# Tracking for Mobile 3D Augmented Reality Applications 



Fong Wee Teck<br>NUS Graduate School for Integrative Sciences and Engineering<br>National University of Singapore

A thesis submitted for the degree of
Doctor of Philosophy (PhD)
September 2010


#### Abstract

This thesis presents the PhD research carried out on tracking for mobile 3D augmented reality applications. Augmented Reality (AR) is the superposition of the virtual and the real environments, so that both the virtual elements and the real world can be interactively perceived by the user at the same time. The research focus is on robust, wide-area tracking for high precision 3D AR applications in non-prepared environments. The main motivation is to move AR out of laboratory, so as to achieve mobile AR in outdoor environments. Tracking allows the AR system to determine the segments of the real world that the user is looking at, so that virtual 3D objects can be inserted to appear visually coherent to the user. This allows computer systems to augment the users reality while he is on the move.

Wide-area applications require the tracking systems to operate in a wide range of conditions, and over a wide range of motion. Robustness, precision, low latency and jitter are important requirements for successful and satisfactory augmentation of the user's reality. This research takes a multidisciplinary approach towards solving the research question, through investigating three different but complementary tracking systems, namely Com-


puter Vision (CV), Inertial Measurement and Global Positioning Systems (GPS), to derive a hybrid wide-area tracker, known as the Augmented Reality TRackIng SysTem, or ARTIST. This approach is chosen based on the observation that no single property and its associated sensors are able to meet the requirements of robustness and precision. Sensors with complementing strengths and weaknesses can be combined together to approximate a perfect sensor.

As Inertial and GPS function well over a large area, the approach taken is to first improve the precision of both sensors, so that they can work reasonably well in regions where CV fails. The research on inertial measurement focused on the calibration of MEMS-based sensors, so that they can be used as independent orientation trackers. Calibration methods for tri-axial accelerometers and gyroscopic systems that are completely independently of external equipment have been developed. This allows end-users to perform calibration on-site, which has not been achieved for gyroscope calibration. For GPS, a novel method for GPS positioning based on the Differential Single Difference of GPS carrier phase measurement has been formulated. It is suitable for AR positioning with an accuracy of 10 cm , and avoids the computationally expensive resolution of integer ambiguity. However, the level of precision achieved is not comparable to CV.

The research on CV focuses on marker-less 6DOF tracking using natural features. A CV tracker with accurate 3D augmentation and good robustness
against illumination changes, partial occlusion and extreme object poses, has been developed and tested. Simultaneous augmentation of three objects at 15 fps was achieved through efficient system design, as well as improvements to underlying algorithms. Specifically, the keypoint signature is improved with a proposed matching method, which maintains the matching accuracy with lower computation load. New models were also proposed for Efficient Second-order Minimization (ESM) that allows for handling of radial distortion, shadows, specular glares and partial occlusion. Finally, two methods are developed to combine the sensor output. The first is a loosely coupled configuration where standalone GPS and inertial measurements are used to limit the search set for initializing the computer vision tracker. This system only works in areas where there are sufficient features for CV tracker. The second is a Kalman Filter based hybrid tracker, where low level sensor outputs, consisting of differential GPS carrier phases, acceleration and angular velocity and CV measure are combined. The second system is a true wide area tracker, with degraded precision with GPS and inertial tracking when CV fails.

## Contents

List of Figures ..... xiii
List of Tables ..... xix
1 Introduction ..... 1
1.1 Augment Reality TRackIng SysTem (ARTIST) ..... 4
1.2 Contributions ..... 5
1.3 Thesis Organization ..... 8
2 Design of ARTIST hybrid tracker for mobile Augmented Reality ..... 9
2.1 Background Information ..... 10
2.1.1 Augmented Reality Tracking ..... 12
2.1.2 Mobile Hybrid Tracking Systems ..... 13
2.2 Requirements for registering graphics in Augmented Reality ..... 16
2.2.1 Tracker spatial precision ..... 16
2.2.2 Occlusion and shading ..... 20
2.3 Design rationales for ARTIST ..... 22
2.4 Mathematical Framework for Hybrid Tracking ..... 24
2.4.1 Orientation Filter ..... 28
2.4.2 Position Filter ..... 31
3 Inertial sensors: Calibration for orientation sensing ..... 35
3.1 Introduction ..... 36
3.2 Background ..... 37
3.2.1 MEMS Accelerometer ..... 37
3.2.2 MEMS Gyroscope ..... 38
3.2.3 Strapdown Inertial Measurement Units ..... 39
3.2.4 Usage in Virtual and Augmented Reality Applications ..... 39
3.2.5 Coordinate Frames ..... 40
3.2.6 Strapdown IMU Computations ..... 41
3.2.7 Sensor Calibration ..... 43
3.2.8 Sensor Performance and Error Characteristics ..... 44
3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment ..... 46
3.3.1 Motivation ..... 48
3.3.2 Development of the Methods ..... 49
3.3.2.1 Tri-axial accelerometer error model ..... 50
3.3.2.2 Gyroscope bias removal during calibration ..... 52
3.3.2.3 Tri-axial gyroscopic system error model ..... 54
3.3.3 Quasi-static detection for measurement of static sensor outputs ..... 57
3.3.4 Proposed Calibration Procedures ..... 58
3.3.4.1 Controlled collection of accelerometer calibration data (Procedure 1) ..... 59
3.3.4.2 Controlled collection of gyroscope calibration data (Pro- cedure 2) ..... 60
3.3.4.3 In-field collection of calibration data (Procedure 3) ..... 63
3.3.5 Calibration Results and Analysis ..... 65
3.3.5.1 Accelerometer calibration with controlled data ..... 67
3.3.5.2 Gyroscope calibration with controlled data ..... 69
3.3.5.3 Calibration with data collected using handheld and head- mounted IMU ..... 71
3.3.5.4 Analysis ..... 74
3.4 Concluding Remarks ..... 76
3.4.1 Future Developments ..... 77
4 Global Positioning System: Differential carrier phase for open-area positioning ..... 79
4.1 Introduction ..... 80
4.2 Background ..... 81
4.2.1 The Global Positioning System ..... 81
4.2.2 Applications in Augmented Reality ..... 82
4.2.3 Differential Global Positioning System ..... 83
4.2.4 Coordinate Frames ..... 84
4.2.4.1 Earth-Centered Earth-Fixed ( $\boldsymbol{E C E F}$ ) Frame ..... 85
4.2.4.2 North-East-Down (NED) Frame ..... 85
4.2.5 Precise Ephemeris ..... 87
4.2.6 GPS Carrier Phase Measurement ..... 90
4.2.7 Differential Positioning ..... 92
4.2.7.1 Single Difference ..... 92
4.2.7.2 Double Difference ..... 93
4.2.7.3 Triple Difference ..... 95
4.2.8 Integer Ambiguity ..... 96
4.3 Precise Positioning using Differential Single Difference ..... 98
4.3.1 Motivation for Precise Positioning using Differential Single Dif- ference (DSD) ..... 98
4.3.2 Development of the Method ..... 99
4.3.2.1 Drift Correction using Linear Regression ..... 103
4.3.3 Experimental Setup ..... 104
4.3.4 Experimental Results ..... 106
4.3.4.1 Experiment E1 ..... 106
4.3.4.2 Experiment E2 ..... 110
4.4 Concluding Remarks ..... 111
4.4.1 Issues ..... 113
5 Computer Vision: High precision positioning on textured planar sur-
faces ..... 115
5.1 Introduction ..... 116
5.2 Tracking System Organization ..... 121
5.2.1 Systems Perspective ..... 121
5.2.2 Algorithms Perspective ..... 125
5.3 Computer Vision Tracker Components ..... 134
5.3.1 Feature Detection ..... 134
5.3.1.1 Adaptive thresholding ..... 135
5.3.1.2 Feature orientation assignment ..... 138
5.3.2 Feature Matching ..... 140
5.3.2.1 Ferns ..... 141
5.3.2.2 Experimental determination of Fern parameter values ..... 149
5.3.2.3 Comparison with the sparse keypoint signature ..... 159
5.3.3 Pose Refinement ..... 163
5.3.3.1 Image warping using homography ..... 165
5.3.3.2 ESM and computation of Jacobian matrices ..... 170
5.3.3.3 ESM reference image ..... 173
5.3.3.4 Tolerance to illumination changes and partial occlusion ..... 175
5.4 Experimental Setup and Results ..... 178
5.4.1 Experimental Setup and Implementation Details ..... 178
5.4.2 Experimental Result ..... 179
5.4.2.1 Tracking of a single object ..... 179
5.4.2.2 Tracking of multiple objects ..... 181
5.4.2.3 Types of surfaces which can be tracked ..... 183
5.4.2.4 Comparison of the proposed illumination model with the discrete illumination model ..... 184
5.5 Concluding Remarks and Future Works ..... 186
6 ARTIST hybrid tracker: Experimental results ..... 189
6.1 Introduction ..... 190
6.1.1 Loosely Coupled Configuration ..... 190
6.1.2 Distinctive Planar Surface for Outdoor AR ..... 191
6.1.3 Phases of System Operations ..... 192
6.1.4 Tightly Coupled Configuration using Kalman Filter ..... 194
6.2 Experimental Setup and Results ..... 195
6.2.1 Experimental Setup ..... 195
6.2.2 Experimental Results ..... 195
6.2.2.1 CV Tracking in outdoor environment ..... 195
6.2.2.2 Campus walkthrough using Loosely Coupled Configu- ration ..... 197
6.2.3 Kalman filter results ..... 198
6.2.3.1 Position Filter ..... 198
6.2.3.2 Orientation Filter ..... 201
6.3 Concluding Remarks ..... 204
7 Conclusion ..... 207
7.1 Analysis ..... 209
7.2 Recommendation for Future Research ..... 211
8 Publications ..... 213
References ..... 215
A Results videos ..... 229

## List of Figures

3.1 A simple accelerometer. ..... 38
3.2 IMU sensor setup. ..... 40
3.3 The misalignment of sensor axes from the ideal orthogonal configuration. ..... 51
3.4 The Allan Variance plot of the three gyroscopes in the prototype IMU. ..... 53
3.5 Plots of the three axes of $\boldsymbol{u}_{\mathrm{a}}$ (jagged lines) and $\boldsymbol{u}_{\mathrm{g}}$ (smooth lines) in an uncalibrated IMU. ..... 56
3.6 The process of quasi-static detection, showing the original signal going through high pass filtering, followed by rectification and low pass filtering. ..... 58
3.7 Three cases of the IMU with one axis parallel to the rotation hinge (dark edge of the grey surface) for gyroscope calibration. ..... 61
3.8 Six cases of the IMU with one axis perpendicular to the rotation hinge (dark edge of the grey surface) for gyroscope calibration. ..... 62
3.9 The procedural flow for in-field calibration data collection ..... 66
3.10 The custom-built IMU used in the experiments. ..... 67
4.1 Illustration of a right-handed frame. ..... 84
4.2 The Earth-Centered Earth-Fixed ( $\boldsymbol{E C E F}$ ) coordinate frame. ..... 85
4.3 A plot of five processed Double Difference signals obtained using raw measurements from six satellites. ..... 94
4.4 A comparison between measured and simulated Double Differences (DD) for two satellites, generated using the same reference satellite. ..... 100
4.5 Illustration of the variables used in DSD computation. ..... 101
4.6 Plot of the drift of position vector from the initial position of a stationary receiver, $\boldsymbol{r} \mathbf{1}$ against time $t$. ..... 107
4.7 Plots of position vector from the initial position of mobile receiver $r, r \boldsymbol{Z}$ derived using D2 against time $t$. ..... 108
4.8 Drift corrected position vector from the initial position of mobile receiver $r, \boldsymbol{r} \mathscr{Z}^{\prime}$ derived using D2 against time $t$. ..... 108
4.9 Drift corrected position vector from the initial position of mobile receiver $r, \boldsymbol{r} 3$ derived using D3 against time $t$. ..... 110
4.10 Plot of position vector from the initial position of mobile receiver $r, r 4$ against time $t$. ..... 111
4.11 Augmentation using the proposed Differential GPS tracker and IMU (The checker board is used to indicate the drift). ..... 112
5.1 Summary of the computer vision module of ARTIST and an overview of the algorithms running within each stage of the computer vision tracking operation ..... 133
5.2 The arrangement of a ring of 16 pixels for detection of features in FAST. 135
5.3 Comparison of the FAST-9 feature detection for the same image frame with different thresholding schemes. ..... 137
5.4 A sequence of three images for visual illustration of the stability of ori- entation assignment used in ARTIST. ..... 140
5.5 An illustration of the Fern testing process on a feature ..... 142
5.6 Superposition of 100 signatures from random projective warps of a key- point. ..... 146
5.7 The probability $p_{i}$ that a base class occurs in the $s_{k}$ for two features. ..... 147
5.8 The changes of $p_{1}$ to $p_{50}$ of a feature as training progresses. ..... 148
5.9 The six images used for extracting the features for training of the generic ferns and feature matching tests. ..... 151
5.10 The three test sets for the matching test using real images. ..... 152
5.11 The comparison of matching rates for the proposed peak probabilitiesmethod and the sparse signatures method for simulated and real testimages.161
5.12 The comparison of matching rates for the proposed peak probabilitiesmethod and sparse signatures for real image test sets.161
5.13 For this figure and Figure 5.14, the reference image is the tracked regionis enclosed within the blue square.166
5.14 Examples of the image warping process. The warped image of example 2has greater blur and errors, due to greater change in scale than example 1.167
5.15 The ESM reference image selection process where the sub-grids with high image gradient are rendered. ..... 174
5.16 Augmentation of a cube for checking the accuracy of the normal vector obtained. ..... 175
5.17 Augmentation of a teapot using ESM onto a planar surface with illumi- nation interferences and extreme object pose. ..... 180
5.18 Plots of $x, y$ and $z$ motions for video frames with occlusions similar to those shown in Figure 5.17(b) ..... 181
5.19 Augmentation of teapots onto three objects. ..... 182
5.20 Augmentation on a surface with rich and varied patterns. ..... 183
5.21 Tracking of a high contrast surface with large changes in scales. ..... 183
5.22 Augmentation on high gloss surfaces. ..... 184
6.1 The selection of surfaces for feature matching based on the GPS position and the $\boldsymbol{N E} \boldsymbol{D}$ orientation of the camera. (Selected surfaces are darkened)193
6.2 Summary of the hybrid tracking system. ..... 194
6.3 The experimental setup consisting of the Dragonfly camera, InertiaCube and LEA-4T GPS module with antenna. ..... 196
6.4 Examples of surfaces that can be augmented using ARTIST. ..... 196
6.5 Augmentation in the presence of camera rotation and large scale changes. ..... 197
6.6 The GPS positions of the seven augmentation sites, where the orientation is the direction from the smiling mouth towards the eyes of the icon. ..... 198
6.7 Augmentation at the seven selected sites. ..... 199
6.8 Reference computer vision tracking data for tightly coupled tracker in the camera frame ..... 202
6.9 The GPS Differential Single Difference tracking data corresponding to the motion shown in Figure 6.8 ..... 202
6.10 The comparison of performance of position filter tracking against the reference computer vision data. ..... 203
6.11 Plot of the variation of the four elements of the orientation quaternion during the orientation filter test. ..... 204
6.12 The errors in augmentation during computer vision tracking failure. ..... 205

## List of Tables

3.1 Mean and standard deviation of all the parameter values (dimensionless) obtained from calibration using 30 sets of data. The scale factor and bias parameters are emphasized. ..... 68
3.2 The average and maximum observed magnitudes of errors from the ideal $1 g$ for the measured static accelerations. The angular errors are shown in brackets. ..... 68
3.3 The mean and standard deviations of gyroscope model parameter values (dimensionless). ..... 69
3.4 Average magnitude of divergence and angular deviation between $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$, with and without applying the gyroscope error model in Eq. 3.22. ..... 71
3.5 The mean and standard deviations of the accelerometer error model parameter values (dimensionless) ..... 72
3.6 Comparison of the average and maximum magnitudes of errors for the same accelerometer test data, compensated with model parameters ob- tained using the three procedures. ..... 73
3.7 The mean and standard deviations of the gyroscope error model param- eter values (dimensionless). ..... 75
3.8 Comparison of the average magnitudes of divergence, angular deviation between $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$ for the same gyroscope test data sets, compensated with model parameters obtained using the three procedures. ..... 75
4.1 The World Geodetic System 1984 (WGS 84) reference ellipsoid ..... 86
4.2 Closed-form solution used to convert of $\boldsymbol{E C E F}$ coordinates to geodetic latitude. ..... 88
5.1 Summary of the stages of tracker operations and comparisons between ARToolkit, PTAM, SIFT and ARTIST ..... 126
5.2 Variations of matching rate, run time per feature and memory required with respect to the number of Ferns and tests per Fern. ..... 154
5.3 The variation of matching rate, run time per feature and memory re- quired, with respect to the number of random warps. ..... 156
5.4 The variation of matching rate and time per feature with respect to the value of $k$. ..... 157
5.5 The variation of matching rate, time per feature and memory usage, with respect to the number of base classes. ..... 159
5.6 The variation of matching rate, time per feature and memory usage, with respect to the number of base classes. ..... 160
5.7 Values of the camera intrinsic parameters and radial distortion coefficients. 179
5.8 Average computational times for key tracking components. ..... 182
5.9 The rms pixel error and average processing time per video frame. ..... 185

## 1

## Introduction

## 1. INTRODUCTION

This thesis presents the PhD research carried out on tracking for mobile 3 D augmented reality applications. The primary research focus is on wide-area, unassisted, robust and precise tracking in non-prepared environments. Here, wide-area can be best described as being the opposite of local-area. In particular, wide-area refers to the lack of a physical boundary for the tracker operation. This is analogous to the cellular network being wide-area, while WiFi is local-area. A wide-area tracker should work in whole area, rather than in isolated spots, which would be considered as local-areas separated by large distances. Unassisted means to track independently, without additional infrastructure, therefore non-prepared environments. This involves the determination of the orientation and translation of the tracker relative to a predefined world coordinate system. Specifically, it allows the system to determine the segments of the scene of the real world that the user is looking at. This allows the insertion of virtual threedimensional (3D) objects so that they appear visually coherent to the user. In a sense, this research aims to align the coordinate systems of the real and virtual worlds.

Augmented Reality (AR) can be broadly defined as the superposition of virtual elements, mainly 3 D graphics, onto the real world so that both the virtual elements and the real world can be perceived by the user at the same time (Milgram et al., 1994). It is generally accepted that AR consists of (1) a combination of both the virtual and real worlds with (2) real-time interactivity and (3) registration in the 3D space. An important aspect is interactivity, where the appearance of the virtual element reacts to the changes in the view point of the user in real time. Therefore, for the movie industry, where computer graphics are inserted realistically into live footages, it would
not be considered as AR. This is because there is no interactively, but the concept of superpositioning the virtual onto the real is similar. This interactivity entails realtime performance from the tracking systems, which in turn necessitates novel hardware architectures and software algorithms. The level of performance of AR tracking and registration systems has to reach one that is both natural and comfortable to the majority of users. This is a necessary condition for AR to be used as a new form of computer interface, where the computer output is fused with the real world, instead of being separated as they are now. It will also facilitate the development of new interaction techniques, which at present often appear unwieldy due to the nature of the current tracking systems.

The problem of precise tracking has been effectively solved for small, local-area applications. The solutions range from the trackers and sensors developed in the field of Virtual Reality (VR) to the popular ARToolkit. For wide-area applications, the user moves beyond a locally controlled environment, thus requiring the tracking systems to operate in a wider range of conditions, and over a wider range of motion. Robustness, precision, low latency and jitter are important requirements for successful and satisfactory augmentation of the users reality. Robustness refers to continuous operation in the presence of interference, as well as rapid recovery from complete tracking failure. Ideally, the user does not notice any failures. Precision refers to the low errors of the position and orientation measurements, which can be less than one millimetre and one degree respectively, for augmentation within an arm's length. Latency is the time between the actual measurement and tracker output. If the latency is significant,

## 1. INTRODUCTION

the virtual objects will appear to detach and lag behind the real object during motion. Jitter refers to the 'shaking' of the virtual object, which should not be perceptible to the user. The actual precision, latency and jitter requirements are dependent on the actual AR application. These requirements often present significant challenges in uncontrolled environments. Furthermore, the tracking systems would often be required to operate without modifications to the environment. Therefore, both active and passive marker-based systems are not applicable. Consequently, these systems have to work unassisted, relying on the properties of the environment to perform the tracking. Much effort had been expended in the preceding research in AR, navigation and robotics, on finding properties common to a majority of environments that can be robustly utilized for tracking purposes.

### 1.1 Augment Reality TRackIng SysTem (ARTIST)

This research takes the approach of a detailed study into Computer Vision (CV), Inertial Measurement and Global Positioning System (GPS) to derive a hybrid wide-area tracker, known as the Augmented Reality TRackIng SysTem, or ARTIST. This approach is chosen based on the observation that no single property and its associated sensors are able to meet the requirements of robustness and precision. Sensors with complementing strengths and weaknesses can be combined together to approximate a perfect sensor. Inertial systems utilize the Newtonian laws of motion, which are applicable to practically all environments on Earth, making them robust. They also have high precision and low latency, but suffer from drift errors. GPS is complementary in
that its errors are bounded, but it has high latency and low precision. CV systems are relatively less robust, but provide high precision and increased versatility beyond the tracking, especially as a HCI device. For example, CV can be used for object identification, hand gesture interactions and facial recognition.

### 1.2 Contributions

Research work on the three tracking technologies resulted in contributions in each area. Research on CV-based trackers resulted in the development a wide-area, six degrees of freedom (6DOF) tracker that can obtain the camera pose relative to multiple planar textured surfaces in real-time. Recently developed algorithms for feature detection, matching and pose refinement are improved and combined to form a robust CV tracker for ARTIST, which has accurate registration and low jitter. The tracker is designed to search for multiple patches in the time between video frames. This enables the tracker to operate over large environments, using distinctive planar patches as tracking beacons. The GPS and inertial system are used to reduce the number of patches to search for, enabling ARTIST to operate over a wide area. Other improvements include the compensation for lens radial distortion, illumination changes and tolerance to partial occlusion. This results in high precision and good robustness to external interferences. Grid-based initialization automatically selects good regions within the area designated by the user for tracking. This tracker has been tested indoors and outdoors.

Initial difficulties with the calibration of the inertial sensors led to the development

## 1. INTRODUCTION

of novel methods to calibrate sensor errors, particularly the gyroscope scale factors and axes misalignments, without the use of external equipment. This removes the need for high precision calibration machines or standards. All that is required for calibration is the local gravity and the inertial sensor themselves. Experiments show that it is possible to perform the calibration by holding the inertial measurement unit in the hand, and moving it randomly.

Research work on GPS resulted in the development a novel differential GPS carrier phase method, based on the Differential Single Difference (DSD), suitable for outdoor AR applications. It differs from existing differential GPS methods commonly used in geo-surveying as it avoids solving for integer ambiguity, which is the main difficulty in real-time precision GPS systems. The proposed system achieves an accuracy of 10-20 cm using low cost GPS modules. This is the accuracy level of current real-time systems but without the need for heavy computation to resolve the integer ambiguity. This accuracy is significantly improved as compared to standalone GPS, which has several meters of error. The jitter is significantly reduced as well, but the proposed method suffers from a drift of around 1 millimeter per second. This drift is highly linear within a period of several minutes and can be removed using linear regression.

All three tracking components of ARTIST are integrated onto a hybrid tracker for outdoor environments. Two solutions are presented here. The first is a loosely coupled configuration that allows for greater hardware flexibility. For all environments, the inertial system provides accurate orientation with respect to the local level earths surface. For outdoor environments, this orientation information can be combined with

GPS positioning to form a coarse 6DOF tracker. The typical usage scenario is, when the user is outdoors, the standalone GPS provides a position that has an accuracy of $10-20$ meters. The system uses this coarse position and the orientation from the inertial system to define a set of potential patches for initializing the CV tracker. After initialization, augmentation is mainly reliant on the ARTIST CV component. GPS and inertial are used when CV tracking fails, or to acquire new patches to track. This loosely coupled configuration is possible as the CV tracker is sufficiently robust for independent operation over extended period of time. It also demonstrates the applicability of the ARTIST framework to commodity hardware, such as mobile phones, where low-level data, such as GPS carrier phase measurement, required for tight coupling of components is not available.

The second hybrid solution is a tightly coupled configuration, where the GPS DSD position tracking and IMU orientation are combined using Kalman filters with highly accurate CV tracking data. Two separate filters are used for position and orientation respectively. The position filter combines the CV and GPS DSD positions using a constant velocity system model. The orientation filter combines the CV and IMU orientations using the Multiplicative Extended Kalman Filter (MEKF). The acceleration measurements from the IMU were not used for integration to velocity and position as they contained significant random errors. However, they are used for detecting static states, which allows the filters to determine the values of part of the system state.

## 1. INTRODUCTION

### 1.3 Thesis Organization

The thesis is organized as follows. Chapter 2 present the overview of the design of the ARTIST hybrid tracker, including design considerations and the Kalman Filters for combining the various sensor information. The next three chapters focus on each of the three sensor types, namely inertial (chapter 3), GPS (chapter 4) and CV (chapter 5). These chapters consider each sensor in isolation from the others, while Chapter 6 presents the combination of the three trackers to form the ARTIST tracker, by implementing the ideas presented in Chapter 2. The thesis is concluded in chapter 7, with analysis on the ARTIST and possible future works to extend it

## 2

## Design of ARTIST hybrid tracker for mobile Augmented Reality

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

In this chapter, the motivation and overview of the design of the Augmented Reality TRackIng SysTem (ARTIST) is described. This is meant to provide better appreciation of the rationale behind the choices made in the design of ARTIST from the perspective of prior works and requirements imposed by Augmented Reality applications. In particular, the design requirements of high accuracy and robustness, as well as low latency and jitter, are presented in relation to fidelity of graphics required for registration, to convince the human user that the virtual object is indeed part of the real scene. In addition, this chapter also presents the mathematical formulation of Kalman Filtering for fusing the sensor data from inertial, GPS and Computer Vision (CV). In summary, the chapter start with a survey of AR and tracking in Section 2.1, followed by requirements for registering AR graphics in Section 2.2, leading to the design rationales in Section 2.3 and finally the Kalman Filter in Section 2.4. Hopefully, presenting the overall design early will bring coherence to the seemingly disjointed chapters on the three individual components of ARTIST. The actual implementation and test results will be delayed to Chapter 6.

### 2.1 Background Information

Augmented Reality (AR) is a relatively new research area that has been developed as a variation of the much more established field of Virtual Reality (VR). An often-cited set of overview of the requirements, design, problems and applications of AR systems were presented by Azuma (1997) and Azuma et al. (2001). It is generally accepted that AR consists of (1) a combination of both the virtual and real worlds with (2) real-time
interactivity and (3) registration in the 3D space. This particular definition does not limit the display and tracking technologies used and emphasizes on the interactivity aspect of AR. A recent survey by Zhou et al. (2008) gives an indication of the variety of tracking systems, and interaction and display techniques being reported. From the trends presented, the main AR tracking methods continue to be CV based, or hybrids between CV and inertial. AR platforms are also becoming increasingly more mobile, with working demonstrations on mobile phones (Wagner et al., 2008b).

The majority of the research and application of AR has been focused on either enabling or utilizing AR as an interface. Bowman et al. (2004) presented AR as a 3D user interface (UI) that can form the basis of future ubiquitous computing platforms. Users can tap on computing resources at all locations even while on the move. This is possible with AR, as the user is able to perceive the dynamic real environment while operating the UI. The ability to operate computing resources and access digital information without having to switch between the real world and the UI is one of the main factors driving the adoption of AR in numerous applications. Some applications include medical, manufacturing (Ong and Nee, 2004), annotation and visualization (Vlahakis et al., 2002), robot path planning (Chong et al., 2007), entertainment, military heads up display (HUD), outdoor mobile AR, and collaborative AR. Among these, the HUD is the most established, after having been used by fighter pilots for decades, and demonstrated its effectiveness in reducing cockpit workload.

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

### 2.1.1 Augmented Reality Tracking

Among the various issues involved in the development of AR applications, accurate tracking and registration remain the most critical issues to be resolved (Azuma, 1997; Zhou et al., 2008). This is mainly driven by the human visual perceptual capabilities, which will be explored in detail in Section 2.2. High accuracy tracking for local-area applications has been well developed, in part due to the developments in VR research. Welch and Foxlin (2002) presented an overview of the various physical phenomena employed for tracking purposes. These include mechanical linkages, inertial, sound, light and magnetic sensors. The most notable tracker used for AR is the ARToolkit (2010), which requires only a low cost camera and easily printed square markers. With its low cost, ease of use and ease of software development, ARToolkit is perhaps the most commonly used tracker. However, it requires a line of sight to the marker and suffers from jitter under non-optimal lighting conditions.

For local-area applications, ARToolkit has enabled a great increase in research output for AR. This is because for the first time, there is a simple and effective way to experiment with new interfaces enabled by AR. However, the reliance of ARToolkit on markers renders it a less than satisfactory solution. Another example is the ArcheoGuide (Vlahakis et al., 2002), which uses Fourier-based 2D image registration to accurately augment missing parts of archeological sites. However, the method in ArcheoGuide limits the user to stand at several predetermined locations to view the augmented buildings. The works presented by Wagner et al. (2008a) and Wagner et al. (2008b) demonstrate recent advances in computational efficiency of tracking algorithms and processing ca-
pabilities of mobile phones. It is expected that the mobile phones will be the first AR platform to become popular among general users. This is due to the wide availability, low cost, mobility and ease of use of the mobile phone form factor.

There is an impetus to develop trackers that are not only marker-less, but also operate in environments which users normally move in, such as their work place, home and various locations that they visit. These environments are uncontrolled and dynamic, making it difficult to achieve the accuracy, robustness, jitter-free and latency requirements. Therefore, the significance of achieving the primary goal of wide-area precision tracking is to enable users to utilize AR in their normal operating environments. This would truly reveal the potential of AR as a new form of 3D interface (Bowman et al., 2004) for ubiquitous and mobile computing.

### 2.1.2 Mobile Hybrid Tracking Systems

There are several reported hybrid tracking systems used in research prototypes for mobile AR applications, which are designed to be worn by a user. These systems typically include a portable computer, trackers and a Head Mounted Display (HMD). The Touring Machine (Feiner et al., 1997) uses a GPS and compass for registering buildings. It is used for navigation and display of interesting information about buildings. Due to the limited accuracy of GPS (approximately 100 metres), the Touring machine is only suitable for coarse augmentation on building at large distances. The wearable AR kit presented by Ribo et al. (2002) used a laptop with 3D graphics capabilities and hybrid trackers. It was rather bulky due to the limitations of the technologies then. Furthermore, the tracker they developed would fail when there are insufficient image

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

features. A recent system by Peternier et al. (2006) used a PDA instead of a laptop and resulted in a total weight of less than half a kilogram.

Azuma (1993) as well as Welch and Foxlin (2002) highlighted that none of the current tracking technologies can effectively operate in unprepared environment. The most promising approach seems to be the use of a combination of sensing techniques, so as to compensate the weaknesses of one sensor with the strengths of another sensor. The combination of inertial and GPS (Farrel and Barth, 1999; Grewal et al., 2001; Jekeli, 2000), as well as those of inertial and marker-less CV (Foxlin and Naimark, 2003; Jiang et al., 2004; Kotake et al., 2005; Ribo et al., 2002; Yokokohji et al., 2000; You and Neumann, 2001; You et al., 1999), are some of the most commonly used approaches. This can be attributed to the fact that inertial, marker-less CV and GPS are practically source-less, i.e., they do not require a specialized emitters to work, as. Specifically, inertial tracking is truly source-less and can work independently in any environment. The GPS depends on the radio frequency emitting satellites to function. However, due to the way the system is designed, GPS can work anywhere on earth that has a clear view of the sky. An ideal marker-less CV system should work with any unmodified scene, depending only on natural scene structures. Therefore, these tracking systems do not have any inherent range limits that are usually associated with ultrasonic, magnetic or marker-based trackers. Furthermore, they have complementary strengths and weaknesses.

The combination of inertial and GPS is mainly used for navigation purposes, typically over fairly large distances. The fusion of the low frequency GPS data with the
high frequency inertial data is commonly achieved using Kalman Filters (Kalman, 1960; Maybeck, 1979), which optimally combine sets of noisy measurements. Such systems are robust and fast, but do not work well indoors, as the satellite signal strength is reduced when indoors. They only have the level of precision for annotating buildings but not the finer features. On the other hand, marker-less CV is typically based on natural features, such as points, lines and textures. Such algorithms are often able to provide precise translation and orientation information. However, they are not robust in outdoor environment. Unlike GPS, Kalman Filters are not typically used for fusing CV and inertial systems; rather, inertial information is used to constrain the CV algorithms to improve the robustness and processing speed.

The approach of combining all three tracking systems into a single hybrid tracker is presented in recent work from Reitmayr and Drummond (2006) and Kim et al. (2007). Their approaches are similar in the use of GPS and inertial to define an initial search space for the CV component to track buildings. Reitmayr and Drummond (2006) used textured 3D models with line tracking, while Kim et al. (2007) used online user 3D modeling with the aid of aerial photos to improve the robustness of tracking. Both systems represent a loose coupling of the three systems, which is more flexible on hardware choices, but depends heavily on the performance of the CV tracker. It is expected that further improvements in such hybrid trackers will be mainly derived from improvements in CV tracking, as well as tighter integration through sensor fusion and self-calibration. In particular, as CV is not expected to work well in areas with little image features, the improvements in the GPS and inertial trackers will allow continued

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

tracking in such areas. This is the approach taken in this research.

### 2.2 Requirements for registering graphics in Augmented Reality

The requirements for AR trackers are mainly driven by the visual perception capabilities of the human users. Although augmentation can be applied to other senses such as hearing and touch, the research here is primarily concerned with visual aspects of AR. This is because vision is the main sensory input for humans and arguably the most demanding on the precision of trackers. The main concern is to convince the user that the graphics augmented onto their reality are indeed parts of the real world. The requirements to create such an illusion depends on several factors. The ones considered are the location and appearance of the augmented object, the relative motion between the user and object, the AR setup, whether it is video or optical see-through, and the nature of the AR application. As this research is primarily on tracking, the main focus is on positional and motion accuracy and stability. However, it should be noted that other cues, such as occlusion and shading, can destroy the illusion. Although out of the scope of this research, they are discussed in the following sub-section to provide further areas of work for AR trackers.

