Verifying Stereo Vision using Structure from Motion

by

Barry-Michael Morné van Wyk

Thesis presented in partial fulfilment of the requirements for the degree of

Master of Science in Engineering

at the University of Stellenbosch

Department of Mathematical Sciences,
University of Stellenbosch,
Private Bag X1, 7602 Matieland, South Africa.

Supervisors:

Dr. K.M. Hunter
Applied Mathematics
Department of Mathematical Sciences
University of Stellenbosch

Dr. N.L. Muller
Applied Mathematics
Department of Mathematical Sciences
University of Stellenbosch

March 2008
Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: ..............................
           B.M. van Wyk

Date: .................................

Copyright © 2008 University of Stellenbosch
All rights reserved.
Abstract

Verifying Stereo Vision using Structure from Motion

B.M. van Wyk

Department of Mathematical Sciences,
University of Stellenbosch,
Private Bag X1, 7602 Matieland, South Africa.
Thesis: MScEng (Applied Mathematics)
March 2008

The medical radiation treatment facility at iThemba Labs requires a precise and robust patient positioning system. The current system makes use of an accurately calibrated multi-camera stereophotogrammetry (SPG) setup that is vulnerable to physical disruptions that invalidate system calibration. The task in this thesis is to design a vision system that can be used to verify the correct operation of the SPG system. We propose an unscented Kalman filter (UKF) based structure from motion (SFM) system for this purpose. Our SFM system does not rely on calibration information used by the SPG system and provides accurate reconstruction for verification purposes. The system is critically evaluated against a set of synthetic and real motion sequences.
Uittreksel

Gebruik van Struktuur uit Beweging om Stereo Visie te verifieer

(Verifying Stereo Vision using Structure from Motion)

B.M. van Wyk

Departement Wiskundige Wetenskappe,
Universiteit van Stellenbosch,
Privaatsak X1, 7602 Matieland, Suid-Afrika

Tesis: MScIng (Toegepaste Wiskunde)
Maart 2008

Die mediese bestralingsbehandeling fasilititeit by iThemba Labs vereis ’n eksakte en betroubare pasint posisionering stelsel. Die huidige stelsel maak gebruik van akkuraat gekalibreerde stereo kameras. Enige fisiese versteuring sal die kalibrasie van hierdie stelsel ontwrig. Die doel van hierdie tesis is om ’n struktuur-uit-beweging stelsel te ontwerp wat die korrektheid van die stereo stelsel kan bevestig. Ons stelsel is gebaseer op ’n Kalman filter (unscented) en gebruik nie die stereo stelsel se kalibrasie inligting nie, maar voorsien ’n akkurate rekonstruksie vir verifikasie doeleindes. Die stelsel is krities gevalueer teen sintetiese en werklike data.
Acknowledgements

I would like to express my sincere gratitude to the following people and organisation who have contributed to this work:

- My study leaders Dr. Karin Hunter and Dr. Neil Muller for their guidance, ideas, motivation and patience
- iThemba Labs who provided me with the funding for my post graduate studies
- Mr. Evan de Kock who made me feel very welcome at iThemba Labs
- The Applied Mathematics division of the department of Mathematical Sciences
- Prof. BM Herbst for his help and knowledge
- Pieter Rautenbach for his advice and experience
- My Family for their support and motivation
- My girlfriend, Jolene
- All my friends
Dedications

This thesis is dedicated to my parents and sisters
# Contents

Declaration ................................................................. i  
Abstract ................................................................ ii  
Uittreksel ................................................................ iii  
Acknowledgements ......................................................... iv  
Dedications ................................................................ v  
Contents ................................................................ vi  
List of Figures ................................................................ ix  
List of Tables ................................................................ xii  
Nomenclature ................................................................ xiii  
Acronyms ................................................................ 1  

## 1 Introduction  
1.1 Patient positioning at iThemba Labs ......................... 2  
1.2 Aim of this thesis ...................................................... 2  
1.3 Proposed solution ...................................................... 3  
1.4 Contribution ............................................................ 4  
1.5 Thesis outline .......................................................... 4  

## 2 Literature Study  
2.1 Stereo vision ............................................................ 5  
2.2 Shape from shading .................................................. 5  
2.3 Structured light ......................................................... 6  
2.4 Structure from motion (SFM) ...................................... 6  
2.4.1 Matrix Factorisation (MF) ..................................... 7  
2.4.2 Filtering ............................................................ 8  
2.5 Summary ............................................................... 10
## CONTENTS

### 3 Quaternions

3.1 Fundamentals .................................................. 11
3.2 Quaternions and rotations ................................... 13
3.3 Derivative of a quaternion .................................... 14
3.4 Summary ...................................................... 16

### 4 The Kalman filter (KF)

4.1 Basic assumptions ............................................ 18
4.2 The linear Kalman filter (LKF) ............................... 18
4.3 The extended Kalman filter (EKF) ......................... 20
4.4 The unscented Kalman filter (UKF) ....................... 23
  4.4.1 The unscented transform (UT) ......................... 24
  4.4.2 The unscented Kalman filter algorithm ............... 27
4.5 Summary ...................................................... 29

### 5 Structure from motion

5.1 Observation modelling ....................................... 31
  5.1.1 Initialisation and setup of the observation model ... 34
5.2 Structure and motion models .................................. 35
  5.2.1 Structure model ........................................ 35
  5.2.2 Motion model .......................................... 35
  5.2.3 Combining the structure and motion models .......... 37
5.3 Initialisation and setup of structure and motion models . 37
  5.3.1 Initialisation with no knowledge of object’s structure or motion ........................................... 37
  5.3.2 Using pre-computed estimates of structure and motion for initialisation .................................... 38
5.4 Setup of noise models ....................................... 39
5.5 Dimensions of our state space and covariance matrices ... 40
5.6 Summary ...................................................... 40

### 6 Implementation Issues

6.1 Unscented Kalman filter .................................... 42
  6.1.1 Initialisation and setup ................................ 42
  6.1.2 Sigma structure ......................................... 44
6.2 Lens distortion ............................................... 44
6.3 Marker detection ............................................ 45
6.4 Euclidean 3-dimensional reconstruction via ground truth ... 46
6.5 Real time performance ....................................... 48
6.6 Summary ...................................................... 48


7 Tests and results

7.1 Error calculation ............................................. 49
7.2 Synthetic cube test sequences ................................. 50
  7.2.1 Test sequences without noise ............................ 51
  7.2.2 Test sequences with noise ............................... 58
  7.2.3 Summary of results for synthetic test sequences .... 66
7.3 Real sequences ................................................ 66
  7.3.1 Real sequence 1 - CT-SPG ............................... 67
  7.3.2 Real sequence 2 - Proton vault SPG system .......... 86
  7.3.3 Real sequence 3 - More on using the UKF to calculate
      initialisation data ....................................... 100
  7.3.4 Summary of results for real sequence 2 and 3 ...... 104
7.4 Summary ..................................................... 104

8 Conclusions .................................................. 106

A Supplementary information ................................. 109
A.1 Required Software ........................................ 109
A.2 Module description ...................................... 110
A.3 Graphical user interface (GUI) ........................... 111
A.4 Plotting Data ............................................. 115

B Supplementary CD ......................................... 117

Bibliography .................................................. 118
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Camera setup around CT scanner at iThemba Labs</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Camera setup in treatment vault at iThemba Labs</td>
<td>3</td>
</tr>
<tr>
<td>4.1</td>
<td>Diagram showing the Kalman filter loop</td>
<td>21</td>
</tr>
<tr>
<td>4.2</td>
<td>The UKF predicts the state more accurately, the EKF needs additional correction</td>
<td>24</td>
</tr>
<tr>
<td>5.1</td>
<td>Linear perspective projection of a point $P$ onto the image plane $Y$ at focal length $-f$</td>
<td>31</td>
</tr>
<tr>
<td>5.2</td>
<td>The points $P_1^{ccs}$ and $P_2^{ccs}$ yield the same observation</td>
<td>32</td>
</tr>
<tr>
<td>5.3</td>
<td>Linear perspective projection of a point $P$ onto the image plane $Y$ at focal length $-f$ in terms of the OCS</td>
<td>33</td>
</tr>
<tr>
<td>6.1</td>
<td>The effect of radial lens distortion on straight lines</td>
<td>45</td>
</tr>
<tr>
<td>6.2</td>
<td>Distortion object used at iThemba Labs for lens distortion correction</td>
<td>46</td>
</tr>
<tr>
<td>6.3</td>
<td>Special retro-reflective radiopaque markers used at iThemba Labs</td>
<td>47</td>
</tr>
<tr>
<td>7.1</td>
<td>Observed and estimated coordinates - pure rotation</td>
<td>52</td>
</tr>
<tr>
<td>7.2</td>
<td>Estimated structure parameters and RMS error of UKF - pure rotation</td>
<td>52</td>
</tr>
<tr>
<td>7.3</td>
<td>Estimated motion of synthetic cube - pure translation</td>
<td>53</td>
</tr>
<tr>
<td>7.4</td>
<td>Extract from the reconstruction sequence of synthetic cube - pure rotation</td>
<td>53</td>
</tr>
<tr>
<td>7.5</td>
<td>Observed and estimated coordinates - pure translation</td>
<td>54</td>
</tr>
<tr>
<td>7.6</td>
<td>Estimated structure parameters and RMS error of UKF - pure translation</td>
<td>54</td>
</tr>
<tr>
<td>7.7</td>
<td>Estimated motion of synthetic cube - pure translation</td>
<td>55</td>
</tr>
<tr>
<td>7.8</td>
<td>Observed and estimated coordinates - rotation and translation</td>
<td>56</td>
</tr>
<tr>
<td>7.9</td>
<td>Estimated structure parameters and RMS error of UKF</td>
<td>57</td>
</tr>
<tr>
<td>7.10</td>
<td>Estimated motion of synthetic cube - rotation and translation</td>
<td>57</td>
</tr>
<tr>
<td>7.11</td>
<td>Observed and estimated coordinates - noise variance 0.0001</td>
<td>59</td>
</tr>
<tr>
<td>7.12</td>
<td>Estimated structure parameters - noise variance 0.0001</td>
<td>59</td>
</tr>
<tr>
<td>7.13</td>
<td>Estimated motion of synthetic cube - noise variance 0.0001</td>
<td>60</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

7.14 Estimated structure parameters and RMS error of UKF - noise variance 0.001 ........................................ 61
7.15 Estimated motion of synthetic cube - noise variance 0.001 ........................................ 62
7.16 Observed and estimated coordinates - noise variance 0.01 ........................................ 63
7.17 Estimated structure parameters and RMS error of UKF - noise variance 0.01 ........................................ 64
7.18 Estimated motion of synthetic cube - noise variance 0.01 ........................................ 64
7.19 Extract from the reconstruction sequence of synthetic cube - noise variance 0.01 ........................................ 65
7.20 Detected markers tracked during real sequence 1 ........................................ 68
7.21 Observed and estimated coordinates - Test 1 ........................................ 69
7.22 Estimated structure parameters and RMS error of UKF - Test 1 ........................................ 69
7.23 Estimated motion of marker carrier - Test 1 ........................................ 70
7.24 Extract from the reconstruction of marker carrier - Test 1 ........................................ 70
7.25 Estimated structure parameters and RMS error of UKF - Test 2 ........................................ 71
7.26 Estimated motion of marker carrier - Test 2 ........................................ 72
7.27 Estimated structure parameters and RMS error of UKF - Test 3 ........................................ 73
7.28 Estimated motion of marker carrier - Test 3 ........................................ 74
7.29 Observed and estimated coordinates - Test 4 ........................................ 76
7.30 Estimated structure parameters and RMS error of UKF - Test 4 ........................................ 76
7.31 Estimated motion of marker carrier - Test 4 ........................................ 77
7.32 Estimated structure parameters and RMS error of UKF - Test 5 ........................................ 78
7.33 Estimated motion of marker carrier - Test 5 ........................................ 79
7.34 Estimated structure parameters and RMS error of UKF - Test 6 ........................................ 80
7.35 Estimated motion of marker carrier - Test 6 ........................................ 81
7.36 Estimated structure parameters and RMS error of UKF - Test 7 ........................................ 82
7.37 Estimated motion of marker carrier - Test 7 ........................................ 83
7.38 Reconstructed marker carrier from PMF system - Test 8 ........................................ 84
7.39 Estimated structure parameters and RMS error of UKF - Test 8 ........................................ 85
7.40 Estimated motion of marker carrier - Test 8 ........................................ 85
7.41 Detected markers tracked during real sequence 2 ........................................ 87
7.42 Observed and estimated coordinates - Test 9 ........................................ 88
7.43 Estimated structure parameters and RMS error of UKF - Test 9 ........................................ 88
7.44 Estimated motion of marker carrier - Test 9 ........................................ 89
7.45 Extract from the reconstruction of marker carrier - Test 9 ........................................ 89
7.46 Estimated structure parameters and RMS error of UKF - Test 10 ........................................ 90
7.47 Estimated motion of marker carrier - Test 10 ........................................ 91
7.48 Estimated structure parameters and RMS error of UKF - Test 11 ........................................ 92
7.49 Estimated motion of marker carrier - Test 11 ........................................ 93
7.50 Observed and estimated coordinates - Test 12 ........................................ 94
7.51 Estimated structure parameters and RMS error of UKF - Test 12 ........................................ 95
7.52 Estimated motion of marker carrier - Test 12 ........................................ 95
7.53 Estimated structure parameters and RMS error of UKF - Test 13 ........................................ 96
LIST OF FIGURES

7.54  Estimated motion of marker carrier - Test 13 .......................... 97
7.55  Reconstructed marker carrier from PMF system - Test 14 .......... 98
7.56  Estimated structure parameters and RMS error of UKF - Test 14 99
7.57  Estimated motion of marker carrier - Test 14 .......................... 99
7.58  Detected markers tracked during real sequence 3 ....................... 100
7.59  Observed and estimated coordinates - Test 15 ........................... 101
7.60  Estimated structure parameters and RMS error of UKF - Test 15 102
7.61  Estimated motion of marker carrier - Test 15 ............................ 102
7.62  Extract from the reconstruction of marker carrier - Test 15 ...... 103

A.1   SFM GUI - main frame ...................................................... 111
A.2   SFM GUI - Select video sequence ......................................... 112
A.3   SFM GUI - remove markers that mat become occluded ................. 113
A.4   SFM GUI - select frame to obtain structure from .................... 114
A.5   SFM GUI - using mayavi to view 3-dimensional structure ......... 115
A.6   SFM GUI - select corresponding markers for Euclidean recon- struction ................................................................. 116
# List of Tables

6.1 Real time performance of our system ............................................. 48

7.1 Summary of results from synthetic test cases ................................. 66

7.2 Summary of results from real sequence 1 ...................................... 86

7.3 Summary of results from real sequence 2 and 3 .............................. 104
Nomenclature

Constants
\[ f \] focal length
\[ m \] number of features being tracked
\[ n \] augmented state dimension

Variables
\[ s \] structure depth
\[ \theta_{dim} \] rotation angle in radians
\[ \omega_{dim} \] angular velocity (radians per frame)
\[ t_{dim} \] translation
\[ d_{dim} \] translation velocity (units per frame)
\[ W_j^m \] Sigma point weights for mean
\[ W_j^c \] Sigma point weights for covariance

Scalars, Vectors and Matrices
\[ a \] scalar quantity
\[ n \] vector
\[ q \] quaternion
\[ I \] Identity matrix
\[ M \] matrix
\[ \chi \] sigma point distribution

Functions and Operators
\[ E[\cdot] \] expectation operator
\[ f(\cdot) \] vector function
\[ p(\cdot) \] general probability density function
\[ \eta(\cdot) \] Gaussian probability density function
\[ O(\cdot) \] computational complexity
Subscripts

\( i \) \( i \)-th time-step
\( j \) \( j \)-th time-step
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCS</td>
<td>Camera coordinate system</td>
</tr>
<tr>
<td>COP</td>
<td>Centre of projection</td>
</tr>
<tr>
<td>CT</td>
<td>Computer tomography</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td>HOT</td>
<td>Higher order term</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman filter</td>
</tr>
<tr>
<td>LKF</td>
<td>Linear Kalman filter</td>
</tr>
<tr>
<td>MF</td>
<td>Matrix factorisation</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum mean squared error</td>
</tr>
<tr>
<td>OCS</td>
<td>Object coordinate system</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability density function</td>
</tr>
<tr>
<td>PF</td>
<td>Particle filter</td>
</tr>
<tr>
<td>PMF</td>
<td>Projective matrix factorisation</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root mean squared error</td>
</tr>
<tr>
<td>RV</td>
<td>Random variable</td>
</tr>
<tr>
<td>SFM</td>
<td>Structure from motion</td>
</tr>
<tr>
<td>SFS</td>
<td>Shape from shading</td>
</tr>
<tr>
<td>SUT</td>
<td>Scaled unscented transform</td>
</tr>
<tr>
<td>UT</td>
<td>Unscented transform</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman filter</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Cancer is one of the leading causes of death worldwide. From a total of 58 million deaths worldwide in 2005, cancer accounted for 7.6 million (or 13% of all deaths [1]). Cancer is broadly defined as the uncontrolled growth and reproduction of a group of cells, called a tumour. Cancer cells from the primary tumour can metastasise (move to other parts of the body); 30% of patients have metastases at diagnosis.

The three main forms of cancer treatment are surgery, chemotherapy and radiotherapy. These treatments can be used in conjunction with each other or in isolation, depending on the particular form of cancer being treated. The success of a cancer treatment is determined by the survival rate of the patients. A patient surviving for 5 years after treatment without further symptoms is said to be cured. The five year survival rate of treated cancer sufferers is about 45% [2].

