target initiation instants can not be synchronized and hence the velocity of the target is not exactly equal to  $\vec{0}$  m/s when the first frame is captured. Additionally the metric measurements for acquiring the target position w.r.t the center of the first camera can only be performed from the outer core of the camera bulk. Since the initial target state can be estimated accurately by RB-EKF method, calibration parameters are also estimated more accurately.

The RB-EKF is tested by using a single ballistic trajectory for parallel, perpendicular and arbitrary placement of the cameras in sections 7.5, 7.6 and 7.7. Thus the capability of the proposed method is presented by investigating the calibration performance of the algorithm with various camera poses. In all of these experiments final estimates of the calibration parameters satisfy our expectations. Especially section 7.7 proves the practical applicability of the algorithm since the poses of arbitrarily placed cameras are estimated successfully by using partial observation of the target trajectory.

Finally multiple target trajectories are used for increasing the calibration accuracy in section 7.8. Although the re-projection pixel error is not reduced compared to a single trajectory case, the estimations are expected to be more accurate as shown in 6.5.

In conclusion physical experiments demonstrate that the ease of applicability and the successful calibration performance of the RB-EKF method makes it particularly beneficial for any application where frequent re-calibration is required such as outdoor field measurement applications using multiple-cameras.

## CHAPTER 8

### Conclusions and Future Work

We propose a procedurally effective and high performance method of calibrating a multiple-camera measurement system by making use of the known dynamics of a target object. The proposed method promises to considerably facilitate the field installation of a camera based ground truth measurement system. In fact it needs only a ballistically thrown simple calibration target in the multi-camera joint-FOV to calibrate the complete system. Extension of the two camera case considered for the formulation to a multi-camera system is straightforward in two ways: first, we can either insert the calibration parameters and measurements taken from the additional cameras into the state and measurement vectors and use the algorithm as it is, or we can calibrate the cameras two by two, i.e., Cam1 – 2, Cam1 – 3, Cam1 – 4, etc. The latter approach may be more feasible since it does not increase the dimensionality of the problem.

The simulation results validate that both R-EKF and RB-EKF algorithms can converge rapidly and asymptotically yield the desired camera parameters. However for successful calibration R-EKF needs an accurate estimate of the initial target state which is quite difficult to obtain in a real case. On the other hand RB-EKF can tolerate significant initial state uncertainty when sufficient number of *throw iterations* are performed. This removes the need for manual geometric measurements performed by the experimenter (such as 3D-2D correspondence measurements) and hence results in the removal of the human related measurement errors that can affect the calibration process. Although our primary motivation is to generate an easily applicable and fast calibration algorithm but not to surpass the performances of the standard methods we feel that the eliminated human factor may enable us to get superior calibration performance in practice compared to the methods that need 3D-2D correspondence measurements.

The testing of the algorithms on a real imaging system shows that RB-EKF method can be successfully used for determining the external poses of the cameras in a multi-camera setup. The cameras are placed in parallel, perpendicular and arbitrary positions in order to test the RB-EKF method for free fall and ballistic motion cases by either using a single or multiple target trajectories. In all of the physical experiments the calibration parameters are estimated around their expected values.

Although re-projection pixel error is used to demonstrate the success rate of our method one should not forget that pixel error is computed on the 2D image plane, and it is not always proportional with the actual 3D position errors. Our method tries to balance the information coming from the dynamic motion model with the information due measurements. Some of the present methods only use the measurements obtained from a static scene and minimize the re-projection pixel error in order to calibrate a camera setup. Hence even when RB-EKF yields higher re-projection pixel errors compared to these algorithms, the actual poses of the cameras may still be estimated better. Moreover the presence of process and measurement noise terms in the EKF algorithm compensate for modeling errors and unknown inputs to the system, reducing the sensitivity to such factors as wind disturbance, image noise, ball center detection errors etc.

An interesting extension to our study is to provide a full calibration (both internal and external) by means of the proposed method. Even though internal calibration does not have the same speed requirements of the external calibration and hence is not a critical candidate for this method, these static internal parameters can also be made part of the state vector and obtained similar to the external parameters. But it is clear that with the increasing dimensionality of the problem, the convergence properties of EKF may change for the worse in terms of speed, accuracy and stability.

## REFERENCES

- [1] I. Kakadiaris and D. Metaxas, “Model-Based Estimation of 3D Human Motion,” 2000.
- [2] DM Gavrilu and LS Davis, “3-D model-based tracking of humans in action: a multi-view approach,” *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR’96, 1996 IEEE Computer Society Conference on*, pp. 73–80, 1996.
- [3] J.M. Rehg and T. Kanade, “Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking,” *ECCV(2)*, pp. 35–46, 1994.
- [4] R. Stolkin, A. Greig, and J. Gilby, “A calibration system for measuring 3D ground truth for validation and error analysis of robot vision algorithms,” *Measurement Science and Technology*, vol. 17, no. 10, pp. 2721–2730, 2006.
- [5] V. Lippiello, B. Siciliano, and L. Villani, “3D pose estimation for robotic applications based on a multi-camera hybrid visual system,” *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pp. 2732–2737, 2006.
- [6] Vicon MX Motion Capture Systems, “<http://www.vicon.com/products/viconmx.html>,” (accessed August 19, 2008).
- [7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [8] F. Remondino and C. Fraser, “Digital camera calibration methods: considerations and comparisons,” *Proc. ISPRS Commission V Symposium*, pp. 266–272, 2006.
- [9] U. Saranlı, M. Buehler, and D.E. Koditschek, “RHex: A Simple and Highly Mobile Hexapod Robot,” *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616, 2001.
- [10] C. Everitt, “Projective Texture Mapping,” *White paper, NVidia Corporation*, 2001.
- [11] J. Bloomenthal and J. Rokne, “Homogeneous Coordinates,” 1993.
- [12] YI Abdel-Aziz and HM Karara, “Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry,” *Proceedings of the Symposium on Close-Range Photogrammetry*, pp. 1–18, 1971.
- [13] KW Wong, “Mathematical formulation and digital analysis in close-range photogrammetry,” *Photogrammetric Eng. Remote Sensing*, vol. 41, no. 11, pp. 1355–1373, 1975.
- [14] W. Faig, “Calibration of close-range photogrammetry systems: Mathematical formulation,” *Photogrammetric Engineering and Remote Sensing*, vol. 41, no. 12, pp. 1479–1486, 1975.