### 2.2.1 Tracker spatial precision

The requirements imposed by the location and appearance of augmented objects are generally concerned with spatial precision of the tracker. The precision required is related to how well the human user can detect errors in registration. This in turn depends
on how far the augmented object is from the user, and whether there exist features on the surface that serves as references for detection mis-registration. Therefore, a near object with sharp, high contrast lines and features, requires the most precision, while a faraway, uniformly coloured building requires low precision. As CV operates on similar principles to human vision, such trackers works best in conditions where there are a lot of image features. This is the most probable reason for CV to become the dominant tracker in AR, as it is compatible with human vision. The spatial precision is generally described by the tracker's position and orientation accuracy. The resultant registration error induced by position and orientation errors are important at different distances (Foxlin, 2002). For nearby objects, typically within several metres, positional errors results in greater mis-registration, while at larger distances, orientation errors create greater offsets. In the related surveys on tracking by Foxlin (2002) and Welch and Foxlin (2002), the cited resolutions required are 1 mm for position, and $\frac{1}{10}$ of a degree for orientation. Azuma (1993) reported that humans are able to perceive minute misalignments as small as $\frac{1}{60}$ of a degree. This higher reported precision is probably only applicable to the scenario where the graphics is augmented on a flat surface with fine, high-contrast lines. These lines provide the guides for human subjects to easily determine mis-registration. These high spatial accuracy requirements will only be required for high precision augmentation applications using optical see-through displays.

The requirements imposed by relative motions are generally related to the latency and robustness. If there is significant latency, the augmented graphics will lag behind the real world and therefore appear detached, when there are large and high speed

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

relative motions. This is particularly important for optical see-through AR, where the real world is instantly visible. It is possible to determine the detection threshold and build trackers to meet the latency requirement. However, a better method would be to reach a latency value that is low enough for the prediction algorithms to be highly accurate. In this way, it may be possible to build AR systems with zero latency. On another note, trackers tend to fail when there are drastic changes, which can be caused by large, high-speed or highly irregular motions. The effects of tracker failure can be catastrophic and totally disrupt the AR experience. Therefore, the ideal tracker should be completely robust or only fails infrequently. However, this is not practical at this point of research, the main goal for ARTIST is to recover from failures as quickly as possible. Therefore, it is important to recover within the time frame where prediction is still effective. This is the main disadvantage of CV trackers, as it often fails abruptly and possibly completely.

As pointed earlier, the optical see-through AR places greater demands on spatial accuracy and latency requirements than video see-through. This is because, the human is able to perceive the real world at greater resolution and lower latency with optical seethrough. The earlier discussion is mainly applicable to the case of optical see-through. For video see-through, the requirements for spatial accuracy is determined greatly by the image resolution. Although users can detect sub-pixel errors, mis-registration of 1 to 2 pixels and degrees has not been found to disrupt the augmentation in practice. As the video is used for both tracking and display, the latency for CV is zero by default. Therefore, the main latency requirements for other sensors will have to be defined
with respect to the CV tracker. The robustness requirements for both types of AR is generally similar.

The type of applications places different requirements on the tracker. The most difficult case is the one where a 3D graphic is augments onto a nearby object with fine features, such as surgical application. The next type of application is annotation of text onto nearby object with fine features. Although, both types of applications requires accurate tracking, the registration of 3D objects is typically more demanding than pointing the annotations to specific points on the objects. This is because there is often some tolerance to the area the annotation can point to. Following this discussion, it can be surmised that annotation of faraway buildings places low demand on the tracker, which allowed the Touring Machine to succeed using highly inaccurate GPS and compass. As pointed out by Foxlin (2002), there is one more class of applications, where the virtual graphics are not attached to the real world, such as AR games. As the virtual objects are not anchored, trackers with low spatial accuracy would suffice. If the objects are animated, which is common in games, it is plausible that the motion can mask the effects of a limited amount of latency and jitter.

Finally, the jitter requirement is discussed, as it does not appear to be highly dependent on the factors described earlier. This is because jitter is temporal in nature, it is possible for the user to detect it by comparison with the previous position. However, it may be possible that fine features on the augmented object will provide the reference for the user to detect jitter that would be undetectable otherwise. In comparison to VR, jitter is less of a problem for AR, as it manifests mainly as the shaking of virtual

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

objects. Although this can affect the AR experience, it does not lead to simulator sickness as the surrounding world, which is real, is still visible and stable. This also applies for video see-through, except for the case of handheld AR, where the shaking of the camera can cause nausea, but it is not due to the tracker or graphics. The jitter in position is only important for augmentation of nearby objects. However, the shaking due to orientation jitter will increase with increasing distance to the augment object. The ARToolkit has high jitter for the normal to the marker plane, while maintaining good positional stability. In contrast, the ARTIST CV tracker uses a much larger number of pixel to reduce the jitter.

### 2.2.2 Occlusion and shading

For humans, both occlusion and shading cues are important for building up the 3D perception of the real world. However, the current focus of AR tracking is typically about finding the pose of the human head (or camera) with respect to a coordinate system. The handling of occlusion and shading is in general, predicated on the assumption that the graphics are already augmented at the correct position and appears correct to the user. Therefore, much of the focus is on solving the plain tracking problem for now.

In order to handle occlusion in AR , the depth information of the real environment is needed, so as to determine the portions of the virtual graphics to be shown or hidden. The depth information is typically obtained using three methods. The first is to use a prior 3D model of the real world, which can be purposed-built for AR, or available from other areas of work, such as engineering models, 3D medical scans and urban planning. This method cannot handle any changes of the real world, which are often inevitable
in uncontrolled environment. In contrast, the following two methods obtain the depth on-line to react to changes in real time. The second method determines whether the real world object being augmented is itself occluded. This is done at coarse level in this research, and Pilet et al. (2007) demonstrated fine boundary detection of the occlusion. This is similar to foreground background segmentation in videos. This method would only work if the virtual graphics augments only the surface of the real object. If the virtual component has depth, it becomes possible that the virtual part also occludes the real object in front of the augmented object. The third method is to build the 3D representations in real time, which is most general and difficult. It may be possible to use dense optical flow methods to obtain the depth, but to the best of the author's knowledge, none exist that are robust and runs in real time. It may be possible that low cost depth cameras that are beginning to appear at the time of writing may be used for local area imaging. As the depths obtained are of low resolution, it may be combined with normal cameras to achieve real time high resolution depth imaging. However, the correct display of virtual objects in the presence of occlusion is outside the scope of this work, this area of work was not pursued here. Occlusion is a highly dominant depth cue, this area of research is likely to attract greater interests when the plain tracking problem is satisfactorily solved for most AR applications.

In comparison to occlusion, the effects of shading is less detrimental. In normal circumstances, illumination is expected to be from above. Therefore in this research, the illumination is simply a light source that is above the user and shining toward the 3D model. In works related to the wide baseline matching by Lepetit and Fua (2006),

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

the reference image is used to collect approximate illumination conditions. In the work on real time augmentation of deformable objects, the illumination information is better approximated using a uniformly colored sphere, which reveals the intensity and direction of illumination (Pilet et al., 2008). However, as real-time photorealistic rendering is not currently feasible, it is unlikely that more accurate illumination information will improve realism further than above methods. However, when rendering does become more realistic, then shading become much more critical. This in turn requires solving the problem of extracting the unknown number of light sources in the environment. There is recent work by Lalonde et al. (2009) with combines a number of weak indicators in video streams to obtain a more robust illumination field. If the final goal of the AR application is to be indistinguishable from reality, then the geometry and reflectance properties of the real world are needed, in addition to illumination sources. On another note, the work by Klein and Murray (2008a) added blurring and various artifacts common to video captured by low cost camera to the generated 3 D graphics. This is done so that virtual objects appear less distinguishable from the video that they are augmenting on. Although not directly related to shading, this work represents an effort to improve the visual coherence between the AR graphics and real world it is augmenting.

### 2.3 Design rationales for ARTIST

The primary reason for selecting inertial sensing, GPS and CV, is to combine the strengths of each sensor to achieve the goals of ARTIST. One of these goals is to be
able to track over a large physical area with robustness, this necessitated the use of GPS and inertial sensing. However, ARTIST will also need to be able to accurately augment 3D virtual objects onto real ones at arm length distances. Therefore, a typical user work flow for ARTIST would be to first navigate to a location, several tens of metres away. Upon reaching the target location, the user would become interested in details and thus ARTIST will perform accurate augmentation. Therefore, ARTIST is required to handle changes in scale smoothly. The initial approach taken in this research was to start by improving the spatial accuracy of inertial measurement and GPS. In particular, inertial sensing is of higher priority as it can work in all environments, while GPS is limited to outdoor areas. Finally, CV will be used to meet the high accuracy requirements. This is reflected in the order of chapters in this thesis, starting with inertial sensing, GPS, CV and the implementation of hybrid tracking.

In this research, it is found that error characteristics of current microchip inertial sensors limit their application to orientation sensing. The inertial measurement unit can be relied on to provide accurate absolute orientation, to a fraction of a degree, with reference to the level ground when it is static. The accuracy will decrease when drifts accumulates during continuous motions, which generally last no more than several seconds for human motion sensing. For position sensing, the large random noise in accelerations measured during motion can cause large drifts in velocity and positions, rendering these sensors unsuitable, except for very short periods of time. In contrast, the differential GPS carrier phase method developed in this research can achieve centimetre level of accuracy with slow drift and little jitter. However, the method can

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

only measure the change in position from an initial point. Therefore, the tracker still requires accurate initial positions for augmentation in the reference coordinate frame.

This is solved by using CV to initialize ARTIST using feature rich planar surfaces as landmarks. When augmenting in featureless areas, tracking is performing primarily using differential GPS position and inertial orientation. In the course of this research, CV algorithms has improved in accuracy, robustness and scalability that it is possible to use it as the central tracker. This is demonstrated in this research using the loosely coupled version of ARTIST (Section 6.1.1), where only high level position from the GPS and orientation from the inertial system is used to limit the search set for initializing the CV tracker. However, current CV algorithms is not completely robust in all environments. This can occur in large open flat areas with uniform surface colour where there is little practical features. Highly regular and repetitive features, such as lines on glass facaded building, can often confuse CV algorithm by causing feature matching errors. Therefore, the tightly coupled configuration of ARTIST is also implemented and tested in Section 6.2.3.

### 2.4 Mathematical Framework for Hybrid Tracking

The mathematical framework for combining the measurements from the three components of ARTIST is based on the Kalman Filter (Kalman, 1960). The filter system is designed based on the required outputs, available sensor measurements, and sensor error characteristics. The required outputs for AR are the position, $\boldsymbol{p}_{\mathrm{A}}$, and orientation, $\boldsymbol{q}_{\mathrm{A}}$, which is parameterized using quaternion. Therefore, $\boldsymbol{p}_{\mathrm{A}}$ and $\boldsymbol{q}_{\mathrm{A}}$, are part of
the ARTIST Kalman Filter system state, and have the subscript, A, to differentiate them from parameters of the three tracker components. The available sensor measurements and error characteristics are those found in this research to be applicable. The measurements are used as observables in the Kalman Filter, while the error characteristics are used for modeling the state transitions They are introduced here and detailed descriptions can be found in the respective chapters.

For inertial measurement, the senor outputs are the acceleration, $\boldsymbol{a}_{\mathrm{IMU}}$, and angular velocity, $\boldsymbol{\omega}_{\text {IMU }}$. The main sensor error modeled is the gyroscope bias, $\boldsymbol{b}_{\mathrm{g}}$, which varies with time and cannot be compensated using calibration. For GPS, the main measurement considered is the Differential Single Difference position, $\Delta \boldsymbol{p}_{\mathrm{DSD}}$, which is the change in position in one time step, computed using the GPS carrier phase measurements from a stationary and a mobile GPS receivers. The derivation of $\Delta p_{\text {DSD }}$ is the result of this research on differential GPS positioning described in Section 4.3. The main systematic error for GPS is the slow drift in position. However, this error is not modeled in the filter, as the drift has low values and causes significantly lower errors than the random noise in DSD computations. For CV, the position, $\boldsymbol{p}_{\mathrm{CV}}$, and orientation, $\boldsymbol{q}_{\mathrm{CV}}$, are directly obtained. Error sources that are not explicitly modeled are considered as random noise sources. The final type of measurement is available when the tracker is stationary, which can be accurately detected using the quasi-static filter (Section 3.3.3)(Saxena et al., 2005) on accelerometer measurements. In this case, it is certain that the accelerations and velocities of the tracker are zero. The orientation with respect to the Earth's local level can also be accurately obtained when the

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

accelerometers are static.

The design consists of two simple filters, one for position and one for orientation. This design is chosen after it is found that the acceleration readings from MEMS based accelerometers contain significant transient errors during motion. These errors are further increased during the conversion to acceleration in navigation frame for integration to velocity and position, because the errors are coupled with orientation errors. As the accelerometer readings contribute little for position tracking, it is omitted from the filter, and is mainly used for quasi-static state detection and measurement of the gravity vector in static state. Therefore, unlike general kalman filters used for high end IMU, it is assumed that orientation errors do not couple significantly into position measurements and two smaller filters can be used instead. This simplification is also chosen for its low computational complexity and is found to be sufficient for this research. The two filters are presented in the following subsections.

As both filters are based on the Extended Kalman Filter, the generic equations are presented first. Let $\boldsymbol{x}_{s}$, and $\mathbf{P}_{s}$ be the system state vector and covariance matrix respectively. The associated noise vectors are omitted from the following equations for simplicity in presentation. The state transition function, $\mathrm{f}_{s}$, from time $t_{k}$ to $t_{k+1}$ is,

$$
\begin{equation*}
\boldsymbol{x}_{s(k+1 \mid k)}=\mathrm{f}_{s}\left(\boldsymbol{x}_{s(k \mid k)}\right) \tag{2.1}
\end{equation*}
$$

The covariance propagation equation is

$$
\begin{equation*}
\mathbf{P}_{s(k+1 \mid k)}=\mathbf{F}_{s(k)} \mathbf{P}_{s(k)} \mathbf{F}_{s(k)}^{\mathrm{T}}+\mathbf{Q}_{s(k)} \tag{2.2}
\end{equation*}
$$

where $\mathbf{F}_{s(k)}$ is the Jacobian of $\mathrm{f}_{s}$ with respect to $\boldsymbol{x}_{s}$, and $\mathbf{Q}_{s}$ is the system noise covari-
ance matrix.

The general form of the measurement equation of a sensor, $m$, is:

$$
\begin{equation*}
z_{m}=\mathrm{h}_{m}\left(x_{\mathrm{s}(\mathrm{k}+1 \mid \mathrm{k})}\right) \tag{2.3}
\end{equation*}
$$

where $h_{m}$ is the measurement function. The measurement residual equation is,

$$
\begin{equation*}
\boldsymbol{y}_{m}=\boldsymbol{z}_{m}-\mathrm{h}_{m}\left(\boldsymbol{x}_{\mathrm{s}(\mathrm{k}+1 \mid \mathrm{k})}\right) \tag{2.4}
\end{equation*}
$$

where $\boldsymbol{z}_{m}$ represents the measurement from the sensor, while $\mathrm{h}_{m}\left(\boldsymbol{x}_{\mathrm{s}(\mathrm{k}+1 \mid \mathrm{k})}\right)$ represents the estimate by the filter. The Kalman gain is computed using,

$$
\begin{equation*}
\mathbf{K}_{m}=\mathbf{P}_{s(k+1 \mid k)} \mathbf{H}_{m}^{\mathrm{T}}\left(\mathbf{H}_{m} \mathbf{P}_{s(k+1 \mid k)} \mathbf{H}_{m}{ }^{\mathrm{T}}+\mathbf{R}_{m}\right)^{-1} \tag{2.5}
\end{equation*}
$$

where the measurement matrix, $\mathbf{H}_{m}$, is the Jacobian of $\mathrm{h}_{m}$, and $\mathbf{R}_{m}$ is the sensor noise covariance matrix. The state vector and covariance matrix are updated using,

$$
\begin{align*}
\boldsymbol{x}_{s(k+1 \mid k+1)} & =\boldsymbol{x}_{s(k+1 \mid k)}+\mathbf{K}_{m} \boldsymbol{y}_{m}  \tag{2.6}\\
\mathbf{P}_{s(k+1 \mid k+1)} & =\left(\mathbf{I}-\mathbf{K}_{m} \mathbf{H}_{s}\right) \mathbf{P}_{s(k+1 \mid k)} \tag{2.7}
\end{align*}
$$

Both noise covariance matrices, $\mathbf{Q}_{s(k)}$, and $\mathbf{R}_{m}$, are diagonal matrices, $\sigma \mathbf{I}$, unless explicitly stated. This reflects the assumption that the errors in the state parameters and individual measurements are independent. Although this is not completely true, the assumption does not affect filter performance in practice, while simplifying the computation. In this research, the noise covariance matrices are tuned manually to allow the filter to converge.

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

### 2.4.1 Orientation Filter

The orientation filter is a Multiplicative Extended Kalman Filter (MEKF) (Markley, 2003), which combines the measurements from the IMU and CV. The system state, $\boldsymbol{x}_{\Omega}$, consists of the orientation error $\boldsymbol{a}_{\Omega}$ and the gyroscope bias, $\boldsymbol{b}_{\mathrm{g}}$. The orientation error, $\boldsymbol{a}_{\Omega}$, is modeled as a Gibbs vector of three small angular errors between the current estimated and true quaternions, as presented by Markley (2003). Therefore, the state vector is as follows:

$$
\begin{equation*}
x_{\Omega}=\binom{a_{\Omega}}{b_{\mathrm{g}}} \tag{2.8}
\end{equation*}
$$

Following the MEKF design, the orientation quaternion is updated outside the filter using the bias compensated angular velocities, $\boldsymbol{\omega}_{\mathrm{A}}=\boldsymbol{\omega}_{\mathrm{IMU}}-\boldsymbol{b}_{\mathrm{g}}$, and the second order quaternion integration method presented in Eq. 3.6. This updating step takes place at the same high constant rate as the gyroscope readings. Both the orientation error, $\boldsymbol{a}_{\Omega}$, and bias, $\boldsymbol{b}_{\mathrm{g}}$, are modeled as random processes. As such, the state propagation is simply $\boldsymbol{x}_{\mathrm{A}(k+1 \mid k)}=\boldsymbol{x}_{\mathrm{A}(k \mid k)}$. For the propagation of the state covariance matrix, $\mathbf{P}_{\Omega}$, the result by Markley (2003) is used.

$$
\mathbf{F}_{\Omega(k)}=\left(\begin{array}{cc}
-\left\lfloor\boldsymbol{\omega}_{\mathrm{A}} \mathrm{x}\right\rfloor & \mathbf{I}_{3 \times 3}  \tag{2.9}\\
\mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3}
\end{array}\right)
$$

where $\boldsymbol{\omega}_{\mathrm{A}}=\left[\omega_{x}, \omega_{y}, \omega_{z}\right]^{\mathrm{T}}$, and $\lfloor\boldsymbol{\omega} \mathrm{x}\rfloor=\left(\begin{array}{ccc}0 & -\omega_{z} & \omega_{y} \\ \omega_{z} & 0 & -\omega_{x} \\ -\omega_{y} & \omega_{x} & 0\end{array}\right)$ is the skew symmetric matrix.

As presented by Markley (2003), the orientation error, $\boldsymbol{a}_{\Omega}=\mathbf{0}$, at the start of every
propagation step. The orientation error obtained after each measurement, is integrated into $\boldsymbol{q}_{\mathrm{A}}$ using quaternion multiplication, hence the name MEKF. The error is reset to zero after each measurement update. The multiplicative step is achieved using the following equations, where $\otimes$ represents quaternion multiplication.

$$
\begin{align*}
\rho & =\binom{2}{a_{\Omega}} \otimes q_{\mathrm{A}}^{-}  \tag{2.10}\\
q_{\mathrm{A}}^{+} & =\frac{\rho}{|\rho|} \tag{2.11}
\end{align*}
$$

There are three measurements for this filter. The first is the vector error measurement using the earth magnetic and gravity vectors. The magnetic vector is available at the constant rate at which the magnetometer is read, while the gravity vector is only available during quasi-static state. The second measurement is the bias measurement, which are the gyroscope readings during quasi-static state. The third measurement is the quaternion obtained using CV, which is only available when at least one feature rich planar surface is in front of the camera. Therefore, measurements from each sensor is incorporated one at a time, in a manner similar to the Single Constraint At A Time (SCAAT) EKF implemented in the HiBall tracker by Welch and Bishop (1997).

For the vector error measurement, let $\boldsymbol{v}_{\mathrm{B}}$, be the vector measured by the magnetometer or accelerometer in the sensor frame. As the quaternion, $\boldsymbol{q}_{\mathrm{A}}$, maintains the orientation with respect to the North-East-Down $\boldsymbol{N E D}$ frame 4.2.4.2, the estimated $\hat{\mathbf{v}}_{\mathrm{B}}$, can be obtained by rotating the known $\boldsymbol{v}_{\mathrm{I}}$ in the $\boldsymbol{N} \boldsymbol{E} \boldsymbol{D}$ frame using $\boldsymbol{q}_{\mathrm{A}}$ and Eq. 3.7. As the orientation error, $\boldsymbol{a}_{\Omega}$, is assumed to have small values, the rotation from

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

$\hat{\mathbf{v}}_{\mathrm{B}}$ to $\boldsymbol{v}_{\mathrm{B}}$, can be approximated using skew symmetric matrix as follows.

$$
\begin{equation*}
\boldsymbol{v}_{\mathrm{B}} \approx\left(\mathbf{I}_{3 \times 3}-\left\lfloor\boldsymbol{a}_{\Omega \times} \times\right\rfloor \hat{\mathbf{v}}_{\mathrm{B}}\right. \tag{2.12}
\end{equation*}
$$

Therefore, the error measurement vector is derived as follows,

$$
\begin{aligned}
\boldsymbol{y}_{\mathrm{B}} & =\boldsymbol{v}_{\mathrm{B}}-\hat{\mathbf{v}}_{\mathrm{B}} \\
& =-\left\lfloor\boldsymbol{a}_{\Omega} \times\right\rfloor \hat{\mathbf{v}}_{\mathrm{B}} \\
& =\left\lfloor\hat{\mathbf{v}}_{\mathrm{B}} \times\right\rfloor \boldsymbol{a}_{\Omega}
\end{aligned}
$$

The last step is a property of the skew-symmetric matrix. Therefore, the measurement matrix, $\mathbf{H}_{\mathrm{B}}$ is

$$
\begin{equation*}
\mathbf{H}_{\mathrm{B}}=\left(\left\lfloor\hat{\mathbf{v}}_{\mathrm{B}}\right\rfloor \mathbf{0}_{3 \times 3}\right) \tag{2.13}
\end{equation*}
$$

For bias measurement, the biases are directly measured as the gyroscope readings in static state. The measurement equations are as follows.

$$
\begin{align*}
z_{\mathrm{b}} & =\boldsymbol{\omega}_{\mathrm{IMU}, \text { static }}  \tag{2.14}\\
\mathbf{H}_{\mathrm{b}} & =\left(\begin{array}{ll}
\mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3}
\end{array}\right) \tag{2.15}
\end{align*}
$$

For CV, the measured quaternion, $\boldsymbol{q}_{\mathrm{cv}}$, is in the $\boldsymbol{N E D}$ frame. This is obtained by quaternion multiplication of the quaternion of the planar surface with respect to the NED frame, and its current quaternion with respect to the camera. The quaternion of the planar surface with respect to $\boldsymbol{N E D}$ frame is in turn determined using the IMU orientation during the preparation phase. As the orientation errors, $\boldsymbol{a}_{\Omega}$, is represented by the Gibbs vector, the following is true.

$$
\begin{equation*}
0.5 * \boldsymbol{a}_{\Omega}=\boldsymbol{q}_{\mathrm{cv}} \otimes \boldsymbol{q}_{\mathrm{A}}^{-1} \tag{2.16}
\end{equation*}
$$

where $\boldsymbol{q}_{\mathrm{A}}^{-1}$ is the quaternion conjugate. As the current estimate of $\boldsymbol{a}_{\Omega}$ is zero,

$$
\begin{align*}
\boldsymbol{z}_{\mathrm{cv}} & =\boldsymbol{q}_{\mathrm{cv}} \otimes \boldsymbol{q}_{\mathrm{A}}^{-1}  \tag{2.17}\\
\mathbf{H}_{\mathrm{cv}} & =\left(\begin{array}{ll}
0.5 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3}
\end{array}\right) \tag{2.18}
\end{align*}
$$

### 2.4.2 Position Filter

The position filter combines the position measurements from the IMU, GPS, and CV. The system state, $\boldsymbol{x}_{\mathrm{p}}$, consists of the position, $\boldsymbol{p}_{\mathrm{A}}$, and, $\boldsymbol{v}_{\mathrm{A}}$, which are both in the $\boldsymbol{N E} \boldsymbol{D}$ frame. Therefore, the state vector is as follows:

$$
\begin{equation*}
x_{\mathrm{p}}=\binom{\boldsymbol{p}_{\mathrm{A}}}{\boldsymbol{v}_{\mathrm{A}}} \tag{2.19}
\end{equation*}
$$

The constant velocity model is used here for state propagation. Therefore,

$$
\begin{align*}
\boldsymbol{v}_{\mathrm{A}(k+1 \mid k)} & =\boldsymbol{v}_{\mathrm{A}(k \mid k)}+\boldsymbol{n}_{\mathrm{V}}  \tag{2.20}\\
\boldsymbol{p}_{\mathrm{A}(k+1 \mid k)} & =\boldsymbol{p}_{\mathrm{A}(k \mid k)}+\boldsymbol{v}_{\mathrm{A}(k \mid k)} \delta t \tag{2.21}
\end{align*}
$$

where $\boldsymbol{n}_{\mathrm{v}}$ is the system process noise, and $\delta t$, is the time since the last state estimate. The zero mean, Gaussian noise, $\boldsymbol{n}_{\mathrm{v}}$, is explicitly shown here to reflect the assumption that human motion is constant within a short period of time, and is perturbed by random changes in velocity. For the propagation of the state covariance matrix, the following state transition matrix is used.

$$
\mathbf{F}_{p(k)}=\left(\begin{array}{cc}
\mathbf{I}_{3 \times 3} & \delta t \mathbf{I}_{3 \times 3}  \tag{2.22}\\
\mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3}
\end{array}\right)
$$

As with the orientation filter, there are three measurements available. The first is the position measured by CV, the second is the change in position measured by GPS

## 2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

DSD, and the last is the static readings in quasi-static state. Unlike the orientation filter, the errors here are additive and can be simply incorporated into the filter.

For CV, the position, $\boldsymbol{p}_{\text {cv }}$ is obtained using the position of the planar surface in $\boldsymbol{N E D}, \boldsymbol{p}_{\text {NED }}$, its position relative to the camera, $\boldsymbol{p}_{\text {cam }}$, scaled by a factor $s_{c v}$, and rotated using the camera orientation in $\boldsymbol{N E D}$, which is also $\boldsymbol{q}_{\mathrm{A}}$, maintained by the orientation filter. Both $\boldsymbol{p}_{\text {NED }}$, and $s_{c v}$, for each planar surface are pre-determined during the preparation phase, by comparing the CV output with GPS readings.

$$
\begin{equation*}
\boldsymbol{p}_{\mathrm{cv}}=s_{c v} \mathbf{R}\left(\boldsymbol{q}_{\mathrm{A}}\right) \boldsymbol{p}_{\mathrm{cam}}+\boldsymbol{p}_{\mathrm{NED}} \tag{2.23}
\end{equation*}
$$

where $\mathbf{R}\left(\boldsymbol{q}_{\mathrm{A}}\right)$, is the rotation matrix represented by $\boldsymbol{q}_{\mathrm{A}}$, obtained using Eq. 3.7. Although the orientation errors can couple into the position measurement, it is treated as random noise, as the orientation filter errors do not appear to cause significant errors here. Therefore, the measurement equations are,

$$
\begin{align*}
\boldsymbol{y}_{\mathrm{cv}} & =\boldsymbol{p}_{\mathrm{cv}}-\mathbf{H}_{\mathrm{pcv}} \boldsymbol{x}_{\mathrm{p}}  \tag{2.24}\\
\mathbf{H}_{\mathrm{pcv}} & =\left(\begin{array}{ll}
\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3}
\end{array}\right) \tag{2.25}
\end{align*}
$$

For GPS DSD, the measurement is the change in position, $\Delta \boldsymbol{p}_{\mathrm{DSD}}$, from the previous position estimate, $\boldsymbol{p}_{\mathrm{DSD}}$, at which the previous GPS carrier phase measurement is made. As GPS measurements are not synchronous with other position measurements, $\boldsymbol{p}_{\mathrm{DSD}}$ is maintained separately outside the filter, which is the filter position estimate, $\boldsymbol{p}_{\mathrm{A}}$, at the time of the last GPS measurement. As such, the measurement equations are,

$$
\begin{align*}
\boldsymbol{y}_{\mathrm{DSD}} & =\boldsymbol{p}_{\mathrm{DSD}}+\Delta \boldsymbol{p}_{\mathrm{DSD}}-\mathbf{H}_{\mathrm{DSD}} \boldsymbol{x}_{\mathrm{p}}  \tag{2.26}\\
\mathbf{H}_{\mathrm{DSD}} & =\left(\begin{array}{ll}
\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3}
\end{array}\right) \tag{2.27}
\end{align*}
$$

Finally, by using the accelerometers to determine the quasi-static state, we can determine the instances where there is no change in position, and the velocity is zero. The measurement equations are,

$$
\begin{align*}
\boldsymbol{y}_{\mathrm{qS}} & =\binom{\boldsymbol{p}_{\mathrm{A}}}{\mathbf{0}_{3 \times 1}}-\mathbf{H}_{\mathrm{qs}} \boldsymbol{x}_{\mathrm{p}}  \tag{2.28}\\
\mathbf{H}_{\mathrm{qS}} & =\mathbf{I}_{6 \times 6} \tag{2.29}
\end{align*}
$$

This concludes the design of the hybrid tracking system. The experimental results using the above filter is presented in Section 6.2.3.
2. DESIGN OF ARTIST HYBRID TRACKER FOR MOBILE AUGMENTED REALITY

## 3

## Inertial sensors: Calibration for orientation sensing

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

### 3.1 Introduction

Inertial measurement is the use of the laws of classical Newtonian mechanics to measure the motion of a body. Due to the sensing mechanisms available, the linear acceleration and angular velocity are the quantities measured. The linear acceleration can be accurately measured using accelerometers. By measuring the linear acceleration acting on a mass, the velocity can be obtained by integration, which can be further integrated to give the position. The angular velocity is measured using gyroscopes, which can be integrated to give the orientation. Theoretically, these steps give the displacement from the initial point, and the orientation with respect to the reference frame. As both accelerometers and gyroscopes measure the inertial quantities, they are completely selfcontained and do not require external signals. In contrast, positioning systems such as GPS and most VR trackers require external emitting sources, such as satellites and beacons transmitting radio, infra-red and ultrasound signals, to operate. Therefore, highly accurate instruments have been used for space, submarine and aviation navigation, where there is no available aiding signal. However, they are both too large and expensive for consideration in this research.

In this research work, silicon-based Micro-Electro-Mechanical System (MEMS) sensors are used. They are highly portable with small size, low weight and low power consumption. Furthermore, they have low latency and jitter, and high robustness to external interferences. One of the main problems of using MEMS sensors is the sensitivity to temperature changes. This is due to the underlying material properties of
the silicon substrate that the sensors are built from. This in turn affects the sensor bias and drift, which results in orientation errors and position errors. This is presented in detail in section 3.2.8. Therefore, with current MEMS sensors error characteristics, they are not suitable for tracking positions.

However, the systematic sensor errors can be well mitigated with sensor calibration. For the sensors used in this research, they can be calibrated to function well as independent orientation sensors. Thus, inertial measurement can provide independent and robust orientation information with respect to the local Earth level surface. When combined with position trackers, such as GPS and CV, inertial measurements can be used to reduce jitter and latency. As mechanization equations are well developed in the navigation fields, this research focuses on sensor calibration methods that do not require the use of external equipments.

### 3.2 Background

### 3.2.1 MEMS Accelerometer

The basic design of an accelerometer is essentially a suspended proof mass. By measuring the forces acting on this known mass, the acceleration can in turn be measured. A mass suspended with springs functions as a simple accelerometer. This is shown in Figure 3.1. The displacement gives a measure of the acceleration experienced by the proof mass.

MEMS-based accelerometers are miniature versions that are machined onto silicon using the same technique for manufacturing computer chips. Numerous designs ex-

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING



Acceleration

Figure 3.1: A simple accelerometer.
ist. For example, some designs use capacitance or piezoelectric effect to measure the displacement. Currently, tri-axial accelerometers packaged as single chips less than one-centimeter square are commonly available. Conceptually, it consists of a single proof mass suspended by three sets of orthogonal springs, allowing measurement in the three axes.

### 3.2.2 MEMS Gyroscope

A conventional gyroscope consists of a spinning mass suspended using gimbals. The conservation of angular momentum keeps the gyroscope pointing in the same direction in inertial space, and this allows for measurement of orientation as the body rotates. In contrast, MEMS gyroscopes are vibratory sensors that use a different phenomenon known as the Coriolis Effect (Titterton and Weston, 1997). They measure the angular rate instead of orientation. A structure is made to vibrate along a specific plane. Rotation about an axis orthogonal to this plane induces the proportional Coriolis acceleration, which moves the vibratory structure as in the case of accelerometers, and the angular rate can be measured. The structure used may be shaped like a tuning fork, a drum or a disc. MEMS gyroscopes made using silicon and quartz are commonly
available, and the silicon-based gyroscopes are used in this research work.

### 3.2.3 Strapdown Inertial Measurement Units

An Inertial Measurement Unit (IMU) consists of an array of accelerometers and gyroscopes for measuring linear and angular motions. The operating principles and usage for navigation can be found in references by Farrel and Barth (1999); Grewal et al. (2001); Jekeli (2000). The development of the highly precise inertial sensors has been driven by the military, space and aviation industries. This is because inertial navigation is the only practical self-contained tracking system that can operate in all environments. In the early systems, the inertial sensors move independently of the body of the vehicle using a gimbaled system, and remain stationary with respect to the inertial frame through the use of the gyroscopes. Current systems typically utilize the strapdown configuration, where the sensors are rigidly attached to the body of the vehicle and are thus non-stationary in the inertial frame. The resulting systems are lighter, mechanically less complex and more robust. However, the computation of the orientation and position is more involved than the gimbaled system, and is presented in detail by Savage (1998a,b), as well as Titterton and Weston (1997).

### 3.2.4 Usage in Virtual and Augmented Reality Applications

Due to the large size of early inertial systems, they were limited to large vehicles, such as submarines and aircrafts. Recently, inertial sensors based on MEMS are readily available in the millimeter size range. This resulted in the widespread use for VR and AR applications. As there are no moving components, these IMUs are of the strapdown

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

configuration. Foxlin et al. (1998) presented an example of using an IMU to track the head of a user. However, critical performance parameters, such as the sensor zero bias drift, are several orders of magnitude worse than those required for independent position tracking. This results in rapid and unbounded position drifts of MEMS-based IMUs. Therefore, most reported uses of inertial sensors for AR applications are in combination with CV trackers in the form of hybrid trackers. They are presented in section 2.1.2.