Radiotherapy is a localised form of treatment that is used to treat the primary tumour and is (possibly combined with surgery) responsible for about 40% of all cancer cures [2]. Radiotherapy uses ionising radiation for the treatment of cancerous tumours as the cancer cells are more sensitive to ionising radiation than normal cells within the same organ. There are many types of radiation including photons (X-rays), electrons, neutrons and protons. For radiotherapy to be effective an accurate beam delivery system, precise tumour and critical structure localisation, accurate and reproducible patient positioning and accurate 3D treatment planning is required.

The Medical Radiation Group (MRG) at iThemba Labs has treatment facilities for neutron, proton and photon radiotherapy.


1.1 Patient positioning at iThemba Labs

The current patient positioning system at iThemba Labs makes use of a motorised treatment chair to which the patient is fixed with an immobilisation device. The patient is then moved into position for treatment by a unique system based on multi-camera stereophotogrammetry (SPG).

The patient wears a custom-made plastic mask that carries radiopaque and retro-reflective markers. The mask is worn during the treatment planning phase at which time the position of the markers on the mask relative to the tumour are acquired from computer tomography (CT) data. The SPG system accurately calculates the position of the markers detected by the CCD cameras and uses this to automatically position the treatment chair. This positioning system aligns the tumour and treatment beam with an accuracy of approximately 1mm.

There is a new patient positioning system planned at iThemba Labs which will make use of a commercial robot together with a rigid chair or couch for patient support. SPG techniques will be employed for automatic patient positioning and work is being done on the automatic verification and correction of the patients position at treatment.

1.2 Aim of this thesis

The SPG system used at iThemba Labs for patient positioning needs to be very accurately calibrated to position the patients within the required precision. Currently there is no way of verifying the accuracy of the SPG system.
CHAPTER 1. INTRODUCTION

Figure 1.2: Camera setup in treatment vault at iThemba Labs

at treatment or pre-treatment phase without using a calibration object. If a camera were to be moved or knocked by mistake or something untoward occurred to the camera rig, the calculated calibration data would no longer be accurate and this would not be detected until tests are performed using the calibration object.

The SPG system at iThemba Labs relies on the multi-camera aspect to aid in robustness for small changes in position. The aim of this thesis is to develop an independent vision system to verify the operation of and the marker positions calculated by the SPG system. This vision system needs to be accurate, but does not require the same accuracy as the SPG system since its intended use is it to identify large inaccuracies in the reconstruction. The system also needs to run at near real-time speed and remain independent of the current SPG system: not relying on any of the SPG calibration data.

1.3 Proposed solution

In this thesis we use a Kalman filter (KF) based structure from motion system to estimate the structure of the tracked markers. The KF will run at near real time giving a structure estimation at each frame. The KF will estimate the structure and motion of the mask up to a scale factor as we do not use calibration information from the SPG system. The Euclidean structure will be recovered from the projective reconstruction using the ground truth method with real world coordinates of the markers on the mask provided from the CT data. Doing this allows us to use a small set of ground truth observations to validate all the markers.
We will investigate methods to improve the speed of convergence of the KF and the performance of the KF when the assumptions made about our structure and motion models are not valid.

1.4 Contribution

In this thesis we build on the work of Rautenbach [3] and move on to develop a practical implementation for use in the patient positioning system at iThemba Labs. We further look to improve the performance of the UKF with regards to convergence speed and structure estimation by using pre-computed initialisation information and by allowing the UKF to adaptively adjust the covariance values associated with noise encountered in our system.

1.5 Thesis outline

Background information and concepts important to structure from motion and computer vision are discussed in Chapter 2. We then move on to discuss quaternions in Chapter 3 which are very important in our motion model and representation of rotations in the KF. The linear Kalman filter (LKF), extended Kalman filter (EKF) and unscented Kalman filter (UKF) are investigated in Chapter 4. The models used for our observation process, structure and motion are presented in Chapter 5 with the implementation issues dealt with in this thesis addressed in Chapter 6. We show tests and results for our system in Chapter 7 and end the thesis with our conclusions in Chapter 8.
Chapter 2

Literature Study

Three dimensional scene reconstruction is part of the field of study known as computer vision. Extensive research has been done in this field and it has become of huge interest to society with many applications in surveillance and security, robotics, virtual reality, autonomous vehicle navigation and sports analysys.

Computer vision research started in the 1970’s where computers were used to enhance input images and detect image features. In this chapter we discuss the most commonly used methods and applications applicable to this thesis for 3-dimensional reconstruction.

2.1 Stereo vision

Stereo vision is a reconstruction method in which at least two images from different views of a scene are used to reconstruct the scene’s 3-dimensional structure. For this method of reconstruction we require corresponding points to be detected in each image and we assume we have knowledge of the camera matrices associated with each frame. If calibrated cameras are used we are able to reconstruct the scene accurately to its Euclidean structure. If we choose not to use calibrated cameras we are able to reconstruct the scene up to an unknown projective transform, requiring further knowledge of the 3-dimensional scene to move the reconstruction up to a Euclidean reconstruction.

As mentioned in the introduction the patient positioning system at iThemba Labs uses stereo vision methods using multiple cameras [4] to establish the position of markers on the treatment mask within 1mm of their actual position. To reach this accuracy the cameras are accurately calibrated and accurate marker detection techniques are employed. The marker detection will be discussed in Chapter 6 when we discuss the implementation issues we dealt with
in this thesis.

An example of an uncalibrated stereo vision system is [5] where Mohr et al. use multiple uncalibrated images to do a relative reconstruction of a scene using point correspondences between images. Tomasi and Kanade did pioneering work in the field of computer vision publishing papers on stereo vision and image registration [6], feature tracking [7] and structure from motion techniques based on matrix factorisation [8; 9] which we discuss in Section 2.3.1.

2.2 Shape from shading

Shape from shading (SFS) allows the recovery of a scene’s structure using only one image of the scene. SFS uses shading information in the image to extract structure information in the scene. Research in this field started in the late 1970’s when Horn introduced the idea in [10]. He was the first to formulate the shape from shading problem simply and rigorously as that of finding the solution of a nonlinear first-order partial differential equation known as the brightness equation [11].

In [12] Atick, Griffin and Redlich proposed a SFS system in which they suggest the brain classifies objects into lower dimensional object classes according to their shape. They use this to simplify the extraction of shape from shading to the problem of parameter estimation in a lower dimensional space. They did not provide any conclusive results.

2.3 Structured light

The idea behind structured light systems is to replace one of the cameras in a stereo vision system with a projector. The projector is used to display a known pattern over a scene which is captured by a camera. Stereo vision methods can be used to reconstruct the scene using the projected pattern and captured image as the stereo image pair. The advantage of using a structured light system is that it simplifies the task of identifying corresponding features between frames and allows corresponding features to be identified very accurately.

Lequellec et al. in [13] proposes a structured light system that uses a CCD camera and light beam matrix to reconstruct the 3-dimensional surface shape of a vehicle cockpit. In [14] Scharstein et al. develop a structured light system for the calculation of high accuracy stereo depth maps. Their motivation for this is the need of accurate ground truth information of more challenging 3-dimensional scenes.
2.4 Structure from motion (SFM)

Structure from motion attempts to reconstruct the structure and motion of an object or 3-dimensional scene from the observed 2-dimensional motion of the object. SFM is similar to stereo vision as it uses interframe differences of corresponding features to estimate structure. Stereo vision methods are more computationally complex than SFM methods but SFM needs to process many frames compared to a minimum of two for stereo vision.

There are two well known categories of SFM systems: those based on a filtering algorithm that minimise the error in the predicted observation and the actual observation (such as Kalman filters and particle filters) and methods based on matrix factorisation techniques such as [15; 16; 9]. We discuss both these approaches next.

2.4.1 Matrix Factorisation (MF)

In principle three orthographic images of four points are sufficient to recover the positions of the points relative to each other (shape) and the viewpoints from which the images were taken (motion) [16]. The matrix factorisation (MF) method makes use of this principle to recover the shape and motion of an object from a stream of images. It is important to note that under an orthographic projection, pure translation is never enough to recover structure.

Tomasi and Kanade proposed a method in [16; 15] that allows them to recover the shape and motion of an object from a stream of images, initially only considering planar motion and then moving on to 3-dimensional motion. This method involves tracking corresponding features between frames and arranging them in a measurement matrix of dimension $2F \times P$, where $P$ is the number of points and $F$ the number of frames in the sequence. This matrix gathers the horizontal and vertical coordinates of $P$ points through $F$ frames. They show that if all objects are measured with respect to their centroids that the measurement matrix is of rank three under an orthography. Using this they cast the SFM problem as a matrix factorisation problem and make use of a singular value decomposition based algorithm to solve it. They present their results and show that the method gives accurate results. In 1996 Sturm and Triggs [17] updated the algorithm for the projective camera. In [9] Han and Kanade build on the MF method to extract motion information for multiple objects in a scene.

The MF method requires batch processing of the complete set of frames to calculate the structure and motion for the sequence and thus does not lend itself to real time applications.
We make use of an implementation of the MF method by de Vaal [18] to provide our system with initialisation information for the structure and motion models.

2.4.2 Filtering

Filtering based SFM methods require models for the structure and motion characteristic of the object being tracked. We require models for the camera’s projective geometry and the relative motion of the camera, these are referred to as the state transition and observation model. The observation model uses the state transition model to predict an observation. The difference between the predicted and actual observation is then minimised to provide a better estimate. The accuracy of filtering based SFM methods depends heavily on these models.

There are two popular filtering methods used in computer vision, Kalman filtering which is a parametric method that estimates the parameters of a probability density function (PDF), usually a Gaussian PDF, and particle filters (PF) which are non-parametric. PF’s fall in the class of sequential Monte-Carlo methods (SMC) that estimate the unknown PDF using samples or particles from which statistics of the PDF are drawn [19; 20].

Particle filters (PF)

PF’s are widely used in computer vision for visual tracking with applications in surveillance, teleconferencing, human computer interaction and visual speech recognition.

Theoretically PF’s are able to represent any non-linearity or PDF and if enough particles are chosen the estimated PDF will approach the true PDF. For PF’s to be effective one needs to design a proposal distribution that fits the posterior distribution reasonably well [21]. A common method is to use an EKF or UKF to propose the prior PDF the particles are chosen from. These particles are then used to calculate the particles that estimate the posterior PDF. In [21] van der Merwe et al. combines a UKF with a PF that allows the PF to incorporate the latest observations in the prior updating routine.

The main issues affecting PF’s are particle degeneration, which is when one particle gains most of the weight while the rest become insignificant, and computational expense as the number of particles used increases exponentially as the state dimension of the PF increases.
A good introduction to PF’s is provided by Hoffmann in [20] where he combines PF’s and active appearance models to track human movement.

**Kalman filters (KF)**

Kalman filters (KF) are popular in the control system and computer vision community for their ease of implementation, robustness and low computational complexity. There are three main flavours of the KF, namely the linear Kalman filter (LKF) used for linear problems, the extended Kalman filter (EKF) and unscented Kalman filter (UKF) which are used for non-linear problems such as SFM.

Detailed research has been done in this field by Azarbayejani et al. [22], and Jabara et al. [23] [24]. They formulated a method for the recursive recovery of point wise structure and motion of a tracked object and the focal length of the camera being used. Their approaches were based on the EKF with the results provided showing their system is robust and accurate.

Julier et al. [25; 26; 27; 28] proposed the UKF for use in structure from motion problems. The UKF is based on the unscented transform that allows the probability distribution of the KF to be propagated through a non-linear transform rather than trying to linearise the transform itself as in the EKF.

Rautenbach [3], Magaia [29], Malan [30] and Venter [31] use the UKF proposed by Julier et al. to perform structure and motion estimation. In [29] a video based traffic monitoring system is developed to accurately identify and track vehicles at traffic intersections. Malan [30] develops a system to track the motion of satellites relative to a structure (earth). They use monocular vision to estimate the structure and pose of the cameras (satellites) relative to the structure(earth) to calculate the satellite’s position in order to maintain formation flight. Venter [31] develops a SFM system to reconstruct tracked features in a scene. Rautenbach [3] builds on the work of Venter to develop a facial reconstruction system. Rautenbach adopts a simpler motion model to Venter and investigates the occlusion of tracked features during a sequence.

The work in this thesis builds onto that of [3] where we investigate methods to improve the convergence speed of the KF and to improve the performance of the KF in situations where the motion models fail. We discuss the KF in detail in Chapter 4.
CHAPTER 2. LITERATURE STUDY

2.5 Summary

In this chapter we presented the most common methods in computer vision for structure estimation. We discussed stereo vision, shape from shading and structure from motion. We provided an overview of these methods and learnt from their characteristics that stereo vision and structured light do not lend themselves to the task at hand as each of these systems rely on calibration. Shape from shading did not offer the accuracy we required in our reconstruction.

During our discussion of structure from motion methods we learnt there were two popular approaches to this task, those based on filtering and those on factorisation methods. The factorisation approach requires a batch processing of the image sequence and therefore is not eligible for real time SFM estimation. We did decide to make use of this method to provide our system with initialisation information to attempt to speed up convergence.

The filtering approaches are again split into two classes, particle filters (PF) and Kalman filters (KF). PF’s are used widely in tracking application [20], often combined with KF’s to improve performance or provide initialisation information for the particles. PF’s are very good when the noise characteristics of a system do not adhere to standard probability distribution functions (PDF), but do get very computationally expensive when many particles are used to estimate the PDF or the state of the system becomes large.

KF’s have been used widely for a long time and are well understood. They are robust and computationally inexpensive compared to stereo vision, but do need to process many more frames to recover structure and motion information. We choose to use a KF based SFM system for this thesis.
Chapter 3
Quaternions

Quaternions were invented by William Rowan Hamilton in 1843 in his effort to construct hypercomplex numbers, or higher dimensional generalisations of complex numbers [32]. They offer an efficient and simple method for representing the rotation of a tracked object and have been very important in the fields of structure from motion (SFM) [3] and computer graphics [33].

A single quaternion can represent a transformations in $\mathbb{R}^3$ and according to [34] offers the following advantages over homogeneous transformations for representation of rotations:

- They are simpler, more intuitive representations of rotations.
- They are small in size compared to the Euler rotation matrix representation.
- They are not ambiguous
- They do not suffer from gimbal lock
- They offer easier rotation interpolation.

We start this chapter with a brief overview of the fundamental mathematics behind quaternions. We then discuss how we relate a rotation to a quaternion and describe a formula for the calculation of the time derivative of the quaternion.

3.1 Fundamentals

In this section we present a general discussion on quaternions and will discuss the specific relationship between a quaternion and rotation in section 3.2. A
quaternion is defined as a four dimensional vector over the quaternion space $\mathbb{H}$,

$$\mathbf{q} = [q_0, q_1, q_2, q_3].$$  \hfill (3.1)

It is convenient to visualise as a quaternion as a 3-dimensional vector extended by a real number. We can write a quaternion as

$$\mathbf{q} = [v, \mathbf{w}]$$  \hfill (3.2)

where $v$ is a scalar and $\mathbf{w}$ is a 3-dimensional vector to represent this.

In this thesis we use the convention that we write quaternions of the form $[v, 0]$ or $[v, 0, 0, 0]$ as just $v$ where $v \in \mathbb{R}$ and $[v, 0, 0, 0] \in \mathbb{H}$ to aid the readability our discussion.

Given two quaternions $\mathbf{q}_1 = [v_1, \mathbf{w}_1]$ and $\mathbf{q}_2 = [v_2, \mathbf{w}_2]$ we use the following definitions from [32]:

**Definition 1: Addition**

$$\mathbf{q}_1 + \mathbf{q}_2 = [(v_1 + v_2), (\mathbf{w}_1 + \mathbf{w}_2)].$$  \hfill (3.3)

**Definition 2: Multiplication**

$$\mathbf{q}_1 \mathbf{q}_2 = [(v_1 v_2 - \mathbf{w}_1 \cdot \mathbf{w}_2), (v_1 \mathbf{w}_2 + v_2 \mathbf{w}_1 + \mathbf{w}_1 \times \mathbf{w}_2)].$$  \hfill (3.4)

where $\mathbf{w}_1 \cdot \mathbf{w}_2$ is the dot product, and $\mathbf{w}_1 \times \mathbf{w}_2$ is the cross product of the two 3-dimensional vectors.

From this definition it is important to note that the multiplication of two quaternions is not commutative as the cross product of two vectors is not commutative. We can write (3.4) in matrix notation as [30]

$$\mathbf{q}^a \mathbf{q}^b = \begin{bmatrix} q_0^a & -q_1^a & -q_2^a & -q_3^a \\ q_1^a & q_0^a & q_3^a & -q_2^a \\ q_2^a & -q_3^a & q_0^a & q_1^a \\ q_3^a & q_2^a & -q_1^a & q_0^a \end{bmatrix} \begin{bmatrix} q_0^b \\ q_1^b \\ q_2^b \\ q_3^b \end{bmatrix},$$  \hfill (3.5)

which is very useful for computational purposes.

**Definition 3: The conjugate of a quaternion $\mathbf{q} = [v, \mathbf{w}]$ is**

$$\mathbf{q}^c = [v, -\mathbf{w}].$$  \hfill (3.6)

**Definition 4: The norm of a quaternion $\mathbf{q} = [v, \mathbf{w}]$ is**

$$N(\mathbf{q}) = \mathbf{q} \mathbf{q}^c = [v^2 + \mathbf{w} \cdot \mathbf{w}, 0] = v^2 + \mathbf{w} \cdot \mathbf{w}.$$  \hfill (3.7)
The norm is a real-valued function with the norm of a product of quaternions satisfying the property $N(pq) = N(p)N(q)$ and the norm of the conjugate of a quaternion satisfying $N(q^*) = N(q)$ [35].

**Definition 5:** A *unit quaternion* $q = [v, w]$ is

$$Q_1 := \{q|N(q) = 1\}.$$  \hfill (3.8)

**Definition 6:** A *pure quaternion* $q = [v, w]$ is

$$Q_0 = \{q|q = [0, w]\}.$$  \hfill (3.9)

**Definition 7:** The multiplicative inverse of a quaternion $q = [v, w]$, is

$$q^{-1} = \frac{[v, -w]}{N(q)},$$  \hfill (3.10)

where $(N(q) \neq 0) \Rightarrow q \neq 0$.