- [15] J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a non-linear distortion model [CCD sensory]," *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 1, 1990.
- [16] R.Y. TSAI, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *Radiometry*, 1992.
- [17] J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.
- [18] G.Q. Wei and S. De Ma, "Implicit and explicit camera calibration: theory and experiments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 5, pp. 469–480, 1994.
- [19] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [20] J. Bastian and A. van den Hengel, "Computing Image-Based Reprojection Error on Graphics Hardware," *Proc. VIIth Digital Image Computing: Techniques and Applications*, 2006.
- [21] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit imagecorrection," *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 1106–1112, 1997.
- [22] P. Sturm and S. Ramalingam, "A generic concept for camera calibration," *European Conference on Computer Vision*, vol. 2, pp. 1–13, 2004.
- [23] Camera Calibration Toolbox for MATLAB, "[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)," (accessed August 19, 2008).
- [24] F. Pedersini, A. Sarti, and S. Tubaro, "Multi-camera systems," *Signal Processing Magazine, IEEE*, vol. 16, no. 3, pp. 55–65, 1999.
- [25] S. Ganapathy, "Decomposition of transformation matrices for robot vision," *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, vol. 1, 1984.
- [26] C. Wiles and A. Davison, "Calibrating a Multi-Camera System for 3D Modelling," *Proceedings of the IEEE Workshop on Multi-View Modelling and Analysis of Visual Scenes*, pp. 29–36, 1999.
- [27] I.O. Sebe and G.Q. Chen, "Multi-camera Calibration," Tech. Rep., STMicroelectronics Technical Report, 2002.
- [28] M. Han and T. Kanade, "Creating 3D models with uncalibrated cameras," *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pp. 178–185, 2000.
- [29] T. Svoboda, D. Martinec, and T. Pajdla, "A Convenient Multicamera Self-Calibration for Virtual Environments," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.
- [30] H.G. Maas, "Image sequence based automatic multi-camera system calibration techniques," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 5-6, pp. 352–359, 1999.

- [31] P. Baker and Y. Aloimonos, "Complete calibration of a multi-camera network," *Omnidirectional Vision, 2000. Proceedings. IEEE Workshop on*, pp. 134–141, 2000.
- [32] B. Moghaddam et al., *Probabilistic visual learning for object detection*, Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1997.
- [33] CP Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," *Computer Vision, 1998. Sixth International Conference on*, pp. 555–562, 1998.
- [34] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
- [35] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," 2001.
- [36] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," *Proc. ECCV*, vol. 4, pp. 113–130, 2002.
- [37] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. CVPR*, vol. 1, pp. 511–518, 2001.
- [38] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.
- [39] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," *DAGM 25th Pattern Recognition Symposium*, pp. 297–304, 2003.
- [40] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: The importance of good features," *Proc. CVPR*, vol. 2, pp. 53–60, 2004.
- [41] H. Schneiderman and T. Kanade, "Statistical method for 3 D object detection applied to faces and cars," *Proc IEEE Comput Soc Conf Comput Vision Pattern Recognition*, vol. 1, pp. 746–751, 2000.
- [42] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *Robotics and Automation Magazine*, vol. 13, pp. 99–110, 2006.
- [43] T. Bailey and H. Durrant-Whyte, "Simultaneous Localisation and Mapping (SLAM): Part II State of the Art," *Robotics and Automation Magazine*, vol. 13, pp. 108–117, 2006.
- [44] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758–767, 2000.
- [45] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1.
- [46] B. Bose and E. Grimson, "Ground plane rectification by tracking moving objects," *IEEE International Workshop on Visual Surveillance and PETS*, 2003.

- [47] C. Stauffer and K. Tieu, “Automated multi-camera planar tracking correspondence modeling,” *Computer Vision and Pattern Recognition*, pp. 259–266, 2003.
- [48] A. Azarbayejani and A.P. Pentland, “Recursive estimation of motion, structure, and focal length,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, 1995.
- [49] F.M. Mirzaei and S.I. Roumeliotis, “A Kalman filter-based algorithm for IMU-camera calibration,” *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2427–2434, 2007.
- [50] E. Stringa and C.S. Regazzoni, “A novel camera calibration algorithm based on Kalman filter,” *Proceedings of 15th International Conference on Pattern Recognition*, vol. 1, pp. 872–875, 2000.
- [51] Rudolph Emil Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [52] T. PAEK and D.M. CHICKERING, “Evaluating the Markov assumption in Markov Decision Processes for spoken dialogue management,” *Language resources and evaluation*, vol. 40, no. 1, pp. 47–66, 2006.
- [53] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, Wiley New York, 2001.
- [54] S. Thrun, W. Burgard, and D. Fox, “Probabilistic Robotics (Intelligent Robotics and Autonomous Agents),” 2005.
- [55] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” *ACM SIGGRAPH 2001 Course Notes*, 2001.
- [56] D.C. Brown, “Close-range camera calibration,” *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.