### 3.2.5 Coordinate Frames

An IMU generally consists of orthogonally mounted accelerometers and gyroscopes. In this work, the IMU is a conventional, strapdown 6DOF tracker (Titterton and Weston, 1997). It consists of a platform with a three-axis, right-handed coordinate system, known as the sensor frame $(\boldsymbol{S})$, associated with it. Three accelerometers and three gyroscopes are rigidly mounted, such that the sensitive axes of one accelerometer and one gyroscope are aligned along each of the three axes of the platform. This is shown in Figure 3.2.


Figure 3.2: IMU sensor setup.

The North-East-Down frame (NED), which is similar to $\boldsymbol{S}$, is another useful frame of reference. The $\boldsymbol{N E D}$ is the common frame of reference between the inertial system and the GPS (Section 4.2.4). In this frame, the $X$-axis and $Y$-axis are parallel to the earth's surface, and aligned to the North and East respectively, while the $Z$-axis points downwards.

### 3.2.6 Strapdown IMU Computations

In a traditional non-strapdown IMU, the platform moves independently of the vehicle such that $\boldsymbol{S}$ coincides with $\boldsymbol{N E D}$ (Farrel and Barth, 1999; Grewal et al., 2001; Jekeli, 2000). In contrast, for an IMU in the strapdown configuration, the platform is strapped to the structure of the vehicle and the two frames do not coincide. This necessitates computations (Jekeli, 2000; Savage, 1998a; Titterton and Weston, 1997) to first determine the transformation from $\boldsymbol{S}$ to $\boldsymbol{N E D}$, so as to transform the accelerations measured in $\boldsymbol{S}$ to the accelerations in $\boldsymbol{N E D}$, as the IMU navigation requires acceleration measurements in $N E D$ and not $S$. As the origins of the two frames coincide, only pure rotations are required to align the axes of the two frames.

This section describes the process of obtaining the orientation of the IMU with respect to $\boldsymbol{N E D}$. Specifically, this means obtaining the rotation matrix $\mathbf{R}_{\mathrm{S}}^{\mathbb{N}}$ for the transformation from $S$ to $\boldsymbol{N E D}$. The goal of computing $\mathbf{R}_{\mathrm{S}}^{\mathrm{N}}$ is to obtain the velocity vector $\boldsymbol{v}_{\mathrm{N}}$ and position vector $\boldsymbol{p}_{\mathrm{N}}$ in $\boldsymbol{N E} \boldsymbol{D}$ using the following procedures. Let $\boldsymbol{a}_{\mathrm{S}}$ and $\boldsymbol{a}_{\mathrm{N}}$ be the acceleration vectors measured in $\boldsymbol{S}$ and $\boldsymbol{N E D}$ respectively. Let $s \boldsymbol{f}_{\mathrm{N}}$ be the

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

specific force acting in $\boldsymbol{N E D}$. Therefore,

$$
\begin{align*}
\boldsymbol{a}_{\mathrm{N}} & =\mathbf{R}_{\mathrm{S}}^{\mathrm{N}} \boldsymbol{a}_{\mathrm{S}}  \tag{3.1}\\
\boldsymbol{s} \boldsymbol{f}_{\mathrm{N}} & =\boldsymbol{a}_{\mathrm{N}}-\boldsymbol{g}_{\mathrm{N}} \tag{3.2}
\end{align*}
$$

where $\boldsymbol{g}_{\mathrm{N}}$ is Earth's gravity in $\boldsymbol{N E D} . \boldsymbol{v}_{\mathrm{N}}$ and $\boldsymbol{p}_{\mathrm{N}}$ are computed by,

$$
\begin{align*}
& \boldsymbol{v}_{\mathrm{N}}=\int s f_{\mathrm{N}}-\boldsymbol{v}_{\text {initial }}  \tag{3.4}\\
& \boldsymbol{p}_{\mathrm{N}}=\int \boldsymbol{v}_{\mathrm{N}}-\boldsymbol{p}_{\text {initial }} \tag{3.5}
\end{align*}
$$

In this research, $\mathbf{R}_{\mathrm{S}}^{\mathrm{N}}$ is parameterized using a quaternion, $\boldsymbol{q}$. The second order numerical integration algorithm in section 4.2.3.1.1 Jekeli (2000) is used. At each time step $k$, the current quaternion $\boldsymbol{q}_{k}$ is related to the quaternion $\boldsymbol{q}_{k-1}$ of the previous time step $k-1$, using Eq. 3.6.

$$
\begin{equation*}
\boldsymbol{q}_{k}=\left(\cos (0.5|\delta \boldsymbol{b}|) \mathbf{I}_{4}+\frac{1}{|\delta \boldsymbol{b}|} \sin (0.5|\delta \boldsymbol{b}|) \mathbf{B}\right) \boldsymbol{q}_{k-1} \tag{3.6}
\end{equation*}
$$

where $\delta \boldsymbol{b}=\boldsymbol{w}_{k} \Delta t=\left(\begin{array}{c}\delta b_{x} \\ \delta b_{y} \\ \delta b_{z}\end{array}\right),|\delta \boldsymbol{b}|=\sqrt{\left(\delta b_{x}\right)^{2}+\left(\delta b_{y}\right)^{2}+\left(\delta b_{z}\right)^{2}}$, and $\mathbf{B}=\left(\begin{array}{cccc}0 & \delta b_{x} & \delta b_{y} & \delta b_{z} \\ -\delta b_{x} & 0 & \delta b_{z} & -\delta b_{y} \\ -\delta b_{y} & -\delta b_{z} & 0 & \delta b_{x} \\ -\delta b_{z} & \delta b_{y} & -\delta b_{x} & 0\end{array}\right)$,
where $\Delta t$ is the time interval between the time steps and $\mathbf{I}_{4}$ is a $4 \times 4$ Identity matrix.
The vector $\delta \boldsymbol{b}$ is obtained from the angular velocity $\boldsymbol{w}_{k}$, which is measured using the gyroscopes. This is used in Eq. 3.6 to compute the current quaternion $\boldsymbol{q}_{k}$ from the previous quaternion $\boldsymbol{q}_{k-1}$.

Using the initial quaternion $\boldsymbol{q}_{0}$ and the sequence of $\boldsymbol{w}_{k}$, from $k=0$ to $k=n$, the quaternion, $\boldsymbol{q}_{n}=(a, b, c, d)^{\mathrm{T}}$ at time step $k=n$ can be obtained. The rotation matrix $\mathbf{R}_{\mathrm{S}}^{\mathrm{N}}$ is obtained from $\boldsymbol{q}_{n}$ using Eq. 3.7.

$$
\mathbf{R}_{\mathrm{S}}^{\mathrm{N}}=\left(\begin{array}{ccc}
a^{2}+b^{2}-c^{2}-d^{2} & 2(b c+a d) & 2(b d-a c)  \tag{3.7}\\
2(b c-a d) & a^{2}-b^{2}+c^{2}-d^{2} & 2(c d+a b) \\
2(b d+a c) & 2(c d-a b) & a^{2}-b^{2}-c^{2}+d^{2}
\end{array}\right)
$$

### 3.2.7 Sensor Calibration

High-end inertial sensor calibration and error modeling are well-established fields (Titterton and Weston, 1997). The basic idea is to compare the sensor output with known values generated using calibration instruments. Researchers have also used optical trackers (Kim and Golnaraghi, 2004) to calibrate low-end inertial sensors for less demanding applications. In previous methods, the main difficulties are in the generation of accurate external calibration values, as well as precisely mounting and moving the IMU. These calibration procedures often require costly, specialized and high precision equipment, which may not be available to researchers who are seeking to use the IMU for basic orientation measurements. Furthermore, the low cost sensors do not justify the high cost of the calibration instruments. Therefore, calibration methods that can be carried out by the users with minimum amount of equipment are desired.

Calibration methods that can be carried out by the users with minimum amount of equipment typically fall into two classes. The first class uses Kalman filters and carefully designed error models with special maneuvers to expose the various model parameters (Foxlin and Naimark, 2003; Grewal et al., 1991). The filters developed are relatively complex as the model parameters are difficult to separate and the maneuvering needs to be fairly precise. Another class of calibration methods (Lötters et al., 1998; Saxena et al., 2005; Skog and Händel, 2006), utilizes the property that the magnitude

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

of the acceleration measured using a static tri-axial accelerometer is always exactly 1 g , regardless of the orientation.

The work by Lötters et al. (1998) provided a method to calibrate the biases and scale factors of a tri-axial accelerometer using the gravity vector as a stable and accurate standard. The accelerometer can be fully calibrated by placing it in various orientations without the need to be precise. They illustrated that it is possible to eliminate the need for physical precision by using mathematical constraints. This idea is further extended by Skog and Händel (2006) to calibrate for axis misalignment and the gyroscope. As their method for calibrating the gyroscope requires the use of an accurate turn rate table, it fails to achieve complete independence from external equipment, as with the accelerometer. A recent work on IMU and GPS integration by Syed et al. (2007) utilized 26 positions to calibrate the tri-axial accelerometer. The rotation of the Earth and a turntable are used to calibrate the gyroscope. The requirement of the use of external equipment is removed in this research (Fong et al., 2008a), and the details are described in detail in Section 3.3.

### 3.2.8 Sensor Performance and Error Characteristics

The types of applications that an inertial sensor can be used for depend largely on its error characteristic and accuracy. Due to the use of integration for inertial measurements, the errors accumulate very quickly over time. For orientation tracking using rate gyroscope, the error grows linearly with time. For example, if there is a constant minute error of $0.01^{\circ} / \mathrm{sec}$ in angular velocity measurement, the error in the integrated orientation becomes $0.6^{\circ}$ in one minute and $36^{\circ}$ in an hour. Due to the double in-
tegration, the rate of error growth is proportional to the square of the elapsed time for position tracking using accelerometers. For example, consider the case where the constant error in acceleration measurement is $0.01 \mathrm{~m} / \mathrm{sec}^{2}$. After one minute, the errors in velocity from single integration and position from double integration become $0.6 \mathrm{~m} / \mathrm{sec}$ and $0.5 \times 0.01 \times 60^{2}=18 \mathrm{~m}$ respectively. The problem is further complicated when the error is not constant and varies randomly with time. Such dynamic errors cannot be compensated by calibration. Thus, the dynamic error characteristics become more critical than the static accuracy.

The main errors for MEMS accelerometers are the zero bias, scale factor errors, cross-axis sensitivity. Zero bias error is the non-zero error reading that the sensor gives when it is at the zero position. For accelerometers, this bias varies predictably with temperature and shows little random drifts. Scale factor error occurs when the scale used for converting the electrical voltage output from the sensor to physical measurement is different from the specified value. This error can also be different along the full scale of measurement, resulting in non-linearity. Finally, the sensor can be sensitive to acceleration orthogonal to its sensitive axis. For MEMS gyroscopes, errors such as zero bias, scale factor and sensitivity to cross axis rotation are present. In addition, there are earth's gravity, or $g$-dependent bias and random zero bias drift. The $g$-dependent bias is the zero bias varying with the acceleration acting on the gyroscope. The random zero bias is the random walk of the zero bias, which is unpredictable as it changes in random steps. This random bias error is the primary factor limiting the use of MEMS-based IMU.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

In this research work, most sensor errors are systematic and it is possible to calibrate and compensate for these errors. This is the focus for the inertial measurement aspect in this research work. However, the significant random bias drift of the gyroscope prevents the use of inertial sensing for position tracking. This is because for position tracking, accurate orientation of the accelerometers is required so that the acceleration can be transformed from the body frame to the navigation frame. However, the gyroscope bias results in errors in the orientation, and thus the acceleration in the navigation frame. As an illustration, the bias of ring laser gyroscopes used for long term navigation varies by $0.001^{\circ}$ in an hour, while MEMS gyroscopes commonly available have a drift of more than a thousand degrees in an hour (Titterton and Weston, 1997). This represents a difference of at least seven orders of magnitude. It is possible to use the accelerometers and magnetometer to directly measure the earths gravity and magnetic field respectively. This allows the orientation to be obtained in low dynamics conditions. As the human user rarely maintains constant motions, the gyroscopes can maintain the orientation satisfactorily in these short durations of high dynamics. Therefore, inertial sensing can be used as an independent orientation tracker.

### 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment

Most calibration procedures start with the development of error models for each type of sensor. This is followed by fitting the models to the sets of data collected for calibration, in order to obtain the error compensation parameters. Error models are generally available for traditional high-end inertial sensors (Farrel and Barth, 1999; Grewal et al.,

# 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment 

2001; Jekeli, 2000; Titterton and Weston, 1997). Compared to high-end sensors, current MEMS-based sensors have low signal to noise ratios. Bias, scale factor and temperature effects dominate the errors. Therefore, earlier sophisticated error models cannot be directly applied. Barshan and Durrant-Whyte (1995) demonstrated an early method to fit error models of gyroscope bias to early solid-state sensors. In the following, sensor error models and proposed methods to calibrate the inertial sensors without the use of external equipment are presented.

This section presents published methods (Fong et al., 2008a) that were developed during the course of this PhD research. They are designed to calibrate and compensate for non-zero biases, non-unit scale factors, axis misalignments and cross-axis sensitivities of MEMS-based IMU. The methods depend on the Earths gravity as a stable physical calibration standard. Specifically, the calibration of gyroscopes is significantly improved by comparing the difference in the outputs of the static accelerometer and the IMU orientation integration algorithm after arbitrary motions. The derived property and the proposed cost function allow the gyroscopes to be calibrated without external equipment. A custom-made prototype IMU is used to demonstrate the effectiveness of the proposed methods. Two types of calibration data are collected. The first type is carefully obtained using prescribed motions. The second type is less rigorously collected from the IMU when it is mounted on the head of a user or hand held with brief random movements. With calibration, the observed average static angular error is less than a quarter of a degree and the dynamic angular error is reduced by a factor of two to five.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

### 3.3.1 Motivation

Although calibrated IMUs are commercially available, it can be advantageous to build custom IMUs, e.g., to achieve smaller sizes and better ergonomics, to take timely advantage of newer, higher performance sensors, to circumvent the limitations of commercial units or to reduce costs. Outside the inertial navigation field, calibration can be challenging due to the lack of certified calibration equipment. The purpose of the proposed methods is to demonstrate simple, yet effective accelerometer and gyroscope calibration. This is to improve the accuracy of the orientation measured by the IMU.

Traditional calibration equipments cost many times more than MEMS-based sensors. Therefore, it is not economical to procure such equipments to quantify the errors of these sensors, so that they can meet the requirements of human scale inertial measurement. This forms the motivation to develop calibration methods that do not rely on high physical precision to fully exploit the available accuracy of current MEMSbased sensors. As an illustration, the maximum zero bias and sensitivity of a MEMS accelerometer at the time of writing are 0.1 g and 0.001 g respectively. One way to directly measure the zero bias of an accelerometer is to mount it with its axis perpendicular to the Earths gravity vector, such that a zero reading can be expected. In such a position, the accelerometer can detect deviations as small as $\sin ^{-1}(0.001)=0.06^{\circ}$ from the gravity vector. Therefore, to fully exploit the sensitivity of the accelerometer so as to accurately determine the zero bias, the combined tolerance of the test equipment and the mounting should be tighter than $0.01^{\circ}$, with respect to the gravity vector.

### 3.3.2 Development of the Methods

The Earth's local gravity vector is used as the physical standard for calibrating the IMU. It is readily available and is a very stable quantity. A tri-axial accelerometer is calibrated using the following property:
(p1): The magnitude of the static acceleration measured must equal that of the gravity (Lötters et al., 1998).

This is a tri-axial orthogonal constraint, where values measured on each axis are not independent. For a tri-axial gyroscopic system, the proposed property is:
( $\mathbf{p 1} \mathbf{)}$ : The gravity vector measured using a static tri-axial accelerometer must equal the gravity vector computed using the IMU orientation integration algorithm, which in turn uses the angular velocities measured using the gyroscopes.

This property holds whenever the IMU is static after arbitrary motions. Both properties ( $\mathbf{p 1 )}$ and ( $\mathbf{p 2 )}$ impose the physical and mathematical constraints on the sensor outputs, which are used to calibrate the sensor errors. As precise motions and externally generated calibration standards are not required, it is possible to calibrate the IMU by hand holding it and moving it for a few minutes, as shown in section 3.3.5.3. This greatly reduces the time and difficulty involved in calibrating the IMU, especially for the gyroscope.

Lötters et al. (1998) proposed the use of (p1) to calibrate accelerometer biases and scale factors. Their method does not require precise inclinations and the model parameters are fitted using robust estimation techniques. This was extended by Skog and Händel (2006) to include sensor axis misalignment. In this case, the sensor axes in

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

the sensor frame $(\boldsymbol{S})$ are now non-orthogonal. As the misalignment is of a small angle, the orthogonalization of the axes can be performed linearly. Let the orthogonalized axes form the platform frame $(\boldsymbol{P})$. Eq. 3.8 shows the tri-axial accelerometer error model proposed by Skog and Händel (2006) to convert the $k$-th acceleration vector $\boldsymbol{a}_{\mathrm{S}, k}$ measured in $\boldsymbol{S}$, to $\boldsymbol{a}_{\mathrm{P}, k}$ measured in $\boldsymbol{P}$.

$$
\begin{equation*}
\boldsymbol{a}_{\mathrm{P}, k}=\mathbf{M S}\left(\boldsymbol{a}_{\mathrm{S}, k}-\boldsymbol{b}_{\mathrm{a}}\right) \tag{3.8}
\end{equation*}
$$

The bias vector is $\boldsymbol{b}_{\mathbf{a}}$, the misalignment matrix is $\mathbf{M}=\left(\begin{array}{ccc}1 & -\alpha_{y z} & \alpha_{z y} \\ 0 & 1 & -\alpha_{z x} \\ 0 & 0 & 1\end{array}\right)$, and the scale matrix is $\mathbf{S}=\left(\begin{array}{ccc}s_{x x} & 0 & 0 \\ 0 & s_{y y} & 0 \\ 0 & 0 & s_{z z}\end{array}\right) \cdot \alpha_{i j}$ is the small rotation of the $i$-th axis of the sensor about the $j$-th axis in $\boldsymbol{P}$, in order to align with the $i$-th axis in $\boldsymbol{P}$. The misalignments of the axes are illustrated in Figure 3.3. Skog and Händel (2006) proposed a similar model for calibrating the gyroscopes, but the derived cost function would require a turntable with turn rates accurate to within $0.1^{\circ}$ per second. Therefore, their gyroscope calibration method is not independent of external equipment. In this research, ( $\mathbf{p} 2$ ) is proposed to eliminate the use of turntables, or any other specialized equipment for calibrating the gyroscope.

### 3.3.2.1 Tri-axial accelerometer error model

In this research, the model in Eq. 3.8 is improved by considering the cross-axis sensitivities, which can be up to five percent of the full measurement scale in practical MEMS accelerometers (Titterton and Weston, 1997). $\mathbf{S}$ is modified as $\mathbf{S}^{*}=\left(\begin{array}{lll}s_{x x} & s_{x y} & s_{x z} \\ s_{y x} & s_{y y} & s_{y z} \\ s_{z x} & s_{z y} & y_{z z}\end{array}\right)$, where $s_{i j}$ is the sensitivity of the $i$-th axis of the accelerometer to the accelerations in


Figure 3.3: The misalignment of sensor axes from the ideal orthogonal configuration.
the $j$-th axis. Ideally, $\mathbf{S}^{*}$ is an Identity matrix, meaning that there is no scaling error along the axis, and the sensor is not sensitive to cross-axis acceleration.

As the effects of the minor cross-axis sensitivity and the sensor misalignment are similar, and there is no requirement to obtain them as separate quantities, $\mathbf{M}$ is multiplied with $\mathbf{S}^{*}$ to give matrix $\mathbf{E}$ in Eq. 3.9.

$$
\mathbf{E}=\left(\begin{array}{ccc}
s_{x x}-s_{y x} \alpha_{y z}+s_{z x} \alpha_{z y} & s_{x y}-s_{y y} \alpha_{y z}+s_{z y} \alpha_{z y} & s_{x z}-s_{y z} \alpha_{y z}+s_{z z} \alpha_{z y}  \tag{3.9}\\
s_{y x}-s_{z x} \alpha_{z x} & s_{y y}-s_{z y} \alpha_{z x} & s_{y x}-s_{z z} \alpha_{z x} \\
s_{z x} & s_{z y} & s_{z z}
\end{array}\right)
$$

Ignoring the products between the off-diagonal terms of both $\mathbf{M}$ and $\mathbf{S}^{*}$, which have small values, an approximation of $\mathbf{E}, \mathbf{E}^{*}$, can be obtained as follows in Eq. 3.10:

$$
\mathbf{E}=\left(\begin{array}{ccc}
s_{x x} & s_{x y}-s_{y y} \alpha_{y z} & s_{x z}+s_{z z} \alpha_{z y}  \tag{3.10}\\
s_{y x} & s_{y y} & s_{y x}-s_{z z} \alpha_{z x} \\
s_{z x} & s_{z y} & s_{z z}
\end{array}\right)=\left(\begin{array}{ccc}
e_{00} & e_{01} & e_{02} \\
e_{10} & e_{11} & e_{12} \\
e_{20} & e_{21} & e_{22}
\end{array}\right)
$$

$\mathbf{E}^{*}$ is a diagonally dominant correction matrix. The proposed error model for a tri-axial accelerometer setup is given in Eq. 3.11.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

$$
\begin{equation*}
\boldsymbol{a}_{\mathrm{P}, k}=\mathbf{E}^{*}\left(\boldsymbol{a}_{\mathrm{S}, k}-\boldsymbol{b}_{\mathrm{a}}\right) \tag{3.11}
\end{equation*}
$$

The model parameters in matrix $\mathbf{E}^{*}$, and the bias vector $\boldsymbol{b}_{\mathrm{a}}$, are collected to form $\boldsymbol{\theta}_{\mathrm{a}}=\left\{e_{00}, e_{01}, e_{02}, e_{10}, e_{11}, e_{12}, e_{20}, e_{21}, e_{22}, b_{x}, b_{y}, b_{z}\right\}$ to define the function in Eq. 3.12.

$$
\begin{equation*}
\mathrm{h}\left(\boldsymbol{a}_{\mathrm{S}, k}, \boldsymbol{\theta}_{\mathrm{a}}\right)=\mathbf{E}^{*}\left(\boldsymbol{a}_{\mathrm{S}, k}-\boldsymbol{b}_{\mathrm{a}}\right)=\boldsymbol{a}_{\mathrm{P}, k} \tag{3.12}
\end{equation*}
$$

Assuming that the magnitude of gravity is unity, the cost function proposed by Skog and Händel (2006) to measure the amount of deviation from the ideal $1 g(\mathbf{p} \mathbf{1})$ for $K$ sets of measurements is shown in Eq. 3.13.

$$
\begin{equation*}
\mathrm{L}\left(\boldsymbol{\theta}_{\mathrm{a}}\right)=\sum_{k=0}^{K-1}\left(1-\left\|\mathrm{h}\left(\boldsymbol{a}_{\mathrm{S}, k}, \boldsymbol{\theta}_{\mathrm{a}}\right)\right\|^{2}\right)^{2} \tag{3.13}
\end{equation*}
$$

### 3.3.2.2 Gyroscope bias removal during calibration

The most significant source of error for gyroscopic systems is the random bias drift. Random gyroscope bias drifts can be characterized using the Allan Variance, $\sigma_{a}^{2}$ (ElDiasty et al., 2007; Niu and El-Sheimy, 2005; Sabatini, 2006), which measures the variance of the differences between consecutive interval averages. It is originally used to study clock drifts, and is defined in Eq. 3.14.

$$
\begin{equation*}
\sigma_{a}^{2}=\frac{1}{2} \sum_{k=1}^{K}(y(t, k)-y(t, k-1))^{2} \tag{3.14}
\end{equation*}
$$

where $y(t, k)$ is the $k$-th interval average which spans $t$ seconds.

In this research work, static gyroscope signals were collected for one hour, and analyzed by varying $t$ from 1 second $(K=3600)$ to 400 seconds $(K=9)$. As $t$ increases, the effects of noise are reduced, and the value of $\sigma_{a}^{2}$ decreases and converges to the average of the random drifts. The least value of $K$ is chosen as 9 so that the number of samples or interval averages is not too small for statistical reasoning to be applied.

Figure 3.4 shows the Allan Variance plot of the three gyroscopes in the prototype IMU described in section 3.3.5. The drift characteristic of the x -axis of the gyroscope is the worst, as its $\sigma_{a}^{2}$ takes 20 seconds to converge. This can be due to defects in the manufacturing of this gyroscope or the assembly of the IMU. This implies that the gyroscope bias should be averaged over a period of at least 20 seconds so that the average bias will not change significantly in the next few 20 seconds interval. For the purpose of calibration, averaging the static gyroscope signals over a period of time determined using the Allan Variance analysis above would keep the bias drift minimal during the following time period when the calibration data is collected.

$t(\mathrm{sec})$

Figure 3.4: The Allan Variance plot of the three gyroscopes in the prototype IMU.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

### 3.3.2.3 Tri-axial gyroscopic system error model

The misalignments, scale factors and cross-axis sensitivities are modeled next. For brevity, the general notation of the accelerometer model is used, except for the turn rate vector $\boldsymbol{w}=\left(w_{x}, w_{y}, w_{z}\right)^{\mathrm{T}}$, in radians per second. The error model is as shown in Eq. 3.15.

$$
\begin{equation*}
\boldsymbol{w}_{\mathrm{P}, k}=\mathbf{M}_{\mathrm{g}} \mathbf{S}_{\mathbf{g}}^{*}\left(\boldsymbol{w}_{\mathrm{S}, k}\right) \tag{3.15}
\end{equation*}
$$

where $\mathbf{S}_{\mathrm{g}}^{*}=\left(\begin{array}{llll}s_{x x} & s_{x y} & s_{x z} \\ s_{y x} & s_{y y} & s_{y z} \\ s_{z x} & s_{z y} & s_{z z}\end{array}\right)$ and $\mathbf{M}_{\mathrm{g}}=\left(\begin{array}{ccc}1 & -\alpha_{y z} & \alpha_{z y} \\ \alpha_{x z} & 1 & -\alpha_{z x} \\ -\alpha_{x y} & \alpha_{y x} & 1\end{array}\right)$.
In this model, $\boldsymbol{w}_{\mathrm{S}, k}$ is assumed to have zero biases, i.e., the existing biases have been removed using a separate gyroscope bias model. This is because gyroscope biases can change over time, while the other model parameters remain relatively constant. Therefore, the gyroscope biases have to be modeled separately. For the short duration of the calibration in this research, the random gyroscope biases are effectively removed by averaging the static signals over 20 seconds. An example of a more sophisticated bias compensation technique has been reported by Sabatini (2006).
$\mathbf{M}_{\mathrm{g}}$ is the full misalignment correction matrix, where there is no predefined alignment unlike for $\mathbf{M}$ (Skog and Händel, 2006). As with the accelerometers, minor misalignments and cross-axis sensitivities are not distinguished, and Eq. 3.16 is obtained.

$$
\begin{equation*}
\boldsymbol{w}_{\mathrm{P}, k}=\mathbf{E}_{\mathrm{g}}\left(\boldsymbol{w}_{\mathrm{S}, k}\right) \tag{3.16}
\end{equation*}
$$

### 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment

A new cost function is proposed here using (p2). First, define $\Psi$ as the operator that converts a sequence of $\boldsymbol{w}_{\mathrm{P}, k}$, from $k=0$ to $k=n$, and the initial gravity vector $\boldsymbol{u}_{0}$, to the gyroscope computed gravity vector $\boldsymbol{u}_{g}$. Therefore,

$$
\begin{equation*}
\boldsymbol{u}_{\mathrm{g}}=\Psi\left(\boldsymbol{w}_{\mathrm{P}, k}, \boldsymbol{u}_{0}\right) \tag{3.17}
\end{equation*}
$$

$\Psi$ can be any algorithm that computes the rotation matrix $\mathbf{R}$ through integrating the angular velocities $\boldsymbol{w}_{\mathrm{P}, k}$. The method used in this research is outlined in section 3.2.6. The computed gravity vector $\boldsymbol{u}_{\mathrm{g}}$ is obtained from the starting gravity vector $\boldsymbol{u}_{0}$, using Eq. 3.18.

$$
\begin{equation*}
\boldsymbol{u}_{\mathrm{g}}=\mathbf{R} \boldsymbol{u}_{0} \tag{3.18}
\end{equation*}
$$

Let $\boldsymbol{u}_{\mathrm{a}}$ be the gravity vector measured using the static accelerometer. Figure 3.5 shows the divergence of $\boldsymbol{u}_{\mathrm{a}}$ and $\boldsymbol{u}_{\mathrm{g}}$ in an uncalibrated IMU that is rotated $180^{\circ}$ about a single axis. The jagged lines are the values of each of the three axes of $\boldsymbol{u}_{\mathrm{a}}$, while the smooth lines are the corresponding values for $\boldsymbol{u}_{\mathrm{g}}$. From Figure 3.5, it is clear the gyroscope sensor errors have accumulated and caused the divergence between the jagged and smooth lines to increase as the rotation continues.

The nine elements of $\mathbf{E}_{\mathrm{g}}$ are collected to form $\boldsymbol{\theta}_{\mathrm{g}}$ for the definition of the proposed cost function in Eq. 3.19.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING


$t(\mathrm{sec})$

Figure 3.5: Plots of the three axes of $\boldsymbol{u}_{\mathrm{a}}$ (jagged lines) and $\boldsymbol{u}_{\mathrm{g}}$ (smooth lines) in an uncalibrated IMU.

$$
\begin{equation*}
\mathrm{L}\left(\boldsymbol{\theta}_{\mathrm{g}}\right)=\sum_{k=0}^{K-1}\left\|\boldsymbol{u}_{\mathrm{a}}-\boldsymbol{u}_{\mathrm{g}}\right\|^{2} \tag{3.19}
\end{equation*}
$$

As state in (p2), the relation $\boldsymbol{u}_{\mathrm{g}}-\boldsymbol{u}_{\mathrm{a}}=0$ is true for all the arbitrary motions between the static states, thus enabling the gyroscopes to be calibrated without the aid of accurate reference turn rates or precise maneuvers. This is an improvement over the previous calibration methods presented in Section 3.2.7, because the calibration process of the gyroscopes is now totally independent of external calibration values. The calibrated accelerometers, which already exist in the IMU, provide the required gravity vector measurements. This means that the IMU can be fully calibrated as it is, without the need for any precision mounting on another instrument. However, due to sensor errors and algorithm errors in $\Psi, \boldsymbol{u}_{\mathrm{a}}$ measured using the accelerometers and $\boldsymbol{u}_{\mathrm{g}}$ computed using $\Psi$ on the gyroscope measurements, are not equal in practice. The
goal is to find the values for the error parameters presented in this section, so that the difference between $\boldsymbol{u}_{\mathrm{a}}$ and $\boldsymbol{u}_{\mathrm{g}}$ is minimum.

### 3.3.3 Quasi-static detection for measurement of static sensor outputs

During calibration, the quasi-static detector (Saxena et al., 2005) is used to ensure that the IMU is not subject to minute low frequency motions and vibrations that are imperceptible to the user. The detector proposed by Saxena et al. (2005) uses both the accelerometer and gyroscope. In this research, only the accelerometers are required to determine the quasi-static state of the IMU. The output of each axis of the accelerometer is first high-pass filtered, followed by a rectification and then low-pass filtered. Let $\boldsymbol{a}$ be the vector of the output of a tri-axial accelerometer, $\operatorname{HPF}()$ be the high pass filter, $\operatorname{RECT}()$ be the rectification operator and $\operatorname{LPF}()$ be the low pass filter, the quasi-static state vector $s$ is given by Eq. 3.20.

$$
\begin{equation*}
s=\operatorname{LPF}(\operatorname{RECT}(\operatorname{HPF}(\boldsymbol{a}))) \tag{3.20}
\end{equation*}
$$

The square of the magnitude of the vector $s$ can be used to detect the motions and vibrations that are imperceptible to the user. The gyroscopes are found to be redundant as the tri-axial accelerometer can detect the angular motion as well. When the IMU is rotated, the direction of the gravity vector with respect to the IMU changes. This causes the output of each axis of the accelerometer to change, which leads to the square of the magnitude of $s$ to increase above a preset detection threshold. This detector improves the accuracy of the static gravity vector measurements, which are required

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

for both the accelerometer and gyroscope calibrations. The quasi-static detector is also essential when the data collection is performed in the field, where it is almost impossible to control the disturbances on the IMU.In this research, $\operatorname{HPF}()$ is implemented as a digital 2 Hz order 1 Butterworth high pass filter. $\operatorname{LPF}()$ is implemented using a 2 Hz order 1 Butterworth low pass filter. The detection process for one axis is shown in Figure 3.6. The $\operatorname{HPF}()$ removes the gravity component from each axis, which is a low frequency signal when the IMU is quasi-static. $\operatorname{RECT}()$ prevents false detections of the quasi-static state when the high-passed signal crosses the zero axis. The LPF() smoothes the signal and removes the spikes.


Figure 3.6: The process of quasi-static detection, showing the original signal going through high pass filtering, followed by rectification and low pass filtering.

### 3.3.4 Proposed Calibration Procedures

This section is divided into three sub-sections. Sections 3.3.4.1 and 3.3.4.2 present the proposed controlled data collection procedures for the accelerometers and the gyroscopes respectively. These procedures have been designed to increase the probability that the non-linear optimization process will arrive at the correct parameter values. It is important to provide more data points than the number of parameter values to be

# 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment 

fitted, and with maximum data variability so as to avoid pathological conditions in the function space of the cost functions. This occurs because the function is highly nonlinear and typically has several local minima. If the data does not have high variability, the minimization procedure may be trapped in one of the local minima and miss the global minimum, which results in calibration errors. The numbers of data points proposed in the latter sections are those that have been found to work well in simulations conducted in this study and with real data. Section 3.3.4.3 presents a proposed data collection procedure that is less rigorous, in which the users alternate between moving and keeping the IMU stationary. In this case, only one data set is required to calibrate both the accelerometers and the gyroscopes. The variability of the data is measured statistically to ensure that the change in the orientation of the IMU is large enough to overcome the effects of the sensor noise. In the experiments, an angular difference of $10^{\circ}$ between the static states has been found to be a good minimal value for the non-linear optimization to produce results with the standard deviations reported in Section 3.3.5.3.