The multiplicative inverse has the property that $qq^{-1} = q^{-1}q = [1, 0] = 1$. The inverse of this operation satisfies $(q^{-1})^{-1} = q$ and $(pq)^{-1} = p^{-1}q^{-1}$ [35].

### 3.2 Quaternions and rotations

Quaternions are very useful in representing rotations in terms of the Euler axis $\hat{e} = [e_x, e_y, e_z]$ and angle $\theta$ [30]. If we restrict $\hat{e}$ such that $||\hat{e}|| = 1$ the Euler axis and angle representation can be written as an equivalent quaternion $q = [v, w]$ with $w = [w_0, w_1, w_2]$ by

$$q = \left[\cos \left(\frac{\theta}{2}\right), \hat{e} \sin \left(\frac{\theta}{2}\right)\right].$$  \hfill (3.11)

On expansion of the above we get a quaternion $q = [v, w]$ with

$$v = \cos \left(\frac{\theta}{2}\right),$$
$$w_0 = e_x \sin \left(\frac{\theta}{2}\right),$$
$$w_1 = e_y \sin \left(\frac{\theta}{2}\right),$$
$$w_2 = e_z \sin \left(\frac{\theta}{2}\right)$$  \hfill (3.12)

where $v$ describes the rotation angle and $w_1, w_2$ and $w_3$ the axis of rotation [34].

This quaternion representation is close to the minimal axis-and-angle rotation
and can be transformed into a rotation matrix. The $3 \times 3$ rotation matrix $R$ can be constructed from a quaternion $q = [v, w]$ as follows

$$R(q) = \begin{bmatrix}
  v^2 + w_0^2 - w_1^2 - w_2^2 & 2(w_0w_1 + vw_2) & 2(w_0w_2 - vw_1) \\
  2(w_0w_1 - vw_2) & v^2 - w_0^2 + w_1^2 - w_2^2 & 2(w_1w_2 + vw_0) \\
  2(w_0w_2 + vw_1) & 2(w_1w_2 - vw_0) & v^2 - w_0^2 - w_1^2 + w_2^2
\end{bmatrix}.$$  

As a result of the restriction that $||\hat{e}|| = 1$ we see that

$$v^2 + w_0^2 + w_1^2 + w_2^2 = 1$$

which indicates the parametrisation satisfies the definition of a unit quaternion in (3.8).

As a result of this normality constraint the rotation matrix $R$ is orthonormal,

$$RR^T = I = R^T R,$$

with

$$\det(R) = 1$$

constraining the rotation matrix not to allow reflection [3].

We have shown how we can relate a quaternion to the Euler axis and angle, but how do we combine two quaternions (rotations) together? This is done in the same way we combine rotation matrices, we multiply the quaternions [30]. Note should be taken that the order in which the rotations are applied in (the order the multiplication operations are performed) is very important as multiplication of quaternions is not commutative.

### 3.3 Derivative of a quaternion

The time-derivative of a quaternion is very important in this thesis as it forms part of the motion dynamics model for the tracked object. We will follow the same convention for formulation as in [3] which is based on the assumptions made in [36].

Let the quaternions $q(t)$ and $q(t + \Delta t)$ represent the orientation of a rigid body with respect to a reference coordinate system at subsequent steps in time. Let the quaternion $q(\Delta t)$ relate $q(t)$ to $q(t + \Delta t)$ such that

$$q(t + \Delta t) = q(\Delta t)q(t).$$

(3.17)
We can express $q(\Delta t)$ in Euler axis and angle form using (3.11)

$$q(\Delta t) = \left[ \cos \frac{\Delta \theta}{2}, \hat{e} \sin \frac{\Delta \theta}{2} \right].$$  \hspace{1cm} (3.18)

The change in $\hat{e}$ over the time interval $\Delta t$ causes second order terms that can be ignored [3; 36].

Using (3.5) we rewrite (3.18) as

$$q(t + \Delta t) = \left[ \cos \left( \frac{\Delta \theta}{2} \right) I + N \sin \left( \frac{\Delta \theta}{2} \right) \right] q(t)$$  \hspace{1cm} (3.19)

where

$$N = \begin{bmatrix}
0 & -e_x & -e_y & -e_z \\
e_x & 0 & -e_z & e_y \\
e_y & e_z & 0 & -e_x \\
e_z & -e_y & e_x & 0
\end{bmatrix}$$  \hspace{1cm} (3.20)

is an antisymmetrical matrix and $I$ is the identity matrix.

The instantaneous angular velocity of the rigid body is given by

$$\omega = \lim_{\Delta t \rightarrow 0} \frac{\Delta \theta}{\Delta t}.$$  \hspace{1cm} (3.21)

In the case where $\Delta t$ is small we define

$$\Delta \theta = \omega \Delta t.$$  \hspace{1cm} (3.22)

The angular velocity vector is given by

$$\hat{\omega} = \omega \hat{e}.$$  \hspace{1cm} (3.23)

$$\hat{\omega} = [\omega_x, \omega_y, \omega_z].$$  \hspace{1cm} (3.24)

We can simplify (3.19) by using the following small angle approximation (from first order Taylor expansions)

$$\cos \frac{\Delta \theta}{2} \approx 1$$  \hspace{1cm} (3.25)

and

$$\sin \frac{\Delta \theta}{2} \approx \frac{1}{2} \omega \Delta t.$$  \hspace{1cm} (3.26)

to get

$$q(t + \Delta t) = \left[ I + \frac{1}{2} \Omega(\hat{\omega}) \Delta t \right] q(t).$$  \hspace{1cm} (3.27)
with
\[
\Omega = \begin{bmatrix}
0 & -\omega_z & -\omega_y & -\omega_x \\
\omega_x & 0 & -\omega_z & \omega_y \\
\omega_y & \omega_z & 0 & -\omega_x \\
\omega_z & -\omega_y & \omega_x & 0
\end{bmatrix}.
\] (3.28)

The definition for the derivative of a quaternion is
\[
\frac{d}{dt}q(t) = \lim_{\Delta t \to 0} \frac{q(t + \Delta t) - q(t)}{\Delta t}
\] (3.29)
and from (3.27) and (3.29) it follows
\[
\frac{d}{dt}q(t) = \frac{1}{2} \Omega(\dot{\omega})q(t).
\] (3.30)

This represents a homogeneous system of linear first order differential equations. We can solve this exactly if we are given the initial condition \(q_0 = q(t_0)\) and have a constant angular velocity using
\[
q(t) = e^{\frac{1}{2}(t-t_0)\Omega(\omega)}q_0.
\] (3.31)

The matrix exponential for an arbitrary matrix \(A\) is defined as
\[
e^{tA} = \sum_{n=0}^{\infty} \frac{(tA)^n}{n!}
\] (3.32)
\[
= I + tA + \frac{t^2}{2!}A^2 + \frac{t^3}{3!}A^3 + \frac{t^4}{4!}A^4 + \ldots
\] (3.33)

### 3.4 Summary

In this chapter we introduced the quaternion and discussed its advantages above homogeneous representations of rotations for rigid bodies. We then summarised the fundamental mathematics surrounding quaternions and showed how we can relate quaternions to rotations, specifically the Euler axis and angle representation of rotations. We finally proceeded to derive a formula for the time-derivative of a time-dependent quaternion under the assumption of a constant velocity that forms part of our dynamic model for the motion of a rigid object.
Chapter 4

The Kalman filter (KF)

If we have a noisy dynamical system with internal state \( x_i \) at time \( i \) that we cannot measure directly, we need an approach that estimates the internal state given (possibly noisy) observations \( y_i \) at time \( i \). When viewed in this way, this problem is filtering in the sense of [3] and is important as it affects many fields of study. The Kalman filter offers us a solution to this problem.

The Kalman filter, first introduced in 1960 by Kalman [37], is a mean squared error minimiser that estimates the state of a dynamic system from a series of measurements under the influence of Gaussian noise. The Kalman filter has a proven track record in practice and as stated in [38] is theoretically attractive because it can be shown that of all possible filters it is the only one to minimise the variance of the estimation error.

The Kalman filter has been used in many fields of engineering; from aircraft tracking to the rectification of electro-magnetic signals and structure from motion. It also has been applied to problems outside these fields such as commerce with an example being the development of a regional economic activity index for the Chicago metropolitan area [39].

When the Kalman filter was first described in 1960 it could only be used for linear system estimations. Since then much work has been done on techniques to expand the scope of the Kalman filter to deal with non-linear systems. In this section we will discuss the linear Kalman filter (LKF) as well as two extensions to it, namely the extended Kalman filter (EKF) used in [22] and the unscented Kalman filter (UKF) [25].
4.1 Basic assumptions

The Kalman filter (KF) uses the first order Markov assumptions. These assumptions are that the current state $x_i$ only depends on the previous state $x_{i-1}$, in terms of a probability function $p$. It is given by

$$p(x_i|x_{i-1})$$

(4.1)

and that the current observations $y_i$ only depend on the current state $x_i$

$$p(y_i|x_i).$$

(4.2)

We will use the notation

$$\mathbf{x}_{ij}=E(x_i|y_1,...,y_{i-1})$$

(4.3)

to indicate that the predicted state mean, $\mathbf{x}_{ij}$, is the expected value of the current state $x_i$, given the observations $y_1,...,y_{i-1}$. In the same way

$$\mathbf{y}_{ij}=E(y_i|x_1,...,x_{i-1})$$

(4.4)

indicates that the predicted observation mean $\mathbf{y}_{ij}$ is the expected value of the current observation $y_i$ given the states $x_1,...,x_{i-1}$.

As the prediction and observation steps of the KF are modelled by Gaussian random processes each will have a covariance associated with it. The covariance associated with the prediction step is $P_{x_{ij}}$, and the covariance associated with the observation step is $P_{y_{ij}}$. The KF also assumes that any noise encountered in the system is Gaussian with a zero-mean.

4.2 The linear Kalman filter (LKF)

We first discuss the linear Kalman filter (LKF) as it is the simplest of all implementations and it provides us with an understanding of the operation of the extended filters. We will use the same convention for formulation as in [40] and [3]. For the derivation of the LKF see [40].

From (4.1) we define our linear state transition equation as

$$x_i = F_{i-1}x_{i-1} + \omega_{i-1}$$

(4.5)

and from (4.2) the observation transform is defined as

$$y_i = H_i x_i + v_i$$

(4.6)
where \( i \) is the current time step. \( F_{i-1} \) and \( H_{i-1} \) are matrices that govern the dynamics and observation process of the system. The state transition and observation noise terms are \( \omega_{i-1} \) and \( \nu_i \) respectively. In the general form of this equation there is a control term \( B_{i-1} u_{i-1} \) present but this will be ignored. The noise vectors \( \omega_{i-1} \) and \( \nu_i \) are both additive with Gaussian distributions, uncorrelated and have an expected value of zero. For a given problem it is assumed that we know of \( \omega_{i-1} \) and \( \nu_i \). We can define \( \omega_{i-1} \) and \( \nu_i \) as

\[
\omega_{i-1} \sim \mathcal{N}(0, Q) \quad (4.7)
\]
\[
\nu_i \sim \mathcal{N}(0, R) \quad (4.8)
\]

to show that \( \omega_{i-1} \) and \( \nu_i \) are taken from a zero mean Gaussian PDF with respective covariance matrices \( Q \) and \( R \).

The choice of these covariances is very important as they determine how the KF will react to the difference between the predicted and observed state. This difference is known as the innovation.

The Kalman gain, which is the term used by the KF to correct its prediction, can be seen as a weighting that determines how the error in the prediction is used to correct the predicted state. This weighting is controlled by the chosen covariances.

If we choose the process noise, \( \omega_{i-1} \), large compared to the observation noise, \( \nu_i \), the KF will trust its observations more than its predictions and the innovation in the filter will have a large effect on the prediction correction. If we choose the observation noise large compared to the process noise the KF will trust its predictions more than the observations and the innovation will have less of an effect during the prediction correction.

Balancing the process and observation noise is of utmost importance to ensure the filter combines observations and predictions in the estimate to optimally converge. This process of choosing the correct noise parameters is known as tuning the filter.

Our state transition and observation transition equations are (4.5) and (4.6). The LKF equations proceed as follows:

state prediction

\[
\tilde{x}_{i|i-1} = F_i \tilde{x}_{i-1|i-1}, \quad (4.9)
\]
\[
P_{x_{i|i-1}} = F_i P_{x_{i-1|i-1}} F_i^T + Q_{i-1}, \quad (4.10)
\]

observation prediction

\[
\tilde{y}_{i|i-1} = H_i \tilde{x}_{i|i-1}, \quad (4.11)
\]
\begin{equation}
P_{y|i-1} = H_i P_{x|i-1} H_i^T + R_{i-1},
\end{equation}

Kalman gain calculation

\begin{equation}
K_i = P_{x|i-1} H_i^T P_{y|i-1}^{-1}.
\end{equation}

We combine our observation and prediction to calculate the state correction

\begin{equation}
\mathbf{x}_i = \mathbf{x}_{i|i-1} + K_i (\mathbf{y}_i - \mathbf{y}_{i|i-1}),
\end{equation}

\begin{equation}
P_{x|i} = P_{x|i-1} - K_i H_i K_i^T.
\end{equation}

For the first iteration of the LKF loop we initialise the covariance matrices, \( P_{x0} \) and \( P_{y0} \), with the prior knowledge we assume to have regarding the process and observation models and noise characteristics. An iteration of the LKF loop is described next and illustrated in Figure 4.1.

1. Predict state \( \mathbf{x}_{i|i-1} \) and state covariance \( P_{x|i-1} \) with the best estimate of the previous state, \( \mathbf{x}_{i-1|i-1} \), and state covariance, \( P_{x|i-1|i-1} \), using (4.9) and (4.10).

2. Calculate predicted observation, \( \mathbf{y}_{i|i-1} \), as well as the covariance of the predicted observations, \( P_{y|i-1} \), using (4.11) and (4.12) given the state and covariance estimates \( \mathbf{x}_{i|i-1} \) and \( P_{x|i-1} \).

3. Calculate the Kalman gain with the predicted state and observation covariance matrices, \( P_{x|i-1} \) and \( P_{y|i-1} \), using (4.13).

4. Calculate the corrected state, \( \mathbf{x}_{i|i} \), and state error covariance matrix, \( P_{x|i} \), using the Kalman gain, \( K \), and the difference between the measured and predicted observations, \( \mathbf{y}_i \) and \( \mathbf{y}_{i|i-1} \) respectively.

It is also important to note that the KF only makes provision for zero mean white Gaussian noise. This means that any noise that is not of this form that influences the system will not be correctly taken into account by the filter. An example of this type of noise is lens distortion from cameras.

### 4.3 The extended Kalman filter (EKF)

The extended Kalman filter (EKF) is an approach widely used to help the KF deal with non-linear problems. The EKF has been used widely in the field of structure from motion but more recently the unscented Kalman filter (UKF) has been preferred. The EKF uses a Taylor series expansion, generally of the first order but higher order expansions are possible, to linearise non-linear
CHAPTER 4. THE KALMAN FILTER (KF)

1. State Prediction
   Calculate prediction for the next state $X_{i|i-1}$ using (4.9)
   Calculate covariance $P_{x|i|i-1}$ for the predicted state $X_{i|i-1}$

2. Observation Prediction
   Predict observation $Y_{i|i-1}$ with predicted state $X_{i|i-1}$ using (4.11)
   Calculate covariance $P_{y|i|i-1}$ for the predicted observation using (4.12)

3. Kalman Gain Calculation
   Calculate $K_i$ using (4.13)

4. Correct Prediction
   Correct prediction $X_{i|i-1}$ using (4.14)
   Calculate covariance $P_{x|i}$ for the corrected state $X_{i|i}$

Start with Initial Conditions
$X_{i|i}=0$
$P_{x|i}=0$

Observation Input to KF
$Y_i$

Figure 4.1: Diagram showing the the Kalman filter loop
models. We use the same convention for formulation of the EKF as in [40] and
[3]. For the derivation of the EKF see [40].

For the EKF our state transition equation for the non-linear system becomes
\[ x_i = f(x_{i-1}) + \omega_{i-1} \]  \hspace{1cm} (4.16)
with the observation transform
\[ y_i = h(x_i) + \nu_i. \]  \hspace{1cm} (4.17)

The Taylor series expansion of the function \( f \) in (4.16) excluding higher order
terms \((\text{HOTs})\) yields
\[ x_i = f(\mathbf{x}_{i-1}|_{i-1}) + f'(\mathbf{x}_{i-1}|_{i-1})(\mathbf{x}_{i-1} - \mathbf{x}_{i-1}|_{i-1}) + \text{HOTs} + \omega_{i-1} \]  \hspace{1cm} (4.18)
where \( f' \) is the Jacobian derivative of the vector function \( f \) defined as
\[ f'(\mathbf{x}_{i-1}|_{i-1}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_{i-1}|_{i-1}}. \]  \hspace{1cm} (4.19)

The higher order terms in (4.18) can be expanded to form a higher order EKF
[40] but this is seldom done. A similar procedure holds for the expansion of
the function \( h \) in (4.17).