### 3.3.4.1 Controlled collection of accelerometer calibration data (Procedure 1)

For the tri-axial accelerometer, the IMU is placed in 18 positions, i.e., $K=18$, to obtain 18 readings to compute the cost function $\mathrm{L}\left(\boldsymbol{\theta}_{\mathrm{a}}\right)$. These 18 positions consist of resting the IMU on its six flat faces and 12 edges. This proposed arrangement allows the gravity vector measurements to be spread evenly over the unit sphere about the center of the platform frame ( $\boldsymbol{P}$ ) (Lötters et al., 1998). As there are 12 model parameters to fit, it

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

is prudent to have more than 12 measurements. Next, the static signals are averaged over a period of one second to reduce noise. $L\left(\boldsymbol{\theta}_{\mathrm{a}}\right)$ is minimized using the Downhill Simplex optimization method (Press et al., 1992). The initial parameter values are set using the nominal values. This procedure is hereafter referred to as Procedure 1.

### 3.3.4.2 Controlled collection of gyroscope calibration data (Procedure 2)

After the accelerometer has been calibrated, it can be used to serve as a static gravity vector sensor for calibrating the gyroscopes. 18 sets of continuous accelerometer and gyroscope samples are taken. Each set consists of 20 seconds of quasi-static samples for measuring the gyroscope biases and gravity vector. This is followed by a rotation to a new orientation, and a further one-second of quasi-static samples to measure the gravity vector in the new orientation. The period of 20 seconds has been determined using the Allan Variance analysis in section 3.3.2.2. The IMU is mounted on a hinged surface to ease rotations. The precision of the mounting is not critical so long as the mounting is secured. Any rotation speed is acceptable as long as it spans and stays within the measurement range of the gyroscope, e.g., $300^{\circ} / \mathrm{sec}$ for the prototype IMU used in this research. The main requirement is that the IMU must be rotated through large angles, so that the minute systematic errors are accumulated by the integration of the angular velocities to cause significant divergence between $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$. From Eq. 3.15 , the effects of the various model parameters can only be observed when the angular velocities are non-zero. Therefore, the motion prescribed in the following paragraph is meant to ensure that the errors due to each model parameter are accumulated such that its effect in the divergence is significant. If the amount of rotation is small, the

### 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment

accumulated errors will be of smaller values, and can be overwhelmed by the effect of random noise. This in turn can adversely affect the non-linear optimization process.

The IMU is mounted in nine different positions, where three of the positions have one of the axes parallel to the hinge, as illustrated in Figure 3.7. If the IMU were precisely mounted, it would measure zero readings about the other two axes. However, precise mounting is not required in this calibration method as the angular velocities about all three axes are measured simultaneously. The use of property ( $\mathbf{p} \mathbf{2}$ ) provides the constraint to compute the model parameters. As illustrated in Figure 3.8, each of the remaining six positions has one axis perpendicular to the hinge and the other two axes at an angle of approximately $45^{\circ}$ to the hinge. This gives non-zero measurements about the two axes and better exposes the misalignment and cross-axis sensitivity parameters. The clockwise and counter-clockwise rotations of approximately $180^{\circ}$ about the hinge give a total of 18 sets of data. For ease of reference, this procedure is denoted as Procedure 2. The process of computing the gyroscope error model parameter values uses the same non-linear optimization method as in the case of the accelerometers.


Figure 3.7: Three cases of the IMU with one axis parallel to the rotation hinge (dark edge of the grey surface) for gyroscope calibration.


Figure 3.8: Six cases of the IMU with one axis perpendicular to the rotation hinge (dark edge of the grey surface) for gyroscope calibration.

# 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment 

### 3.3.4.3 In-field collection of calibration data (Procedure 3)

The proposed gyroscope calibration method imposes no restrictions on the type of rotations between the static states. This method, together with the use of the quasistatic detector, and the fact that accelerometer calibration does not require precise inclinations, enables the IMU to be calibrated using data that is collected less rigorously. The following proposed data collection and processing method, hereafter referred to as Procedure 3, consists of first keeping the IMU stationary for a period of 20 seconds to remove the random gyroscope bias. Next, the IMU is moved and paused for at least 24 times, which is twice the number of accelerometer error model parameters, so as to obtain the static acceleration measurements and gyroscope readings for calibrating both types of sensors. The quasi-static detector is used to indicate to the user that the IMU has been kept below a pre-defined quasi-static threshold for a preset period of time, after which the IMU can be moved again. In practice, a threshold of $1.16 \times 10^{-4} g^{2}$ and a period of 0.25 seconds are found to be suitable values for obtaining good calibration data. To provide additional buffer against bad data points, the IMU is moved and paused for a total of 30 times, instead of 24 times. Finally, two variations are considered, viz., the first variation involves the IMU being held in the hand (Procedure 3(hand)) and the second with the IMU mounted on a users head (Procedure 3(head)). For Procedure 3(hand), the IMU is left stationary for 20 seconds for the initial gyroscope bias measurement. For Procedure 3(head), this stationary period is reduced to two seconds, as it is found that it is difficult for a user to keep his head still for 20 seconds.

The resultant value of the accelerometer cost function $\mathrm{L}\left(\boldsymbol{\theta}_{\mathrm{a}}\right)$ in Eq. 3.13 provides

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

an indication of the residue error, as well as the quality of the data. The quasi-static threshold is varied from $5.8 \times 10^{-5} g^{2}$ to $3.5 \times 10^{-4} g^{2}$ in steps of $5.8 \times 10^{-5} g^{2}$, and the static time is varied from 0.05 to 0.5 seconds, in steps of 0.05 seconds, in order to find the best values to use for a set of data. The search across a range of quasi-static threshold values and static times is performed because the motion profile is uncontrolled and the residue error is used to find the least noisy set of static gravity vector measurements. The number of quasi-static sets will vary depending on the threshold and the length of the static time. Generally, shorter time lengths will give a greater number of quasistatic pauses, which tend to be noisier as well. Higher thresholds can result in a lower number of quasi-static stages. This is because there are cases where the IMU is moved very slightly, which would cause the data to be separated into two sections when the threshold is low, and considered to be continuously static when the threshold is high.

The ranges mentioned above are determined empirically to give good calibration results. When the quasi-static threshold is set below $5.8 \times 10^{-5} g^{2}$, the number of sets of calibration data obtained is frequently less than 30 , as the IMU is continuously subject to minute vibrations of the users hand or head. Furthermore, sets of measurement with the lowest residue error are rarely obtained with quasi-static threshold above $3.5 \times 10^{-4} g^{2}$. Therefore, setting this upper threshold reduces the amount of computation required. Setting the step size lower than $5.8 \times 10^{-5} g^{2}$ generally produces slight improvement at a large increase in the computation time. The variances of the accelerometer readings for each of the three axes are used to ensure that the IMU is moved sufficiently. In this case, a minimum variance of $0.2 g$ for each axis is found to be sufficient to ensure

# 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment 

that most motions in the data set are at least $10^{\circ}$ apart. If the range of motion is too small, the non-linear optimization process may not produce meaningful results.

The above process simultaneously determines the values of the accelerometer error model parameters, the quasi-static threshold and the length of the static time, which gives the lowest value of the cost function $\mathrm{L}\left(\boldsymbol{\theta}_{\mathrm{a}}\right)$. After performing these steps, the accelerometer is considered to be calibrated. Next, the gyroscope data is separated into sets using the quasi-static states determined using the optimal threshold and time. The gyroscope biases measured using the initial long pause of the IMU allows the bias to be effectively removed from all the gyroscope readings. The non-random errors in the accelerometer readings are compensated using the accelerometer error model parameter values to obtain the gravity vectors. The bias-free gyroscope readings and the gravity vectors are used to compute the parameter values of the gyroscope error model. In summary, the data collected using the proposed procedure is first analyzed to obtain the static accelerometer readings for the calibration method presented in section 3.3.4.1. The result from the first step is then used to analyze the gyroscope readings to provide the data required by the method presented in section 3.3.4.2. Figure 3.9 illustrates the process graphically.

### 3.3.5 Calibration Results and Analysis

The results in this section are obtained from the data collected using a custom-built IMU, measuring $51 \mathrm{~mm} \times 35 \mathrm{~mm} \times 12 \mathrm{~mm}$. The tri-axial accelerometer and gyroscopic system give rise to a total of six sensor outputs. Each sensor output is sampled at a rate of 1000 Hz , implying that 6000 sensor readings are collected per second. Figure

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING



Figure 3.9: The procedural flow for in-field calibration data collection.
3.10 shows a picture of the custom-built IMU.


Figure 3.10: The custom-built IMU used in the experiments.

### 3.3.5.1 Accelerometer calibration with controlled data

To study the effects of noise in the measurements, 30 sets of 18 measurements were collected to obtain the mean and the standard deviation of the model parameter values. This is shown in Table 3.1. The large number of repetitions of Procedure 1 and the resultant low standard deviation values provide evidence that proper data for the non-linear optimization process can be collected using the proposed controlled data collection process. Each of the 30 sets of data provides a noise-contaminated measurement of the model parameters. To make the best use of all the data collected; the mean values in Table 3.1 are used in Eq. 3.21 to define the resultant error model for the tri-axial accelerometers in the prototype IMU.

$$
\boldsymbol{a}_{\mathrm{P}, k}=\left(\begin{array}{ccc}
1.001 & 0.01 & 0.005  \tag{3.21}\\
-0.01 & 1.009 & 0.013 \\
0.014 & 0.015 & 1.000
\end{array}\right)\left(\boldsymbol{a}_{\mathrm{S}, k}-\left(\begin{array}{c}
0.025 \\
-0.022 \\
0.019
\end{array}\right)\right)
$$

To show that the errors are effectively compensated, 100 measurements were taken

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

Table 3.1: Mean and standard deviation of all the parameter values (dimensionless) obtained from calibration using 30 sets of data. The scale factor and bias parameters are emphasized.

|  | Mean | Standard Deviation |
| :---: | :---: | :---: |
| $e_{00}$ | 1.001 | 0.002 |
| $e_{01}$ | 0.010 | 0.008 |
| $e_{02}$ | 0.005 | 0.012 |
| $e_{10}$ | -0.010 | 0.008 |
| $e_{11}$ | 1.009 | 0.002 |
| $e_{12}$ | 0.013 | 0.011 |
| $e_{20}$ | 0.014 | 0.012 |
| $e_{21}$ | 0.015 | 0.010 |
| $e_{22}$ | 1.000 | 0.002 |
| $b_{x}$ | 0.025 | 0.002 |
| $b_{y}$ | -0.022 | 0.002 |
| $b_{z}$ | 0.019 | 0.002 |

with the IMU mounted in various positions. The average and maximum magnitudes of the errors from the ideal $1 g$ are shown in Table 3.2. The average error and consequently the errors of the inclination angle are reduced by approximately five times.

Table 3.2: The average and maximum observed magnitudes of errors from the ideal $1 g$ for the measured static accelerations. The angular errors are shown in brackets.

|  | Average Error | Max Observed Error |
| :---: | :---: | :---: |
| No calibration | 20.1 mg | 44.9 mg |
|  | $\left(1.15^{\circ}\right)$ | $\left(2.57^{\circ}\right)$ |
| With calibration | 4.0 mg | 28.1 mg |
|  | $\left(0.23^{\circ}\right)$ | $\left(1.61^{\circ}\right)$ |

The results in Table 3.2 show that for either the pitch or roll angles, the prototype IMU has an average angular error of $0.23^{\circ}\left(\sin ^{-1}(0.0040)\right)$ and a maximum error of $1.61^{\circ}\left(\sin ^{-1}(0.0281)\right)$ after calibration. As the magnitude of the gravity vector is assumed to be the only quantity known, the angular error here is calculated for the worst case, where the full error appears on a single accelerometer axis that is perfectly

### 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment

horizontal, i.e., perpendicular to the gravity vector. An error of $\sin (\theta) g$ will result in the pitch or roll angle being measured as $\theta^{\circ}$ instead of zero.

### 3.3.5.2 Gyroscope calibration with controlled data

For the gyroscope calibration, Procedure 2 was repeated to ascertain its stability in the presence of measurement noise. After four repetitions, the computed values of the gyroscope error model parameter remained constant and thus no further repetitions were made. Table 3.3 shows the mean and standard deviations of each parameter.

Table 3.3: The mean and standard deviations of gyroscope model parameter values (dimensionless).

|  | Mean | Standard Deviation |
| :---: | :---: | :---: |
| $e_{00}$ | 0.944 | 0.006 |
| $e_{01}$ | 0.000 | 0.001 |
| $e_{02}$ | -0.008 | 0.002 |
| $e_{10}$ | -0.015 | 0.001 |
| $e_{11}$ | 0.947 | 0.008 |
| $e_{12}$ | -0.008 | 0.003 |
| $e_{20}$ | -0.015 | 0.001 |
| $e_{21}$ | 0.004 | 0.004 |
| $e_{22}$ | 0.998 | 0.003 |

As in the case of the accelerometer, to make use of all the data statistically, the mean values are used to define the gyroscope error model for the prototype IMU in Eq. 3.22 .

$$
g_{\mathrm{P}, k}=\left(\begin{array}{ccc}
0.944 & 0.000 & -0.008  \tag{3.22}\\
-0.015 & 0.947 & -0.008 \\
-0.015 & 0.004 & 0.998
\end{array}\right)\left(g_{\mathrm{S}, k}\right)
$$

To illustrate that the model represented by Eq 3.22 effectively compensates the gyroscope errors, the average magnitude of the error vector between $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$ is used.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

The original raw data collected for the calibration, denoted as the calibration set, is used. In addition, two test sets of raw data are used. The first test set, test set 1, consists of 26 samples, of which 18 samples were obtained using the same motion profile described in Procedure 2, except that the angle of rotation is $90^{\circ}$ instead of $180^{\circ}$. For the remaining eight samples, the IMU is mounted so that none of the axes is parallel to the rotation axis. A rotation of $90^{\circ}$ in one direction about the rotation axis causes the gyroscopes to measure non-zero angular velocities on all three axes. The second test set, test set 2, consists of 30 samples, which are divided into six groups. Each group consists of five samples where the motion is a simple rotation of $180^{\circ}$ in a single direction about one axis of the IMU. As there are three axes and two directions of rotation, there are a total of six combinations. All the three data sets have 20 seconds of static data to determine the gyroscope bias.

By validating the gyroscope error model against test set 1 and test set 2, which are not collected using Procedure 2, it can be shown that the effectiveness of the gyroscope error model is not limited to the prescribed motions used during calibration, but applies to general cases as well. Table 3.4 shows the results. The angular deviation is determined by forming an isosceles triangle with vectors $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$ as the two equal sides, and vector $\left(\boldsymbol{u}_{\mathrm{g}}-\boldsymbol{u}_{\mathrm{a}}\right)$ as the base. When the IMU is in motion, the dynamic orientation is maintained using only the gyroscope as the accelerometer measures the accelerations in addition to the gravity. Without applying the sensor error model, the divergence of $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$ can cause an angular error greater than $10^{\circ}$ for all three sets of data, which is visually perceptible. The error is reduced five times after applying the model in Eq.

### 3.3 Methods for In-Field User Calibration of Inertial Measurement Unit without External Equipment

3.16 for the calibration set and the test set 1, making the error less perceptible. A lower divergence allows the dynamic orientation to be accurately maintained over a longer period of time. For test set 2, the parameter values have changed due to temperature changes, as pointed out by El-Diasty et al. (2007). However, the error is still reduced more than two times.

Table 3.4: Average magnitude of divergence and angular deviation between $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$, with and without applying the gyroscope error model in Eq. 3.22.

|  | Calibration set | Test set 1 | Test set 2 |
| :---: | :---: | :---: | :---: |
| Without gyroscope | 213.7 mg | 342.0 mg | 236.7 mg |
| error model | $\left(12.30^{\circ}\right)$ | $\left(19.70^{\circ}\right)$ | $\left(13.60^{\circ}\right)$ |
| With gyroscope | 37.5 mg | 65.4 mg | 98.1 mg |
| error model | $\left(2.15^{\circ}\right)$ | $\left(3.75^{\circ}\right)$ | $\left(5.62^{\circ}\right)$ |

### 3.3.5.3 Calibration with data collected using handheld and head-mounted IMU

The experimental results presented in Sections 3.3.5.1 and 3.3.5.2 demonstrate that both the accelerometer and gyroscope model-fitting procedures perform consistently with the data collected in pre-determined manners. In this section, the relatively imprecise motion profiles for Procedures 1 and 2 are further disregarded. As the cost functions for both sensors do not require known motion profiles, the main goal now is to demonstrate that the non-linear model fitting process can work with data of arguably lower quality. As discussed in section 3.3.5.2, Procedure 3 is repeated to ascertain its stability, and since the data is expectedly noisier, 10 repetitions are made. Table 3.5 and Table 3.7 show the mean and standard deviations of the error model parameter values for the accelerometer and the gyroscope respectively.

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

Table 3.5: The mean and standard deviations of the accelerometer error model parameter values (dimensionless).

|  | Procedure 1 |  | Procedure 3(Hand) |  | Procedure 3(Head) |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Parameters | Mean | Std Dev | Mean | Std Dev | Mean | Std Dev |
| $e_{00}$ | 1.001 | 0.002 | 1.000 | 0.002 | 0.997 | 0.005 |
| $e_{01}$ | 0.010 | 0.008 | 0.003 | 0.012 | -0.017 | 0.025 |
| $e_{02}$ | 0.005 | 0.012 | 0.004 | 0.011 | 0.017 | 0.024 |
| $e_{10}$ | -0.010 | 0.008 | -0.006 | 0.010 | 0.014 | 0.024 |
| $e_{11}$ | 1.009 | 0.002 | 1.008 | 0.002 | 1.004 | 0.014 |
| $e_{12}$ | 0.013 | 0.011 | 0.007 | 0.016 | 0.015 | 0.011 |
| $e_{20}$ | 0.014 | 0.012 | 0.013 | 0.010 | 0.002 | 0.024 |
| $e_{21}$ | 0.015 | 0.010 | 0.022 | 0.014 | 0.010 | 0.010 |
| $e_{22}$ | 1.000 | 0.002 | 0.999 | 0.001 | 0.996 | 0.006 |
| $b_{x}$ | 0.025 | 0.002 | 0.024 | 0.001 | 0.026 | 0.008 |
| $b_{y}$ | -0.022 | 0.002 | -0.022 | 0.001 | -0.017 | 0.015 |
| $b_{z}$ | 0.019 | 0.002 | 0.019 | 0.001 | 0.021 | 0.006 |

From Table 3.5, the results for Procedure 3(hand) and Procedure 1 agree well. A standard statistical hypothesis test, which is the two-sample $t$-test, is used to determine the presence of any significant difference through examining the two-tail $P$-values. From the test, it is found that only $e_{01}$ and $e_{21}$ have $P$-values less than 0.1 , which indicates the presence of significant statistical difference. Ten out of twelve parameters are not statistically different; this can be attributed to the fact that both procedures are performed with the IMU held in the hand. Procedure 1 specifies positions for placing the IMU and a longer static time. The Procedure 3(Hand) compensates for the random placements of the IMU through using more static samples, checking that the IMU is moved sufficiently, as well as searching for the best quasi-static threshold and static time to use. For Procedure 3(Head), the $P$-values for the two-sample $t$-test for eight out of the twelve parameters are less than 0.1, indicating that this procedure does not perform well. We can also observe in Table 3.5 that the standard deviations of the
parameter values for Procedure 3(Head) are generally larger than those for Procedure 1 and Procedure 3(Hand). The lower calibration accuracy of Procedure 3(Head) is mainly due to the restricted range of motion and is frequently the case for non-linear optimization where the effect of noise is more significant when the spread of the input data is low.

To further elucidate the difference in the performance of the three procedures, the raw data used to obtain the results in Table 3.2 is reused. Table 3.6 shows the average magnitude of the error and the maximum observed error from $1 g$, with error compensation using parameter values obtained from each of the three procedures. The results show that the average errors are reduced by at least a factor of three, and the difference in the performance among the three procedures is small relative to the overall error reduction. Although there is degradation in the performance when the random motions are used, the results show that the additional data processing described in Section 3.3.4.3 mitigates the detrimental effects well.

Table 3.6: Comparison of the average and maximum magnitudes of errors for the same accelerometer test data, compensated with model parameters obtained using the three procedures.

|  | Average Error | Max Observed Error |
| :---: | :---: | :---: |
| No calibration | 20.1 mg | 44.9 mg |
|  | $\left(1.15^{\circ}\right)$ | $\left(2.57^{\circ}\right)$ |
| Procedure 1 | 4.0 mg | 28.1 mg |
|  | $\left(0.23^{\circ}\right)$ | $\left(1.61^{\circ}\right)$ |
| Procedure 3(Hand) | 4.4 mg | 29.4 mg |
|  | $\left(0.25^{\circ}\right)$ | $\left(1.68^{\circ}\right)$ |
| Procedure 3(Head) | 5.3 mg | 30.0 mg |
|  | $\left(0.30^{\circ}\right)$ | $\left(1.72^{\circ}\right)$ |

For the gyroscopes, the results in Table 3.7 show that the parameter values for

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

Procedure 3(Hand) and Procedure 3(Head) have larger standard deviations. This is expected due to the randomness of the motion profile. For the case of Procedure 3(Head), the large standard deviations can also be attributed to the restricted range of motion, and the shorter initial static time for measuring the gyroscope biases. The results from the two-sample $t$-test are inconclusive for Procedure 3, and only one parameter shows significant statistical difference from Procedure 2. As with the accelerometers, the raw data used to obtain the results in Table 3.4 are reused to study the performance of the three procedures. The results are shown in Table 3.8. The parameter values obtained using all the three procedures reduce the systematic gyroscope errors to similar levels for each data set. The only exception is when Procedure 2 is tested with the calibration set; the parameter values have been specifically fitted to the data, and the angular error is reduced by a factor of 5.7. The reduction factor for Procedure 3(Hand) and Procedure $3($ Head ) are 3.1 and 2.6 respectively for the calibration set. For test sets 1 and 2, all three procedures have similar reduction factors of five and two respectively. Therefore, it is evident that both Procedure 3(Hand) and Procedure 3(Head) have similar performance as Procedure 2 despite having larger standard deviations. One possible reason is that the scale factor error is the dominant gyroscope error in this case and the values obtained using all the three procedures are similar. For the custom-built IMU, all three procedures can be used for effective gyroscope calibration.

### 3.3.5.4 Analysis

In the final analysis, the cost function derived in this research has eliminated the requirement for comparison with precise external inclinations and turn rates. However,

Table 3.7: The mean and standard deviations of the gyroscope error model parameter values (dimensionless).

|  | Procedure 2 |  | Procedure 3(Hand) |  | Procedure 3(Head) |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Parameters | Mean | Std Dev | Mean | Std Dev | Mean | Std Dev |
| $e_{00}$ | 0.944 | 0.006 | 0.949 | 0.005 | 0.944 | 0.006 |
| $e_{01}$ | 0.000 | 0.001 | -0.003 | 0.013 | 0.014 | 0.029 |
| $e_{02}$ | -0.008 | 0.002 | -0.004 | 0.011 | 0.009 | 0.028 |
| $e_{10}$ | -0.015 | 0.001 | -0.007 | 0.013 | -0.022 | 0.025 |
| $e_{11}$ | 0.947 | 0.006 | 0.948 | 0.019 | 0.945 | 0.021 |
| $e_{12}$ | -0.008 | 0.003 | -0.034 | 0.027 | -0.032 | 0.014 |
| $e_{20}$ | -0.015 | 0.001 | -0.014 | 0.011 | -0.028 | 0.024 |
| $e_{21}$ | 0.004 | 0.004 | 0.026 | 0.032 | 0.033 | 0.017 |
| $e_{22}$ | 0.998 | 0.003 | 1.003 | 0.017 | 1.001 | 0.009 |

Table 3.8: Comparison of the average magnitudes of divergence, angular deviation between $\boldsymbol{u}_{\mathrm{g}}$ and $\boldsymbol{u}_{\mathrm{a}}$ for the same gyroscope test data sets, compensated with model parameters obtained using the three procedures.

|  | Calibration set | Test set 1 | Test set 2 |
| :---: | :---: | :---: | :---: |
| No calibration | 213.7 mg | 342.0 mg | 236.7 mg |
|  | $\left(12.30^{\circ}\right)$ | $\left(19.70^{\circ}\right)$ | $\left(13.60^{\circ}\right)$ |
| Procedure 2 | 37.5 mg | 65.4 mg | 98.1 mg |
|  | $\left(2.15^{\circ}\right)$ | $\left(3.75^{\circ}\right)$ | $\left(5.62^{\circ}\right)$ |
| Procedure 3 | 70.2 mg | 69.0 mg | 107.9 mg |
| (hand) | $\left(4.02^{\circ}\right)$ | $\left(3.95^{\circ}\right)$ | $\left(6.19^{\circ}\right)$ |
| Procedure 3 | 83.2 mg | 73.2 mg | 108.2 mg |
| (head) | $\left(4.76^{\circ}\right)$ | $\left(4.19^{\circ}\right)$ | $\left(6.20^{\circ}\right)$ |

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

there is need to provide low noise data that is well distributed over the range of inputs. This is well illustrated by the increased variability of the computed parameter values as the quality of the data is degraded. In the case of accelerometer calibration, Procedure 1 does not require significantly more effort than Procedure 3. Therefore, the recommended accelerometer calibration technique is to provide a list of positions to place the IMU, as in Procedure 1, and apply the data processing steps in Procedure 3 , to capitalize on the best features of both techniques. In contrast, Procedure 2 for the gyroscope is more laborious, as it requires remounting the IMU and longer data collection times, often taking more than an hour to perform all the 18 rotations. For Procedure 3, there is no additional effort for gyroscope calibration as all the data is collected while calibrating the accelerometer, and each set of data only requires a few minutes to collect.

The above results and analysis show that if sufficient repetitions of Procedure 3 (Hand) are made, 10 in this case, the performance in error compensation can approach that of the controlled Procedures 1 and 2. This result is significant for casual users as it means that IMU can be calibrated by simply moving the IMU held in the hands.

### 3.4 Concluding Remarks

The proposed error models and calibration methods compensate for sensor errors and thereby improve the accuracy of the readout of low cost sensors. This is achieved without the need for comparison with generated truth-values. By applying Procedure

1 to the accelerometers in the custom-built prototype IMU, the average observed pitch or roll angle error is $0.23^{\circ}$, which represents a reduction ratio of five as compared to the errors without calibration. From Table 3.2, the observed maximum pitch or roll error is also reduced by a factor of 1.6 , from $2.57^{\circ}$ to $1.61^{\circ}$. Using Procedure 2 , there is a minimum reduction factor of two for the dynamic angular divergence due to the gyroscope sensor errors.

These methods are accessible to and can be easily performed by IMU developers who do not have specialized calibration equipment. As the IMU is mainly used as an orientation sensor in a hybrid AR tracker, an accuracy of $0.23^{\circ}$ is adequate for maintaining the illusion of proper augmentation. Procedure 3, which can handle random motions, enables IMUs to be easily calibrated by causal users, so as to reduce the average static error to $0.30^{\circ}$ and reduce the dynamic angular divergence by more than a factor of two. This level of ease of inertial sensors calibration has not been achieved in previous methods.

### 3.4.1 Future Developments

It is possible to automate the data collection process and calibrate the IMU automatically without user intervention. This has been applied for accelerometers (Lötters et al., 1998) and is now possible for gyroscopes with methods proposed in this research. This can reduce the production time and cost during mass production, as the IMU can self-calibrate during use. Procedure 3 is a step in the direction towards self-calibrating IMUs. The sensor data can be collected while the IMU is in normal use, and these data can be analyzed in a fashion similar to Procedure 3 to extract sets of data for

## 3. INERTIAL SENSORS: CALIBRATION FOR ORIENTATION SENSING

calibration purposes. As the optimization process is relatively slow, the data can be stored for later analysis, which can be done offline. Although it is not possible to make these low cost sensors into navigation grade instruments, they can be easily made into accurate orientation sensors and short-term positional trackers as part of the hybrid tracking systems.

The availability of miniature accelerometers and gyroscopes enables inertial sensing to be applicable to human scale tracking for VR and AR applications. The performance of these sensors, in terms of sensor errors, is not sufficient for independent position tracking in the foreseeable future. However, through the use of calibration, the systematic sensor errors can be compensated for. This enables a MEMS-based IMU to function as an independent orientation sensor, with low jitter, low latency and high robustness. Therefore, inertial sensing forms an important component in a hybrid AR tracker, as it is a reliable source of orientation information. When used with GPS, which is a pure position tracker, both systems form a complete 6DOF tracker. Inertial sensing can aid CV tracking by adding stability and maintaining the orientation in the presence of interference to vision tracking, such as motion blur due to rapid camera motion. With reduction in cost and size, it is expected that inertial sensing will become increasingly more common in AR trackers. At the time of writing, they are becoming quite common in consumer electronics, such as mobile phones and game controllers, for providing orientation information.

# Global Positioning System: Differential carrier phase for open-area positioning 

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

### 4.1 Introduction

The Global Positioning System (GPS) is a satellite navigation system that enables users to determine their absolute positions with respect to the Earth. It is built and maintained by the United States to serve the countrys military needs. In addition, there is a civilian service, known as the Standard Positioning Service (SPS), which is freely available globally. This SPS service has been widely used for non-military applications, such as aviation and maritime navigation, providing driving directions and land surveys. The SPS is specified to provide an accuracy of 13 metres or better in the horizontal plane for $95 \%$ of the time. The height accuracy is specified at 22 m for $95 \%$ of the time (Cosentino et al., 2006).

With respect to Augmented Reality (AR) tracking, the addition of GPS to the tracking assembly provides a global absolute coordinate frame with respect to the Earth. Thus, the operating volume of the tracker can be expanded to include the whole of the Earths surface that provides a clear view of the sky. GPS is robust to environmental interferences, as it is developed to meet the requirements for all weather and all condition military operations. The receiver equipment is also widely available at low cost. However, the accuracy and jitter levels do not meet the requirements for AR, except for augmentation at distances of hundreds of metres from the users. The jitter is particularly detrimental as the position measurement can change by more than a metre per second. Therefore, to achieve better accuracy and lower jitter, the low noise carrier phase measurement of the GPS signal is used, instead of the code measurement
used in GPS SPS. Furthermore, two receivers are employed to remove common mode errors to further increase accuracy.

This chapter introduces a novel differential precise GPS positioning method developed in the course of this research (Fong et al., 2008b). This method processes the simultaneous carrier phase measurements from two low cost GPS receiver modules to achieve an accuracy of 10 cm , with low jitter, and without the need to expend large computation resources to resolve the integer ambiguity of carrier phase measurements.

### 4.2 Background

This section provides a brief introduction to the components and operating principles of the GPS and the coordinate frames used. Specialized topics include the use of precise ephemeris to determine the GPS satellite positions and the mathematical models for differential GPS positioning. The current methods for solving carrier phase integer ambiguity are presented to clarify the difficulty it presents to current real time GPS precision positioning systems.

### 4.2.1 The Global Positioning System

The GPS consists of satellites that transmit structured radio signals to the earths surface. The number of satellites, orbital radius and their arrangements in space were chosen to ensure global coverage with at least six satellites visible at all location, and at all time (Kaplan and Hegarty, 2005). One of the most important features of GPS satellites is the onboard rubidium and cesium atomic clock standards. These clocks allow the GPS signals transmitted by every satellite to be highly synchronized and

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

with very high frequency stability, both of which are important in achieving the stated accuracy with low cost passive receivers. These clocks are used to generate transmitted radio signals at two frequencies, namely the L 1 at 1.5754 GHz , and L 2 at 1.2276 GHz . The L1 frequency is used in this research as it is the only one currently available from low cost receivers. All satellites transmit at the same frequencies. In order for the GPS receiver to distinguish between transmissions from different satellites, the information is modulated with Pseudo Random Number (PRN) codes before transmission. These codes serve to allow for simultaneous satellite transmissions and measurement of the range between the satellite and the receiver. However, the errors in GPS positioning using PRN code measurements can be more than 10 metres and have high jitter. Therefore, the L1 carrier phase is used. As modern radios have phase lock loops that can measure the phase of the carrier to within $5 \%$ of the wavelength (L1 wavelength is 190.3 millimetre), the accuracy of the carrier phase measurements is often within one centimetre. This is the mechanism used for improving the accuracy of GPS positioning in ARTIST.

### 4.2.2 Applications in Augmented Reality

One of the earliest applications of GPS in an AR system is the Touring Machine Feiner et al. (1997). This prototype is used for augmentation when navigating in a city. In this application, the accuracy of the GPS receiver is sufficient. A recent example of a lightweight wearable system reported byPeternier et al. (2006) illustrates the rapid reduction of weight and power consumption of GPS receivers. Their work allows the GPS to be used as a robust and lightweight absolute position tracker. However, the
accuracy and jitter levels are not suitable for stable augmentation at distances within several meters, which is required by many AR applications.

### 4.2.3 Differential Global Positioning System

The use of Differential GPS is typically required to achieve centimeter level of accuracy. Many works have reported the use of differential GPS carrier phase for determining the relative positions between two or more GPS receivers (Cosentino et al., 2006; HofmannWellenhof et al., 2004; Leick, 2003). The most common application of differential GPS is for land surveys. Therefore, most presented methods deal with relative distances of several kilometers or more, and are not directly applicable to AR applications. Outdoor AR applications are expected to work using shorter baselines, as raw measurements are most likely to be transmitted in real-time using wireless links with limited ranges. Therefore, techniques for short baselines are more applicable (Chang et al., 2005a; Cosentino et al., 2006; Hayward et al., 1998). When the baseline is less than a kilometer, the differential GPS measurements can be approximated using interferometry (Cosentino et al., 2006; Hayward et al., 1998). Such an approximation is widely used in GPS attitude determination and it is used in this research.

The main research issue in differential GPS is the real-time resolution of the integer ambiguity in the presence of measurement noise (Chang et al., 2005b; Cosentino et al., 2006; Hofmann-Wellenhof et al., 2004). The method proposed in this research does not require the resolution of the integer ambiguities. A similar work by How et al. (2002) tracks the relative position from an initial point rather than from the stationary GPS receiver. This method uses the difference between two simultaneous GPS Doppler

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

measurements to obtain accurate velocities, which are in turn integrated to give the position. The method proposed in this research uses carrier phase measurements instead.

### 4.2.4 Coordinate Frames

Two coordinate frames are used in this GPS research; they are the Earth-Centered Earth-Fixed $(\boldsymbol{E C E F})$ frame and the North-East-Down (NED) frame. Both frames are right-handed Cartesian coordinate frames. To visualize a right-handed frame, one can imagine a flat right hand curling to form a thumbs up hand sign. The four fingers of a flat right hand are pointing towards the $x$-axis. As the fingers curl, they rotate and point towards the $y$-axis. Finally, the thumb points towards the $z$-axis. An illustration of a right-handed frame is shown in Figure 4.1.


Figure 4.1: Illustration of a right-handed frame.