Using the state and observation transition equations (4.16) and (4.17) the
equations for the EKF proceed as follows:

state prediction
\[ \mathbf{x}_{i|i-1} = f(\mathbf{x}_{i-1|i-1}), \]  \hspace{1cm} (4.20)
\[ P_{\mathbf{x}_{i|i-1}} = F_{i|i-1}P_{\mathbf{x}_{i-1|i-1}}F_{i|i-1}^T + Q_{i-1}, \]  \hspace{1cm} (4.21)
\[ F_{i|i-1} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_{i-1|i-1},} \]  \hspace{1cm} (4.22)

observation prediction
\[ \mathbf{y}_{i|i-1} = h(\mathbf{x}_{i|i-1}), \]  \hspace{1cm} (4.23)
\[ P_{\mathbf{y}_{i|i-1}} = H_{i|i-1}P_{\mathbf{x}_{i|i-1}}H_{i|i-1}^T + R_{i-1}, \]  \hspace{1cm} (4.24)
\[ H_{i|i-1} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_{i|i-1},} \]  \hspace{1cm} (4.25)
Kalman gain calculation
\[ K_i = P_{\mathbf{x}_{i|i-1}}H_{i|i-1}^T P_{\mathbf{y}_{i|i-1}}^{-1}, \]  \hspace{1cm} (4.26)
We combine our observation and prediction to calculate the state correction
\[
\tilde{x}_{i|i} = \tilde{x}_{i|i-1} + K (\tilde{y}_i - \tilde{y}_{i|i-1}),
\]
\[ (4.27) \]
\[
P_{x_{i|i}} = (I - K_i H_{i|i-1}) P_{x_{i|i-1}}.
\]
\[ (4.28) \]

An iteration of the EKF follows the same order as that of the LKF mentioned in Section 4.2.

The EKF suffers from two well known drawbacks. These drawbacks (as stated in [26]) are:

1. Linearisation can produce highly unstable filters if the assumptions of local linearity are violated.
2. The derivation of the Jacobian matrices is non-trivial in most applications and often leads to significant implementation issues.

The UKF, discussed in the next section and used in this thesis, addresses these drawbacks for weakly non-linear gaussian functions.

### 4.4 The unscented Kalman filter (UKF)

The unscented Kalman filter (UKF) was first proposed by [25] in 1996 and was designed to replace the EKF. It has the ability to model up to the fourth Gaussian moment where the EKF can only model the first two [3]. To quote [41], “The UKF consistently outperforms the EKF in terms of prediction and estimation error, at an equal computational complexity of $O(n^3)$ for general state-space problems” and it is shown in [25] to be superior to the EKF in the following respects:

- It has a smaller expected error for all absolutely continuous functions.
- It can be applied to non-differentiable functions where the EKF fails.
- It avoids the derivation of Jacobian matrices.
- It is a more direct generalisation of the LKF than the EKF.
- It provides results at least as good as a well tuned EKF and is more robust than the EKF when stressed.

These points make the UKF a very attractive option to explore for weakly non-linear system estimation.
The basic idea behind the UKF is that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary non-linear function [26], in Figure 4.2 we show that the UKF predicts state $t_i$ more accurately than the EKF. The UKF accomplishes this by employing the unscented transform (UT). The UT provides a means for calculating the statistics of a random variable under the influence of a non-linear transformation.

**Figure 4.2:** The UKF predicts the state more accurately, the EKF needs additional correction

We start this section by discussing the UT and will then move on to the UKF.

### 4.4.1 The unscented transform (UT)

The unscented transform (UT) represents the probability density function (PDF) of a random variable with a set of points, known as sigma points, each having a weighting associated with it. The sigma points are chosen so that their mean and covariance match those of the PDF. Each sigma point undergoes the non-linear transformation after which the mean and covariance of the transformed set can be calculated. In general $2n + 1$ points are needed.
to describe a random variable of dimension $n$.

For a given random variable $\mathbf{x}$ of dimension $n$ with mean $\overline{\mathbf{x}}$ and covariance $P_{\mathbf{x}}$, that is to be propagated through an arbitrary non-linear function $H$, we calculate the sigma points, $\chi_i$, as follows:

\begin{align}
\chi_0 &= \overline{\mathbf{x}}, \\
\chi_i &= \overline{\mathbf{x}} + \Theta_i, \ i = 1, ..., n. \\
\chi_{i+n} &= \overline{\mathbf{x}} - \Theta_i, \ i = n + 1, ..., 2n,
\end{align}

where $\Theta_i$ is the $i$th column of the matrix $\Theta$ given by

\begin{equation}
P_{\mathbf{x}} = \frac{1}{n} \Theta^T \Theta.
\end{equation}

In this thesis we make use of the Cholesky decomposition [42, Lecture 23] for the calculation of the matrix factorisation. Cholesky decomposition factorises a positive-definite matrix into a triangular matrix times its transpose,

\begin{equation}
Q = L L^T
\end{equation}

Work has been done investigating the effects of different techniques that use both more and less sigma points to describe a PDF. In [43] additional sigma points are used to improve the estimated mean and covariance of a transformed distribution at a cost of computation complexity, while in [28] a reduced sigma point method is proposed using only $n + 1$ asymmetrically distributed sigma points. This reduced number of sigma points decreases the computational complexity of the algorithm, but suffers from the introduction of a small bias and covariance error.

In 1999 [27] introduced the scaled unscented transform (SUT). The SUT provides us with additional scaling parameters that allow us to fine tune the unscented transform to deal with various probability distributions and systems that are highly non-linear. These scaled sigma points, $\chi'_j$, are of the form

\begin{equation}
\chi'_j = \chi_0 + \alpha (\chi_j - \chi_0),
\end{equation}

where $\alpha$ is a scaling parameter constrained such that $0 < \alpha < 1$. This scaling parameter determines the spread of the sigma points around the mean. It is only used if there are strong non-linearities in the system and is set to 1 in our case.
The weights associated with these scaled sigma points are calculated by

\[ W_0^{(m)} = \frac{\lambda}{n + \lambda}, \quad (4.35) \]

\[ W_0^{(c)} = \frac{\lambda}{n + \lambda} + \left(1 - \alpha^2 + \beta\right), \quad (4.36) \]

\[ W_j^{(m)} = W_j^{(c)} = \frac{1}{2(n + \lambda)}, \quad j = 1, \ldots, 2n. \quad (4.37) \]

\[ \lambda = \alpha^2 (n + k) - n, \quad (4.38) \]

where \( \beta \) and \( k \) are the remaining scaling parameters introduced by the SUT. The parameter \( \beta \) is a positive scaling factor with \( \beta \geq 0 \). This parameter influences the higher order moments by giving additional weight to \( W_0^{(c)} \). \( \beta \) is set to 2, which is recommended by \cite{27} for a Gaussian PDF. The additional scaling parameter \( k \) is used to guarantee semi-definiteness of the covariance matrices. It is chosen such that \( k \geq 0 \). We set \( k = 0 \).

The weights \( w^{(m)} \) are used for the calculation of the mean of the transformed variable while the weights \( w^{(c)} \) are employed in the calculation of the covariance associated with the transformed variable.

The procedure for applying the SUT is

1. Calculate the sigma points using (4.29) to (4.32).
2. Select values for the scaling parameters \( \alpha \) and \( \beta \).
3. Apply scaling procedure in (4.34).
4. Apply the non-linear transformation over \( \chi' \)

\[ \gamma' = H \left[ \chi' \right], \quad (4.39) \]

5. Calculate the weights associated with \( \chi' \) using (4.35) to (4.38).
6. The mean of the transformed variable is given by

\[ \overline{\gamma} = \sum_{j=0}^{2n} W_j^{(m)} \gamma'_j, \quad (4.40) \]

7. The covariance of the transformed variable is given by

\[ P_y = \sum_{j=0}^{2n} W_j^{(c)} \left[ \gamma' - \overline{\gamma} \right] \left[ \gamma' - \overline{\gamma} \right]^T. \quad (4.41) \]
4.4.2 The unscented Kalman filter algorithm

We use the same convention for formulation of the unscented Kalman filter (UKF) as in [3]. For the derivation of the UKF see [26].

The state and observation transition equations are the same as those for the EKF

\[ x_i = f(x_{i-1}) + \omega_{i-1}. \]

\[ y_i = h(x_i) + v_i. \]

We use augmented data structures for the state, covariance and sigma state of the UKF for ease of implementation and computation. These augmented data structures are

\[ \pi = \begin{bmatrix} \bar{x} \\ 0 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} P & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \chi \\ \Omega \\ \Psi \end{bmatrix}. \]

(4.44)

Where \( \bar{x} \) is the state mean, \( P, Q \) and \( R \) are the covariance matrices as in the EKF and \( \chi \) is the transformed sigma state with \( \Omega \) and \( \Psi \) the sigma point state and sigma point observation covariances. The dimension of the augmented state vector \( \pi \) is

\[ n = n_x + n_\Omega + n_\Psi \]

(4.45)

where \( n_x, n_\Omega \) and \( n_\Psi \) are the dimensions of the vectors \( \bar{x}, \Omega \) and \( \Psi \). The augmented state covariance \( A \) has dimensions \( n \times n \) with the dimensions of \( P, Q \) and \( R \) being respectively \( n_x \times n_x, n_\Omega \times n_\Omega \) and \( n_\Psi \times n_\Psi \).

We use

\[ \bar{x}_0 = E[x_0], \]

\[ P_0 = E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T], \]

\[ \bar{a}_0 = E[a_0] = [x_0^T, 0^T, 0^T]^T, \]

\[ A_0 = E[(a_0 - \bar{a}_0)(a_0 - \bar{a}_0)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}, \]

(4.49)

to calculate the initialisation values for the UKF.

We then apply the SUT to calculate the sigma state using,

\[ \Lambda_{i-1|i-1} = \left[ \bar{x}_{i-1|i-1}, \bar{x}_{i-1|i-1} \pm \sqrt{(n + \lambda A_{i-1|i-1})} \right]. \]

(4.50)
where $\lambda$ is defined in (4.38) and $n$ in (4.45). The weights associated with the sigma points are calculated using (4.35) to (4.38). This sigma state is then propagated through the non-linear models in the UKF. The prediction and observation steps of the UKF are described below.

Note should be taken of the change in notation due to the inclusion of the sigma points in the UKF algorithm. The terms $\chi_{j,i}$ and $\gamma_{j,i}$ are the sigma points for the state and observation processes respectively, subscript $j$ refers to the particular sigma point being transformed and subscript $i$ the time step.

The state prediction step of the UKF is

$$
\chi_{j,i|i-1} = F(\chi_{j,i-1}, \Omega_{i-1}),
$$

(4.51)

$$
P_{x|i-1} = \sum_{j=0}^{2n} w_j^{(c)} \chi_{j,i|i-1},
$$

(4.52)

$$
P_{x|i-1} = \sum_{j=0}^{2n} w_j^{(c)} \left[ \chi_{j,i|i-1} - \bar{x}_{i-1} \right] \left[ \chi_{j,i|i-1} - \bar{x}_{i-1} \right]^T.
$$

(4.53)

The observation prediction

$$
\gamma_{j,i|i-1} = H(\chi_{j,i|i-1}, \Psi_{i-1}),
$$

(4.54)

$$
P_{y|i-1} = \sum_{j=0}^{2n} w_j^{(c)} \gamma_{j,i|i-1},
$$

(4.55)

$$
P_{y|i-1} = \sum_{j=0}^{2n} w_j^{(c)} \left[ \gamma_{j,i|i-1} - \bar{y}_{i-1} \right] \left[ \gamma_{j,i|i-1} - \bar{y}_{i-1} \right]^T.
$$

(4.56)

We calculate the Kalman gain using

$$
P_{(xy)|i-1} = \sum_{j=0}^{2n} w_j^{(c)} \left[ \chi_{j,i|i-1} - \bar{x}_{i-1} \right] \left[ \gamma_{j,i|i-1} - \bar{y}_{i-1} \right]^T,
$$

(4.57)

$$
K_{i|i-1} = P_{(xy)|i-1} P^{-1}_{y|i-1}.
$$

(4.58)

We combine our observation and prediction to calculate the state correction

$$
\bar{x}_{i|i} = \bar{x}_{i|i-1} + K_{i|i-1} (\bar{y}_i - \bar{y}_{i|i-1}),
$$

(4.59)

$$
P_{x|i} = P_{x|i-1} - K_{i|i-1} P_{y|i-1} K_{i|i-1}^T.
$$

(4.60)
4.5 Summary

In this section we introduced the Kalman filter with a brief discussion of the LKF. We then explored two methods, the EKF and UKF, to allow the KF to handle non-linear problems. We decided to use the UKF in this thesis as it is superior to the EKF in every respect providing a higher statistical accuracy at the same computational complexity. This is due to the sigma point method in which the transformed random variable is propagated through the non-linear model itself in place of attempting to linearise the non-linear model as in the EKF.
Chapter 5

Structure from motion

In the fields of computer vision and photogrammetry structure-from-motion (SFM) is the estimation of an object’s 3-dimensional structure and motion from a sequence of video images (2-dimensional motion) [3; 31; 29].

As stated in the previous chapter we use the unscented Kalman filter (UKF) in this thesis. The UKF minimises the error in its predicted observation calculated from its estimation of the 3-dimensional structure and motion of the rigid body being tracked.

For the UKF to estimate the structure and motion of the tracked object we need to design models for the state and observations. This is very important as the performance of SFM systems depend greatly on these models. We need a model for the state transition process that predicts the new state in terms of the old state, as well as an observation model that transforms the predicted state into a predicted observation. These models will need to be robust and should not depend on prior knowledge of the structure and motion of the tracked object.

Initial work using the KF for SFM was done by [22] where Azarbayejani and Pentland used an extended Kalman filter (EKF) for the estimation of structure and motion of a rigid object. Further research was done in this field by [25; 26] in which Julier and Uhlmann proposed the use of the unscented transform (UT) and UKF. We base our work on the concepts used in [3; 31; 27; 26; 22].

We start the chapter by discussing our observation model and move on to our model that predicts the structure and motion of the tracked object.
5.1 Observation modelling

The observation process we need to model is that of a camera taking an image (or sequence of images) of a 3-dimensional scene. We model the camera as a pinhole camera, this is illustrated in Figure 5.1. In this process there is a projection made of an object in the 3-dimensional scene to the image plane of the camera in 2 dimensions.

Consider a 3-dimensional coordinate in the camera coordinate system (CCS)

\[
P^{ccs} = \begin{bmatrix} p_{ccs}^x \\ p_{ccs}^y \\ p_{ccs}^z \end{bmatrix},
\]

(5.1)

that undergoes a perspective projection onto the image plane at \( Y \). The mathematical relationship between the 3-dimensional point \( P^{ccs} \) and the 2-dimensional projection \( Y \) is given by

\[
Y = \begin{bmatrix} p_{ccs}^x \\ p_{ccs}^y \end{bmatrix} \left( \frac{1}{1 + \frac{p_{ccs}^z}{f}} \right),
\]

(5.2)

where \( f \) is the focal length of the camera. The centre of projection (COP) of the camera, also known as the focus, lies at \(-f\).

Figure 5.1: Linear perspective projection of a point P onto the image plane Y at focal length \(-f\)
Note that objects can only be recovered up to a scale factor. This is illustrated in Figure 5.2 where two objects, $P_1^{ccs}$ and $P_2^{ccs}$ yield the same projection onto the image plane at $Y$. A useful way to visualise this is to think of a large object placed far from the camera and a small object placed near to the camera that are the same size in an image.

![Figure 5.2: The points $P_1^{ccs}$ and $P_2^{ccs}$ yield the same observation](image)

This scale ambiguity is due to the factor $\frac{P_z^{ccs}}{f}$ in our model (5.2). During a sequence, $f$ is assumed constant; changing the focal length $f$ in our model affects the projected coordinate independently of $P_z^{ccs}$ [22]. It is impossible to tell if $P_z^{ccs}$ or $f$ has changed (did we move the object relative to the camera or zoom the camera) as we can only discern relative changes in this ratio since both $P_z^{ccs}$ and $f$ are unknown at the start of a sequence.

We now invert this process (5.2) so we are able to retrieve $P^{ccs}$ in terms of its projection $Y$ and focal length $f$. This relationship is given by

$$P^{ccs} = \begin{bmatrix} P_x^{ccs} \\ P_y^{ccs} \\ P_z^{ccs} \end{bmatrix} = \begin{bmatrix} Y_x(1 + \frac{P_z^{ccs}}{P_z^{ccs}}) \\ Y_y(1 + \frac{P_z^{ccs}}{P_z^{ccs}}) \end{bmatrix}. \quad (5.3)$$
From the above equation we see that we only require the ratio $\frac{P_{ccs}}{z}$ to reconstruct our object up to a scale factor. We assume that we have knowledge of our focal length and we have the coordinates on the image plane thus we are left with only one unknown parameter per coordinate we wish to reconstruct.

Up to this point we have discussed reconstruction in terms of the CCS but the Kalman filter will estimate the structure with respect to the object coordinate system (OCS). The OCS is the coordinate system attached to the object, this is illustrated in Figure 5.3. The CCS and OCS are related via an orthogonal rotation matrix $R$ and translation vector $t$. This relationship is given by

$$P_{ccs} = RP^{ocs} + t,$$  \hspace{1cm} (5.4)

or inversely

$$P^{ocs} = R^{-1}(P_{ccs} - t)$$  \hspace{1cm} (5.5)

where

$$R^T = R^{-1}. \hspace{1cm} (5.6)$$

By substituting (5.3) into the above we can write (5.5) as

$$P^{ocs} = R^T \begin{bmatrix} Y_x(1 + \frac{P_{ccs}}{f}) - t_x \\ Y_y(1 + \frac{P_{ccs}}{f}) - t_y \\ P^{ocs}_z - t_z \end{bmatrix}. \hspace{1cm} (5.7)$$

---

**Figure 5.3:** Linear perspective projection of a point $P$ onto the image plane $Y$ at focal length $-f$ in terms of the OCS
CHAPTER 5. STRUCTURE FROM MOTION

5.1.1 Initialisation and setup of the observation model

As discussed in Chapter 3 we use quaternions to represent rotation in this thesis. In our discussion of the derivative of a quaternion we have a solution to a differential equation (3.32) that requires us to choose an initial (reference) condition for the orientation of the object. We follow [3] and choose to align the OCS and CCS at initialisation. This is done by choosing initial rotation

$$R_0 = I.$$  \hspace{1cm} (5.8)

We also need to choose initialisation values for the translation components of our observation model. To do this we define a feature that is tracked by the image coordinates of the first frame it is detected in. We then choose our translation components $t_x$ and $t_y$ as the centroid of all the detected features in the image. This implies we choose an OCS with the object centroid located at $0$. The assumption made here is inspired by the central limit theorem which states that given an adequate number of randomly distributed features the mean of the observations tend to coincide with the centroid of the object. We set $t_z = 0$ as we have no information about this.