### 4.2.4.1 Earth-Centered Earth-Fixed (ECEF) Frame

The $\boldsymbol{E C E F}$ frame is a right-handed, three-dimensional Cartesian coordinate frame. The origin is at the center of the Earth. The $x$-axis and $y$-axis lie on the equatorial plane, with the $x$-axis and $y$-axis passing through the equator at $0^{\circ}$ longitude and $90^{\circ}$ East longitude respectively. The $z$-axis coincides with the rotational axis of the Earth and points towards the North. This is illustrated graphically in Figure 4.2. The satellite position and the various positioning algorithms, including the method developed in this work, are represented in the $\boldsymbol{E C E F}$ frame. In general, various points in the $\boldsymbol{E C E F}$ frame are given in the familiar $(x, y, z)$ Cartesian notation.


Figure 4.2: The Earth-Centered Earth-Fixed (ECEF) coordinate frame.

### 4.2.4.2 North-East-Down (NED) Frame

The $\boldsymbol{N E D}$ frame is a right-handed local level coordinate frame, with its center at a specified latitude and longitude on the Earths surface. The orientation measured using inertial sensing is with respect to the $\boldsymbol{N E D}$ frame as well (Section 3.2.5). The $x$-axis

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

and $y$-axis are on the local level or tangential plane, with the $x$-axis pointing North, the $y$-axis pointing East. The $z$-axis points Down, parallel to the normal of the local tangential plane. This is a more natural coordinate frame for AR applications, as the user moves around in the local environment.

In order to convert a position measured in the $\boldsymbol{E C E F}$ frame to the $\boldsymbol{N E D}$ frame, one would need to obtain the latitude and longitude of this position and use that to rotate the $\boldsymbol{E C E F}$ axes to align with the $\boldsymbol{N E D}$ axes. The conversion procedure, in turn, requires a reference ellipsoid that defines the average Earths surface, which would coincide with a perfectly smooth Earth. The local level plane in the $\boldsymbol{N E D}$ frame is tangential to the reference ellipsoid at the specified longitude and latitude. In this research, the reference used is known as the World Geodetic System 1984 (Kaplan and Hegarty, 2005). The relevant parameters are given in Table 4.1.

Table 4.1: The World Geodetic System 1984 (WGS 84) reference ellipsoid

| Equatorial cross section | Circular |
| :--- | :--- |
| Mean radius in equatorial plane, or semi-major axis, $a$ | $6,378.137 \mathrm{~km}$ |
| Polar radius, or semi-minor axis, $b$ | $6,356.752 \mathrm{~km}$ |
| ${\text { (Eccentricity })^{2}, e^{2}=1-b^{2} / a^{2}}^{(\text {Second Eccentricity })^{2}, e^{2}=\left(a^{2} / b^{2}\right) e^{2}}$ | 0.00669437999014 |

Using the reference ellipsoid, the $\boldsymbol{E C E F}$ position is interchangeable with the geodetic Latitude, Longitude and Height ( $\boldsymbol{L L H}$ ) coordinates, commonly used for geographic disciplines. The position $\boldsymbol{p}_{\text {ECEF }}$ in $\boldsymbol{E C E F}$ is $\boldsymbol{p}_{\text {ECEF }}=\left(x_{p}, y_{p}, z_{p}\right)$. For $\boldsymbol{L L H}, \boldsymbol{p}_{\mathrm{LLH}}=$ ( $\varphi, \Lambda, h$ ), where $\varphi$ is the latitude, $\Lambda$ is the longitude, and $h$ is the normal height above the ellipsoid surface. The convention used is as follows. The latitude, $\varphi$ is positive in the northern hemisphere and negative in the south. The longitude, $\Lambda$ is positive
to the east of the Greenwich meridian and negative to the west. In this research, the conversion from $\boldsymbol{E C E F}$ to $\boldsymbol{L L H}$ coordinates is shown in Table 4.2. It is based on the closed form solution reported by Zhu (1994), and shows the computations necessary for obtaining $\varphi$ and $\Lambda$. Height is omitted as only the longitude and latitude are required to rotate the $\boldsymbol{E C E F}$ to the $\boldsymbol{N E D}$ coordinate frame. Furthermore, current AR applications are only expected to operate on or near the Earth's surface.

The position vector can be rotated from $\boldsymbol{E C E F}$ to the $\boldsymbol{N E D}$ frame, at longitude $\Lambda$, and latitude $\varphi$, using the following rotation matrix, $\mathbf{R}_{\mathrm{E}}^{\mathrm{N}}$ (Jekeli, 2000) in Eq. 4.1.

$$
\mathbf{R}_{\mathrm{E}}^{\mathrm{N}}=\left(\begin{array}{ccc}
-\sin (\varphi) \cos (\Lambda) & -\sin (\varphi) \sin (\Lambda) & \cos (\varphi)  \tag{4.1}\\
-\sin (\Lambda) & \cos (\Lambda) & 0 \\
-\cos (\varphi) \cos (\Lambda) & -\cos (\varphi) \sin (\Lambda) & -\sin (\varphi)
\end{array}\right)
$$

### 4.2.5 Precise Ephemeris

The ephemeris is a set of satellite orbit parameters that allows the $\boldsymbol{E C E F}$ position of a satellite, at a point in time, to be accurately determined. As satellite orbits can be perturbed to a small degree by unpredictable forces, such as thermal effects, the ephemeris is updated every few hours to enable accurate positioning. Two types of ephemeris are considered. First is the broadcast ephemeris that is downloaded directly from the GPS satellites. Each satellite broadcasts the ephemeris of every other GPS satellite. The data format is specified in the report by Arinc Research Corporation (Arinc, 2000) and the computation of the orbit position is presented by Xu (2007). The second is the precise ephemeris. Although the broadcast ephemeris is always available, the high precision ephemeris downloaded from the International Global Navigation

Table 4.2: Closed-form solution used to convert of $\boldsymbol{E C E F}$ coordinates to geodetic latitude.

First obtain $\Lambda$,

$$
\Lambda=\left\{\begin{aligned}
\tan ^{-1}\left(\frac{y_{p}}{x_{p}}\right) & \text { if } x_{p} \geq 0 \\
180^{\circ}+\tan ^{-1}\left(\frac{y_{p}}{x_{p}}\right) & \text { if } x_{p}<0 \cap y_{p} \geq 0 \\
-180^{\circ}+\tan ^{-1}\left(\frac{y_{p}}{x_{p}}\right) & \text { if } x_{p}<0 \cap y_{p}<0
\end{aligned}\right.
$$

To obtain $\varphi$,

$$
\begin{aligned}
r & =\sqrt{x_{p}^{2}+y_{p}^{2}} \\
E^{2} & =a^{2}+b^{2} \\
F & =54 b^{2} z_{p}^{2} \\
G & =r^{2}+\left(1-e^{2}\right) z_{p}^{2}-e^{2} E^{2} \\
C & =\frac{e^{4} F r^{2}}{G^{3}} \\
S & =\sqrt[3]{1+C+\sqrt{C^{2}+2 C}} \\
P & =\frac{F}{3(1+1 / S+S)^{2} G^{2}} \\
Q & =\sqrt{1+2 e^{4} P} \\
r_{0} & =\frac{-P e^{2} r}{1+Q}+\sqrt{\frac{1}{2} a^{2}\left(1+\frac{1}{Q}\right)-\frac{P\left(1-e^{2}\right) z_{p}^{2}}{Q(1+Q)}-\frac{1}{2} P^{2}} \\
V & =\sqrt{\left(r-e^{2} r_{0}\right)^{2}+\left(1-e^{2}\right) z_{p}^{2}} \\
Z_{0} & =\frac{b^{2} z_{p}}{a V}
\end{aligned}
$$

Finally,

$$
\varphi=\tan ^{-1}\left(\frac{z_{p}+e^{\prime 2} Z_{0}}{r}\right)
$$

Satellite System (GNSS) Service (IGS) is used in this work (Moore and Neilan, 2005). The main purpose of using precise ephemeris is to eliminate the ephemeris error as possible residue error in the models developed in this research.

The IGS is a voluntary international network of stations with high performance GPS receivers. There are several IGS processing nodes, which process the raw measurements from all over the world to derive accurate satellite orbits, clock errors and ionospheric corrections. For satellite orbits, three types of data or products are available, namely Final, Rapid and Ultra-Rapid products (IGS, 2010). The Final and Rapid products are used for post-processing as they are released weekly and daily respectively, after the data is recorded. The orbit accuracy is within 5 cm and the satellite clock error is within 0.1 ns .

For real time purposes, the Ultra-Rapid product, specifically the predicted half, is used to compute the satellite position. The stated accuracy is approximately 10 cm and 5ns for satellite orbit and clock errors respectively. In comparison, the broadcast ephemeris has an accuracy of approximately 160 cm and 7 ns . Each Ultra-Rapid product has 48 hours of orbit data, which consisted of processed data from the past 24 hours as well as predicted satellite orbit for the next 24 hours. It is released every six hours, with a three-hour delay. Due to the delay, each downloaded data file has at most 21 hours of usable predicted satellite orbits. The Ultra-Rapid product is released in the SP3 format (Hilla, 2007). The data consist of predicted $\boldsymbol{E C E F}$ positions and clock errors of all the GPS satellites at 15 -minute intervals. In order to obtain accurate position within each interval, the Lagrange interpolation is used in this work (Xu, 2007). The

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

general form for $N$-th order interpolation is as shown in Eq. 4.2, where $\boldsymbol{p}(t)$ is the interpolated value at $t$, and $f\left(t_{k}\right)$ is the value of the $k$-th sample point of $t$.

$$
\begin{equation*}
\boldsymbol{p}(t)=\sum_{k=0}^{N} l_{k}(t) f\left(t_{k}\right) \tag{4.2}
\end{equation*}
$$

where the Lagrange basis function is $l_{k}(t)=\frac{\left(t-t_{0}\right) \ldots\left(t-t_{k-1}\right)\left(t-t_{k+1}\right) \ldots\left(t-t_{N}\right)}{\left(t_{k}-t_{0}\right) \ldots\left(t_{k}-t_{k-1}\right)\left(t_{k}-t_{k+1}\right) \ldots\left(t_{k}-t_{N}\right)}$
In this work, time is scaled, such that 15 minutes is equal to a value of 1.0. Time $t$ is further translated, such that it has a value in the range of 3.0 to 4.0 . In this way, the orbit is interpolated using four points before and five points after $t$. Furthermore, $t_{k}$ has integer values from 0 to 8 . As the order of the interpolation is fixed in this case, the basis functions can be partially pre-computed. The specific interpolation used in this work is shown in Eq. .

$$
\begin{align*}
\boldsymbol{p}(t)= & \frac{\gamma}{40320 t}-\frac{\gamma}{5040(t-1)}+\frac{\gamma}{1440(t-2)}-\frac{\gamma}{720(t-3)}+\frac{\gamma}{576(t-4)} \\
& \quad-\frac{\gamma}{720(t-5)}+\frac{\gamma}{1440(t-6)}-\frac{\gamma}{5040(t-7)}+\frac{\gamma}{40320 t} \tag{4.3}
\end{align*}
$$

where $\gamma=\prod_{k=0}^{N}(t-k)$.

### 4.2.6 GPS Carrier Phase Measurement

The GPS L1 carrier signal has a frequency of 1.5754 GHz , which implies a wavelength of 0.1903 m . As GPS receivers are radios, a Phase Lock Loop (PLL) can be used to measure the fractional phase of the carrier signal. Therefore, it is possible to measure the position within a single wavelength or cycle. In modern equipment, the noise level is $0.2-5 \mathrm{~mm}$ (Hofmann-Wellenhof et al., 2004). With the high stability of GPS frequencies,
this represents a highly accurate mechanism to measure distances. The main difficulty is that every carrier cycle is identical to the other, which results in ambiguity in the number of full carrier cycles between the satellite and the receiver. With accurate Doppler measurements, or frequency shift due to the high relative velocity between the satellite and the receiver, it is possible to determine the change in cycles since the start of the measurement by integrating the Doppler. As such, only the number of carrier cycles between the satellite and receiver at the start of the measurement is unknown, and this is known as the integer ambiguity. If the integer ambiguity is known, all the carrier cycles at latter times can be found by adding the integrated Doppler to the integer ambiguity. Therefore, integer ambiguity remains constant with time, even though the number carrier cycle changes with time. Carrier phase measurement is rarely used in standalone configurations; this is further limited by the fact that most GPS noise sources can result in errors of several cycles.

The models for the carrier phase measurement and the associated errors can be found in numerous references (Chang et al., 2005a; Cosentino et al., 2006; HofmannWellenhof et al., 2004; Leick, 2003). Eq. 4.4 is the model for the phase measurements from satellite $i$, to receiver $s$ at time $t_{k}$, which accounts for most of the significant errors. The measurement unit is the number of carrier cycles.

$$
\begin{equation*}
\Phi_{s}^{i}\left(t_{k}\right)=\lambda^{-1} \rho_{s}^{i}\left(t_{k}\right)+N_{s}^{i}+f \tau_{s}\left(t_{k}\right)+f \tau^{i}\left(t_{k}\right)-\beta_{\mathrm{iono}}\left(t_{k}\right)+\delta_{\mathrm{tropo}}\left(t_{k}\right)+\mu_{s}^{i}\left(t_{k}\right) \tag{4.4}
\end{equation*}
$$

$\Phi_{s}^{i}\left(t_{k}\right)$ is the carrier phase measurement. $\rho_{s}^{i}\left(t_{k}\right)$ is the change in the range since the start of measurement, which can be measured by integrating the Doppler shift. $N_{s}^{i}$ is the

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

starting integer ambiguity, and it is constant with respect to time except for cycle slips. $\tau_{s}\left(t_{k}\right), \tau^{i}\left(t_{k}\right)$ are the receiver and satellite clock errors respectively. $\beta_{\text {iono }}\left(t_{k}\right), \delta_{\text {tropo }}\left(t_{k}\right)$ are the errors due to transmission through the Earth's ionosphere and troposphere respectively. $\mu_{s}^{i}\left(t_{k}\right)$ includes both random noise and un-modeled errors. $\lambda$ and $f$ are the wavelength and frequency respectively.

### 4.2.7 Differential Positioning

In this research, differential GPS positioning refers to the use of carrier phase measurements from two or more GPS receivers to cancel out the common mode errors. The use of carrier phase measurement is common for GPS surveying systems, and it can typically achieve an accuracy of 20 cm in real-time, and 1 mm with post-processing. There are several established methods to combine the carrier phase measurements to remove certain errors (Hofmann-Wellenhof et al., 2004; Xu, 2007). The following sections present the commonly used Single, Double and Triple Differences.

### 4.2.7.1 Single Difference

The Single Difference (SD) is the difference between two simultaneous measurements of the carrier phase by the receivers $s$ and $r$, for satellite $i$. If the baseline is less than 20 km , and the receivers are at the same height above the sea level, the ionospheric error, $\beta_{\text {iono }}\left(t_{k}\right)$, and the tropospheric error, $\delta_{\text {tropo }}\left(t_{k}\right)$, are common to both, and can be removed by differencing (Cosentino et al., 2006; Hofmann-Wellenhof et al., 2004). For most AR applications, the distance and height separation between receivers are expected to be sufficiently small for the above errors to be effectively removed by Single

Differencing. The satellite clock error, $\tau^{i}\left(t_{k}\right)$ is also common between the two receivers, and is removed. For SD, the noise level of $\mu_{s r}^{i}\left(t_{k}\right)$ is double that of the measurement noise $\mu_{s}^{i}\left(t_{k}\right)$, but remains uncorrelated between the satellites (Hofmann-Wellenhof et al., 2004). The model for SD is shown in Eq. 4.5.

$$
\begin{align*}
S D_{s r}^{i}\left(t_{k}\right)= & \Phi_{r}^{i}\left(t_{k}\right)-\Phi_{s}^{i}\left(t_{k}\right) \\
= & \lambda^{-1}\left[\rho_{r}^{i}\left(t_{k}\right)-\rho_{s}^{i}\left(t_{k}\right)\right]+\left[N_{r}^{i}-N_{s}^{i}\right]+f\left[\tau_{r}\left(t_{k}\right)-\tau_{s}\left(t_{k}\right)\right]+f\left[\tau^{i}\left(t_{k}\right)-\tau^{i}\left(t_{k}\right)\right] \\
& -\left[\beta_{\text {iono }}\left(t_{k}\right)-\beta_{\text {iono }}\left(t_{k}\right)\right]+\left[\delta_{\text {tropo }}\left(t_{k}\right)-\delta_{\text {tropo }}\left(t_{k}\right)\right]+\left[\mu_{r}^{i}\left(t_{k}\right)-\mu_{s}^{i}\left(t_{k}\right)\right] \\
= & \lambda^{-1} \rho_{s r}^{i}\left(t_{k}\right)+N_{s r}^{i}+f \tau_{s r}\left(t_{k}\right)+\mu_{s r}^{i}\left(t_{k}\right) \tag{4.5}
\end{align*}
$$

### 4.2.7.2 Double Difference

Double difference is commonly used in GPS surveying. The SD for satellites $i$ and $j$ are differenced, which removes the common inter-receiver clock error $\tau_{s r}\left(t_{k}\right)$. The model for DD is shown in Eq. 4.6.

$$
\begin{align*}
D D_{s r}^{i j}\left(t_{k}\right) & =S D_{s r}^{j}\left(t_{k}\right)-S D_{s r}^{i}\left(t_{k}\right) \\
& =\lambda^{-1}\left[\rho_{s r}^{j}\left(t_{k}\right)-\rho_{s r}^{i}\left(t_{k}\right)\right]+\left[N_{s r}^{j}-N_{s r}^{i}\right]+f\left[\tau_{s r}\left(t_{k}\right)-\tau_{s r}\left(t_{k}\right)\right]+\left[\mu_{s r}^{j}\left(t_{k}\right)-\mu_{s r}^{i}\left(t_{k}\right)\right] \\
& =\lambda^{-1} \rho_{s r}^{i j}\left(t_{k}\right)+N_{s r}^{i j}+\mu_{s r}^{i j}\left(t_{k}\right) \tag{4.6}
\end{align*}
$$

The removal of the inter-receiver clock error results in a processed signal with low noise levels, and this can be observed in Figure 4.3. The data used to generate Figure 4.3 was collected by two stationary receivers placed 2 metres apart. Therefore, the

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

changes in the DD is purely due to satellite motion. The removal of the receiver clock errors in DD enables the computation of the "float solution", where the DD integer ambiguities can be computed as real values and not integers. The accuracy is often quoted as within the decimeter level.


Figure 4.3: A plot of five processed Double Difference signals obtained using raw measurements from six satellites. (A different color is used for each satellite and the yellow line along the horizontal is a plot of the noise level of the reference satellite.)

The signal plots in Figure 4.3 have been processed by the software developed in this research to perform the carrier cycle integer increment using only the raw measurements. The plots show that the low cost receivers have a low combined noise level for $\mu_{s r}^{i j}\left(t_{k}\right)$, despite the fact that the noise level in $\mu_{s r}^{i j}\left(t_{k}\right)$ is increased by four times over the measurement noise $\mu_{s}^{i}\left(t_{k}\right)$. Furthermore, the low cost receivers can only provide accurate fractional phase measurement within a cycle. The integer cycle count from the receivers used is unreliable, resulting in constant cycle slip. The clean DD signals
allow for the determination of the increment or decrement of the integral cycles in low dynamics conditions.

Although real-time tracking with four or more visible satellites is possible using DD , it is not utilized in the proposed method. This is because the noise, $\mu_{s r}^{i j}\left(t_{k}\right)$, becomes correlated across satellites. Furthermore, as the DDs in each time epoch are computed against one common reference satellite, this causes the measurement to be overly dependent on the noise level of the reference satellite measurement (Chang et al., 2005a). This is observed in the experiments carried out in this research, and requires repeated computation of the residue errors with every satellite as the reference, so as to determine the satellite with the lowest noise to be used as the reference satellite. Furthermore, these techniques can be complicated by the signal outage or setting of the reference satellite. This is especially true for techniques where some forms of data from the previous epochs are kept.

### 4.2.7.3 Triple Difference

The DD at well separated time epochs, $t_{k}$ and $t_{k}$, can be differenced to form the Triple Difference (TD), which removes the time invariant integer ambiguities. This in turn requires that there is no cycle slips in the integrated Doppler (Section 4.2.6), which would cause the change in the number of carrier cycles since the start of measurement to be wrong. The model is shown in Eq. 4.7.

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

$$
\begin{align*}
T D_{s r}^{i j}\left(t_{k}, t_{k^{\prime}}\right) & =D D_{s r}^{i j}\left(t_{k^{\prime}}\right)-D D_{s r}^{i j}\left(t_{k}\right) \\
& =\lambda^{-1}\left[\rho_{s r}^{i j}\left(t_{k^{\prime}}\right)-\rho_{s r}^{i j}\left(t_{k}\right)\right]+\left[N_{s r}^{i j}-N_{s r}^{i j}\right]+\left[\mu_{s r}^{i j}\left(t_{k^{\prime}}\right)-\mu_{s r}^{i j}\left(t_{k}\right)\right] \\
& =\lambda^{-1} \rho_{s r}^{i j}\left(t_{k}, t_{k^{\prime}}\right)+\mu_{s r}^{i j}\left(t_{k}, t_{k^{\prime}}\right) \tag{4.7}
\end{align*}
$$

TD is a robust method to determine the static baseline vector to an accuracy of 1 m . However, the noise levels increase by eight times due to differencing and become highly time correlated. Therefore, a minimum of one hour of static data is often recommended in practice (Hofmann-Wellenhof et al., 2004). This renders TD as an ineffective method for dynamic real-time AR tracking.

### 4.2.8 Integer Ambiguity

A general scheme for differential positioning is to use DD to solve for both the position and integer ambiguity as a linear system. The integer property of the ambiguity is ignored at this stage and the solution is a float solution, where integer ambiguities are returned as real numbers or decimal values. The float solution values are resolved to true integer values using Integer Least Squares techniques, such as the Least-squares AM-Biguity Decorrelation Adjustment (LAMBDA) and Fast Ambiguity Search Filter (FASF) (Chang et al., 2005a; Hofmann-Wellenhof et al., 2004). These methods give the fixed solution with centimeter accuracy level.

The resolution of the integer ambiguities in the presence of noise is non-trivial. The short wavelength of the GPS carrier implies a large search space, and therefore a high computational load. For example, with a short baseline of 1 m , the DD integer ambiguity
can range from -5 to +5 for each satellite. This means that there are 11 possible values per satellite; and with six satellites, the search space has a size of $(11)^{6}=1,771,561$. A brute force search is computationally infeasible due to the exponential increase in the search space. Furthermore, due to noise, the point in the search space with the least residue error may not be the actual solution. This increases the difficulty of obtaining the actual integer ambiguities. An improved method is to only search the region around the float solution. The size of the region is defined by the covariance in the ambiguity solution. The LAMBDA method is most commonly used because it de-correlates noise, resulting in smaller search regions. Data collected over several minutes to an hour is typically required to average out the noise and increase the confidence in the accuracy (Hofmann-Wellenhof et al., 2004).

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

### 4.3 Precise Positioning using Differential Single Difference

This section presents a novel differential GPS carrier phase technique for 3D outdoor position tracking in mobile AR applications (Fong et al., 2008b). It has good positioning accuracy, low drift and jitter, and low computational requirement. The proposed method differs from previous differential methods (Section 4.2.7) with the use of a different differential quantity, namely, the Differential Single Difference (DSD). The DSD is used to compute the relative position of the mobile GPS receiver from its initial position, without having to determine the baseline vector relative to the stationary GPS receiver. This method achieves the accuracy of current real-time carrier-based precision GPS trackers without the need for heavy computing resources required for resolving the integer ambiguities. There is a resultant linear drift due to the accumulation of the minute errors in the actual GPS modules. However, the drift rate is less than $0.001 \mathrm{~ms}^{-1}$, and it varies slowly and is highly linear within a period of several minutes. Therefore, the drift can be compensated using linear regression. Experimental results using low cost GPS receivers show that the position error is 10 cm , and the drift is $0.001 \mathrm{~ms}^{-1}$.

### 4.3.1 Motivation for Precise Positioning using Differential Single Difference (DSD)

Differential carrier phase GPS positioning was developed using low cost modules to provide position tracking at centimeter levels. However, initial work with SD and DD proved unsuccessful. In static tests with known baseline vector and precise satellite
orbits, the length of the baselines obtained did not agree with the known values. As it is possible to generate simulated DD plots in these cases, a comparison was made with the measured DD values. Figure 4.4 shows the comparison between the simulated and measured DD for two satellites, against the same reference satellite. The measured DD plots are generated in the same manner as those in Figure 4.3. Both the simulated and measured DD plots have similar forms, except for the gradients and measurement noise. This indicated the existence of a drift in the DD signal due to the lower quality of the phase measurements.

For static receivers, the position is constant over time, but the drift in DD results in apparent changes or drift in position. By measuring the apparent position drift and compensating this drift directly, the effects of minute errors in individual carrier phase measurements can be compensated as a whole. To avoid the resolution of integer ambiguity, the use of DSD is proposed. Similar to TD, the integer ambiguity is removed by differencing across adjacent time epochs. As it is an accumulative method, it is prone to drift as with inertial navigation. However, the drift is low and thus the method is further developed into a position tracker, presented in the following sections.

### 4.3.2 Development of the Method

The proposed method avoids the resolution of the integer ambiguity through differencing the SD between two consecutive time epochs $t_{k}$ and $t_{k+1}$, for the same satellite i. This is shown in Eq. 4.8. As such, $S D_{s r}^{i}\left(t_{k}, t_{k+1}\right)$ forms the derived GPS quantity described as the DSD. Independent of this research, DSD has been applied in the study of receiver hardware delay (Liu et al., 2004).

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING



Figure 4.4: A comparison between measured (jagged) and simulated (smooth)
Double Differences (DD) for two satellites, generated using the same reference satellite. The simulated DD are generated using the known baseline and satellite positions, and show the first indication of the presence of drift.

$$
\begin{align*}
S D_{s r}^{i}\left(t_{k}, t_{k+1}\right)= & \lambda^{-1}\left[\rho_{s r}^{i}\left(t_{k+1}\right)-\rho_{s r}^{i}\left(t_{k}\right)\right]+\left[N_{s r}^{i}-N_{s r}^{i}\right]+f\left[\tau_{s r}\left(t_{k+1}\right)-\tau_{s r}\left(t_{k}\right)\right] \\
& +\left[\mu_{s r}^{i}\left(t_{k+1}\right)-\mu_{s r}^{i}\left(t_{k}\right)\right] \\
= & \lambda^{-1} \rho_{s r}^{i}\left(t_{k}, t_{k+1}\right)+f \tau_{s r}\left(t_{k}, t_{k+1}\right)+\mu_{s r}^{i}\left(t_{k}, t_{k+1}\right) \tag{4.8}
\end{align*}
$$

In order to obtain position measurements from DSD, the following approximations are used. First, $\rho_{s r}^{i}\left(t_{k}\right)$ is approximated using interferometry principles (Cosentino et al., 2006; Hayward et al., 1998). Consider receivers $s$ and $r$, which are less than a kilometer apart. As the GPS satellites are approximately $23 \times 10^{6} \mathrm{~m}$ away, the unit vectors, $\boldsymbol{e}_{s}^{i}$ and $\boldsymbol{e}_{r}^{i}$ of the lines of sight from the two receivers to the same satellite $i$ can be assumed to be parallel, i.e., $\boldsymbol{e}_{s}^{i} \approx \boldsymbol{e}_{r}^{i}$. Let $\boldsymbol{b}$ be the baseline vector between the receivers, in metres, and receiver $s$ be stationary. The approximation is given by the
vector dot product, as shown in Eq. 4.9.

$$
\begin{equation*}
\rho_{s r}^{i}\left(t_{k}\right) \approx e_{s}^{i}\left(t_{k}\right) \bullet \boldsymbol{b}\left(t_{k}\right) \tag{4.9}
\end{equation*}
$$

Although the satellite moves several kilometers per second, the approximation, $e_{s}^{i}\left(t_{k}\right) \approx e_{s}^{i}\left(t_{k+1}\right)$, is appropriate due to the large distance between the receiver and the satellite. For example, the closest a satellite can get to a receiver on the ground is 20,200 kilometres. As the satellite travels a distance of approximately four kilometres in a second, the angular change of the vector $\boldsymbol{e}$ is only 0.01 degrees in the same amount time. The various variables used are illustrated in Figure 4.5.


Figure 4.5: Illustration of the variables used in DSD computation.

Substituting Eq. 4.9 into Eq. 4.8, and omitting the noise term, $\mu_{s r}^{i}\left(t_{k}, t_{k+1}\right)$, for greater clarity gives Eq. 4.10.

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

$$
\begin{align*}
S D_{s r}^{i}\left(t_{k}, t_{k+1}\right) & \approx \lambda^{-1}\left[e_{s}^{i}\left(t_{k+1}\right) \bullet \boldsymbol{b}\left(t_{k+1}\right)-e_{s}^{i}\left(t_{k}\right) \bullet \boldsymbol{b}\left(t_{k}\right)\right]+f\left[\tau_{s r}\left(t_{k+1}\right)-\tau_{s r}\left(t_{k}\right)\right] \\
& \approx \lambda^{-1} e_{s}^{i}\left(t_{k+1}\right) \bullet\left[\boldsymbol{b}\left(t_{k+1}\right)-\boldsymbol{b}\left(t_{k}\right)\right]+f\left[\tau_{s r}\left(t_{k+1}\right)-\tau_{s r}\left(t_{k}\right)\right] \\
& =\lambda^{-1} e_{s}^{i}\left(t_{k+1}\right) \bullet \Delta \boldsymbol{b}\left(t_{k+1}\right)+f \Delta \tau_{s r}\left(t_{k+1}\right) \tag{4.10}
\end{align*}
$$

There are four unknowns in Eq. 4.10, namely, the three components in the position change vector, $\Delta \boldsymbol{b}\left(t_{k+1}\right)$ and inter-receiver time drift, $\Delta \tau_{s r}\left(t_{k+1}\right) . S D_{s r}^{i}\left(t_{k}, t_{k+1}\right)$ is derived from raw GPS phase measurements, while $\boldsymbol{e}_{s}^{i}\left(t_{k+1}\right)$ is obtained using precise satellite ephemeris, the receiver position using the standalone GPS measurement and the GPS time measured by the receiver. Although the standalone GPS position has an error of 10 m , applying the same reasoning for the approximation, $e_{s}^{i}\left(t_{k}\right) \approx \boldsymbol{e}_{s}^{i}\left(t_{k+1}\right)$, the large range between the satellite to the receiver causes the resultant error in $\boldsymbol{e}_{s}^{i}\left(t_{k+1}\right)$ to be insignificant. With at least four satellites, the unknowns can be solved using the linear system in Eq. 4.11. Here, $\boldsymbol{e}_{s}^{i}\left(t_{k+1}\right)^{\mathrm{T}}$ is the transpose of $\boldsymbol{e}_{s}^{i}\left(t_{k+1}\right)$.

$$
\left(\begin{array}{c}
S D_{s r}^{1}\left(t_{k}, t_{k+1}\right)  \tag{4.11}\\
\left.S D_{s r}^{2 r} t_{k}, t_{k+1}\right) \\
S D_{s r}^{3 r}\left(t_{k}, t_{k+1}\right) \\
S D_{s r}^{4}\left(t_{k}, t_{k+1}\right) \\
\vdots
\end{array}\right)=\left(\begin{array}{cc}
e_{s}^{1}\left(t_{k+1}\right)^{\mathrm{T}} & 1 \\
\left.e_{s}^{3} t_{k+1}\right)^{\mathrm{T}} & 1 \\
\left.e_{s}^{3} t_{k+1}\right)^{\mathrm{T}} & 1 \\
e_{s}^{4}\left(t_{k+1}\right)^{\mathrm{T}} & 1 \\
\vdots & \binom{\lambda^{-1} \Delta \boldsymbol{b}\left(t_{k+1}\right)}{f \Delta \tau_{s r}\left(t_{k+1}\right)} .
\end{array}\right)
$$

The position change from one epoch to the next can be accumulated to give the position vector of the receiver $r$ from its starting position. This is in contrast to measuring the baseline vector from receiver $s$ to receiver $r$, and it avoids the resolution of integer ambiguities. The main issue with accumulative approaches is that minute errors and biases are also accumulated, resulting in positional drift. Furthermore, the
noise also becomes time correlated. However, the experimental results in the Section 4.3.4 show that with phase measurements from low cost GPS receivers, the drift is less than $0.001 \mathrm{~ms}^{-1}$ and highly linear with time. The position derived is in the $\boldsymbol{E C E F}$ frame. The position vector can be rotated from $\boldsymbol{E C E F}$ to the local level $\boldsymbol{N E D}$ frame using Eq. 4.1.

### 4.3.2.1 Drift Correction using Linear Regression

This sub-section presents the linear regression method used to correct the drift in the proposed method. From the plots of the apparent position drift of the static receivers, such as the plot presented in section 4.3.4, the variations of the positions in the $x, y$ and $z$ axes are linear within a period of several minutes. As there is no clear relationship among the variations about each axis, simple linear regression is used for each axis to compensate for the drift.

The goal of simple linear regression is to fit a straight line along the data points, such that the variation of the data about this line is the minimum. In this case, it is to derive the gradients and intercepts of the three best fit lines along the position drifts of the $x, y$ and $z$ axes, with respect to time. As the procedure is the same for all three axes, only the linear regression for the $x$-axis is shown. Let $x_{k}$ and $t_{k}$ be the $x$-axis position and time respectively for the $k$-th data sample. The gradient $m_{x}$, and intercept $c_{x}$ of the line of best fit among $N$ data samples is shown in Eq. 4.12.

$$
\begin{equation*}
x_{k}=m_{x} t_{k}+c_{x} \tag{4.12}
\end{equation*}
$$

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

The gradient $m_{x}$, is obtained using Eq. 4.13, where the mean of $x$ and $t$, are $\bar{x}$ and $\bar{t}$ respectively.

$$
\begin{equation*}
m_{x}=\frac{\sum_{k=1}^{N}\left(x_{k}-\bar{x}\right)\left(t_{k}-\bar{t}\right)}{\sum_{k=1}^{N}\left(t_{k}-\bar{t}\right)^{2}} \tag{4.13}
\end{equation*}
$$

By using $N \bar{x}=\sum_{k=1}^{N} x_{k}$ and $N \bar{t}=\sum_{k=1}^{N} t_{k}$, Eq. 4.13 can be converted to the form as shown in Eq. 4.14.

$$
\begin{equation*}
m_{x}=\frac{N\left(\sum_{k=1}^{N} x_{k} t_{k}\right)-\left(\sum_{k=1}^{N} x_{k}\right)\left(\sum_{k=1}^{N} t_{k}\right)}{N\left(\sum_{k=1}^{N} t_{k}^{2}\right)-\left(\sum_{k=1}^{N} t_{k}\right)^{2}} \tag{4.14}
\end{equation*}
$$

The form in Eq. 4.14 shows that the gradient $m_{k}$, can be computed by accumulating four values, namely, $x_{k}, t_{k},\left(x_{k} t_{k}\right)$ and $t_{k}^{2}$, as they are obtained in each time epoch. This means that the linear regression can be computed efficiently. By substituting a known point on the line, $(\bar{x}, \bar{t})$ into Eq. 4.12, the intercept $c_{x}$, is obtained as:

$$
\begin{equation*}
c_{x}=m_{x} \bar{t}-\bar{x} \tag{4.15}
\end{equation*}
$$

### 4.3.3 Experimental Setup

To determine the effectiveness of the proposed method, two LEA-4T GPS modules from U-Blox are used to collect raw carrier phase measurements. The data is recorded using serial links and the vendor supplied software. The maximum GPS measurement rate is 10 Hz . Two experiments, E1 and E2 were conducted in an open area, where there are minimal obstructions from buildings and trees, which can cause signal outages and multipath errors.