Our final model relates 2-dimensional image coordinates with our object’s structure. As we assume our object to be rigid this is also the current estimate of our tracked object’s unchanging structure. We obtain our final model by incorporating the initial conditions into (5.7) and writing it in a time-dependent form. The equation for a single tracked feature then becomes

$$P_{ocs_{i|i-1}} = \begin{bmatrix} Y_x(1 + \frac{p_{ocs_{z,i-1}}}{P_{ocs_{z,i-1}}}) - Y_{x0} \\ Y_y(1 + \frac{p_{ocs_{z,i-1}}}{P_{ocs_{z,i-1}}}) - Y_{y0} \end{bmatrix}.$$  \hspace{1cm} (5.9)

We simplify this notation by substituting $P_{ocs_{z,i-1}}$ with $s$ for convenience,

$$P_{ocs_{i|i-1}} = \begin{bmatrix} Y_x(1 + \frac{s_{i|i-1}}{s_{i|i-1}}) - Y_{x0} \\ Y_y(1 + \frac{s_{i|i-1}}{s_{i|i-1}}) - Y_{y0} \end{bmatrix}.$$  \hspace{1cm} (5.10)

From the above equation we see that our model depends only on the estimate of $P_{ocs_{z}}$ and the initial observations $Y_0$ and its mean $\overline{Y}$, any errors introduced in these initial conditions will be propagated through to future estimates.

We assume the observations are corrupted by an additive, zero-mean white Gaussian noise

$$v \sim \eta (0, R)$$  \hspace{1cm} (5.11)

with 0 indicating a zero mean and $R$ the observation covariance matrix.
5.2 Structure and motion models

The structure and motion models form the state estimation process of the Kalman filter. We assume noise for the state estimation process to be zero-mean, white Gaussian described by

\[ \mu \sim \eta(0, Q) \]  

where 0 denotes a zero mean and \( Q \) the state covariance matrix.

5.2.1 Structure model

In this thesis we attempt to accurately calculate the structure of a tracked object. As stated in the previous section there is one unknown parameter per tracked feature we need to estimate to recover the structure. We place the unknown parameter for each tracked feature in a structure vector

\[ s = [s_0, s_1, \ldots, s_m], \]  

where \( m \) is the number of features we are tracking and \( s_k = P_{z,k}^{ocs} \).

The structure state transition equation that governs the structure estimate is

\[ s_i^{|i-1} = s_{i-1}^{|i-1} + \mu_{structure} \]  

which states that the current structure estimate is similar to the previous structure estimate. This is a valid assumption if we are tracking a rigid object.

In using this structure state transition equation we are relying on the Kalman gain in the Kalman filter to adjust the structure and absorb any modelling errors.

5.2.2 Motion model

Quaternions (as discussed in Chapter 3) are used in this thesis to represent the rotation of the tracked object relative to the object’s coordinate system (OCS). Using (3.28) and (3.30) we can write the rotation transition equation as

\[ q_i^{|i-1} = q_{i-1}^{|i-1} + \Delta t \frac{1}{2} \Omega(\omega_i^{|i-1}) q_{i-1}^{|i-1} + \mu_{rotation}. \]  

In the above equation we have the term \( \Omega(\omega_i^{|i-1}) \) that is a function of the angular velocity of the tracked object. We therefore need to incorporate the angular velocity of the tracked object relative to the OCS into our model.
For this we use the angular velocity transition equation

\[ \omega_{i|i-1} = \omega_{i-1|i-1} + \mu_{\text{rotation-velocity}} \]  

(5.16)

that states the angular velocity of the tracked object between frames is constant.

The rotation transition equations, (5.15) and (5.16), act on the rotation

\[ q = [q_0, q_1, q_2, q_3] \]  

(5.17)

and rotational velocity vectors

\[ \omega = [\omega_x, \omega_y, \omega_z] . \]  

(5.18)

The formulation for the translation transition equations follow a similar procedure. The translation relates the origin of the OCS with respect to the origin of the camera coordinate system (CCS).

We model the translation transition with

\[ t_{i|i-1} = t_{i-1|i-1} + \Delta t d_{i|i-1} + \mu_{\text{translation}} \]  

(5.19)

where \( d \) is the model of the translation velocity given by

\[ d_{i|i-1} = d_{i-1|i-1} + \mu_{\text{translation-velocity}} . \]  

(5.20)

The translational velocity \( d \) is related to the translation \( t \) via

\[ d = \frac{dt}{dt} . \]  

(5.21)

that states the velocity is the time derivative of the translation.

The translation transition equation (5.19) states that the translation in the current step is equal to that of the previous step plus the change in translation and (5.20) states that the translational velocity is constant between steps.

The translation transition equations act on the translation vector

\[ t = [t_x, t_y, t_z] \]  

(5.22)

and translational velocity vector

\[ d = [d_x, d_y, d_z] . \]  

(5.23)
By assuming we have a constant velocity in our system (and therefore no acceleration) any acceleration of the object will be handled as noise effects. This is only valid if we have very small changes in acceleration which implies that the change of the object’s position between frames will be constant. We are able to detect when there is no movement in the video sequence by monitoring the change in detected feature coordinates. When this occurs the tracked object is either stopping or the direction of movement is changing.

5.2.3 Combining the structure and motion models

Our final step in the modelling process is to combine the statevectors for structure and motion into the full state vector,

\[ x = [s, q, \omega, t, d]. \] (5.24)

Our state vector has dimensions \( m + 13 \) where \( m \) is the number of features we are tracking.

5.3 Initialisation and setup of structure and motion models

In this section we discuss how to initialise our structure and motion models. We provide a scheme by which to initialise the structure and motion models if no information of the scene is available. We then propose two methods to initialise the structure and motions models from pre-computed knowledge of the scene. The first method utilises the structure and motion estimate from a previous run of the UKF on a particular video sequence. The second method makes use of structure and motion estimates obtained from a matrix factorisation algorithm implemented by de Vaal [18] that is run on a small subset of frames from a particular video sequence.

5.3.1 Initialisation with no knowledge of object’s structure or motion

If we have no knowledge of the 3-dimensional scene we will use the following default initialisation values.

We set the entries of the structure vector in (5.13) to zero,

\[ s = [0, 0, 0, ..., 0]. \] (5.25)
This corresponds to the tracked object being initialised as a plane parallel to the image plane.

The estimated rotation of the tracked object is a relative measure of the alignment of the OCS and CCS. In Section 5.1.1 we chose to align the OCS and CCS at initialisation. This is equivalent to initialising the rotation vector with

\[ q_0 = [1, 0, 0, 0] . \] (5.26)

This corresponds to the identity matrix. We also set the initial rotational velocity vector to zero,

\[ \omega_0 = [0, 0, 0] . \] (5.27)

We have some information for the initialisation of the translational components. We choose to set the initial translational components \( t_x \) and \( t_y \) to the mean of the initial set of 2-dimensional observations in Section 5.1.1. This is valid as the translation of the tracked object is defined by the translation of its centroid. We set \( t_z = 0 \). Our initial translation vector is then

\[ t = [Y_x, Y_y, 0] . \] (5.28)

We set the translational velocity components to zero as we did with the rotational velocity components,

\[ d = [0, 0, 0] . \] (5.29)

The initial structure we use for the tracked object coincides with a plane at the image plane of the camera with the centroid located at the mean of the 2-dimensional observations. This is also the origin of the OCS which is the point around which rotation of the object occurs.

Note that if the initial structure estimate does not coincide with the real centroid of the object, possibly due to an uneven distribution of detected features, problems will be encountered in the estimation of the objects rotation [3].

### 5.3.2 Using pre-computed estimates of structure and motion for initialisation

We attempt to improve the convergence of the Kalman filter by initialising the structure and motion models with pre-computed structure estimates. We investigate two methods to pre-compute this structure estimate.

The first method initialises the structure and motion models with the estimate from a previous run of the Kalman filter. We investigate variations of
this technique by changing the number of frames we would use to estimate
this initialisation structure as well as the number of passes we would allow the
Kalman filter (KF) to have over the frame set.

Using this method we have good alignment of the coordinate system of our
initialisation structure and the OCS of our system. We also have the estimated
motion parameters for rotation and translation to use in initialisation of our
system.

The second method uses a structure from motion system based on matrix
factorisation to calculate the initialisation estimates. This system will batch
process a set of frames from our video sequence and return an estimate of the
structure and pose of the object.

To use the structure estimate from the matrix factorisation method we need
to align the coordinate system of the initialisation structure with that of the
OCS of our system. We also need to adjust the scale of the structure we use
for initialisation to better suit our system as the scale of reconstruction is not
consistent between the two systems.

We tested this method with varying numbers of frames from our video se-
quence.

5.4 Setup of noise models

There are three covariance matrices in our system, $P$, $Q$ and $R$. We stated
previously in this chapter the $Q$ and $R$ are the covariance matrices that deal
with the structure and observations modelling respectively, $P$ is our state co-
variance matrix.

Each of these are initialised as spherical covariance matrices (entries along
the diagonal). This is valid as the noise in our system is assumed to be white
and Gaussian (Section 4.2). The matrix $P$ is initialised with the same value
in each of its diagonal entries. The matrices $Q$ and $R$ are initialised with each
entry in the diagonal being fixed for a particular sequence. We will discuss the
choice of these values in Chapter 6 where we deal with implementation issues
come across in this thesis.

We attempt to improve the performance of the Kalman filter by allowing the
filter to change the fixed covariance values in the $R$ and $Q$ covariance matri-
ces. This is done as the tracked object does not always move with a constant velocity (the core assumption of our motion model). When this occurs, our motion model fails and the performance of the filter is adversely effected.

We are able to detect when there is a stop in motion during a sequence (by monitoring the change in marker positions) and if this occurs we alter the values in the covariance matrix that deal with the structure and velocity components of the object to cope with the situation in a better manner.

### 5.5 Dimensions of our state space and covariance matrices

In this section we described the structure and motion models used in our system. These models make up the state vector of our system. Our state vector (5.24) has dimensions $m+13$ where $m$ is the number of features we are tracking. This state has noise vectors for the state and observation associated with it. These vectors, $\nu$ and $\mu$, have dimensions $m+13$ and $2m$ respectively and are combined with our state vector to form our augmented state vector (4.44). Therefore the augmented state vector has dimensions

$$n = 2(m+13) + 2m = 4m + 26. \quad (5.30)$$

The covariance matrices associated with the state of our system, $P$, $Q$ and $R$, are all square matrices with dimensions $(m+13) \times (m+13)$, $(m+13) \times (m+13)$ and $2m \times 2m$, respectively. The dimension of the augmented covariance matrix is then $n \times n$, $n$ is described by (5.30).

The number of sigma points we require to represent this with the unscented transform is (see Section 4.4.1)

$$2n + 1 = 2(4m + 26) + 1 = 8m + 43. \quad (5.31)$$

From this we see that for a large number of tracked features we require a large number of sigma points and the dimensions of the covariance matrices are also very large. From Section 4.4 we know the UKF runs at a computational complexity of $O(n^3)$ and thus will be slowed down significantly if the matrices become too large due to too many points being tracked.

### 5.6 Summary

In this chapter we discussed the models we are going to use for the structure, motion and observation process. The model equations are all of the first order
as using higher order transition equations would increase the computational complexity of our system.

The models follow the work of [3; 31] which were based on [27; 26; 22]. Implementation issues of our system are discussed in the next chapter with the results of tests performed on our system presented in Chapter 7.
Chapter 6

Implementation Issues

In this chapter we discuss the issues we encountered in the implementation of our system. These include the initialisation and setup of the UKF, marker detection, Euclidean 3-dimensional reconstruction and lens distortion. We propose two methods to improve the performance of the UKF: we provide initialisation information to speed up convergence of the UKF (discussed in Section 5.3) and allow dynamic changing of the covariance values for the motion model of our object to allow the UKF to perform better when our assumptions about the motion of the object are not valid. We end the chapter with an investigation into the real time performance of our SFM system.

6.1 Unscented Kalman filter

6.1.1 Initialisation and setup

We must be very careful when we setup the UKF as it is very sensitive to the choice of the initial state uncertainty \( P_0 \). To ensure convergence of the UKF we set the entries of \( P_0 \ll 1 \). In choosing the state transition uncertainty \( Q \) we have to balance speed of convergence and filter performance. If we set the values in \( Q \) too large the UKF will not converge and if chosen too small will cause the UKF to converge very slowly.

We also differentiate between the values in \( Q \) dealing with structure, \( Q_s \) and motion \( Q_{r,t} \). This allows us to set the covariance values affecting motion and structure independently of each other. To speed up convergence we can relax the covariance values dealing with the structural components that will allow the UKF more freedom to minimise the error between the actual and predicted observation, this however leads to less accurate structure estimation.

We propose allowing the covariance values dealing with the structure, trans-
CHAPTER 6. IMPLEMENTATION ISSUES

Translation velocity and rotation velocity of the object to be altered when the assumptions made about the motion in our system fail. In our system we assume the object being tracked is moving at a constant velocity, this however is not the case in reality. We are tracking a patient being moved on a robotic chair, this robotic positioning system generally moves with a constant velocity except when changing the direction of motion or stopping in its final position. Before the positioning system can change the direction of movement it needs to come to a complete stop, this is due to its design as the robotic chair has 5 degrees of freedom (axes along which it can move) but can only move along one of these at a time.

Each time the positioning system comes to a stop it causes a discontinuity in our motion model, this in turn will cause a larger error in our predicted observation. This error will be taken into account by the Kalman gain when the error between the predicted and actual observation is minimised. The Kalman gain controls the structural estimate and both rotational and translational velocity components of our system. The attempt by the system to minimise the error caused by our motion model failing will then distort the estimated structure.

We attempt to counteract this by adjusting the covariance values that control the structure and velocity components of our object when we detect such a stop in motion in the video sequence. This stop in motion is detected by investigating the difference in the positions of detected markers in two consecutive frames. If the difference in marker positions is not large enough we assume the marker carrier is stopping or changing its direction of motion. This is a valid assumption as during normal operation of the positioning system the chair maintains a smooth motion, only accelerating or decelerating when stopping or changing direction.

Care must be taken in choosing the values for the covariance matrix $Q$. If the ratio of entries of $Q_s$ to $Q_{r,t}$ is too large it will cause the Cholesky decomposition to fail due to ill conditioning and lead to numerical instability [3].

If we assume a one pixel error in detection of marker coordinates the observation noise variance in our system can be based on the respective reciprocals of the resolution of the width and height of the image frame, i.e.

$$
\sigma^2_r = \frac{1}{2}(\frac{1}{w} + \frac{1}{h}).
$$

We use $\sigma^2_r$ for the entries along the diagonal of the covariance matrix $R$. 

6.1.2 Sigma structure

During the observation prediction step (4.54) of the first iteration of the UKF we use sigma points to reconstruct the object’s structure. The observation prediction step involves reconstructing the object’s structure relative to the OCS (5.10), the structure is then rotated and translated (5.4) to align it with the CCS. The sigma structure is then projected onto the image plane from which the predicted observation mean and covariance are calculated (4.56 and 4.57).

The problem we have is the initial observation $y_0$ and observation mean $\overline{y_0}$ are not from a set of sigma points and we reconstruct our object from a sigma point set. We cannot create a set of sigma points from the initial observation as that implies we know the state sigma points that produced it. What we do is use the initial observation and observation mean as the sigma points, we rewrite (5.10) as

$$P_{ocs}^{j,i-1} = \begin{bmatrix}
y_x,0(1 + \frac{\chi_{s,j,i-1}}{j}) - \overline{y}_x,0 \\
y_y,0(1 + \frac{\chi_{s,j,i-1}}{j}) - \overline{y}_y,0 \\
\chi_{s,j,i-1}
\end{bmatrix}, j = 0..2n \tag{6.2}$$

where $\chi_{s,j,i-1}$ is the $j$-th predicted sigma structure parameter.

To calculate our prediction from the sigma structure we need to rotate and translate this sigma structure to align it with the CCS. We must be careful to rotate each sigma feature with its corresponding sigma quaternion $\chi_{q,j,i-1}$. We require that the sigma quaternion has a unit norm to be a valid rotation, we therefore normalise each sigma quaternion to prevent unintended scaling of the sigma structure. This sigma structure is projected onto the image plane from which the predicted observation mean and covariance are calculated.

6.2 Lens distortion

It has been assumed up to this point in this thesis that the linear prospective projection model is an accurate description of the pinhole-camera, i.e the world point, image point and optical centre are collinear and that world lines are imaged as lines. This is not true in practice however, where the most common deviation is lens distortion. Lens distortion becomes more pronounced at lower focal lengths and when poor quality lenses are used. The effect of radial lens distortion on straight lines is illustrated in Figure 6.1.

The cure for this distortion is to correct the image measurements to those that...
would have been obtained using a perfect linear camera. According to [44] radial lens distortion is modelled as

\[ P_{\text{corr}} = P_{\text{cen}} + L(r)(P_{rd} - P_{cen}), \]  

(6.3)

where \( P_{\text{corr}} = [x_{\text{corr}}, y_{\text{corr}}] \) are the corrected image coordinates, \( P_{rd} = [x_{rd}, y_{rd}] \) the coordinates under influence of radial distortion and \( P_{cen} = [x_{cen}, y_{cen}] \) the centre of radial distortion. The function \( L(r) \) determines the distortion factor and is a function of the radial distance

\[ r = \sqrt{(x_{rd} - x_{cen})^2 + (y_{rd} - y_{cen})^2}. \]  

(6.4)

The function \( L(r) \) is only defined for positive values of \( r \) and \( L(0) = 1 \). An approximation of any arbitrary function \( L \) can be obtained by using a Taylor expansion of the form

\[ L(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots \]  

(6.5)

The coefficients \([k_1, k_2, k_3, \ldots, x_{cen}, y_{cen}]\) are considered part of the cameras internal calibration.

Distortion correction at iThemba Labs is calculated using the distortion object in Figure 6.2. The distortion object is a grid structure with the centroid of each marker a point in the grid. The distortion coefficients are calculated by minimising the error between the observed lines of the grid and straight lines. The method employed at iThemba Labs is fully discussed in [45].

### 6.3 Marker detection

Marker detection at iThemba Labs is aided with the use of retro-reflective radio-paque markers and special filters fitted to the cameras. The special filters combined with the reflective markers cause the markers to show up very brightly on the captured image, see Figure 6.3, that allows for a simple thresholding technique to detect the markers in the image.
The thresholding allows the detection of the clusters of pixels that represent a marker. The clusters of pixels will be ellipsoidal in shape as the projection of a circle is an ellipse. We use the centroid of an ellipse fitted to the pixel cluster as the centroid of the marker. We are able to do this as the deviation from the true centroid is small at the angles and scales of interest. This method is fully described in [46].

6.4 Euclidean 3-dimensional reconstruction via ground truth

To move to a Euclidean reconstruction from the projective reconstruction we need to calculate the transformation \( W \) that relates the projective points \( X_p \) to the real world points \( X_w \). This transformation is of the form

\[
\begin{bmatrix}
    x_w \\
y_w \\
z_w \\
1
\end{bmatrix} = W \begin{bmatrix}
    x_p \\
y_p \\
z_p \\
1
\end{bmatrix},
\]

(6.6)

The 4 \( \times \) 4 transformation matrix \( W \) relates the projective structure to the real world structure via a scaling, rotation and translation and is a restricted case of a general homography known as an affine transformation. The matrix \( W \) is
We require a minimum of four point correspondences to solve for the transformation $W$. The corresponding points are arranged in measurement matrices $X_w$ and $X_p$ and need to be chosen so that no set of four are coplanar. This is done to ensure the existence of the right pseudo-inverse of $X_p$. To solve for $W$ we multiply (6.6) through from the right with the right pseudo-inverse of $X_p$ yielding

$$X_wX_p^+ = WX_pX_p^+ = WI,$$

which we can arrange

$$W = X_wX_p^+.$$(6.9)

Equation (6.9) provides us the solution to the transformation $W$ that we use to move from our projective to the metric reconstruction of the tracked object. The ground truth data was obtained from scanning the marker carrier with the CT scanner at iThemba Labs and is accurate to within 0.5mm.
6.5 Real time performance

Our system runs on Ubuntu Linux powered by an Intel Pentium M (Dothan core) 1.6GHz processor accompanied with 1GB DDR2-533 ram. We summarise the real time performance statistics of our system in Table 6.1. The difference in time per frame between the synthetic and real performance is due to marker detection and an increase in state size (due to more markers being tracked).

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Synthetic sequences</th>
<th>Real sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Markers tracked</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>Time to initialisation</td>
<td>0.22s</td>
<td>4.46s</td>
</tr>
<tr>
<td>Time per frame</td>
<td>0.18s</td>
<td>0.64s</td>
</tr>
<tr>
<td>Image load time</td>
<td>n/a</td>
<td>0.003s</td>
</tr>
<tr>
<td>Marker detection time</td>
<td>n/a</td>
<td>0.19s</td>
</tr>
<tr>
<td>UKF state prediction</td>
<td>0.05s</td>
<td>0.06s</td>
</tr>
<tr>
<td>UKF observation prediction</td>
<td>0.12s</td>
<td>0.31s</td>
</tr>
<tr>
<td>UKF correction</td>
<td>0.01s</td>
<td>0.04s</td>
</tr>
</tbody>
</table>

Table 6.1: Real time performance of our system

We use an effective frame rate of 6 frames per second (using every fourth frame from the PAL video sequence captured at 25 frames per second). Our system processes about two frames per second tracking 19 markers. The number of markers tracked during a sequence greatly influence the processing time per frame.

We implemented this system in Python, an interpreted language and therefore has an extra layer of abstraction causing larger processing times if compared to a compiled language such as C. The real time performance would improve if we were to port our implementation to such a language.

6.6 Summary

In this section we discussed the most significant implementation issues dealt with in this thesis. We looked at initialisation of the noise parameters of the UKF and proposed a method to aid the performance of the UKF when assumptions in the motion models fail. We then discussed lens distortion and the marker detection scheme used in this thesis and at iThemba Labs. We ended with a summary of the real-time performance of our system.
Chapter 7

Tests and results

In this chapter we present results obtained from tests run on our system using both synthetic and real video sequences. Our synthetic test sequences are generated using Python and NumPy and are designed to test the correctness of our system under ideal and noisy conditions. We use noiseless synthetic test sequences containing pure rotation, pure translation and a combination of rotation and translation for the motion of our object to verify the correctness of our system. We then corrupt our synthetic test sequence with varying levels of additive Gaussian noise to investigate the performance of our system in noisy conditions.

The real sequences we test our system on are obtained from iThemba Labs. Marker detection and radial lens distortion correction is performed by iThemba’s SPG software. We only track markers visible for the entire video sequence as we require an accurate structural estimate and methods that accommodate for feature occlusion distort the estimated structure when the lost feature is replaced.

We start this chapter by defining our measurement of error for the structural estimation so we are able to compare results from different test sequences. We then present our synthetic test cases and end this chapter with the real test sequences and a summary of our results.

7.1 Error calculation

In this thesis we are interested in the accuracy of our estimated structure. We require a definition for our measure of error so we are able to compare results obtained from the various sequences we used for testing.

We define the distance between the $k$th estimated structure parameter $s_k$ and
its reference position \( s_k^{ref} \) as
\[
\Delta s_k = |a s_k - s_k^{ref}|,  \tag{7.1}
\]
where \( a \) is a scalar used to scale the estimated structure parameter. For the synthetic sequences \( a \) is the factor that scales the \( z \) coordinate of a chosen vertex to its true value of \( \pm 0.5 \). For real sequences we make use of the ground truth method to calculate the transform that moves our projective reconstruction to a Euclidean (metric) reconstruction, this is discussed in Section 7.3.1.

We use the Root-Mean-Squared-Error (RMS) to calculate the error in reconstruction given by
\[
e_s = \sqrt{\frac{1}{m-1} \sum_{k=0}^{m-1} \Delta s_k^2},  \tag{7.2}
\]
where \( m \) is the number of features. We can calculate this error at any frame in our sequence.

We are also interested in the variance of our reconstruction error as it gives us an idea of the distribution of the expected error. This is useful as it indicates the magnitude of deviation we can expect from the true structure: the smaller the variance the smaller the deviation.

The unbiased sample variance of the structure is defined as
\[
\sigma_s^2 = \frac{1}{m-1} \sum_{k=0}^{m-1} (\Delta s_k - \Delta s_k^*)^2,  \tag{7.3}
\]
with the structure mean error
\[
\Delta s_k^* = \frac{1}{m} \sum_{k=0}^{m-1} \Delta s_k.  \tag{7.4}
\]

## 7.2 Synthetic cube test sequences

For our synthetic sequence we generated the projections of a cube’s eight vertices and used these as the coordinate inputs for the UKF. The cube has breadth, width and height of 1 unit. We assume the cube to be transparent so the vertices can be tracked consistently throughout the sequence.

The 2-dimensional projection of the 3-dimensional cube is performed in accord with the observation model for a linear perspective projection camera and is noise free. The focal length of our camera model is set at 10 units and the cube is positioned such that the centroid of the initial 2-dimensional projection is \((0, 0)\) in the camera coordinate system (CCS).
7.2.1 Test sequences without noise

We provide the results of three tests performed on our system using noiseless synthetic data. These tests were performed to verify the correct working of our system in ideal conditions. Our first synthetic test sequence contains only rotation, the second test sequence contains only translation and the third sequence contains a combination of rotation and translation.

Pure Rotation

The first results we present are obtained from running our system on a synthetic test sequence containing rotation around the $Y$ axis of the object’s coordinate system (OCS). The rotation velocity of the cube is constant, performing a complete rotation every 240 frames, this rotation occurs at 0.026 radians/frame.

Figure 7.1 shows the observed sequence with the estimated coordinates overlaid. We note that the sequences are almost identical showing that the UKF correctly estimated the motion of the synthetic cube.

Figure 7.2 (a) shows the estimated structure parameters of the UKF over time. The initial structure parameters of the sequence are aligned with the axis of the OCS and CCS and we expect the structure to converge with half the vertices on the front face of the cube and half on the back face. After fifty frames the UKF converges to the true values of ±0.5.

We see a jump in the structure estimation around frame 240, this is when the cube completes its first rotation. This distortion is due to rounding errors causing acceleration in the cube’s motion. This distortion affects the UKF only slightly being visible at frame 240 when the cube completes its first rotation.

Figure 7.3 (a) and (b) show the estimated quaternion for rotation and the rotational velocity of the synthetic cube. We see the UKF correctly estimates the rotation of the synthetic cube with the quaternion having the expected sinusoidal parameters. The angular velocity, Figure 7.3 (b) is correctly estimated at 0.026 radians/frame. The estimated translation parameters, Figure 7.3 (c) and (d), show that the cube did not translate during the sequence.

Figure 7.2 (b) shows the RMS error over time. After the UKF converges the reconstruction error is small with $\epsilon_s < 0.01$ and structure variance $\sigma_s^2 < 0.0001$, indicating that we can expect our structure estimate to be very close to the true structure. In Figure 7.4 we display some images of the reconstructed cube at various frames during the sequences.
Figure 7.1: Observed and estimated coordinates - pure rotation

Figure 7.2: Estimated structure parameters and RMS error of UKF - pure rotation
CHAPTER 7. TESTS AND RESULTS

Figure 7.3: Estimated motion of synthetic cube - pure translation

Figure 7.4: Extract from the reconstruction sequence of synthetic cube - pure rotation
Pure translation

The second synthetic cube sequence is of a cube translating in the $x$ and $y$ direction. The cube moves at a constant velocity of 0.3 units/frame along the $x$-axis and 0.1 units/frame along the $y$-axis. In Figure 7.5 the estimated coordinates are overlayed on the observed coordinates over time, we notice the UKF again closely follows the actual observations. Figure 7.6 (a) shows that the UKF correctly estimates the cube’s structure converging after 50 frames to the true values of $\pm 0.5$.

Figure 7.7 contains the estimated motion parameters of the cube during the sequence. The UKF correctly estimates that there is no rotation during the sequence and shows the cube translating at the correct velocity (Figure 7.7 (c)). The reconstruction error Figure 7.6 (b) is small with $\varepsilon_s < 0.01$ and variance $\sigma_s^2 < 0.0001$ after UKF convergence.
Figure 7.6: Estimated structure parameters and RMS error of UKF - pure translation

Figure 7.7: Estimated motion of synthetic cube - pure translation
Rotation and translation

In this sequence the cube rotates about its X, Y and Z axis with velocities \( \omega_x = 0.052 \) radians/frame, \( \omega_y = 0.026 \) radians/frame and \( \omega_z = 0.017 \) radians/frame. The rotation is combined with translation along each axis with velocities \( d_x = 0.02 \) units/frame, \( d_y = 0.03 \) units/frame and \( d_z = 0.045 \) units/frame to create the synthetic cube sequence in Figure 7.8. We see that the UKF accurately tracks the complex motion of the cube in the sequence with its estimated coordinates matching the observations. The UKF correctly estimates the structure parameters of cube (Figure 7.9 (a) ) with convergence occurring after 50 frames.

Figure 7.10 shows the estimated motion parameters of the cube. We see the UKF accurately estimates each of these parameters with the RMS error (Figure 7.9 (b)) after convergence of \( e_s < 0.01 \). The structure variance during the sequence is \( \sigma_s^2 < 0.0001 \) indicating the estimated structure is very close to the true structure.

![Figure 7.8: Observed and estimated coordinates - rotation and translation](image-url)
Figure 7.9: Estimated structure parameters and RMS error of UKF

Figure 7.10: Estimated motion of synthetic cube - rotation and translation
7.2.2 Test sequences with noise

We now add varying levels of noise to the synthetic test sequence containing rotation and translation to test the performance of our system in non ideal conditions. The zero mean Gaussian noise is added to the synthetic coordinates before they are processed by our system. We present results from three runs of this synthetic sequence each containing progressively larger levels of noise.

Noise variance $= 0.0001$

The first test, shown in Figures 7.11-7.13, was performed with a zero-mean Gaussian noise with a variance of 0.0001 added to the projected 2-dimensional coordinates of the synthetic cube. This is a relatively small amount of noise and has a negligible effect on the performance of the UKF. The UKF converges after fifty frames to the true structure parameters of the synthetic cube.

Comparing the results from this sequence to those of the noise-free sequence (Figures 7.8-7.10) we see that the UKF correctly estimated the translation and rotation components of the sequence under these noisy conditions.

We can see the effect of the noise in the translation and rotation velocity components in Figure 7.13 (b) and (d). The noise is absorbed in these components as they are directly controlled by the Kalman gain.

The RMS error and structure variance are $e_s < 0.01$ and $\sigma_s^2 < 0.0001$ respectively giving similar results to those in the noise free sequence.
CHAPTER 7. TESTS AND RESULTS

Figure 7.11: Observed and estimated coordinates - noise variance 0.0001

Figure 7.12: Estimated structure parameters - noise variance 0.0001
Figure 7.13: Estimated motion of synthetic cube - noise variance 0.0001
Noise variance = 0.001

We now increase the zero mean Gaussian noise variance to 0.001 for the synthetic sequence, the results are presented in Figures 7.14-7.15. The effects of the noise can be seen in the rotation and translation velocity components (Figure 7.15 (b) and (d)) as the Kalman gain needs to compensate for the noisy observations. The UKF however manages to cope well in this test and correctly estimates the structure (Figure 7.14), translation and rotation components of the cube (Figure 7.15 (a) and (c)).

![Figure 7.14: Estimated structure parameters and RMS error of UKF - noise variance 0.001](image)

(a) Estimated structure parameters  
(b) RMS error
Figure 7.15: Estimated motion of synthetic cube - noise variance 0.001
Noise variance = 0.01

The variance of the zero mean Gaussian noise is increased to 0.01 for the last synthetic test we present (Figures 7.16-7.19). The large noise prevents the UKF from converging, this is indicated by the oscillating estimated structure parameters in Figure 7.17 (a). Structure oscillating occurs if the UKF is not tuned correctly for the noise conditions, if the noise in the system is not Gaussian or the noise is excessively large.

On inspection of the motion parameters of the UKF (Figure 7.18) we see the noise is of such a level it now badly affects the estimation of the rotation and translation parameters as it is absorbed by the respective motion velocities.

The RMS error (Figure 7.17 (b)) $e_s < 0.1$ and structure variance $\sigma_s^2 < 0.05$ are now much larger and indicate the estimated structure can not be trusted. Reconstruction of the cube is presented at various frames in the sequence in Figure 7.19, we can see the the reconstructed object is no longer a perfect cube.

![Observed and estimated coordinates - noise variance 0.01](image.png)

**Figure 7.16:** Observed and estimated coordinates - noise variance 0.01
Figure 7.17: Estimated structure parameters and RMS error of UKF - noise variance 0.01

Figure 7.18: Estimated motion of synthetic cube - noise variance 0.01
Figure 7.19: Extract from the reconstruction sequence of synthetic cube - noise variance 0.01
7.2.3 Summary of results for synthetic test sequences

The results obtained from tests performed using the synthetic test sequences are summarised in Table 7.1. We show the setup of the UKF, number of features tracked, RMS error, structure variance and the frame number at which the RMS error and structure variance were calculated.

The noise free synthetic tests verified that the UKF is working correctly in ideal conditions estimating the motion and structure parameters of the cube accurately. The UKF performs well in the first two sequences we distorted with noise, correctly predicting the structure and motion with a small RMS error and structure variance of less than one percent.

The last test sequence is distorted with a large amount of noise and we see the UKF has trouble converging and incorrectly estimates the structure and motion parameters. This is reflected in the large RMS error and structure variance.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>seq1</th>
<th>seq2</th>
<th>seq3</th>
<th>seq4</th>
<th>seq5</th>
<th>seq6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state covariance $P_0$</td>
<td>0.050</td>
<td>0.050</td>
<td>0.050</td>
<td>0.050</td>
<td>0.050</td>
<td>0.050</td>
</tr>
<tr>
<td>Structure covariance $Q_s$</td>
<td>0.010</td>
<td>0.010</td>
<td>0.010</td>
<td>0.010</td>
<td>0.010</td>
<td>0.010</td>
</tr>
<tr>
<td>Motion covariance $Q_{r,t}$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Observation covariance $R$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Noise variance</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>0.0001</td>
<td>0.001</td>
<td>0.01</td>
</tr>
<tr>
<td>Frame of evaluation</td>
<td>320</td>
<td>150</td>
<td>235</td>
<td>205</td>
<td>205</td>
<td>205</td>
</tr>
<tr>
<td>Number of features</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.005</td>
<td>0.008</td>
<td>0.008</td>
<td>0.009</td>
<td>0.017</td>
<td>0.15</td>
</tr>
<tr>
<td>Convergence (frames)</td>
<td>40</td>
<td>40</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>NA</td>
</tr>
<tr>
<td>Variance of structure</td>
<td>$3.4e^{-5}$</td>
<td>$6.5e^{-5}$</td>
<td>$1.2e^{-4}$</td>
<td>$3.3e^{-4}$</td>
<td>$3.6e^{-4}$</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 7.1: Summary of results from synthetic test cases

7.3 Real sequences

In this section we provide the results from tests performed on our system using real video sequences obtained from iThemba Labs. We use sequences captured in the proton vault with the marker carrier attached to the robotic chair and from the CT-SPG system at the CT scanner with the marker carrier translating on the CT scanner bed. We restrict our marker selection to those we can track consistently through each sequence as we do not deal with marker occlusion in this thesis.
We present results of tests on our system to demonstrate our various improvements. We run each sequence with the UKF in its default setting, not skipping motionless frames or using the proposed adaptive filtering approach. The UKF is then run in the default setting but we skip motionless frames, this is followed by running the UKF with adaptive filtering enabled.