For E1, the receivers were placed three metres apart on a level ground. The direction from the static receiver $s$ to the mobile receiver $r$ with respect to the North was measured using a wireless InertialCube from InterSense. Three sets of data, D1, D2 and D3, were collected in E1. D1 consists of 1 Hz GPS raw measurements collected over a period of one hour with both receivers static. This is to determine the drift characteristics. D2 consists of 10 Hz GPS raw measurements, where the receiver r was first left static for approximately 180 seconds after which it was moved 20 cm along the baseline vector towards the static receiver $s$, before it was returned to the starting position. The same motion profile was repeated but with a distance of one metre instead. D3 consists of 10 Hz GPS raw measurements, where receiver $r$ was first left static for 300 seconds, after which it was moved one metre along the baseline vector towards the static receiver $s$, before it was returned to the starting position. The same motion profile was repeated for ten times to obtain a measure of the accuracy and repeatability of the proposed method.

In E2, the receiver $r$ was mounted rigidly with an InterSense InertialCube and a Firewire video camera. The InertialCube measures the orientation in the $\boldsymbol{N E D}$ frame using inertial sensing. The resultant position and orientation tracking data is used to augment virtual objects onto the video recorded. In this case, the assembly of the receiver $r$, the InertialCube and the camera are handheld and moved over a distance of 0.5 m .

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

### 4.3.4 Experimental Results

### 4.3.4.1 Experiment E1

The position vector from the initial position of receiver $r, r 1$, in the $\boldsymbol{E C E F}$ frame is computed using the proposed method based on the data set D1. The values of $x, y$ and $z$ axes of the vector are shown in Figure 4.6. As both receivers were stationary, Figure 4.6 shows the drift characteristics. The maximum position vector drift is 2.5 m over a period of 3,000 seconds. This translates to a drift of less than $0.001 \mathrm{~ms}^{-1}$, which is sufficient for maintaining the stability of virtual objects augmented onto a real environment. Figure 4.6 also shows that the drift has low jitter and varies slowly with time and is highly linear within a period of several hundred seconds. Although the cause of the linear drift cannot be ascertained using the existing equipment, it is probable that the drift is due to the carrier phase measurements not being perfectly synchronous for the two receivers. It was not possible to set the receivers to output the measurements at specific times. This might have been useful as the receivers used has errors of less than 15 nanosecond for GPS time. So there can be a difference of several milliseconds between the measurements from the two receivers. Attempts to compensate for the difference in time using the doppler shift (Kaplan and Hegarty, 2005) was not successful as doppler noise was more higher than those in the phase measurements. As the algorithms in the receivers are not known, and there was no access to more configurable receivers, the actual cause of the drift was not determined. The variation in the drift in Figure 4.6 is most probably due to minute change in time interval between the measurements. For the various experiments presented here, there
was no discernible pattern in the linear drift due to changes in the spatial configuration of the receivers and satellites. However, more tests and equipment will be required to ascertain that.


Figure 4.6: Plot of the drift of position vector from the initial position of a stationary receiver, r1 against time $t$.

The position vector from the initial position of the receiver $r, \boldsymbol{r} \mathcal{Z}$ in the $\boldsymbol{N E D}$ frame derived using D2 is shown in Figure 4.7. For data set D2, the receiver $r$ was moved 20 cm to and fro, and then one metre to and fro. The initial static period of 180 seconds allows for the determination of the linear drift, which can be effectively removed using linear regression analysis. This is shown in Figure 4.8.

Figure 4.8 shows the plot for $\boldsymbol{r} \boldsymbol{2}^{\prime}$ derived using D2 with the error corrected. The result shows that the linear drift is effectively removed and the prescribed motions are measured with a good level of accuracy. From Figure 4.8, the magnitude of the distance moved is accurate for the first prescribed motion profile of 20 cm , and is within 10 cm for


Figure 4.7: Plots of position vector from the initial position of mobile receiver $r, r \boldsymbol{2}$ derived using D2 against time $t$.


Figure 4.8: Drift corrected position vector from the initial position of mobile receiver $r$, $\boldsymbol{r} \mathscr{Z}^{\prime}$ derived using D2 against time $t$.
the second motion of 1 m . The error after receiver $r$ has been returned to the starting position is 15 cm . These errors are mainly due to the noise introduced by the effects of the motion on the reception of the radio signal by the antenna.

To further illustrate the repeatability of this new method, receiver $r$ was moved one metre to and fro, for ten repetitions to collect data set D3. The position vector from the initial position of receiver $r$ along the baseline vector, $r 3$ is first derived using D3. This position is rotated from the $\boldsymbol{E C E F}$ frame to the $\boldsymbol{N E D}$ frame, and further rotated by the orientation measured by the InertialCube, so as to show the motion along the baseline. There are linear drifts observed along the three axes in the first 300 seconds. Linear regression analysis is used to determine and compensate for these drifts. Figure 4.9 shows the plot of the positions along the baseline against time $t$. The result shows that the linear drift is effectively removed and the prescribed motions are measured with a good level of accuracy. From Figure 4.9, the distance moved along the baseline is accurate within 10 cm for all ten repetitions. This indicates both good precision and repeatability. The error after receiver $r$ was returned to the starting position is 10 cm . There is also motion along the two axes orthogonal to the baseline vector where there is supposed to be none. These errors are mainly due to a change in drift over time and the noise introduced by the effects of motion on the reception of the radio signal by the antenna. Further work will be required to determine the antenna designs to minimize the effects of motion on the reception. The low jitter and high precision indicate that the proposed method is suitable for outdoor AR applications.

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING



Figure 4.9: Drift corrected position vector from the initial position of mobile receiver $r$, $\boldsymbol{r} 3$ derived using D3 against time $t$.

### 4.3.4.2 Experiment E2

Figure 4.10 shows the plot of the linear drift corrected position vector, $\boldsymbol{r} 4$ in the $\boldsymbol{N E D}$ frame, derived using the data collected in experiment E2. For this data set, the linear drift is low and has been effectively removed using linear regression analysis on the initial 60 seconds of the static data. Here, the main motion was the picking up of the camera and panning to record the scene. There was also a certain amount of lateral motion and tilting of the camera. The plot shows that the motion of the camera is tracked with a high level of accuracy and with low jitter. As the motion profile is not exactly known, the effectiveness of the method is demonstrated by augmenting virtual objects onto a video, so as to directly check the effectiveness of the proposed method for outdoor AR. Qualitatively, the video shows that the motion of the camera is well tracked, allowing for fairly realistic augmentation. Figure 4.11 shows the results of
augmenting onto objects at close range, which is not possible using standalone GPS. The resultant drift is visible in Figure $4.11(\mathrm{~d})$, where the teapot appears to float a few centimetres above the box.


Figure 4.10: Plot of position vector from the initial position of mobile receiver $r, \boldsymbol{r} 4$ against time $t$.

### 4.4 Concluding Remarks

This chapter presents the results of a novel use of GPS carrier phase measurements from two GPS receivers for high precision position tracking in outdoor environments with a focus towards AR applications. A quantity, Differential Single Difference (DSD), is derived from raw phase measurements. This research proposed to use DSD to compute the relative position of the mobile receiver from its initial starting position. This method works by accumulating the positional change in each time epoch computed using DSD. The current work shows that the quality of the phase measurements from low cost GPS modules is sufficient to achieve an accuracy of 10 cm in precision tracking. The result

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING



Figure 4.11: Augmentation using the proposed Differential GPS tracker and IMU (The checker board is used to indicate the drift).
obtained using this proposed method shows that the error drifts slowly with time, is highly linear within a period of several minutes and has low jitter. The experimental results also show that the proposed method has an accuracy of 10 cm .

This result has been obtained without sophisticated signal processing or filtering. As the carrier phase can be measured with an accuracy of 1 mm by high-end GPS receiver, a tracker accuracy of 1 cm is likely to be possible with further improvements in the design of the antenna and the receiver, as well as signal processing techniques. For example, choke ring antenna and narrow time correlators can reduce the multipath error in phase measurements, while post processing of differential GPS signals for measuring continental shift have shown that sub-millimetre accuracy can be achieved.

Such a level of precision is comparable to indoor tracking systems, allowing for accurate tracking for new large scale, outdoor AR applications.

The results show that DSD is useful as a derived quantity with good noise characteristics. In GPS surveying, the main goal is to derive accurate measurements of the relative vector between two points that are several kilometers apart, thus limiting the usefulness of DSD. In contrast, the relative position from the initial position is a useful quantity for AR applications.

In a fully developed setup, the static receivers can be parts of an existing infrastructure. The raw measurements are transmitted wirelessly to the mobile unit. On initialization, the starting point may be determined automatically or set by the user, after which tracking continues using the proposed method.

The proposed method has low computational load and is robust as compared to traditional GPS surveying techniques, allowing it to be used in real-time. If the low drift is assumed to be insignificant for the application, the tracking system is immediately usable after the GPS signals are locked on by the receivers. Furthermore, traditional GPS relative positioning techniques can be used to periodically correct the drift through measuring the actual baseline between the two receivers. This allows for highly accurate real-time tracking while avoiding the high computational load associated with integer ambiguity resolution.

### 4.4.1 Issues

The main issues with the proposed method are common to all GPS-based trackers, namely the need for a clear line of sight to the satellites and frequent signal outages.

## 4. GLOBAL POSITIONING SYSTEM: DIFFERENTIAL CARRIER PHASE FOR OPEN-AREA POSITIONING

These are particularly acute for carrier-based techniques, as a good signal-to-noise ratio is required for the phase lock. In GPS, a channel noise measure, $\mathrm{C} / \mathrm{N}_{0}$ is used and a value greater than 35 is generally needed in experiments. Otherwise, this can cause the GPS receiver to lose the signal phase lock due to antenna motions. When the number of phase measurements drops below six, the proposed method is found to be no longer effective. This issue is particularly acute in built-up areas, which signals from near horizon satellites are severely attenuated by buildings and multi-path reflections. Subsequent phase lock will render DSD to be inaccurate as cycle slips would have occurred, and the number of carrier cycle measured is erroneous. Therefore, the preliminary results presented here can be used for developing hybrid trackers through combining GPS with inertial and computer vision based trackers. Under good operating conditions with a clear view of the sky and no nearby obstructions, the proposed method can also be used as a valuable tool for the development and validation of other outdoor trackers.

Finally, with the modernization of GPS with increased signal power levels, the addition of the European Galileo system and the increasing performance of GPS receivers in recent years, GPS is expected to serve as a valuable and easily accessible tool for high precision, real-time and wide area outdoor tracking. However, precision GPS tracking in challenging areas, such as urban canyons and indoor environment, is not feasible at this moment, and we have to rely on computer vision to achieve the highest precision.

## 5

## Computer Vision: High precision positioning on textured planar surfaces

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

### 5.1 Introduction

Computer vision (CV) is one of the most commonly used tracking methods for Augmented Reality (AR) (Zhou et al., 2008). There are several reasons. The first is that the cameras used for obtaining the images for CV analysis are capable of providing rich and high-resolution information about the environment for tracking purposes. This allows for highly accurate augmentation. The second is compatibility with vision as the primary component of human sensory perception. Therefore, the camera serves as the tracking sensor, as well as the means to present the effects of the augmentation to the users. Although there are AR systems which use optical see-through head mounted displays and projectors, and avoid the presentation of the video captured by the camera, augmentation onto live video still remains as the most compelling form of AR. Third is the low cost and availability of the hardware, namely, the cameras and fast computers, for CV processing. Arguably, the success of ARToolkit (2010) in advancing research in AR is due to these factors. ARToolkit (2010) has been the predominant CV marker-based AR tracking technology due to its ease of use on low cost webcams and printable markers. Though the accuracy and jitter levels are lacking and markers are not desirable in many situations, ARToolkit still remains as the entry point for the development of many AR systems. Many marker-based systems have been developed and offer better performance (Naimark and Foxlin, 2002; Wagner et al., 2008a).

Numerous marker-less CV-based AR trackers have been proposed. They belong to two main classes of methods, tracking with known 3D scene structures and tracking with
natural features. The first class of methods often rely on the use of the 3D structures of the scene, which are either available beforehand (David et al., 2004) or obtained while tracking (Davison et al., 2007; Klein and Murray, 2007, 2008b), to improve the robustness of low level CV operations, such as feature tracking. Although the 3D structures are frequently available as CAD models, the lines and corners often do not match those detected by the CV algorithms. Therefore, the latter case is more suitable as the mapped 3D structures contain features that can be directly used for tracking.

For natural feature tracking, the main difficulty is in solving the correspondence problem, or matching of feature points projected in two or more views. After features have been matched, the camera pose can be easily computed. Two of the most promising techniques are Random Trees (RT) (Lepetit and Fua, 2006) and the Scale Invariant Feature Transform (SIFT) (Lowe, 2004). SIFT is a complex feature descriptor, which uses the distribution of gradients to orientate and match features. The main shortcoming of using the original design of SIFT is the slow computation of the descriptors. SIFT is therefore not suitable for real-time AR. However, modifications presented by Wagner et al. (2008b) demonstrate that it is possible to simplify SIFT for real-time tracking on low-powered devices. The RT approaches use tree structures to encode the probability distribution of binary features of key points. Such methods trade memory and prior training time for speed of matching. Recent advances and modifications to the RT methods for low memory devices have been presented by Wagner et al. (2008b). Another recent advance is the proposal of the keypoint signature (Calonder et al., 2008), which overcomes the problem of long training time by using RT on features which these

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

trees have not been trained for. The efficiency of CV based augmentation has also continued to improve, allowing for the augmentation of multiple independently moving objects. Park et al. (2008) demonstrated a multiple object tracking system based on prior 3D object models, as well as RT for feature matching and pose estimation.

The goals for tracking in this research are to achieve wide-area, robust real-time, high accuracy tracking, with low jitter and latency. For the case of ARTIST, the widearea and robustness aspects are mainly provided by the GPS and inertial components, while the CV component was specifically tailored to achieve high accuracy, low jitter and latency. The high resolution of the sensor information from the camera is the main contributor to the high accuracy. However, the dense information requires a large amount of computation to process them, resulting in difficulty in achieving real-time operations with low latency. This is further compounded by the ambiguity caused by the loss of information during the imaging process where 3D data is converted to 2D images. The high density of the visual information also includes a large amount of noise and ambiguities, which prevents early CV systems from being robust. One method to circumvent these issues is to use artificial fiducials, or markers, to limit the amount of information processed and remove the ambiguity. However, there is general consensus that both the researchers and users of AR prefer marker-less CV tracking systems. This is mainly because markers tend to make AR interactions less natural, and cumbersome or even impossible to set up in large environments.

Real-time, robust marker-less CV tracking has advanced rapidly in the recent years with several notable systems. Examples include the machine learning based systems
(Lepetit and Fua, 2006; Özuysal et al., 2007) and Simultaneous Location and Mapping (SLAM) related systems, such as MonoSLAM (Davison et al., 2007) and Parallel Tracking and Mapping (PTAM) (Klein and Murray, 2007). The work by Wagner et al. (2008b) shows that it is possible to make modifications to existing methods to make them run in real-time on mobile phones, which have low processor speeds and limited memory. The efficiency, robustness and accuracy of such systems are mainly achieved through algorithmic means, accompanied by an increase in computational resources. Many early CV tracking algorithms, such as the Kanade-Lucas-Tomasi (KLT) tracker (Shi and Tomasi, 1994), do not automatically achieve the requirements with increase in computational resources. This implies that current and future improvements would rely mainly on the increase in algorithmic and systems sophistication. The study and experimentation with CV systems in this research shows that current successful CV trackers are complex systems, where each component runs sophisticated algorithms of its own, and interacts with other components. In order to build practical CV trackers for AR applications, there is the need to draw on algorithms developed in basic CV research and develop ways to combine them to form efficient and robust systems.

The development of a CV tracker in this PhD research focuses on the experimentation and modification of individual CV algorithms, as well as the ways to combine these algorithms. This chapter is organized as follows. Section 5.2 consists of a discussion on the organization of the CV tracker as a system. Section 5.3 describes each of the algorithms used in constructing the tracker and presents the proposed modifications and improvements made in this research. Specifically, the focus is on feature detection,

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

feature matching and pose refinement. Section 5.4 describes the experimental setup and results for testing the tracking system in real world conditions. Section 5.5 concludes with discussion on the tracker developed, its limitations and further works on improving it. Work on combining this CV tracker with inertial and GPS systems is presented in Chapter 6

### 5.2 Tracking System Organization

The organization of the tracker can be understood from two perspectives. The first being the systems view and the other the algorithms view. The systems perspective consists of the stages of carrying out the tracking, namely (1) preparation, (2) initialization, (3) tracking and (4) relocalization. The algorithms view consists of the individual algorithmic steps, namely (1) feature detection, (2) feature matching, (3) robust pose estimation and (4) pose refinement. The relationship between the two views is that each system component consists of one or more algorithms. As exemplified by the PTAM (Klein and Murray, 2007) and Scale Invariant Feature Transform (SIFT) (Lowe, 2004), the main reason for such complex systems is to achieve real-time operation while achieving high robustness and accuracy. This section primarily describes the system organization of ARTIST, and how it compares with existing systems using the two perspectives. As such, it also serves as a detailed review of these systems.

### 5.2.1 Systems Perspective

The systems perspective is useful for understanding the stages of the operation of the tracker. It is useful to consider this perspective using the examples of ARToolkit, SIFT and PTAM as well as the tracking system, ARTIST that has been developed in the research. As the trackers developed by Wagner et al. (2008b) are similar to SIFT, they are not explicitly compared here, and only interesting characteristics distinct from SIFT are presented.

Preparation - This stage refers to the steps taken outside the tracking process

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

that are required to create conditions that increase the tracker performance. Many AR trackers use a setup consisting of a monocular camera with a fixed focal lens. Therefore, one of the most common preparation steps is the calibration of the camera to determine the intrinsic parameters, such as the focal length, principal point position on the image plane, as well as the radial and tangential distortion parameters. This reduces the number of unknown parameters in the imaging process, which simplifies the algorithms used and improves the efficiency and robustness. For ARToolkit, the preparation stage also includes creating and positioning of markers. By design, SLAM-based systems, such as PTAM, do not typically require further preparation steps. However, for systems based on complex feature descriptors, such as SIFT and ARTIST that are based on machine learning techniques, the preparation stage also consists of obtaining the feature descriptors. Although ARTIST is based on machine learning, the use of keypoint signatures (Calonder et al., 2008) reduces the time required to obtain the feature descriptors as compared to the earlier Random Tree (RT) and Ferns-based systems. Typically, preparation is a one-time process for each hardware and application setup.

Initialization - The initialization, tracking and relocalization steps refer to the operational stages during actual tracking. Initialization steps are employed to create conditions that increase the performance of the tracker. In contrast to the preparation stage, the initialization stage consists of steps for initializing the current tracking session where conditions differ from the previous sessions. This may not be necessary for every tracker. For example, SIFT, ARToolkit and RT based systems do not require initialization and the continuous tracking cycles start immediately. For PTAM, the
initialization consists of moving the camera sideways to obtain an initial 3D map of the features. This map is required for the various operations of PTAM, such as the active search of features, improving both the efficiency and robustness during tracking, as well as relocalization after tracking failure. In this research, ARTIST does not require system wide initialization, such as the one in PTAM. However, there is initialization for each object to be tracked, which consists of the algorithmic steps of feature detection and matching, as well as robust pose estimation. When the active search mode for trackers by Wagner et al. (2008b) is used, similar feature detection and matching are used to initialize the object to be tracked.

Tracking - This refers to the continuous tracking processes to obtain the camera pose for augmenting the virtual objects. Tracking needs to be as efficient as possible. For ARToolkit, this includes thresholding for the detection of the square markers, and identification of the central pattern and the pose of the marker from the four corners of the marker. For PTAM, this includes active feature search and patch matching using the current pose, followed by outlier rejection and pose estimation using robust estimators. The 3D map is refined and extended with new features and keyframes using sparse bundle adjustment. Efficiency is achieved by separating the map building from the tracking process, and running both in parallel on multiprocessors. In contrast, the tracking component of SIFT, which is targeted towards object recognition and not AR tracking, is computationally more intensive and does not permit real-time operations. However, it is able to provide a suitable framework for a more efficient tracker. For SIFT, tracking starts with a scale space analysis for feature detection and the as-

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

signment of an orientation to each feature. A feature descriptor with a recommended vector length of 128 is computed based on the scale and orientation of the feature. This achieves a degree of scale and orientation invariance for the purpose of finding similar features. The similarity between two features is measured through computing the distance between their descriptors. Hough Transform of the scale, orientation and position is used to estimate the object pose robustly.

For ARTIST, the tracking of an initialized object is done using the Efficient Secondorder Minimization (ESM) (Benhimane and Malis, 2007). This iterative method has a convergence region that is sufficiently large for the tracker to converge to the current pose using the pose in the previous frame as the estimate. There is no requirement for feature detection and pose estimation. As ESM can process several thousand pixels per frame, the tracking process of ARTIST is both efficient and accurate. In order to search for new objects that may appear within the field of view of the camera, the initialization process of ARTIST is required for every frame along with tracking. However, to emphasize the difference between the stages of the augmentation of a single object, the algorithmic steps are grouped into two separate stages from the systems perspective. Furthermore, when only one object is tracked, initialization is not required in every frame and only ESM is performed from frame to frame.

Relocalization- This stage refers to the operations to recover from tracking failure, and is only applicable to trackers that use frame-to-frame tracking, such as PTAM and ARTIST. For trackers that discard all the information from the previous frame and track the object within the current frame, relocalization is performed within every frame
and is not explicitly required. This is a more robust setup and applies to ARToolkit and SIFT. However, PTAM and ARTIST use frame-to-frame tracking in order to reduce the amount of computation per frame. For PTAM, two versions of relocalization are used. The first version is based on Random Tree matching, with an emphasis on the reduction of the training time. A later version (Klein and Murray, 2008b) uses blurred, scaled down versions of the map keyframes for relocalization. For ARTIST, relocalization is essentially initialization after the failure of ESM. The subtle difference is in the priority in which the objects are searched. Due to limited processing time, the number of features that can be matched is limited, which in turn limits the number of objects that can be searched. Therefore, the priority for relocalization of the recently lost objects is higher than that for initialization as the probability of success is higher.

Table 5.1 shows a summary of the preceding discussions.

### 5.2.2 Algorithms Perspective

As each of the four trackers, namely, ARToolkit, PTAM, SIFT and ARTIST uses different algorithms for each stage, the algorithms view is presented in terms of the ARTIST with comparison made with the other trackers where appropriate. An analysis of the trackers based on the algorithms used allows for better appreciation of the design decisions of the ARTIST CV tracker. The main algorithmic steps are: (1) feature detection, (2) feature matching, (3) robust pose estimation and (4) pose refinement.

Feature detection - Features refer to the interesting portions of an image, and the commonly used features are corners, lines and blobs. Feature detection is often required to deal with the large amount of information contained within the images by focusing

Table 5.1: Summary of the stages of tracker operations and comparisons between ARToolkit, PTAM, SIFT and ARTIST

|  | Preparation | Initialization | Tracking | Relocalization |
| :---: | :---: | :---: | :---: | :---: |
| ARToolkit | -calibration -markers | -none | -thresholding -central pattern identification -4 point pose | -none |
| PTAM | -calibration | $\begin{aligned} & \text {-approximate 3D } \\ & \text { map } \end{aligned}$ | -active search -patch matching -outlier rejection -pose estimation and refinement -3D map building | -Random Tree relocalization or <br> -Keyframe matching |
| SIFT | -calibration -SIFT descriptors | -none | -feature extraction <br> -feature matching <br> -Hough transform | -none |
| ARTIST | -calibration <br> -keypoint <br> signature | (per object) <br> -feature detection <br> -feature matching <br> -robust pose estimation | $\begin{aligned} & \text { (per object) } \\ & \text {-ESM } \end{aligned}$ | (for recently lost object) -similar to initialization, but with higher priority |

further processing on features. Another reason is that features have specific characteristics that are tailored to facilitate further processing. For example, the texture around point features is used in PTAM, SIFT and ARTIST for feature matching, followed by algorithms for deriving the camera pose from these points. Other types of features will require different downstream algorithms. Therefore, features are used for achieving high efficiency by obtaining useful information quickly from the images, thus improving the robustness through filtering out unnecessary or confusing information. There are two requirements for feature detection. The first requirement is high speed as feature detection is required to be carried out on the entire image. The second requirement is high repeatability where a particular feature is detected well after changes in position, rotation and scale. For ARTIST, feature detection is based on the FAST-9 (Rosten and Drummond, 2006) detector, with the addition of adaptive thresholding and orientation assignment. This detector satisfies the requirements of speed and repeatability.

Feature matching - This refers to the correspondence problem of finding the feature from the object or scene database that matches a particular feature detected in the image. This step is required for obtaining the pose of the object or scene with respect to the camera position. Generally, the approach is that of pattern recognition, where feature descriptors are computed, and the corresponding features are those with the least difference or distance between their descriptors. For actual CV applications, there are several difficult problems to be resolved. The first problem is finding descriptors that are invariant to changes in the appearance of the features, which include changes in position, orientation, scale and illumination. The second problem involves making

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

the descriptor sufficiently discriminative in order for the matching to be correct. The third is enabling the descriptor to be unaffected by clutter and occlusion. For these reasons, local feature descriptors based on the image patch surrounding point features are commonly used and have been shown to perform well.

The use of local descriptors is crucial for handling clutter but places a limit on the discriminating power of the descriptors. This is because there will definitely be features similar in appearance to the features of interest. This is a limitation that cannot be overcome by having better descriptors. Current feature matching techniques are capable of returning a large proportion of correct matches, along with a large number of outliers (Mikolajczyk and Schmid, 2005). The outliers are mainly the result of clutter, and are features detected in the current image that have been erroneously matched to one of the features in the database. The probability of such occurrences increases with the size of the database.

The main weakness of current feature matching techniques is thus the limited ability to reject false positives. Therefore, it is unlikely that feature correspondence can be solved using local information alone. The problem of feature correspondence requires further processing steps that make use of global information, such as the geometry of the features. Therefore, the focus of feature matching should be on efficiency, scalability and obtaining as high a percentage of true positives, with as little false positives or outliers as possible. For SIFT, the feature descriptors are derived using gradient histograms and are matched using best-bin-first, which is an approximate k-nearest neighbor algorithm. For PTAM, the image patch surrounding is used with warping to account for appearance
changes due to motion and lens distortion. Outliers are limited through the use of active search. For ARTIST, the feature descriptors are keypoint signatures (Calonder et al., 2008), which are local patch descriptors obtained using machine learning techniques that requires little training time. A feature matching procedure using the probabilities of peaks appearing within a signature is proposed in this research. The effects of various parameters on the process of generating signatures are also studied to improve the matching accuracy while reducing the computation time and memory requirement. These are presented in Section 5.3.2.2.

Robust Pose Estimation - Due to the limitation on the discriminatory power of local descriptors, the results from feature matching cannot be used directly to compute the pose. An additional step of robust pose estimation is required. It typically uses global information, such as the geometry of the features, to reject outliers and retain correct matches or inliers. RANdom SAmple Consensus (RANSAC) is commonly used. It achieves robustness by randomly selecting a sample of the feature matches for computing the pose, and checking the pose using the remaining matches. The pose with the highest number of matches in agreement, or consensus, is chosen as the estimated pose. For ARTIST, homography is used for checking the geometry of the features. A random sample of four points is used to compute the homography using the algorithm presented by Hartley and Zisserman (2003), in order to re-project features in the database to the current image. The distance between the re-projected and detected features is used to determine the degree of agreement. As homography is limited to planar surfaces, ARTIST is limited to the tracking of such surfaces. If the 3D point positions are

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

known, such as the 3D map in PTAM, RANSAC can be performed using samples of three points. A smaller sample size reduces the probability of the inclusion of outliers and thus reduces the number of RANSAC trials required to obtain the correct pose.

As pointed out by Lowe (2004), the performance of RANSAC greatly deteriorates with increasing proportion of outliers. For example, when half of the matches are correct, one in sixteen samples of four contains no outliers. If only one in ten is correct, the chances of having no outliers are greatly reduced to one in ten thousand. To overcome this, Lowe (2004) used scale, orientation and position for Hough transform to obtain clusters of feature matches with good geometric agreement, and only three matches are required compared to twenty, which is typically required for homography with RANSAC. Wagner et al. (2008b) used three tests for outlier removal. Unlike SIFT, scale information is not available, and only the orientations of potential matches are used to find the dominant orientations to filter out the first set of outliers. The second test used up to 30 lines formed using pairs of features with the best matching scores to check whether the remaining features are on the same side of each line for both the reference and current images. Pairs of features where more than half of the features are mapped to the wrong side of the line are rejected. The third test is RANSAC using homography. For ARTIST, the outlier removal tests described by Wagner et al. (2008b) are implemented to achieve additional robustness during initialization.

Pose Refinement - In order to achieve highly realistic and natural augmentation, the tracker must provide highly accurate camera pose. Trackers based on local features typically exhibit high jitters, which are due to noise and sensitivity of the pose
algorithms to minute inaccuracies of the feature positions. One method to overcome the effects of noise is to use a large number of features, which is not always possible. Each pixel on the camera image sensor measures the intensity of light from the points in the scene. One effective method to measure the accuracy of the computed pose is to measure the intensity error between the actual sensor values and those predicted using a model. The intensity error is in turn measured using the mean square error, and the goal is then to find the pose that minimizes it. Due to the non-linear nature of the projection equations and orientation equations in the camera pose, the minimization problem is non-linear. However, many CV trackers employ linear systems or first-order iterative methods, such as Gauss-Newton, due to limitations in the processing time. Second-order methods give better results, but typically require the computation of Hessian, which is prohibitively expensive to compute.

Recent advances in the development of efficient algorithms for second-order minimization have led to increasing use of such methods in CV trackers, and have been shown to be essential for achieving accurate pose. One example is the bundle adjustment (Engels et al., 2006) used in PTAM. For ARTIST, the ESM (Benhimane and Malis, 2007) is used. Although ESM is different from bundle adjustment, both serves the same purpose to refine the pose given a larger number of measurements, and such methods are the main contributors to the accurate and jitter-free augmentations achieved. However, the main limitation of such methods is that they are iterative, and tend to find the local minimum point instead of the global one. Therefore, they require good initial points to avoid converging to the wrong local minima. For ARTIST, the

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

ESM has a convergence region that is sufficiently large for frame-to-frame tracking, allowing for preceding algorithms to be omitted after the pose is initialized.

In the final analysis, the combination of several algorithms is required to solve the problem of obtaining the camera pose. Each algorithm is required to achieve the goals in AR tracking and compensate for the weak points of the other algorithms. In summary, feature detection is used for filtering out important information from the large amount of image data for efficient downstream processing. Feature matching is required to find the corresponding features in the database, with efficiency as the main consideration. Due to limited information from the localized features as well as problems introduced in the imaging process and the real world scenes, both feature detection and feature matching are not reliable or accurate. Therefore, robust pose estimation is introduced to overcome such problems. However, as the pose is estimated using local features, it is often not accurate and suffers from jitters. Thus, a final pose refinement step is required. Recent CV trackers and ARTIST show that this is an effective framework for solving the tracking problem. Figure 5.1 shows a summary of the systems view of ARTIST and an overview of the algorithms running within each stage of the tracker.


Figure 5.1: Summary of the computer vision module of ARTIST and an overview of the algorithms running within each stage of the computer vision tracking operation

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

### 5.3 Computer Vision Tracker Components

This section details the development and test of the algorithms, namely feature detection based on FAST-9, a proposed keypoint signatures matching method termed as the peak probabilities method, and ESM. The focus is on the modifications and additions to the original algorithms to enable each algorithm to function well as part of the system, and achieve the goals of ARTIST.

### 5.3.1 Feature Detection

For ARTIST, the features are detected using FAST-9, which have been shown to have high efficiency and repeatability (Rosten and Drummond, 2006). For the experimental platform used in this research, which consists of an Intel Core 2 Duo 2.4 GHz processor, an image with a resolution of 512 x 384 pixels can be processed in two milliseconds. The detector works by examining a ring of 16 pixels around the pixel being tested. An illustration similar to Figure 1 of the paper by Rosten and Drummond (2006) is shown in 5.2. A pixel is deemed to be a FAST-9 feature if at least nine of the sixteen ring pixels have pixel intensities that differ from that of the central pixel by a predefined threshold. In Figure 5.2, pixels one to six and twelve to sixteen have a large difference in brightness, while pixels seven to eleven have similar brightness. The original design of FAST requires at least twelve ring pixels to be different for the central pixel to be a feature. However, using machine-learning techniques, both the efficiency and repeatability of FAST can be improved. The authors reported that FAST-9 performs well in both aspects. FAST was also used in PTAM and in the work by Wagner et al.
(2008b).


Figure 5.2: The arrangement of a ring of 16 pixels for detection of features in FAST. (Similar to Figure 1 in paper by Rosten and Drummond (2006) )

There are two additions made to feature detection in this research to enable better feature detection in ARTIST. They are adaptive thresholding and orientation assignment.

### 5.3.1.1 Adaptive thresholding

The FAST-9 detector uses a threshold for determining whether a ring pixel is different from the central pixel. The use of a fixed threshold presents a significant problem for using ARTIST in real world applications where there are large variations in illumination and contrast. If the threshold is set too low, the number of features returned is high, which results in longer processing time and lower robustness due to the increase in the probability of similar features. In contrast, when the threshold is too high, good features may be omitted due to poor contrast, non-optimal illumination conditions, and/or camera image processing. The number of features returned may also be too

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

low for RANSAC to succeed, as a minimum number of inliers are required. Therefore, adaptive thresholding is required to adapt to changing conditions.