We further investigate two approaches to improve UKF convergence by using pre-calculated initialisation data obtained from the UKF and a projective matrix factorisation (PMF) system. We test the system using initialisation data obtained from the UKF using the default and adaptive modes and initialised both an adaptive and default UKF with this data to investigate the performance. We also present tests using initialisation data from the PMF system on each sequence.

We end this section with a summary of our results.

### 7.3.1 Real sequence 1 - CT-SPG

Real sequence 1 was obtained in the CT room at iThemba Labs from the CT-SPG system. We present results obtained from testing our system on this sequence with the UKF run in the default setting, skipping motionless frames, adaptive filtering enabled, using UKF data for initialisation and using initialisation data from the PMF system.

Real sequence 1 consists of 151 frames captured at a resolution of $576 \times 768$ (PAL) with a frame rate of 6 frames per second. During this sequence the marker carrier translates through the CT scanner, the motion is kept as smooth as possible but we still experienced multiple sharp changes in acceleration due to the mechanism used to control the motion of the CT scanner table. The changes in acceleration of the marker carrier start early in the sequence before the UKF has a chance to converge and occur repeatedly throughout the sequence.

We were able to consistently track 12 markers during this sequence, the tracked markers (with radial lens distortion correction) are shown in Figure 7.20.

**Test 1: UKF - default setting**

In Figure 7.21 we display the estimated marker coordinates overlaid on the observed marker coordinates over time, we see that the UKF accurately estimates the observed coordinates. Figures 7.22-7.23 show the marker carrier’s structure and motion components as estimated by the UKF. We see both the rotation velocity (Figure 7.23 (b)) and translation velocity (Figure 7.23 (d))
are affected by a large amount of noise, this is due to the multiple sharp changes in acceleration of the marker carrier in the sequence. As we assume a constant velocity in our motion model these changes have to be taken into account by the Kalman gain, this directly affects the velocity and structure components in our system and results in the observed noise.

The estimated structure parameters of the marker carrier are shown in Figure 7.22 (a), we do not get the parallel lines that we associate with UKF convergence, this is a result of the acceleration in the marker carrier’s motion. On inspection of the RMS error (Figure 7.22 (b)) we see that the error in the structure stabilises after 80 frames at about 2mm with the variance $\sigma_s^2 < 0.09\text{mm}^2$.

We show the reconstructed markers in Figure 7.24, the markers reconstructed are those on the forehead to the bridge of the nose and either side of the nose.
Figure 7.21: Observed and estimated coordinates - Test 1

Figure 7.22: Estimated structure parameters and RMS error of UKF - Test 1
Figure 7.23: Estimated motion of marker carrier - Test 1

Figure 7.24: Extract from the reconstruction of marker carrier - Test 1
Test 2: UKF - Frame skipping

In this test we allowed the UKF to skip over frames with no motion to investigate if this could minimise the effect of the multiple changes in acceleration of the marker carrier during real sequence 1. Results presented in Figures 7.25-7.26.

There is no significant change in the estimated structure parameters of the UKF (Figure 7.25 (a)) if compared to the UKF in default setting (Figure 7.22 (a)). We show the rotation and translation components in Figure 7.26, the velocity components still suffer from large amounts of noise with the main difference between the UKF in default setting and frame skipping mode being the velocity components in frame skipping mode do not need to drop to zero during the stops in motion of the marker carrier and the velocity estimates are smoother over time.

The RMS error for this test are shown in Figure 7.25 (b). We see the RMS error stabilises after about 65 frames of motion (15 motionless frames removed before convergence) to about 2mm with a variance $\sigma_s^2 < 0.09 \text{mm}^2$ that is similar to the UKF in default settings.

![Figure 7.25: Estimated structure parameters and RMS error of UKF - Test 2](image-url)
Figure 7.26: Estimated motion of marker carrier - Test 2
Test 3: UKF - Adaptive mode

The third test on real data we present is of the UKF with adaptive mode enabled run using real sequence 1. In Figure 7.27 (a) we see the estimated structure parameters of the UKF are smoother over time, approaching the parallel lines expected for a UKF that has converged. The effect of the acceleration of the marker carrier is still present in the velocity components of the UKF (Figure 7.28) with a slight improvement in the estimation of the velocity components during and after a change in velocity if compared to the UKF in default setting (Figure 7.23).

The RMS error (Figure 7.27 (b)) and variance are of the same order as the the UKF in default mode with the RMS error about 2mm and variance $\sigma_s^2 < 0.09\text{mm}^2$ stabilising after 80 frames.

![Figure 7.27: Estimated structure parameters and RMS error of UKF - Test 3](image)
Figure 7.28: Estimated motion of marker carrier - Test 3
Initialising the UKF with pre-computed data from a UKF

In these tests we investigate improving UKF convergence using initialisation data obtained from a previous run of the UKF on real sequence 1. We initialise two UKF’s, one in default mode and one in adaptive mode, each with initialisation data obtained from the UKF run in default and adaptive modes (Test 1 and Test 3) on real sequence 1.

Test 4: Default UKF initialising a default UKF

We show the estimated coordinates overlaid on the observed coordinates over time in Figure 7.29. We see that first few predictions of the UKF initialised with pre-computed data do not match up with the actual observations, this is due to a mis-alignment of the initialisation structure. In Figure 7.30 (a) we see this effect more visibly as the estimated structure parameters jump from the initialisation values to the correctly aligned values over the first few frames. The multiple acceleration changes of the marker carrier during real sequence 1 prevent the structure estimate of the UKF from obtaining the parallel lines associated with convergence until late in the sequence.

The rotation and translation parameters of the UKF are presented in Figure 7.31. We again see an offset in the first couple of frames but it soon recovers to the proper estimates. The noise effects on the velocity components is similar to those in Test 1.

The RMS error (Figure 7.30 (b)) of the structure is never larger then 2.6mm with the error stabilising after 80 frames at < 1.9mm with a variance $\sigma_s^2 < 0.09\text{mm}^2$. 
Figure 7.29: Observed and estimated coordinates - Test 4

Figure 7.30: Estimated structure parameters and RMS error of UKF - Test 4
Figure 7.31: Estimated motion of marker carrier - Test 4
Test 5: Default UKF initialised with UKF in adaptive mode

In this test we initialise a UKF in default setting with data from Test 3 (UKF in adaptive mode). We again see the effect from the miss-alignment of the initialisation data in the structure estimate (Figure 7.32 (a)) and the rotation and translation velocities (Figure 7.33 (b) and (d)) with the UKF using the first couple of frames to align itself correctly.

The main difference between between this test and Test 4 is seen in the RMS error (Figure 7.32 (b)) that seems to level out and remain stable at 1.9mm after 100 frames with a structure variance $\sigma_s^2 < 0.09\text{mm}^2$.

Figure 7.32: Estimated structure parameters and RMS error of UKF - Test 5
Figure 7.33: Estimated motion of marker carrier - Test 5
**Test 6: Adaptive UKF initialised with default UKF**

Test 6 involved initialising a UKF in adaptive mode with data from Test 1 (UKF default setting). The structure estimate of the UKF over time (Figure 7.34 (a)) looks quite different from those of tests 4 (Figure 7.30 (a)) and 5 (Figure 7.32 (a)). We were required to adjust the covariance values of the UKF from those of tests 5 and 6 to obtain UKF convergence, this is responsible for the difference in structure estimate graphs.

The translation and velocity components of the UKF (Figure 7.35) again suffer from a small initialisation error and are very similar to those of tests 4 and 5.

The RMS error (Figure 7.34 (b) ) shows the error remains under 2.7mm for the entire sequence and stabilises under 2mm after 80 frames with a structure variance $\sigma_s^2 < 0.1\text{mm}^2$. These results are similar to those of tests 4 and 5.

![Figure 7.34: Estimated structure parameters and RMS error of UKF - Test 6](image-url)
Figure 7.35: Estimated motion of marker carrier - Test 6
Test 7: Adaptive UKF initialised with an Adaptive UKF

In this test we used the same covariance values as in Test 6 to aid UKF convergence. The adaptive UKF is initialised using data obtained from Test 3 (UKF adaptive mode) with results presented in Figures 7.36-7.37.

The structure estimate of the UKF (Figure 7.36 (a)) is closer to those of tests 4 and 5 (Figures 7.30 (a) and 7.32 (a)) and stabilises after 80 frames to a RMS error (Figure 7.36 (b)) of 1.9mm with a maximum error just larger then 2.4mm at 40 frames. The structure variance after stabilisation is $\sigma_s^2 < 0.12\text{mm}^2$.

![Figure 7.36: Estimated structure parameters and RMS error of UKF - Test 7](image-url)
Figure 7.37: Estimated motion of marker carrier - Test 7
CHAPTER 7. TESTS AND RESULTS

Test 8 : PMF used for initialisation data

We present results from tests using a PMF system to pre-calculate initialisation data for the UKF. We batch process 40 frames of real sequence 1 with a PMF system. The PMF system returns the structure and motion of the reconstructed marker carrier over the frames it processed. We need to scale and orientate this structure with the estimated structure of the UKF to minimise the correction the UKF needs to perform to correctly align the initialisation structure during the first couple of frames. We use the UKF in adaptive mode for this test.

In Figure 7.38 we show the structure used to initialise the UKF, the structure is similar to the expected form of markers on the marker carrier but there is still a fair amount of distortion present. The effect of this can be seen in Figure 7.39 (a). The acceleration of the marker carrier combined with the distortion and mis-alignment in the initialisation data prevent the UKF structure estimate from converging quickly.

On inspection of the translation and rotation components of the UKF (Figure 7.40) we see that the effect of the acceleration of the marker carrier in real sequence 1 is reduced and this results in a lower RMS error (Figure 7.39 (b)) after 100 frames less then 1.3mm with a structure variance $\sigma_s^2 < 0.02\text{mm}^2$.

![Figure 7.38: Reconstructed marker carrier from PMF system - Test 8](image)
CHAPTER 7. TESTS AND RESULTS

Figure 7.39: Estimated structure parameters and RMS error of UKF - Test 8

Figure 7.40: Estimated motion of marker carrier - Test 8
Summary of results for real sequence 1

We present a summary of the results from real sequence 1 in Table 7.2. We see the UKF performs well in all the tests with the best results obtained from Test 8 (UKF initialised with PMF) with regards to the RMS error.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>Test 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_s$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_{r,t}$</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>$R$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Num feats</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Convergence (frames)</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td>Max RMSE</td>
<td>29mm</td>
<td>16mm</td>
<td>29mm</td>
<td>2.6mm</td>
<td>2.6mm</td>
</tr>
<tr>
<td>Min RMSE</td>
<td>1.8mm</td>
<td>1.9mm</td>
<td>1.9mm</td>
<td>1.7mm</td>
<td>1.8mm</td>
</tr>
<tr>
<td>feats &lt; 2mm out</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>$\sigma^2_s$ &lt;</td>
<td>0.07mm²</td>
<td>0.1mm²</td>
<td>0.1mm²</td>
<td>0.09mm²</td>
<td>0.1mm²</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Test 6</th>
<th>Test 7</th>
<th>Test 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>0.005</td>
<td>0.005</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_s$</td>
<td>0.0005</td>
<td>0.0005</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_{r,t}$</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>$R$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Num feats</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Convergence (frames)</td>
<td>75</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>Max RMSE</td>
<td>2.7mm</td>
<td>2.4mm</td>
<td>7.2mm</td>
</tr>
<tr>
<td>Min RMSE</td>
<td>1.2mm</td>
<td>1.8mm</td>
<td>1mm</td>
</tr>
<tr>
<td>feats &lt; 2mm out</td>
<td>12</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>$\sigma^2_s$ &lt;</td>
<td>0.1mm²</td>
<td>0.12mm²</td>
<td>0.02mm²</td>
</tr>
</tbody>
</table>

Table 7.2: Summary of results from real sequence 1

7.3.2 Real sequence 2 - Proton vault SPG system

Real sequence 2 was captured in the Proton vault at iThemba Labs. The images are taken at a resolution of 576 × 768 (PAL) at 6 frames per second. Real sequence two consists of 221 frames during which we could consistently track 19 markers on the marker carrier, the tracked markers (with radial lens distortion correction) are shown in Figure 7.41. The marker carrier movement is much smoother in this sequence with 3 changes in direction introducing acceleration components into the system (one occurring before the UKF has time to converge).
We run the UKF with default settings, frame skipping, adaptive mode enabled and using pre-computed data from the PMF system and the UKF to initialise the UKF in adaptive mode.

**Test 9 : UKF - default setting**

The estimated marker coordinates overlaid on the observed coordinates is shown in Figure 7.42. Just as with real sequence 1 the UKF accurately follows the observations and we see how the UKF needs to recover after a change in direction of the marker carrier to again closely match the observations.

There are fewer and less sharp changes in motion during real sequence 2 and we see that most of the estimated structure parameters of the UKF (Figure 7.43 (a)) approach what we would ideally like to see for a convergent UKF. We notice a drift in a few of the structure components over time that was not present in tests performed with real sequence 1. This is due to tracked markers (on the edge of the marker carrier) becoming oblique with the camera during the sequence resulting in an offset in the detected marker’s centroid. This distortion (noise) is not Gaussian and therefore is not effectively taken into account by the UKF, causing the drift in the structure parameters.

The translation and rotation velocity components (Figure 7.44) show much less noise experienced through this sequence and the RMS error (Figure 7.43
(b)) stabilises after about 60 frames to $< 1.6\text{mm}$ with a structure variance $\sigma_s^2 < 0.03\text{mm}^2$.

We show the reconstructed features in (Figure 7.45), the features that are reconstructed are those on the forehead to the bridge of the nose.

![Figure 7.42: Observed and estimated coordinates - Test 9](image1)

![Figure 7.43: Estimated structure parameters and RMS error of UKF - Test 9](image2)
Figure 7.44: Estimated motion of marker carrier - Test 9

Figure 7.45: Extract from the reconstruction of marker carrier - Test 9
Test 10: UKF - Frame skipping

In Test 10 we run the UKF allowing it to skip motionless frames in real sequence 2. The changes in the estimated structure parameters of the UKF over time (Figure 7.46 (a)) are very similar to those of Test 9 (Figure 7.43 (a)) with there being no clear reduction in noise experienced by the rotation and translation velocity components (Figure 7.47) or in the drift of structure components over time. The RMS error (Figure 7.46 (b)) stabilised after about 50 frames to $< 1.6\text{mm}$ with a structure variance $\sigma_s^2 < 0.03\text{mm}^2$.

![Figure 7.46: Estimated structure parameters and RMS error of UKF - Test 10](image)
Figure 7.47: Estimated motion of marker carrier - Test 10
Test 11: UKF - Adaptive mode

For Test 11 we ran the UKF with adaptive mode enabled on real sequence 2. We see the affect of the adaptive mode in the structure parameters of the UKF (Figure 7.48 (a)) with them being smoother over time than those of tests 9 (UKF default, Figure 7.43 (a)) and 10 (UKF frame skipping, Figure 7.46 (a)). There is still a drift in some of the structure components over time due to the offset in the detected marker centroid of oblique markers.

The noise effects of the acceleration in the system can be seen in Figure 7.49, with the noise experienced by the rotation and translation velocities being of the same order as that of tests 9 and 10. The RMS error (Figure 7.48 (b)) improves slightly and stabilises after about 55 frames to < 1.4mm with a structure variance $\sigma_s^2 < 0.03mm^2$.

![Figure 7.48: Estimated structure parameters and RMS error of UKF - Test 11](image-url)
Figure 7.49: Estimated motion of marker carrier - Test 11
UKF - Initialise with UKF data

Test 12: UKF default setting initialised with UKF adaptive

In this test we initialise a UKF in default setting with initialisation data obtained from Test 11 (UKF in adaptive mode) to investigate improving filter convergence and accuracy.

Figure 7.50 shows the estimated coordinates of the tracked markers overlaid on the actual observations. We again suffer from a mis-alignment of our initialisation structure, however it is not as severe as with real sequence 1 (Tests 4-8). The structure estimate of the UKF (Figure 7.51 (a)) shows an ideal case of filter convergence with the structure parameters remaining relatively flat during the sequence. We experience a smaller drift in structure components over time indicating the UKF handles the offset in centroids of oblique markers and changes in acceleration better after convergence. The effects of noise in the translation and rotation velocities (Figure 7.52) are minimal with the UKF closely following the actual motion.

The RMS error of the structure (Figure 7.51 (b)) remains < 1.4mm throughout the sequence with a variance $\sigma_s^2 < 0.04\text{mm}^2$.

![Figure 7.50: Observed and estimated coordinates - Test 12](image)
(a) Estimated structure parameters
(b) RMS error (mm)

Figure 7.51: Estimated structure parameters and RMS error of UKF - Test 12

(a) Rotation - Quaternion
(b) Rotation velocity (radians/frame)

(c) Translation
(d) Translation velocity (units/frame)

Figure 7.52: Estimated motion of marker carrier - Test 12
Test 13: UKF adaptive mode initialised with UKF adaptive mode

We use the UKF in adaptive mode initialised with data from Test 11 (UKF adaptive mode) on real sequence 2 for this test. The adaptive mode again aids in smoothing the estimated structure parameters of the UKF over time which can be seen in Figure 7.53 (a).

The effects of the noise introduced by the acceleration components in this sequence is minimal as the estimated rotation and translation components (Figure 7.54) accurately follow the true motion of the marker carrier. On inspection of the RMS error (Figure 7.53 (b)) we notice that the error remains smaller and more stable than in the case of Test 12, (UKF default initialised with UKF adaptive mode, Figure 7.51 (b)). The maximum error in structure again remains under 1.4mm for the entire sequence with structure variance $\sigma_s^2 < 0.035\text{mm}^2$.