Adaptive thresholding is implemented by adjusting the threshold to achieve a target number of features returned. If the number of features detected in the current frame is above the target, the threshold for the next frame is increased in proportion to the number of features in excess of the target. The converse is true when the number of features is below the target. However, a minimum threshold is used, which is set at 25 in this research, in order to prevent superfluous responses due to noise. For real world tracking, a global adaptive threshold of the entire image is ineffective because the image can have several regions with different contrast and feature density levels. An example is the case where two identical objects are present in the image except for the contrast. The use of a global threshold results in the object with the lower contrast to have a lower number of detected features, which can lead to failure of robust pose estimation, as a minimum number of features is required. It is possible to mitigate this issue effectively by dividing the image into sub-grids, and applying adaptive thresholding independently to each sub-grid. Ideally, the number of features detected is similar for both objects in the above example and the latter stages of the CV module are not affected by contrast. In this research, dividing the 512x384 image frame into $8 \times 6$ sub-grids with a target of twenty FAST-9 features each is found to work well. Figure 5.3 shows a comparison of the thresholding methods. Adaptive thresholding gives a relatively uniform distribution of features and takes up little computational resources.

(a) No adaptive thresholding

(b) Global adaptive thresholding (Less features for low contrast surface on the left)

(c) $8 \times 6$ sub-grid adaptive thresholding (Improved distribution of features)

Figure 5.3: Comparison of the FAST-9 feature (red boxes) detection for the same
image frame for cases with (a) no adaptive thresholding, (b) global adaptive thresholding and (c) sub-grid adaptive thresholding.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

### 5.3.1.2 Feature orientation assignment

For SIFT, the features are assigned orientations so that the feature descriptors can be computed relative to these orientations to achieve rotational invariance. In ARTIST, the feature descriptors are keypoint signatures computed using the Semi-Nave Bayesian machine learning structures known as Ferns (Özuysal et al., 2007). There is no requirement for orientation assignment as the Ferns can be trained to recognize features in various orientations. There are two reasons for assigning feature orientation. The first reason is to increase the distinctiveness of the keypoint signatures by orienting the binary point tests (Section 5.3.2.1) in the Ferns about the orientation of the feature. This has been used by Wagner et al. (2008b) to reduce the memory requirement. This has also been used in ARTIST to reduce the number of Ferns required and speed up feature matching. The second reason is that the orientation information can be used for outlier removal, thus allowing the elimination of matches where the change of orientation is inconsistent with the majority of the matches. This is used in SIFT for object recognition and by Wagner et al. (2008b).

In Figure 5.3, each FAST feature is marked by a small red box. The orientation of each feature is shown by rotating the small red box and drawing a short line from the center of the feature along the dominant direction. Some features may have more than one of such lines due to the possibility of assignment of multiple dominant directions.

In ARTIST, the orientation is assigned using the dominant gradient direction in a manner similar to that in SIFT. There are several differences in the implementation from the original method. First, the gradients, $g_{x}$ and $g_{y}$, are computed using the $3 \times 3$

Prewitt operators $\left(\begin{array}{lll}1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1\end{array}\right)$ for $x$-axis, and $\left(\begin{array}{ccc}1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1\end{array}\right)$ for $y$-axis respectively, instead of the central difference operators, $(10-1)$ and $\left(\begin{array}{c}1 \\ 0 \\ -1\end{array}\right)$. The Prewitt operators are essentially multiple central difference operators and reduce the effects of noise. The second modification is the patch used for orientation assignment is fixed at $15 \times 15$ and centered at the FAST-9 feature, similar to the method used by Wagner et al. (2008b), as FAST-9 does not provide scale information. However, the weighting function is not found to improve the consistency of the orientation assignation significantly and thus omitted. The remaining operations are similar to SIFT. For pixel $\boldsymbol{p}$, the magnitude $m(\boldsymbol{p})$ and orientation $\theta(\boldsymbol{p})$ are computed using the following formulae,

$$
\begin{align*}
& m(\boldsymbol{p})=\sqrt{g_{x}^{2}+g_{y}^{2}}  \tag{5.1}\\
& \theta(\boldsymbol{p})=\arctan \left(\frac{g_{y}}{g_{x}}\right) \tag{5.2}
\end{align*}
$$

For each pixel, $m(\boldsymbol{p})$ is added to one of the 36 bins determined using the orientation $\theta(\boldsymbol{p})$. The bin with the highest sum of magnitudes is chosen as the dominant orientation. If there are other bins that have values more than 0.8 of the highest bin, the feature is assigned additional orientations. This can be observed in Figure 5.3, where some features have multiple boxes with different orientations. However, features with more than three dominant orientations are omitted from further consideration. As orientation assignation is not perfect, allowing multiple orientations for features improves the robustness and increases the chances of correct matches. This is due to the

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

occurrence of non-detected or erroneous orientations assigned in certain camera poses. Figure 5.4 shows three images of the side of an apartment with the camera undergoing $z$-axis rotation. As incorrectly assigned features are detected as outliers by robust pose estimation, and no degradation in final tracker performance has been observed in this research, the quantification of the accuracy of orientation assigned is not attempt here.


Figure 5.4: A sequence of three images for visual illustration of the stability of orientation assignment used in ARTIST.

### 5.3.2 Feature Matching

Keypoint signature, which is first proposed by Calonder et al. (2008), is used in ARTIST for feature matching. It avoids the requirement of long training time, which is one of the main limitations of earlier feature matching methods based on RT (Lepetit and Fua, 2006) and Ferns classifiers (Özuysal et al., 2007). Calonder et al. (2008) observed that the response of the RT classifier for a feature not in the training set consists of several stable peaks. These peaks are stable to rotation, illumination and limited scale changes, thereby forming a sparse signature for feature matching. In this research, the Ferns classifier replaces the RT classifier, as the memory requirement is lower without
significant decrease in classification performance. The proposed method is named the Generic Ferns, analogous to Generic Trees (Calonder et al., 2008).

### 5.3.2.1 Ferns

A Fern (Özuysal et al., 2007) is a classifier that consists of a set of tests on the pixel values of the region surrounding a feature, to determine whether this feature belongs to a class consisting of various warps of the feature in the database. The tests are binary, i.e., the result is either true (1) or false (0). In ARTIST, each test consists of a comparison between the intensity of two randomly chosen pixels within a circular region with a diameter of 41 pixels centered at the feature. As comparisons of single pixel values are susceptible to noise, the feature region is Gaussian filtered to reduce the effects of noise. A small $3 \times 3$ filter $\left(\begin{array}{ccc}1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1\end{array}\right)$ is used in ARTIST to keep the computational load low. As there are 36 orientation bins, the points in the binary tests of the Ferns are pre-rotated at $10^{\circ}$ intervals. This allows the Fern tests to be carried out in a manner that is relatively invariant to orientation changes, and reduces the number of Ferns required to encode the variations in the patch appearance due to orientation changes. The result of each binary test determines the value of a binary digit of an index. For example, when there are ten tests in a Fern, the first test sets the first bit to zero or one depending on the outcome of the test, the second test for the second bit and so on. The resulting number is used to represent the combination of pixel values, which gives a particular set of results for the tests in a Fern. Figure 5.5 illustrates how the index is obtained.

The training of the classifier is similar to that described by Lepetit and Fua (2006).

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES



Figure 5.5: An illustration of the Fern testing process on a feature

For ARTIST, the feature region is artificially warped using homographies instead of affine transforms. The warped feature region is tested to obtain the index and the occurrence of each particular index is counted. After a certain number of training samples, generally around 10000 , the number of occurrence can be converted into probabilities. As trained features have different appearance, the probability of each index is different for each feature. During classification, the feature region is tested to obtain an index, and the trained feature with the highest probability is selected as the matching feature.

The number of tests in each Fern determines the amount of memory required to run the classifier. If there are ten binary tests, there are 1024 indices. The amount of memory required for storing the probabilities doubles with every additional test. The number of tests is typically limited to eighteen, which is not sufficient for feature
matching as only a small number of pairs of pixels are tested. Therefore, several Ferns are used and the probability from each Fern is multiplied together to obtain the final probability (Özuysal et al., 2007). For ARTIST, the amount of memory required is greatly reduced by storing only indices with non-zero probabilities. This would typically reduce the memory required to a quarter of the amount dictated by the number of tests. This large reduction is due to the fact that not all indices are possible for a particular feature even with a large number of warps. As memory accesses are relatively slow for modern computers, reducing the memory used also results in an increase in the computational speed. The Ferns are trained with randomly selected features, and are used as the generic ferns for generating the keypoint signatures described in the next sub-section.

## Keypoint signatures

The keypoint signature $s$ for a keypoint is a vector where each element $s_{i}$ is the response for the $i$-th base class in the training set of the generic ferns. Intuitively, the keypoint signature method reuses a classifier, which takes a long time to build, for classes that it is not trained for. Each class in this case is the set projective warps of image patches around a point feature. Therefore, the keypoint signature method relies on the insight that the output of a classifier for all its classes is a measure of the similarity of any point features to each of the classes that it has been trained for. It turns out that approximately 300 randomly selected features allow the classifier to discriminate other point features (Calonder et al., 2008). For each incoming image, the signature of the image keypoints $s_{\mathrm{I}}$ are computed once and matched against the signatures of the objects

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

keypoints $s_{\mathrm{O}}$. This forms the basis for feature matching.

In the ideal case, all responses, $s_{i}$, in the signature of any point feature remain constant, regardless of the projective transformation. This assumes that the classifier outputs a constant similarity measure for the point feature regardless of the projective warping, which is not true for practical Ferns classifiers. Furthermore, to meet real time requirements, a lower number of Ferns are used, which results in further increases in variability of values of $s_{i}$. This increase in variation of the signature of a point feature in turn increase the chances of mis-classification or false positives. To gain some intuition into these variations, a random keypoint is selected and warp 100 times using random homographies. The signatures obtained are then superimposed to illustrate the variations, as shown in Figure 5.6. The signature of each projective warp is plotted as a line of a single color. To enable easier visualization of this variation, the logarithms of $s_{i}$ are used and normalized so that the signature vector mean is one. Furthermore, the plot is translated such that the minimum value is zero, and only $s_{1}$ to $s_{50}$ are shown. The superposition shows that the signature peaks are stable. However, the superposition does not produce an ideal thin line, and lines of different colors spreads out, particularly at the peaks. As the $s_{i}$ values are not constant relative to one another, the position of color under each peak is not constant and this variations cannot be removed by simple normalization. The different colors at the top of each peak show that the ordering of each peak values is not sufficient for discrimination. This seemingly random variations limits the effectiveness of keypoint correspondence by direct use of the nearest neighbours algorithms.

A similar issue has been described by Wagner et al. (2008b) for their modified version of SIFT matching. They observed variations in descriptor values for each feature that affect matching results, due to modifications to SIFT for real time operations. Their solution was to use Spill Trees, which are essentially $k$ - $d$ trees where the value used to determine the branch to traverse is a range of values instead of a single value. This allows both branches to be searched in the case where the tested descriptor value is near to the decision boundary. For ARTIST, peak probabilities method proposed in the next section similarly provides a way to handle variations. However, encoding variations using peak probabilities avoids the requirements of several megabytes of memory required by the Spill Trees.

## Proposed signature matching method

The peak probability method is proposed in this research to handle the variation in the signatures without requiring increase in classifier complexity and run time. The primary idea is to treat the variation of each $s_{i}$ value in the signature as a random occurrence with a fixed but unknown distribution. The peaks probability method relies on the intuition that certain $s_{i}$ values will have larger value most of the time, which correspond to the peaks in Figure 5.6. In other words, the feature in concern is similar to some of the features classes that the classifier is trained on. Statistically, it means that the distribution of $s_{i}$ values for similar classes are skewed towards larger values. If the feature classes that the classifier is trained on are well separated, a random feature is generally similar to only a subset of $k$ feature classes. Let the probability of occurrence of the $i$-th base class in the set of $k$-largest values in each signature, $\boldsymbol{s}_{k}$,


Figure 5.6: Superposition of 100 signatures from random projective warps (only $s_{1}$ to $s_{50}$ shown) of a keypoint $i$. Each signature is plotted as a line of a single color and the spread of colors indicates the variation of the keypoint signature to projective warping.
be $p_{i}$. For each keypoint, as certain base classes occur within $s_{k}$ with a high $p_{i}$ due to high similarity, this is effective for discriminating between keypoints. Therefore, the peaks probabilities method measures jointed distribution of occurrence of large values among all $s_{i}$, rather than the individual distribution of $s_{i}$ values. As classes in the high similarity subset of the keypoint in concern will occur frequently in $s_{k}$, it requires only a relatively small number of training samples to obtain the joint distribution, compared to the original Ferns classifier. Therefore, the proposed method does not require long training time, nor large memory to hold the distributions of individual $s_{i}$. Probabilities $p_{1}$ to $p_{256}$ of the two keypoints are shown in Figure 5.7. They are obtained with random homographies similar to those used for training. Figure 5.8 shows the plot of the change of the values of $p_{i}$ of a keypoint as training progresses. It shows that conservatively, 500 training warps are sufficient. Therefore, in the peak probabilities method, the features


Figure 5.7: The probability $p_{i}$ that a base class occurs in the $s_{k}$ for two features.
detected in the current image are matched to the features on the planar object using

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES



Figure 5.8: The changes of $p_{1}$ to $p_{50}$ of a feature as training progresses.
the latters' jointed distributions. This can be efficiently achieved by using the $s_{k}$ set of the current image features or, $s_{I k}$. This is because, if the features are similar, then the $p_{i}$ values of the object features corresponding to the classes in $s_{\mathrm{I} k}$, will probably be high as well. Therefore, the sum of $p_{i}$ of these classes, denoted as response, $r$, will be of large value for the matching object feature. Here, the object feature in the database with the maximum value for $r$ is considered as the feature match.

Formally, the training process to obtain the jointed probabilities in the peak probabilities method can be expressed as follows. Let $s_{\mathbf{H}}$ be $s$ of the feature patch warped using homography, $\mathbf{H}$. Let $s_{k, \mathbf{H}}$ be the corresponding $s_{k}$ for this warped patch. Consider $\mathbf{H}^{*}$, as the set of randomly chosen homographies during the training process for warping the features. The probability, $p_{i}$, can be expressed as

$$
\begin{equation*}
p_{i}=\operatorname{Prob}\left(i \in s_{k, \mathbf{H}}\right) \quad \forall \mathbf{H} \in \mathbf{H}^{*} \tag{5.3}
\end{equation*}
$$

Define peak probabilities as, $\boldsymbol{p}=\left\{p_{i}\right\}$. Let $\boldsymbol{p}_{j}$ be the peak probabilities of the $j$-th feature on the planar object in the database. After obtaining $s_{I k}$ of the current image feature, the response $r_{j}$ for the $j$-th object feature is defined as a summation based
similarity measure,

$$
\begin{equation*}
r_{j}=\sum_{i \in s_{k}} p_{i} \quad p_{i} \in \boldsymbol{p}_{j} \tag{5.4}
\end{equation*}
$$

The object feature with the maximum value for $r_{j}$ is chosen as the most probable match to the image feature.

### 5.3.2.2 Experimental determination of Fern parameter values

Due to the requirement for real-time operation, the Fern parameters have to be set such that the feature matching is optimal for both accuracy and efficiency. Important parameters are (1) the number of Ferns and the number of tests in each Fern, (2) the number of training samples, (3) the value of $k$ or the number of peaks in $s_{k}$, (4) the number of base classes for training the generic ferns, and (5) the number of orientation bins. The parameter values are set empirically based on tests using both simulated and real images. The main criterion used is the matching rate, which is defined as the ratio of the correct matches over the number of features to be matched.

The test using the simulated data is also used to generate the generic ferns for the tests using the real images and the actual tracking operations. It is devised such that it is similar to typical operating conditions. Approximately 6500 FAST-9 features from six images with different types of features are used. Figure 5.9 shows the six images used. In order to test the effect of each parameter on the matching performance, random samples of 100 features are selected to simulate the number and types of features detected in typical video images. To simulate the object features, the peak probabilities of the

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

above-mentioned features are obtained as described in Section 5.3.2.1. For matching tests, the feature regions are transformed using 100 different random homographies to simulate the changes in the appearances of the features during camera motion. The average number of correct matches per warp is noted; as it represents the number of correctly matched features for one object for a particular camera position.

The test with real images uses three sets of publicly available test images, namely the Wall, Graffiti and Boat (Available at http://www.robots.ox.ac.uk/vgg/research/affine/). Each test set consists of six images, where the homography between the first and each of the other five images have been accurately determined. Similar to the test performed by Calonder et al. (2008), FAST-9 features are detected with adaptive thresholding and orientation assignment for the first image of each test set. Each feature is projected to the other images in the set using homography, and this prevents the repeatability of the feature detection process from affecting the matching tests. The features in the first image are regarded as object features, and the peak probabilities are obtained as described. The features detected in the other images are matched to those in the first image using the proposed peak probabilities method. Features with multiple orientations are considered as single features, and multiple matches of a single feature due to the several orientations assigned are considered as a single correct match.

For each test set, only images two and three are used for matching with the first image. This is due to the remaining images having viewpoint changes and zoom, which are beyond the range of values used for training the generic ferns. The test images used are shown in Figure 5.10. As the accuracy is dependent on the distribution of


Figure 5.9: The six images used for extracting the features for training of the generic ferns and feature matching tests.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

the feature appearances and the base classes, for both stimulated and real image tests,

10 repetitions are made for each parameter setting and the average matching rate is reported.


Graffiti test set - viewpoint changes with distinct features and well defined edges

(1)
(2)
(3)

Boat test set - rotation and zoom changes with relatively distinct features
Figure 5.10: The three test sets for the matching test using real images.

## Default settings

The default settings for the Ferns are as follows. The number of Ferns is 20 with 12 tests each. The range of the $z$-axis rotation is $0^{\circ}$ to $360^{\circ}$, and the range of the camera tilts, or $x / y$-axis rotation is $-45^{\circ}$ to $+45^{\circ}$ about the vertical. The range of scaling is from 0.75 to 1.5 . Warped images can be translated up to two pixels in the $x$ - and $y$-directions to simulate the inaccuracies of feature detection. The number of training samples is 10000 for Ferns with 10 and 12 tests each. The number of samples is increased to 15000 and 20000 for Ferns with 14 and 16 tests respectively. The number of base classes is 300. Feature region has a size of $41 \times 41$ and value of k is 15 . The number of orientation bins used for assignment is 36 . For the proposed matching method, 500 random warps are used for obtaining the peak probabilities. These settings are used for the actual tracking.

## The number of Ferns and number of tests in each Fern

These two parameters affect the accuracy, run time and memory required for computing keypoint signatures. The number of Ferns is varied from 10 to 30 in steps of 5 , and the number of tests is varied from 10 to 16 in steps of 2 . The matching rate(\%), the average time to compute and match each keypoint signature (run time per feature, msec ), and the memory required ( MB ) for the simulated and real image tests with respect to the number of Ferns and tests per Fern are shown in Table 5.2. The average matching rate for real image tests is shown in brackets below the matching rate for simulated tests. From the results, it can be observed that the matching rate generally

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

increases as more Ferns and more tests are used. This is due to better representation of the underlying distribution of the appearance of a feature undergoing projective distortions. However, the result also indicates that the matching rate stops increasing when more than 14 tests per Ferns are used. The value of 16 represents the upper limit of using the increasing number of tests per Fern to increase the matching rate. Furthermore, the memory requirement will also become increasingly prohibitive for low memory devices. The matching rate for real images are lower than the simulated tests as the real images include a significant amount of noise. The difference in the matching rates is approximately $10 \%$. This will be further explored in Section 5.3.2.3, where the performance of the proposed method is compared with that of the sparse signature (Calonder et al., 2008).

Table 5.2: Variations of matching rate, run time per feature and memory required with respect to the number of Ferns and tests per Fern.

| No. of Ferns <br> No. of Tests | 10 | 15 | 20 | 25 | 30 |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  | $65.02 \%$ | $68.71 \%$ | $71.71 \%$ | $72.31 \%$ | $73.95 \%$ |
| 10 | $(55.97 \%)$ | $(60.85 \%)$ | $(62.28 \%)$ | $(64.64 \%)$ | $(66.61 \%)$ |
|  | 0.0259 msec | 0.0341 msec | 0.0427 msec | 0.0509 msec | 0.0596 msec |
|  | 5.65 MB | 9.59 MB | 11.97 MB | 15.02 MB | 18.78 MB |
| 12 | $67.87 \%$ | $70.29 \%$ | $72.77 \%$ | $73.97 \%$ | $75.20 \%$ |
|  | $(57.81 \%)$ | $(61.28 \%)$ | $(65.44 \%)$ | $(66.73 \%)$ | $(68.94 \%)$ |
|  | 0.0234 msec | 0.0314 msec | 0.0384 msec | 0.0468 msec | 0.0526 msec |
|  | 11.80 MB | 17.38 MB | 23.60 MB | 29.96 MB | 34.32 MB |
|  | $66.26 \%$ | $71.35 \%$ | $73.31 \%$ | $74.75 \%$ | $74.33 \%$ |
|  | $(56.49 \%)$ | $(61.60 \%)$ | $(65.14 \%)$ | $(68.40 \%)$ | $(68.12 \%)$ |
| 14 | 0.0229 msec | 0.0292 msec | 0.0370 msec | 0.0426 msec | 0.0488 msec |
|  | 21.96 MB | 33.74 MB | 45.04 MB | 55.82 MB | 66.30 MB |
|  | $64.68 \%$ | $69.84 \%$ | $71.73 \%$ | $73.85 \%$ | $75.86 \%$ |
|  | $(54.88 \%)$ | $(60.34 \%)$ | $(64.45 \%)$ | $(65.91 \%)$ | $(68.01 \%)$ |
| 16 | 0.0222 msec | 0.0288 msec | 0.0352 msec | 0.0417 msec | 0.0496 msec |
|  | 37.76 MB | 58.63 MB | 79.88 MB | 98.60 MB | 115.76 MB |

By storing only the non-zero probabilities, the memory required is reduced. For example, $10 \times 2{ }^{10} \times 4 \times 300=11.72 \mathrm{MB}$ and $10 \times 2{ }^{12} \times 4 \times 300=44.88 \mathrm{MB}$ are the expected memories required for 10 Ferns with 10 and 12 tests respectively. However, the actual memory usages are 5.65 MB and 11.80 respectively. It can also be observed that the memory required for the Ferns generally doubles for every two tests added, instead of quadruples if all the probabilities are stored. Therefore, the method used for ARTIST is effective in reducing the memory required. The run time increases linearly with an increase in the number of Ferns used. It also shows a slight decrease when the number of tests per Fern is increased. This is because each additional test allows the classifier to better distinguish between warped patches. For example, two different patches may have the same index with the ten point-pair tests of a Fern. However, with an additional test that compare a different pair of points, the two patches can have different indices. Therefore adding tests increase the ability of the classifier to discriminate between patches. Furthermore, as patches that previously had the same index can have different indices due to additional tests, there will be less warp patches assigned to each index and therefore less non-zero probabilities per index. As only non-zero probabilities are processed for computing the signature, there is a decrease in the computational time. This behavior implies that it is feasible to trade memory for speed, for systems with large memory. However, from the results, it can be observed that there is no significant gain in performance when more than 14 tests are used.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

## Number of training samples

A sufficient number of random warps are required for training the generic ferns so that the probabilities obtained are an accurate representation of the actual distribution of the patch appearances. The effects of the number of training samples on the matching rate, run time and memory usage are examined for the default case of 20 Ferns with 12 tests. Table 5.3 shows the results, where the average matching rate for the real image tests are shown in brackets.

Table 5.3: The variation of matching rate, run time per feature and memory required, with respect to the number of random warps.

| No. of random warps | 1000 | 5000 | 10000 | 15000 | 20000 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Matching rate | $70.04 \%$ | $72.62 \%$ | $72.77 \%$ | $72.69 \%$ | $72.12 \%$ |
|  | $(61.94 \%)$ | $(65.37 \%)$ | $(65.44 \%)$ | $(65.63 \%)$ | $(64.89 \%)$ |
| Time per feature (msec) | 0.0320 | 0.0372 | 0.0384 | 0.0402 | 0.0405 |
| Memory(MB) | 9.45 MB | 18.36 MB | 23.60 MB | 26.32 MB | 28.48 MB |

From the results in Table 5.3, it can be observed that the matching rates for both the tests using the simulated and the real images do not increase with more than 5000 random warps. This implies the default value of 10000 is conservative and more than sufficient to ensure good matching rates. As Ferns are trained with more samples, more indices will have non-zero probabilities, which results in greater memory usage and computational time. The memory usage flattens with an increasing number of random warps. This shows that additional random warps are no longer activating new indices. This implies that most of the possible indices that are possible outputs of the Ferns tests for all possible warp patches belonging to a feature have been obtained, and the training is more complete. However, the matching rate results show that the
training does not need to be complete for the Ferns to perform well. This implies with randomness, the indices probabilities of the Ferns would reach their expected values with just 5000 samples. This also implies that additional indices reached using a greater number of random samples would generally have low probabilities and do not improve the matching rate. Therefore, it is possible to reduce the memory usage significantly by using just 5000 warps instead of 10000 .

## The value of $k$ or number of peaks in $s_{k}$

Strictly, $k$ is not a Fern parameter. Its effects on the matching rate and run time are studied here. As we are concerned with the effects of varying the value of $k$, only one set of generic ferns is used for all the tests to remove the effects of random sets of base classes on the matching rate. Each value of $k$ is tested with 10 repetitions to obtain the average. The results are presented in Table 5.4. The average matching rate for real image tests are shown in brackets.

Table 5.4: The variation of matching rate and time per feature with respect to the value of $k$.

| $k$ | 1 | 5 | 10 | 15 | 20 | 25 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Matching rate | $44.17 \%$ | $67.10 \%$ | $71.46 \%$ | $73.47 \%$ | $72.53 \%$ | $72.73 \%$ |
| $(25.86 \%)$ | $(56.15 \%)$ | $(62.62 \%)$ | $(63.94 \%)$ | $(64.14 \%)$ | $64.40 \%$ |  |
| Standard <br> deviation for <br> simulated test | 2.85 | 2.20 | 2.32 | 1.52 | 2.71 | 2.57 |
| Time per <br> feature (msec) | 0.0293 | 0.0329 | 0.0364 | 0.0385 | 0.0400 | 0.0426 |

From Table 5.4, it can be seen that the matching rate does not show significant increase when $k$ is 10 or more. However, the value of 15 is chosen as the default because

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

the matching rates have a lower standard deviation, which implies that the matching performance is more stable. Examination of results from real image tests shows that using $k=15$ provides a good balance between speed and robustness to variation of feature types, as a greater number of peaks is considered. Therefore, despite the slightly higher run time, $k=15$ is used as the default for the current design and setting used for generic ferns.

## The number of base classes

The effect of the number of base classes on the matching accuracy is investigated. The tests conducted by Calonder et al. (2008) showed that using more than 300 randomly chosen features did not significantly increase the matching accuracy. Therefore, the range of test is from 200 to 500 , in steps of 50 . Ten repetitions are made for each test case and the average results are shown in Table 5.5 , where the average matching rate for real image tests are shown in brackets. There is a general increase in the accuracy as the number of base classes is increased. However, the rate of increase reduces as the number of base classes is increased, while the run time and memory usage continue to increase in a relatively linear manner. Therefore, using more than 300 base classes does not significantly improve the matching rate, and the increase in run time is prohibitive for real-time tracking.

## Number of orientation bins

The number of orientation bins that each feature can be assigned to is varied to study its effects on the matching rate, run time and memory usage. The results in Table

Table 5.5: The variation of matching rate, time per feature and memory usage, with respect to the number of base classes.

| No. of <br> base <br> classes | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Matching <br> rate | $68.03 \%$ <br> $(59.96 \%)$ | $70.01 \%$ <br> $(62.13 \%)$ | $72.77 \%$ <br> $(65.44 \%)$ | $73.73 \%$ <br> $(66.79 \%)$ | $74.53 \%$ <br> $(68.04 \%)$ | $75.62 \%$ | 76.43 |
| Time per <br> feature <br> (msec) | 0.0299 | 0.0351 | 0.0384 | 0.0421 | 0.0456 | 0.0506 | 0.0536 |
| Memory <br> (MB) | 15.68 | 19.68 | 23.60 | 27.36 | 31.18 | 35.42 | 38.90 |

5.6 show that the matching rate increases while the run time and memory decrease as more orientation bins are used. However, no further significant improvements are observed when more than 36 bins are used. This is because the orientation assignment scheme has limited resolution and repeatability. The matching rate for real image tests shows a decrease when a large number of orientation bins is used, which implies that the orientation assignment has become unstable when 48 or 60 bins are used. Among the various parameters tested, increasing the number of orientation bins is the only one that increases the matching rate without an increase in run time and memory; the run time and memory are reduced instead.

### 5.3.2.3 Comparison with the sparse keypoint signature

This section compares the performance of the proposed peak probabilities matching method with the sparse signature method (Calonder et al., 2008). The sparse signature is computed using the same generic ferns as in the peak probabilities method, utilizing the non-wrapped features in the simulated tests and the features in the first image of

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

Table 5.6: The variation of matching rate, time per feature and memory usage, with respect to the number of base classes.

| Region <br> size | 1 | 4 | 9 | 18 | 36 | 48 | 60 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Matching <br> rate | $44.38 \%$ | $58.02 \%$ | $64.93 \%$ | $69.72 \%$ | $72.77 \%$ | $73.01 \%$ | 72.04 |
| Time per <br> feature <br> (msec) | 0.0457 | 0.0411 | 0.0393 | 0.0389 | 0.0384 | 0.0372 | 0.0379 |
| Memory <br> (MB) | 53.04 | 35.22 | 30.62 | 26.60 | 23.60 | 22.16 | 20.56 |

each real image test set. The sparse signatures have an average length of more than 150, which is sufficiently long to achieved good matching rates. Although FAST features are used here instead of Difference of Gaussians (Lowe, 2004), the comparison is still considered fair as the same features are used in both methods and the focus is on the improvements from using peak probabilities to represent the variations of the keypoint signatures. The results from the Wall test sets also show that the performance reported here is consistent with those reported by (Calonder et al., 2008). Figure 5.11 shows the average matching rates for both the simulated and real test images, where Ferns with 12 tests each are used for both methods. A range of 10 to 25 Ferns is tested. Figure 5.12 shows the breakdowns of the matching rates for each set of test images, and they are averaged to give the matching rates for the real test images in Figure 5.11.

The plots in Figure 5.11 show that on average, the matching rates achieved using the proposed peak probabilities method is higher than those using the sparse signatures. The main motivation for using the peak probabilities method is to overcome the variations of the sparse signatures due to projective warps. The larger difference in


Figure 5.11: The comparison of matching rates for the proposed peak probabilities method and the sparse signatures method for simulated and real test images.


Figure 5.12: The comparison of matching rates for the proposed peak probabilities method and sparse signatures for real image test sets.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

the matching rates for the simulated test is due to the larger variability in the image warps used, compared to variation of feature appearance changes in the real image. The range of possible image warps is the same as those used for training the generic ferns. Hence, it can be concluded that the peak probabilities are more stable than the sparse signatures over a range of projective warps. Furthermore, even with lower variation in the feature appearance and the presence of noise in real image, the peak probabilities still outperform the sparse signatures.

The results presented in Figure 5.12 provide a better understanding of the differences in performance characteristics of these two methods. For the Graffiti and Boat data sets, the features are more distinct than those in the Wall data set, and the peak probabilities perform significantly better than the sparse signatures, especially when a lower number of Ferns is used. The differences in the matching rates are greater for Graffiti images 1 to 3 and Boat images 1 to 3 , as compared to Graffiti images 1 to 2 and Boat images 1 to 2 , as the third image of each test set is taken with larger changes in the camera pose. This adds further support to the claim that the peak probabilities improve the performance of the keypoint signatures over a wider range of camera poses. For the Wall data set, the matching rate of the sparse signatures is slightly better that those of the peak probabilities. This is likely due to the less distinctive nature of the features in the Wall data set. The use of peak probabilities appears to reduce the distinctiveness of keypoint signatures slightly, resulting in no improvement over the sparse signatures for the Wall data set. However, as the improvement in the matching rate for the other two data sets is more significant, the proposed peak signature method is
deemed to have improved the overall performance of keypoint signatures.
The Wall data set used in the experiment is the same as that used by Calonder et al. (2008) for performance evaluation. The recognition rates that are equivalent to the matching rate reported by Calonder et al. (2008), for Wall images 1 to 2 and Wall images 1 to 3 are approximately $70 \%$ and $60 \%$ respectively. From the matching rates achieved for Wall images 1 to 2 and Wall images 1 to 3 shown in Figure 5.12, it is assumed that the performance of the sparse signatures implemented in this research is consistent with that by Calonder et al. (2008). As the settings for the Random Trees used by Calonder et al. (2008) are not reported, and the type of features used are different, it is not viable to replicate their work in good faith. As the sparse signature has similar performance to SIFT, it implies that the performance of the proposed matching method is also equivalent to SIFT, which is one of the best performing local feature descriptors. The results reported in this research are also consistent with the matching performance reported by Wagner et al. (2008b), which is an average of approximately $60 \%$ for the simulated and real image tests.

### 5.3.3 Pose Refinement

The pose obtained using robust pose estimation typically gives rise to jitters, which greatly affect the AR user experience. Therefore, pose refinement using second-order optimization is required for higher accuracy and lower jitter, so that the virtual objects appear realistically attached to the real world. The ESM developed by Benhimane and Malis (2007) is used in ARTIST. The two main advantages of ESM are high efficiency and large convergence region. The code developed for ARTIST is capable of performing

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

ESM for three objects in real-time. Furthermore, the pose can be refined using the pose from the previous frames, and only ESM is executed for every frame. The ESM is also found to be able to converge using the estimated poses obtained from robust pose estimation thus, allowing for all the algorithms to function together as a single system.

This section presents the tracking of planar surfaces using ESM. The contributions made in this research to ESM are (1) improving the robustness to radial distortion common in wide angle lens, and (2) sub-dividing the reference image into sub-grids for organizing the pixels. This organization strategy is used instead of processing them as a single large patch, or as individual pixels. The proposed sub-grid organization allows for (3) selecting of sub-grids within the ESM reference image that improves the behavior of the ESM iterations. This is achieved by using only sub-grids where the average magnitude of image intensity gradient is higher than a predefined threshold. This is based on the observation that the image gradients are the main information for controlling the ESM iterations. Although it is possible to select individual pixels with high image gradient for this purpose, it is not suitable due to sensitivity of image gradients to image noise and increased complexity to manage a large number of pixel individually. The sub-grid organization also allows for (4) a simple and efficient illumination model for maintaining low ESM error in the presence of illumination changes due to shadows, glares and ambient lighting. The proposed model is shown in this research to be more accurate than the discrete illumination model (Silveira and Malis, 2007), while being more computationally efficient. Furthermore, the combination of the sub-grid organization and good accuracy of the illumination model greatly simplifies the detection
of occlusion, by using a simple threshold. These contributions are presented in detail in the following subsections. The notations and formulation of the ESM are generally faithful to the original work by Benhimane and Malis (2007). The main departures are the addition of the radial distortion (Eq. 5.6 to 5.8 ) in to the image warping model in Section 5.3.3.1, and subsequently the derivation of the Jacobian matrix (Eq. 5.18 to 5.21) in Section 5.4.3.2.