![Figure 7.53: Estimated structure parameters and RMS error of UKF - Test 13](image-url)
Figure 7.54: Estimated motion of marker carrier - Test 13
CHAPTER 7. TESTS AND RESULTS

(a) View 1

(b) View 2

Figure 7.55: Reconstructed marker carrier from PMF system - Test 14

Test 14: PMF used for initialisation

For this test we used the PMF system to calculate initialisation data for a UKF in adaptive mode. We used 50 frames from real sequence 2 to process with the PMF system. We show the initialisation structure obtained from the PMF system in Figure 7.55.

We show the estimated structure parameters of the UKF over time in Figure 7.56 (a), the UKF aligns the initialisation data correctly over the first 50 frames stabilising with an RMS error (Figure 7.56 (b)) < 1.2mm. The effect of the offset in the detected centroids of oblique markers and the acceleration of the marker carrier is more noticeable in this test if compared to tests 12 and 13 (initialising the UKF with UKF data) and causes a small drift in some structure components.

Noise effects on the rotation and translation velocities (Figure 7.57) are of the same order as those of tests 12 and 13 (initialising the UKF with UKF data) and the structure variance remains small at $\sigma^2_s < 0.03\text{mm}^2$. 
Figure 7.56: Estimated structure parameters and RMS error of UKF - Test 14

Figure 7.57: Estimated motion of marker carrier - Test 14
7.3.3 Real sequence 3 - More on using the UKF to calculate initialisation data

In this section we demonstrate the use of the UKF to calculate initialisation data using a subset of frames from the sequence. The video sequence in this test is of the same motion as in real sequence 2 but taken from a different view. We were able to consistently track 16 markers during the sequence, the tracked markers (with radial lens distortion correction) are shown in Figure 7.58.

![Figure 7.58: Detected markers tracked during real sequence 3](image-url)

In this test we ran the UKF in adaptive mode on the first 40 frames of the sequence and then used this calculated structure to initialise a UKF for the remainder of the sequence. By choosing the last frame of the initialisation sequence and the first frame of the actual sequence to be consecutive frames we solve the alignment issue of the initialisation structure encountered in previous tests using the UKF to calculate initialisation data. In Figure 7.59 we see the initialisation structure is optimally aligned as the estimated coordinates match the actual observations from the first frame.

The structure estimate of the UKF (Figure 7.60 (a)) stabilises after about 40 frames to a RMS error (Figure 7.60 (b)) < 1.5mm and structure variance $\sigma_s^2 < 0.05mm^2$. We see the structure estimate is very smooth showing ideal convergence with the effect of the change in motion during the sequence being
very small on the rotation and translation velocity components (Figure 7.61). The markers tracked during this sequence stayed in good view of the camera and we therefore experience less offset in centroid detection, this results in very little drift of the estimated structure components.

We show the reconstructed markers in Figure 7.62, the markers reconstructed are from the top of the forehead to the bridge of the nose on the marker carrier.

![Figure 7.59: Observed and estimated coordinates - Test 15](image-url)
Figure 7.60: Estimated structure parameters and RMS error of UKF - Test 15

Figure 7.61: Estimated motion of marker carrier - Test 15
Figure 7.62: Extract from the reconstruction of marker carrier - Test 15
7.3.4 Summary of results for real sequence 2 and 3

We present a summary of the results from real sequence 2 and 3 in Table 7.3. The UKF gives better results compared to those from real sequence 1, this is due the large amount of constant movement during real sequences 2 and 3 that we did not have in real sequence 1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Test 9</th>
<th>Test 10</th>
<th>Test 11</th>
<th>Test 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_s$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_{r,t}$</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>$R$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Num feats</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>Convergence (frames)</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>Max RMSE</td>
<td>48mm</td>
<td>48mm</td>
<td>48mm</td>
<td>1.4mm</td>
</tr>
<tr>
<td>Min RMSE</td>
<td>0.9mm</td>
<td>1.1mm</td>
<td>0.9mm</td>
<td>0.83mm</td>
</tr>
<tr>
<td>feats &lt; 2mm out</td>
<td>19</td>
<td>18</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>$\sigma^2_s &lt;$</td>
<td>0.03mm$^2$</td>
<td>0.03mm$^2$</td>
<td>0.03mm$^2$</td>
<td>0.03mm$^2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Test 13</th>
<th>Test 14</th>
<th>Test 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_s$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$Q_{r,t}$</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>$R$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Num feats</td>
<td>19</td>
<td>19</td>
<td>16</td>
</tr>
<tr>
<td>Convergence (frames)</td>
<td>0</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>Max RMSE</td>
<td>1.4mm</td>
<td>13mm</td>
<td>10mm</td>
</tr>
<tr>
<td>Min RMSE</td>
<td>0.83mm</td>
<td>0.9mm</td>
<td>1.3mm</td>
</tr>
<tr>
<td>feats &lt; 2mm out</td>
<td>19</td>
<td>19</td>
<td>14</td>
</tr>
<tr>
<td>$\sigma^2_s &lt;$</td>
<td>0.03mm$^2$</td>
<td>0.03mm$^2$</td>
<td>0.05mm$^2$</td>
</tr>
</tbody>
</table>

Table 7.3: Summary of results from real sequence 2 and 3

7.4 Summary

From the results presented in this chapter we see that our system performs well on real and synthetic data. The synthetic tests show the UKF operates correctly in ideal and noisy conditions accurately tracking the complex motion of a synthetic cube. The results from the synthetic tests are presented in Table 7.1. The tests performed on real video sequences captured at iThemba Labs gave good results. We saw that the results from real sequence 2 and 3 are more accurate then those of real sequence 1 (Tables 7.2 and 7.3), and found
this is due to the motion of the marker carrier in real sequence 1 having multiple sharp acceleration changes that our motion model does not handle, forcing the Kalman gain to erroneously adapt the structure and motion parameters of the UKF to agree with the observations. In tests with real sequence 2 we experienced an offset in the detection of the marker centroid of oblique markers that caused a drift in the structure components of the UKF.

We proposed two methods to improve filter performance in this thesis: skipping motionless frames and using adaptive filtering. The tests in which we used frame skipping (Tests 2 and 11) showed no improvement in filter performance with the RMS error and structure variance similar to the UKF in its default setting. The one advantage we notice with frame skipping is that the UKF converged slightly faster if compared to the UKF in defaults settings.

The tests run with the UKF in adaptive mode showed promising results with the adaptive UKF having a definite smoothing and stabilising effect on the estimated structure parameters and the RMS error.

Our attempts at improving the speed of UKF convergence by using pre-computed initialisation data gave us the most accurate results in terms of the structure RMS error. The RMS error of the UKF initialised with pre-calculated data from a UKF remained low throughout the sequence. The UKF responded to changes in the motion of the marker carrier and the offset in centroid detection of oblique markers better using initialisation data. We encountered issues correctly aligning the initialisation data when using both the UKF and PMF methods. We found a solution to alignment error of the UKF initialisation data by choosing the last frame of the initialisation set and first frame of the main set to be consecutive frames from the sequence in question.
Chapter 8

Conclusions

Our goal in this thesis was to develop a structure from motion (SFM) system that accurately estimates the structure of the detected markers while not relying on any calibration information. The intended use of this system being to independently verify the correct operation of the patient positioning system at iThemba Labs.

In Chapter 2 we presented our literature study where we investigated the most popular methods in computer vision used to obtain structure from images. We chose to use a Kalman filter in our SFM system as it does not rely on calibration information (as in the case of stereo vision and structured light), it is more computationally efficient than a particle filter and can run at real time (unlike matrix factorisation methods).

Our discussion of quaternions commenced in Chapter 3 where we presented the basic mathematics surrounding quaternions and showed how to relate the Euler axis and angle representation of rotation to a quaternion. We proceeded to derive a formula for the time derivative of a quaternion that is central to our motion model for rotation.

The Kalman filter was discussed in Chapter 4. We began the discussion with the linear Kalman filter (LKF) to aid us in understanding the working of the Kalman filter. Two methods to extend the LKF to handle non-linear problems were then presented, namely the extended Kalman filter (EKF) and unscented Kalman filter (UKF). We chose to use the UKF for this thesis as it offers a higher statistical accuracy at the same computational complexity as the extended Kalman filter (EKF). The structure, motion and observation models used by the UKF are presented in Chapter 5.

In Chapter 6 we discussed the implementation issues we dealt with during this thesis. We covered the initialisation and setup of the UKF, marker detec-
tion, radial lens distortion and presented the real time performance statistics of our system. The largest influence in the accuracy of our results was found to be radial lens distortion.

We presented a large number of results from tests run using synthetic and real data in Chapter 7. The synthetic tests served to prove the correct working of our system and showed our system works correctly in ideal conditions. We found that the performance of our system on real sequences was best with sequences containing smooth motion (as we expected due the assumptions made in our motion model) but still gave good results with the sequence containing multiple sharp changes in motion (real sequence 1). We further investigated improving the performance of the UKF by using pre-computed initialisation information and allowing the UKF to change its covariance values adaptively. We found that using initialisation data improves convergence time and that the adaptive UKF gave smoother structure estimates lessening the effect of the acceleration in marker carrier motion.

Future work for this system includes

- Extending the motion models to take acceleration into account
- Investigating the use of unscented transforms using fewer sigma points
- Particle filter hybrid systems
- Alignment of initialisation data

Extending our motion models to take acceleration into account would lead to more accurate structure and motion prediction, but cause an increase in the state size of our system and thus an increase in the computational complexity. Using reduced sigma point methods may result in a reduction in reconstruction accuracy but will reduce the computational complexity of our system significantly. Combining these two methods may result in a system with the same accuracy of the UKF in this thesis but at a lower computational complexity, this would improve the real time performance of our system.

Particle filters provide a higher accuracy of statistical modelling than Kalman filter at the cost of computational complexity. The combination of the two systems could prove to be more accurate at a slightly higher computational expense. Alignment of the initialisation data was an issue in this thesis that resulted in an increase in the convergence time of the UKF. Correct alignment of initialisation data would further improve the speed of UKF convergence.

In closing we successfully developed a SFM system that accurately reconstructs
detected markers on the marker carrier that can serve as comparison to test for the correct operation of the SPG system at iThemba Labs. We further investigated four methods to improve UKF performance and convergence. From the results of tests performed on our system we found the optimal setup is to use the UKF in adaptive mode with initialisation information pre-calculated by the UKF. We recommend calculating initialisation data as in Test 15 (using a subset of frames from the entire sequence) as this minimised the effect of the mis-alignment of initialisation data and provided very accurate results.
Appendix A

Supplementary information

The supplementary information is provided as a guide for the project on the CD in Appendix B. The CD contains the Python source code for the SFM system, test video sequences and a copy of this thesis.

The software libraries required by our SFM system are discussed in Section A.1. Section A.2 describes the functions the Python modules in our system perform. In Section A.3 we show the SFM graphical user interface (GUI) and provide an introduction into its use. Section A.5 completes this appendix with the procedures for plotting data gathered by our system.

A.1 Required Software

This system was written in Python 2.4 and requires a compatible Python interpreter. The following Python modules are used in our system an need to be installed:

- wxPython 2.6.3.2
- NumPy 1.0.1
- OpenCV 1.0.0
- MatPlotLib 0.87.7
- PIL (Python Imaging Library) 1.1.6
- SciPy 0.5.2
- pickle Revision: 38432
- PyVtk
- Mayavi 1.5-4
All these required modules are available on and were installed from the Ubuntu Linux software repository.

We also require the installation of the SPG vision code library, this provides the marker detection and lens distortion correction used in this thesis. We include the code library on the supplementary CD in appendix B. The instructions for the compilation and installation of this library are provided in the “readme” file located in source code directory.

### A.2 Module description

We provide a description of the most important Python modules in our system:

- *sfm_gui.py* is the main user interface for the system

- *sfm.py* is the glue in our system combining the UKF with all other components

- *ukf_filter_n.py* is our implementation of the UKF

- *sfm_filter_n.py* creates an instance of *ukf_filter_n.py* and sets it up for SFM

- *ukf_srt.py* is the state update function of the UKF

- *ukf_cm.py* is the observation prediction function of the UKF

- *sfm_rescale.py* transforms the projective structure to a metric (Euclidean) structure via ground truth

- *data_plot.py* plots data about the sequences processed by our system

- *maya_plot.py* uses Mayavi and VTK to generate 3-dimensional plots of our structure
• *SingleBodyReconstruct.py* performs the calculation of our initialisation data using projective matrix factorisation. The projective matrix factorisation system was written by J.H. de Vaal [18]

## A.3 Graphical user interface (GUI)

We start the GUI for our system by running `sfm_gui.py`, which is located in the `source/spg-sfm/` directory on the supplementary CD. The main frame of our GUI is displayed in Figure A.1.

The main frame of our GUI provides us with all the options that we can run our system with. In the top left quadrant of our GUI we can set the covariance values for the UKF. There are three large buttons located in the centre of the GUI. The buttons, namely “Run SFM synthetic”, “Run SFM with tracker” and “run SFM with PF to init” provide the three main ways the system can be run.

![Figure A.1: SFM GUI - main frame](image)

To run a synthetic test case we first need to choose the covariance values
for the UKF and then click the “Run UKF synthetic” button. The default synthetic sequence is that of the cube in complex motion containing rotation and translation. To change the synthetic test sequence we need to edit the `rot_sequence()` function located in the `synth_cube.py` source file. The source file is well commented.

To run our system on a real test sequence we can use either the “Run SFM with tracker” or “Run SFM with PF to init” buttons. For both these options we need to first select a real video sequence. This is done by clicking the “Select Video Sequence” button. A file dialog box will appear (Figure A.2), we then navigate to the desired video sequence directory and select all the frames we need and press “OK”. The camera with which the sequence was captured also needs to be set in the “Camera no.” text box so that the correct lens distortion can be used.

![Figure A.2: SFM GUI - Select video sequence](image)

To run the UKF in the defaults settings we just click on the “Run SFM with tracker” button after choosing the covariance values and selecting a video sequence. After the “Run SFM with tracker” has been clicked the system calculates the lens distortion for the sequence and the prompts the user to deselect any markers that may become occluded during the sequence (Figure A.3).
To run the system with “frame skipping” or “adaptive mode” enabled simply select the corresponding check box, choose covariance values and a video sequence, and click on the “Run SFM with tracker” button. Lens distortion will be calculated and the user will again be prompted to remove any markers which may become occluded (Figure A.3).

To initialise the system with data calculated from the previous run select the “Init KF with final state of previous run” check box, choose the covariance values and click the “Run SFM with tracker” button. For this option to work one must select the same markers to track as used in the previous run.

To use the projective matrix factorisation (PMF) to calculate the initialisation data for our system we use the “Run SFM with PF to init”. For this option we need to choose covariance values, a video sequence and the number of frames the PMF system uses. The options that can be used in the PMF system are displayed in the lower right quadrant of the GUI (Figure A.1). We have options to set camera calibration (we do not use this in this thesis) and the reconstruction type (reconstruction to a similarity gives the best results).
We have four options for plotting the structure of the reconstructed object in our system, these option are in the upper right quadrant of the GUI (Figure A.1). The “Plot PF Structure” button plots the 3-dimensional initialisation structure obtained from the PMF system, “Plot Kalman Structure” plots the projective structure estimated by the UKF and “Plot Euclidean Structure” plots the Euclidean structure calculated using ground truth. The “Plot Animation of Structure” button plots an animation of the the projective structure estimate over time.

If the “Plot Kalman Structure” option is selected the user will be prompted for the frame at which to perform reconstruction (Figure A.4), frame -1 indicates the last frame of the sequence). The user must choose a frame and press “Open”, the structure is then plotted using Mayavi (Figure A.5). To close the plot and return to our system exit Mayavi and press return in the terminal where prompted.

![Figure A.4: SFM GUI - select frame to obtain structure from](image)

To view the Euclidean reconstruction of our object we first need to transform the projective structure using ground truth. To do this click on “Correct Scale” button located on the main GUI (Figure A.1). The user will be prompted for the frame number at which to perform the rescaling of the structure (Figure A.4).
Select the desired frame and press “Continue”. The user is then asked to match corresponding markers (Figure A.6) between the ground truth coordinates and the reconstructed markers. Once corresponding points have been identified, click on “Rescale Reconstruction” button, the reconstruction information is printed in the terminal. We can now view the Euclidean structure by clicking on the “Plot Euclidean Structure” button.

### A.4 Plotting Data

We make use of MatPlotLib to plot 2-dimensional graphs. Unfortunately there were issues combining MatPlotLib into the GUI and we therefore are required to run the following Python modules from the linux terminal to obtain data plots for RMS error, variance, translation, rotation and convergence.

To plot the structure convergence, translation and rotation information obtained by our system the user needs to run `data_plot.py`, this file is located in the `spg_sfm` source code directory and is well commented.

To plot the RMS error and variance of the Euclidean structure over time the user needs to run `sfm_rescale.py`, this file is located in the `spg_sfm` source...
Figure A.6: SFM GUI - select corresponding markers for Euclidean reconstruction
code directory. This plot can only be made after the user has rescaled the
projective structure using the “Correct Scale” option of our GUI.

To plot the RMS error and variance for the synthetic test cases the file `data_plot.py`
needs to be edited and the indicated line of source code uncommented.
Appendix B

Supplementary CD
Bibliography

Available at: http://www.who.int/mediacentre/factsheets/fs297/en/index.html

Available at: http://www.tlabs.ac.za/public/default.htm


[34] Zikic, D. and Wein, W.: 3d rotations and quaternions, seminar: methods and tools in medical imaging. 2004. Available at: http://campar.in.tum.de/twiki/pub/Chair/TeachingSs04SeminarImaging
Available at: http://www.geometrictools.com