### 5.3.3.1 Image warping using homography

The main objective of ESM is to minimize the error between the reference image and the current image, which has been warped using the current camera pose. For images of a planar surface, the $3 \times 3$ homography matrix describes the perspective transformation from one view to another. This warping process is illustrated in Figure 5.13 and 5.14, using a single video sequence. The first figure shows the reference image, where the tracked region is enclosed in a blue square. The latter figure shows two of the subsequent frames, where the tracked region is still enclosed with blue borders, which are no longer square due to camera motion. Figure 5.14 also shows the result of the current image warped using the current homography and the error between the reference and warped images.

In the following, notations, parameters with a superscript asterisk * refer to parameters in the reference image. Let $\boldsymbol{p}=\binom{u}{v}$ be the pixel position in the image coordinates, where the pixel is at the $u$-th row and $v$-th column of the image. Now, $p$ refers to the current pixel while $\boldsymbol{p}^{*}=\binom{u^{*}}{v^{*}}$ refers to the pixel in the reference image. To obtain

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES



Figure 5.13: For this figure and Figure 5.14, the reference image is the tracked region is enclosed within the blue square.
the transformation between the reference and the current images in the presence of radial distortion and perspective warping, the following steps are required. First, $\boldsymbol{p}^{*}$ is transformed to the normalized reference image coordinates $\boldsymbol{m}_{\mathrm{d}}^{*}=\binom{y_{\mathrm{d}}^{*}}{x_{\mathrm{d}}^{*}}$ using the camera intrinsic parameters, as shown in Eq. 5.5.

$$
\begin{equation*}
\binom{u^{*}}{v^{*}}=f\binom{y_{\mathrm{d}}^{*}}{x_{\mathrm{d}}^{*}}+\binom{u_{0}}{v_{0}} \tag{5.5}
\end{equation*}
$$

where $f$ is the focal length of the lens in pixels and $\binom{u_{0}}{v_{0}}$ is the principal point in pixels. The undistorted pixel position $\boldsymbol{m}_{\mathrm{u}}^{*}=\binom{y_{\mathrm{u}}^{*}}{x_{\mathrm{u}}^{*}}$ of $\boldsymbol{m}_{\mathrm{d}}^{*}$ is obtained using the radial distortion model. First the radial distance $r_{\mathrm{d}}$ from the image principal is computed using Eq. 5.6.

$$
\begin{equation*}
r_{\mathrm{d}}=\sqrt{\left(y_{\mathbf{u}}^{*}\right)^{2}+\left(x_{\mathbf{u}}^{*}\right)^{2}} \tag{5.6}
\end{equation*}
$$

Next, the distortion factor $\rho$ is computed using Eq. 5.7.

### 5.3 Computer Vision Tracker Components



Two frames from the video sequence of Figure 5.13, the tracked region is bound by a blue border

(Warped image 1)

(Warped image 2)

The warped images generated using the tracked region (blue border) and current homography

(Error image 1)

(Error image 2)

The error between the warped image and the reference image - brighter areas indicate greater error
Figure 5.14: Examples of the image warping process. The warped image of example
2 has greater blur and errors, due to greater change in scale than example 1.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

$$
\begin{equation*}
\rho=1+K_{1} r_{\mathrm{d}}{ }^{2}+K_{2} r_{\mathrm{d}}{ }^{4} \tag{5.7}
\end{equation*}
$$

Where $K_{1}$ and $K_{2}$ are the two radial distortion coefficients obtained using the camera calibration routine in OpenCV (2010). Finally, the undistorted position is computed using Eq. 5.8.

$$
\begin{equation*}
\binom{y_{\mathrm{d}}^{*}}{x_{\mathrm{d}}^{*}}=\rho\binom{y_{\mathrm{u}}^{*}}{x_{\mathrm{u}}^{*}} \tag{5.8}
\end{equation*}
$$

Next, $\boldsymbol{m}_{\mathrm{u}}^{*}$ is related to the 3D position of the point $\boldsymbol{X}^{*}=\binom{Y^{*}}{X^{*}}$ using Eq. 5.9.

$$
\begin{equation*}
\binom{y_{\mathrm{u}}^{*}}{x_{\mathrm{u}}^{*}}=\binom{\frac{Y^{*}}{Z^{*}}}{\frac{X^{*}}{Z^{*}}} \tag{5.9}
\end{equation*}
$$

Note that the positions of $x^{*}$ and $y^{*}$ are reversed in the pixel vector because the $X$-axis and $Y$-axis in the 3D camera frame are defined to point horizontally towards the right and vertically upwards respectively. Therefore, the camera $X$-axis and $Y$ axis correspond to $v$ (column), and $u$ (row) in the image frame respectively. As the coordinate frame used is right-handed, the $Z$-axis points away from the lens to the image sensor. $\boldsymbol{X}^{*}$ is transformed from the reference camera frame to the current frame using Eq. 5.10, where $\mathbf{R}$ is the rotation matrix and $\boldsymbol{t}$ is the translation vector.

$$
\begin{equation*}
X=\mathbf{R} X^{*}+t \tag{5.10}
\end{equation*}
$$

For the set of points on a plane, the following relationship in Eq. 5.11 holds for rigid camera motion,

$$
\begin{equation*}
\boldsymbol{X} \cong \mathbf{H} \boldsymbol{X}^{*} \tag{5.11}
\end{equation*}
$$

where $\mathbf{H}$ is the $3 \times 3$ homography matrix with determinant equal to one, and it is related to $\mathbf{R}$ and $\boldsymbol{t}$ according to Eq. 5.12 (Benhimane and Malis, 2007), where $d^{*}$ and $\boldsymbol{n}^{*}$ are the perpendicular distance and normal vector, respectively, to the plane in the reference frame. The transpose operator is denoted using ${ }^{T}$.

$$
\begin{equation*}
\mathbf{H}=\mathbf{R}+\boldsymbol{t}\left(\boldsymbol{n}^{*}\right)^{\mathrm{T}} \tag{5.12}
\end{equation*}
$$

As there are $3 \times 3=9$ elements in $\mathbf{H}$ and Eq. 5.11 is defined to scale, there are eight degrees of freedom, namely, three for rotation, three for translation and two for the normal vector that has been normalized. $\mathbf{H}$ is a member of the Special Linear group $\mathbf{S L}(3)$. As the ESM requires small changes in the parameter values in each iteration, $\mathbf{H}$ can be represented using the exponential map that is derived using Lie algebra associated with this group $\operatorname{sl}(3)\left(\right.$ Benhimane and Malis, 2007). Let $\boldsymbol{a}=\left(a_{1}, a_{2}, \ldots, a_{8}\right)^{\mathrm{T}}$ be the $8 \times 1$ vector representing small changes. The exponential map is shown in Eq. 5.13,

$$
\begin{equation*}
\mathbf{H}(\boldsymbol{a})=e^{\sum_{k=1}^{8} a_{k} \mathbf{A}_{k}} \tag{5.13}
\end{equation*}
$$

where $\mathbf{A}_{k}$ are basis matrices in $\mathbf{s l}(3)$ and defined as follows (Benhimane and Malis, 2007).
$\mathbf{A}_{1}=\left(\begin{array}{lll}0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{array}\right)$
$\mathbf{A}_{2}=\left(\begin{array}{lll}0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0\end{array}\right)$
$\mathbf{A}_{3}=\left(\begin{array}{lll}0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{array}\right)$

$$
\mathbf{A}_{4}=\left(\begin{array}{lll}
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0
\end{array}\right)
$$

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

$\mathbf{A}_{5}=\left(\begin{array}{ccc}1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0\end{array}\right)$
$\mathbf{A}_{6}=\left(\begin{array}{ccc}0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1\end{array}\right)$
$\mathbf{A}_{7}=\left(\begin{array}{lll}0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0\end{array}\right)$
$\mathbf{A}_{8}=\left(\begin{array}{lll}0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0\end{array}\right)$

This representation simplifies the derivation of the Jacobian matrices and increases the computational efficiency.

### 5.3.3.2 ESM and computation of Jacobian matrices

The pixel $\boldsymbol{p}$ in the current frame that corresponds to $\boldsymbol{p}^{*}$ is obtained from $\boldsymbol{X}$ using Eq. 5.5, 5.8, 5.9. The transformation between $\boldsymbol{p}$ and $\boldsymbol{p}^{*}$ can be represented as a warping function $w\langle\mathbf{H}\rangle$. If the image brightness constancy assumption holds and $\mathrm{I}^{*}\left(\boldsymbol{p}^{*}\right)$ is the image intensity of the reference image at $\boldsymbol{p}^{*}$, the following relationship in Eq. 5.14 holds,

$$
\begin{equation*}
\mathrm{I}^{*}\left(\boldsymbol{p}^{*}\right)=\mathrm{I}(\boldsymbol{p})=\mathrm{I}\left(w\langle\mathbf{H}\rangle\left(\boldsymbol{p}^{*}\right)\right) \tag{5.14}
\end{equation*}
$$

The homography transforms the current image to match the reference image exactly, i.e., for pixel $i$, each corresponding value in the error vector $\Delta \mathbf{I}$ is zero, as shown in Eq. 5.15.

$$
\begin{equation*}
\Delta \mathbf{I}\left(\boldsymbol{p}_{i}^{*}\right)=\mathrm{I}^{*}\left(\boldsymbol{p}_{i}^{*}\right)-\mathrm{I}\left(\boldsymbol{p}_{i}\right)=\mathrm{I}^{*}\left(\boldsymbol{p}_{i}^{*}\right)-\mathrm{I}\left(w\langle\mathbf{H}\rangle\left(\boldsymbol{p}_{i}^{*}\right)\right)=\mathbf{0} \tag{5.15}
\end{equation*}
$$

However, there are residual errors as it is not possible to model and describe the entire projection process completely. Errors also exist due to random image noise. The objective is to find the set of values of $\boldsymbol{a}$ that gives $\mathbf{H}$, which will result in the least square error. As the transformation function is non-linear, non-linear optimization is
required. The general scheme is to linearize the function around an initial set of values for $\boldsymbol{a}$, and iterate until the least square error is sufficiently small. Thus, the problem is transformed into the modeling of the behavior of the function in the presence of small changes in the parameters. To address this problem, the ESM is developed as a non-linear optimization procedure with a convergence rate similar to the secondorder methods, but with an efficiency of the first-order methods through avoiding the repeated computation of the Hessians. The detailed derivation of ESM is presented by Benhimane and Malis (2007) and the form used in this work is as shown in Eq. 5.16,

$$
\begin{equation*}
\Delta \mathbf{I}=-\frac{1}{2}(\mathbf{J}(\mathbf{0})+\mathbf{J}(\boldsymbol{a})) \boldsymbol{a} \tag{5.16}
\end{equation*}
$$

where $\mathbf{J}(\mathbf{0})$ and $\mathbf{J}(\boldsymbol{a})$ are the current and reference Jacobians respectively. In each iteration in ESM, the Jacobians are computed using the images and Eq. 5.16 is solved using a linear system to obtain $\boldsymbol{a}$. The current estimate of the homography matrix, $\hat{\mathbf{H}}$, is updated from the previous estimate, $\overline{\mathbf{H}}$, using Eq. 5.13 as follows in Eq. 5.17:

$$
\begin{equation*}
\hat{\mathbf{H}}=\overline{\mathbf{H}} \mathbf{H}(\boldsymbol{a})=\overline{\mathbf{H}} e^{\sum_{k=1}^{8} a_{k} \mathbf{A}_{k}} \tag{5.17}
\end{equation*}
$$

Next, $\hat{\mathbf{H}}$ is set as the $\overline{\mathbf{H}}$ for the next iteration. The current image is transformed using these new values and the Jacobians are recomputed to solve Eq. 5.16. This is repeated until the change in error is below a predefined threshold.

The Jacobians are computed for each iteration and differ from those used by Benhimane and Malis (2007) as the radial distortion is modeled. The Jacobians model the effects of small changes in parameter values on the error, and they can be obtained

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

using the chain rule. The forms for both $\mathbf{J}(\mathbf{0})$ and $\mathbf{J}(\boldsymbol{a})$ are shown in Eq. 5.18.

$$
\begin{equation*}
\mathbf{J}=\mathbf{J}_{\mathrm{I}} \mathbf{J}_{m_{\mathrm{d}}} \mathbf{J}_{m_{\mathrm{u}}} \mathbf{J}_{X} \mathbf{J}_{w} \mathbf{J}_{\mathbf{H}} \tag{5.18}
\end{equation*}
$$

$\mathbf{J}$ is a $1 \times 8$ vector for the warping function using homography with respect to $\boldsymbol{a}$. Other than $\mathbf{J}_{\mathrm{I}}$, which is the image gradient, the remaining components are computed using the chain rule. Following the development by Benhimane and Malis (2007), Eq. 5.19 can be obtained.

$$
\begin{equation*}
\mathbf{J}(\mathbf{0})+\mathbf{J}(\boldsymbol{a})=\left(\mathbf{J}_{\mathbf{I}^{*}}+\mathbf{J}_{\mathrm{I}}\right) \mathbf{J}_{\boldsymbol{m}_{\mathrm{d}}} \mathbf{J}_{m_{\mathrm{u}}} \mathbf{J}_{\boldsymbol{X}} \mathbf{J}_{w} \mathbf{J}_{\mathbf{H}} \tag{5.19}
\end{equation*}
$$

Each pixel in the patch results in a row in the Jacobian. For the $i$-th pixel, the structure of the Jacobian in each part of Eq. 5.19 is as follows in Eq. 5.20. The index $i$ is omitted to simplify the notation. All the variable, except $f$ and $\mathbf{J}_{\mathbf{H}}$ are specific to the $i$-th pixel

$$
\begin{align*}
& \left(\mathbf{J}_{\mathrm{I}^{*}}+\mathbf{J}_{\mathrm{I}}\right) \mathbf{J}_{m_{\mathrm{d}}} \mathbf{J}_{m_{\mathrm{u}}} \mathbf{J}_{\boldsymbol{X}} \mathbf{J}_{w} \mathbf{J}_{\mathbf{H}}=\left(\nabla_{p^{*} I^{*}}+\nabla_{\left.w\langle\mathbf{H}\rangle\left(\boldsymbol{p}^{*}\right) \mathrm{I}\right)}\right)\left(\begin{array}{cc}
f & 0 \\
0 & f
\end{array}\right) \\
& \left(\begin{array}{cc}
\rho+2\left(y_{u}^{*}\right)^{2} \frac{d \rho}{d r^{2}} & 2 y_{u}^{*} x_{u}^{*} \frac{d \rho}{d r^{2}} \\
2 y_{u}^{*} x_{u}^{*} \frac{d \rho}{d r^{2}} & \rho+2\left(x_{u}^{*}\right)^{2} \frac{d \rho}{d r^{2}}
\end{array}\right)\left(\begin{array}{ccc}
\frac{1}{Z^{*}} & 0 & -\frac{Y^{*}}{Z^{*}} \\
0 & \frac{1}{Z^{*}} & -\frac{X^{*}}{Z^{*}}
\end{array}\right)\left(\begin{array}{ccc}
\left(\boldsymbol{X}^{*}\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \left(\boldsymbol{X}^{*}\right)^{\mathrm{T}} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \left(\boldsymbol{X}^{*}\right)^{\mathrm{T}}
\end{array}\right) \mathbf{J}_{\mathbf{H}} \tag{5.20}
\end{align*}
$$

where $\nabla_{p^{*}} I^{*}$ and $\nabla_{w\langle\mathbf{H}\rangle\left(p^{*}\right)} \mathrm{I}$ are the image gradient vectors of the reference image and the warped current image respectively. The differential of the distortion factor with respect to the square of the radial distance $r^{2}$ is $\frac{d \rho}{d r^{2}}=K_{1}+2 K_{2} r^{2}$. The Jacobian matrix for the exponential map $\mathbf{J}_{\mathbf{H}}$ is a $9 \times 8$ matrix (Benhimane and Malis, 2007). The
$k$-th column $\mathbf{J}_{\mathbf{H}}$ is represented as $\left(a_{1}, a_{2}, a_{3}, a_{4}, a_{5}, a_{6}, a_{7}, a_{8}, a_{9}\right)^{\mathrm{T}}$, which consists of the reshaped $\mathbf{A}_{k}=\left(\begin{array}{ccc}a_{1} & a_{2} & a_{3} \\ a_{4} & a_{5} & a_{6} \\ a_{7} & a_{8} & a_{9}\end{array}\right)$, where the elements are extracted row by row. After simplification and substitution using Eq. 5.5, Eq. 5.21 can be obtained and used for the implementation.

$$
\left.\begin{array}{c}
\left(\mathbf{J}_{\mathrm{I}^{*}}+\mathbf{J}_{\mathrm{I}}\right) \mathbf{J}_{\boldsymbol{m}_{\mathrm{d}}} \mathbf{J}_{\boldsymbol{m}_{\mathrm{u}}} \mathbf{J}_{\mathbf{X}} \mathbf{J}_{w} \mathbf{J}_{\mathbf{H}}=\left(\nabla_{\boldsymbol{p}^{*}} \mathrm{I}^{*}+\nabla_{w\langle\mathbf{H}\rangle\left(\boldsymbol{p}^{*}\right) \mathrm{I}}\right)\left(\begin{array}{cc}
f & 0 \\
0 & f
\end{array}\right)\left(\begin{array}{cccc}
\rho+2\left(y_{u}^{*}\right)^{2} \frac{d \rho}{d r^{2}} & 2 y_{u}^{*} x_{u}^{*} \frac{d \rho}{d r^{2}} \\
2 y_{u}^{*} x_{u}^{*} \frac{d \rho}{d r^{2}} & \rho+2\left(x_{u}^{*}\right)^{2} \frac{d \rho}{d r^{2}}
\end{array}\right) \\
\left(\begin{array}{ccccccc}
f & 0 & \left(v_{i}^{*}-v_{0}\right) & 0 & \left(u_{i}^{*}-u_{0}\right) & -\left(u_{i}^{*}-u_{0}\right) & \frac{-\left(u_{i}^{*}-u_{0}\right)^{2}}{f} \\
0 & f & 0 & \left(u_{i}^{*}-u_{0}\right) & -\left(v_{i}^{*}-v_{0}\right) & -2\left(v_{i}^{*}-v_{0}\right) & \frac{-\left(u_{i}^{*}-u_{0}^{*}-u_{0}\right)\left(v_{i}^{*}-v_{0}^{*}\right)}{f}
\end{array} \frac{-\left(v_{i}^{*}-v_{0}\right)}{f}\right. \tag{5.21}
\end{array}\right) .
$$

### 5.3.3.3 ESM reference image

As part of the preparation process of ARTIST, the reference image is obtained in the following manner. The user places the planar surface to be tracked within a $192 \times 192$ pixels square at the center of the video frame, which has a size of $512 \times 384$. The plane should be placed as parallel to the camera image plane as possible. This large square is divided into $24 \times 24$ pixels sub-grids. The average image gradient within each sub-grid is computed, and only those sub-grids where the gradient is above 10 grey levels per pixel are used for ESM tracking. This step is carried out, as the image gradient component $\mathbf{J}_{\text {I }}$ is part of the Jacobian matrix. Experimental observation shows that image regions with low gradients do not contribute additional information for ESM convergence, and may cause convergence towards the wrong minima in certain cases. To aid the user in selecting surfaces with high image gradients, only sub-grids with sufficient gradient

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

are rendered during the selection process. This provides a visual indication of the suitability of a surface for ESM tracking. Figure 5.15 shows the sub-grids with high image gradients in the selection process.


Figure 5.15: The ESM reference image selection process where the sub-grids with high image gradient are rendered.

After the selection of the sub-grids, the user is required to move the camera so that the normal of the planar surface can be obtained using the decomposition of the homography computed using ESM. The decomposition method presented by Malis and Vargas (2007) is used, and this results in two possible solutions for the normal. In the formulation of the homography in Eq. 5.12, the normal vector is defined in the reference camera frame and does not change as the camera moves. In the current implementation, the normal vector that does not change when the sideway motion is greater than $0.5 \%$ of the perpendicular distance between the camera and the plane is chosen as the normal of the planar surface. For indoor tracking, this typically means
a lateral motion of a few centimeters. However, for outdoor tracking, the distances encountered can be more than a hundred meters for building facades, and this means a sideway motion of more than half a meter. As a further check, tracking is continued with a virtual 3D object augmented onto the planar surface after the normal vector is determined. This allows the user to visually check the accuracy and the process is illustrated in Figure 5.16. After the normal vector in the reference frame has been obtained, subsequent homography decomposition can be performed using a second algorithm (Faugeras, 1993), which gives only one set of rotation matrix and translation vector required for the augmentation of virtual 3D objects.


Figure 5.16: Augmentation of a cube for checking the accuracy of the normal vector obtained.

### 5.3.3.4 Tolerance to illumination changes and partial occlusion

ESM tracking requires that the image intensity constancy assumption in Eq. 5.15 is valid such that the intensity errors are purely due to object pose errors. In reality,

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

illumination is rarely constant. An illumination model is required for adjusting the pixel intensities in the warped image so that the intensity constancy assumption is valid. For ARTIST, the illumination model is related to the discrete illumination model (Silveira and Malis, 2007), where illumination changes are applied equally within each sub-grid of the reference image presented in Section 5.3.3.3. For the discrete illumination model, the parameters are estimated together with the motion parameters within the ESM iterations. As such, the number of parameters is greatly increased. For example, if $8 \times 8$ sub-grids are used, the total number of parameters is 73 ; eight for motion, 64 for sub-grid illumination coefficients and one for global illumination. This results in very large sparse Jacobian matrices that severely slow down the computation.

For ARTIST, the illumination parameters are estimated directly from the warped and reference images. This is possible as the predicted pose is close to the current one during ESM tracking. The illumination change is modeled as follows. Let $\mathrm{I}_{i, j}$ be the intensity of pixel $i$ in the sub-grid $j$ for the warped image. Let $m_{j}$ and $d_{j}$ be the mean and standard deviation of the pixel intensities in the sub-grid $j$ in the warped image, and $m_{j}^{*}$ and $d_{j}^{*}$ be the corresponding values for the reference image. The modified pixel intensity $\mathrm{I}_{i, j}^{\prime}$ is obtained using the illumination model in Eq. 5.22.

$$
\begin{equation*}
\mathrm{I}_{i, j}^{\prime}=\frac{d_{j}^{*}}{d_{j}}\left(\mathrm{I}_{i, j}-m_{j}\right)+m_{j}^{*} \tag{5.22}
\end{equation*}
$$

The proposed illumination model equalizes $m_{j}$ and $m_{j}^{*}$ as well as $d_{j}$ and $d_{j}^{*}$. There are several advantages as compared to the model by Silveira and Malis (2007). First, the model accuracy is higher as both the mean illumination and the spread of the values
within a sub-grid are adjusted instead of a single scaling coefficient. Second, the computational load is reduced significantly as parameters are directly estimated without the use of large sparse Jacobian matrices. Comparisons with the discrete illumination model using captured video sequences are reported in Section 5.4.2.4. The final advantage is the improved occlusion detection. For the discrete illumination model, parameters can be over adjusted within ESM to compensate for the intensity errors caused by occlusion till such intensity errors reach normal error levels, and this complicates the occlusion detection. For the proposed model, over adjustment is avoided as the parameters are obtained directly from the images. As both the transformation and illumination models are accurate, the occlusion of a sub-grid can be detected easily when its average pixel error is above a pre-defined threshold, which is set as 20 in the current implementation. This result shown in Section 5.4.2.1 demonstrates the effectiveness of the proposed model.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

### 5.4 Experimental Setup and Results

This section describes the test results of the ARTIST CV tracking system.

### 5.4.1 Experimental Setup and Implementation Details

The proposed tracker is implemented using a Macbook Pro with a 2.4 GHz Intel Core 2 Duo processor with 4GB of memory and a DragonFly 2 Firewire camera. The focal length of the lens is 4 mm . The software is written using the C language and compiled using the MinGW GCC suite on Windows XP, as well as GCC on OSX. Specifically, the codes for keypoint signature matching and ESM are implemented from scratch. The greyscale image has an unusual resolution of $512 \times 384$ due to the use of $2 \times 2$ pixel binning mode on the $1024 \times 768$ camera sensor to reduce the noise. The lens is calibrated using the projection model and the calibration routines in OpenCV (2010). The intrinsic parameters of the camera are shown in Table 5.7. Different video sequences are used to test the performance of the proposed tracker. This is done by recording every frame in the video stream as a JPEG file with the quality set at 80 . For Ferns, the default parameter values given in Section 5.3.2.2 are used. The peak probabilities $p_{i}$ for object keypoints are obtained from the $192 \times 192$ ESM reference image using 500 random warps. In order to improve tracker initialization when the camera is further away from the surface to be tracked, the reference image is scaled down by one octave for feature extraction and peak probabilities computation at the lower scale.

For ESM, the image gradients $\nabla_{p^{*}} I^{*}$ and $\nabla_{w\langle\mathbf{H}\rangle\left(p^{*}\right)} \mathrm{I}$ are computed using $3 \times 3$ Prewitt masks, as described in Section 5.3.1.2. As the gradient is computed using three rows or

Table 5.7: Values of the camera intrinsic parameters and radial distortion coefficients.

| Parameter | Value |
| :--- | :---: |
| Focal Length $(f)$ | 439.5 pixels |
| Principal Point $\left(u_{0}, v_{0}\right)$ | $(251,185)$ pixels |
| First radial distortion coefficient $\left(K_{1}\right)$ | -0.425 |
| Second radial distortion coefficient $\left(K_{2}\right)$ | 0.173 |

columns about the pixel at the centre, the gradient computed is larger than expected and this affects the rate of convergence. A multiplication of the gradients computed with a value of 0.16666 results in a significant decrease in the number of iterations, which is critical for achieving good performance.

### 5.4.2 Experimental Result

Three sets of results are presented here to demonstrate that the CV tracker can achieve the goals of high accuracy and high robustness. The first set shows effective working of the various algorithms together as a system. The second set of results illustrates the computational speed of the system; the results show that the tracker can track three objects simultaneously at 16 fps . The final set of results shows the various types of planar objects that can be tracked by this tracker. The results for outdoor tracking will be presented in Chapter 6.

### 5.4.2.1 Tracking of a single object

A flat board with a semi-glossy picture is tracked with smooth motions. For ESM tracking, four images shown in Figure 5.17 are used to demonstrate its robustness against shadows, specular glares, partial occlusion and extreme poses. Only sub-grids with an average image gradient above 10 grey-levels per pixel are rendered. Occluded

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

sub-grids are not rendered to indicate that such conditions are detected and handled robustly. Figure 5.18 shows the plots of the $x, y$ and $z$ motions for frames 1200 to 1500 of this video sequence, where there are occlusions similar to Figure $5.17(\mathrm{~b})$ and the camera is moving. For robustness against partial occlusion, Figure 5.18 shows that for cases where more than 20 sub-grids are non-occluded, the tracked motion is smooth. From frames 1360 to 1390 , there is jitter as fewer than 20 sub-grids are visible, and tracking is lost for frames 1391 to 1434 as most of the picture is covered.


Figure 5.17: Augmentation of a teapot using ESM onto a planar surface with illumination interferences and extreme object pose.


Figure 5.18: Plots of $x, y$ and $z$ motions for video frames with occlusions similar to those shown in Figure 5.17(b)

### 5.4.2.2 Tracking of multiple objects

In the video sequence shown in Figure 5.19, three planar objects are tracked. All four algorithms, namely, feature detection, matching, robust pose estimation and pose refinement are performed for every frame. Figure shows two of the frames where the teapots are augmented onto objects with different poses. The first 800 frames of this sequence, where all three objects are constantly tracked, are used to obtain the average processing times shown in Table 5.8. Each frame requires an average of 63 msec to process, thus giving an average frame rate of 16 fps . This shows that the system is sufficiently fast for multiple objects tracking at 16 fps . For the actual tracker operation where only ESM tracking is used after pose initialization, the frame rate reaches 28 fps , as approximately 36 msec is required per frame. The times required to obtain the keypoint signatures, $s_{\mathrm{I} k}$, for all the image keypoints and match the keypoints of an object are 14 msec and 1 msec per frame respectively. If a set of Ferns is used to directly

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

track one object, the time for each object will be approximately 14 ms as the same algorithm is used. Therefore, the use of the proposed peak probabilities matching method enables more objects to be initialized, as each object requires only an addition of 1 msec for matching. The memory required for peak probabilities is approximately 100 kilobyte for each object instead several megabytes when Ferns are used. The training time is also significantly reduced, as only 500 training samples are required instead of 10000 , which is typically required for training Ferns.


Figure 5.19: Augmentation of teapots onto three objects.

Table 5.8: Average computational times for key tracking components.

|  | Time per frame (msec) | Comments |
| :--- | :---: | :---: |
| Total time | 62.76 | 15.93 fps |
| Feature Detection | 5.96 | With adaptive thresholding <br> and orientation assignment |
| Compute $s_{\text {Ik }}$ | 14.24 | $0.0432 \mathrm{msec} /$ keypoint |
| Signature matching | 2.77 | $0.922 \mathrm{msec} /$ object |
| Outlier removal <br> and RANSAC | 3.67 | $1.222 \mathrm{msec} /$ object |
| ESM | 17.45 | $5.847 \mathrm{msec} /$ object |
| Others | 18.87 | Image loading, OpenGL |

### 5.4.2.3 Types of surfaces which can be tracked

Due to the algorithms used, there are limitations on the types of surfaces that can be tracked. With the use of FAST-9 features and RANSAC, the surfaces are required to have at least 30 point-features per scale. Therefore, for surfaces with predominantly blob and line features, other features detection methods would be required. Furthermore, as ESM requires the presence of varied image gradients for correct convergence, the surface cannot be uniformly colored. However, a variety of surfaces, such as book covers, posters and all kinds of flat images, can be augmented. Image frames from the test video sequences are shown in Figure 5.20, 5.21 and 5.22. Figure 5.17 also shows an example of an augmentation on a low contrast image. Further examples of augmented objects in outdoor environments are presented in Chapter 6.


Figure 5.20: Augmentation on a surface with rich and varied patterns.


Figure 5.21: Tracking of a high contrast surface with large changes in scales.

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES



Figure 5.22: Augmentation on high gloss surfaces.

### 5.4.2.4 Comparison of the proposed illumination model with the discrete illumination model

Five video sequences, labeled 1 to 5, and shown sequentially in Figure 5.17, 5.20, 5.21, 5.22 and 6.4(c) (the side of an apartment block) are used to compare the performance of the proposed illumination model with the discrete illumination model (Silveira and Malis, 2007). The two criteria measured are the model accuracy and the ESM run time. The model accuracy is measured using the root mean square (rms) pixel intensity error obtained after ESM has converged. The pixel intensity errors reported are relative to the 256 grey levels of the images process and not normalized. The visual quality of the augmentation and the number of sub-grids selected for both models are similar for each video sequence to achieve a fair comparison. The sub-grids used are $24 \times 24$ pixels in size. The rms pixel error, average processing time and sub-grids used per frame are shown in Table 5.9.

The run times and rms pixel errors obtained for the discrete illumination model are consistent with or better than those reported by Silveira and Malis (2007). Silveira and Malis (2007) showed that the time required for similar experimental setups, which have approximately 30 parameters and a region area of 20,000 pixels, is around 20 msec per

Table 5.9: The rms pixel error and average processing time per video frame.

|  | Video | 1 | 2 | 3 | 4 | 5 |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| Discrete | RMS pixel error | 6.638 | 7.761 | 8.881 | 12.160 | 12.266 |
| illumination | Time per frame (msec) | 17.17 | 31.21 | 26.72 | 18.55 | 17.102 |
| Model | No. of sub-grids | 27.52 | 39.27 | 33.14 | 25.61 | 21.86 |
| Proposed | RMS pixel error | 6.019 | 5.049 | 7.392 | 6.475 | 5.830 |
| illumination | Time per frame (msec) | 10.57 | 12.08 | 11.87 | 11.64 | 12.21 |
| Model | No. of sub-grids | 30.25 | 40.15 | 36.87 | 27.44 | 21.95 |

iteration. Assuming that the average number of iterations is five, as reported in their paper, the run time per frame is expected to be approximately 100 msec . The hardware configuration is not specifically stated in their paper, but as it is a recent publication, the hardware configuration used is expected to be similar. Therefore, the code and system used in this research is significantly faster. Silveira and Malis (2007) showed that the rms pixel errors range from 5 to 20 , which is consistent with the above results. From Table 5.9, it can be observed that rms pixel error of the proposed illumination model is lower and shows less variation. As the video sequences contain different types of surfaces and illumination changes, it can be concluded that the proposed model is more accurate in adapting to these different variations. This low variation in rms pixel error allows for simple thresholding to be used for detecting occlusions of subgrids described in Section 5.3.3.4. The run times are significantly shorter and have less variation for different videos. This is because for the discrete illumination model, the Jacobian has an additional column for each sub-grid, while the Jacobian in the proposed model has only eight columns regardless of the number of sub-grids (see Eq. 5.21). These results show that while the proposed illumination model is simple, it is

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

more effective and efficient than the discrete model.
The tracking speeds reported in Table 5.9 are faster than those presented in Table 5.8 due to a lower number of sub-grids used for the cases in this section. Furthermore, performing feature detection and matching algorithm in every frame prevents the ESM data from staying in the processor cache memory. This slows down the processing, as the ESM data has to be fetched from the slower main memory.

### 5.5 Concluding Remarks and Future Works

The proposed marker-less CV tracker is able to track multiple textured planar surfaces with high accuracy and low jitter. Robustness against radial distortion, illumination changes, specular glares and partial occlusion were also achieved through the modifications of the keypoint signatures and ESM. The surfaces can be tracked in real world cluttered environments over the full range of rotation with large amount of tilt and scale changes. The advances in CV tracking in ARTIST are brought about by the recent availability of efficient CV algorithms, such as FAST, keypoint signatures and ESM, as well as the focus on the overall system design. Therefore, it is important that the individual algorithms are as efficient and effective as possible; and they must also operate well together as a system. As exemplified by PTAM, SIFT and ARTIST, it is most probable that practical CV trackers of the future will be developed using such a two-tier approach.

The use of machine learning for feature matching adds a degree of flexibility to feature matching. Instead of engineering a feature descriptor, such as SIFT which is
limited to the situations that it performs well, the machine learning approach allows for the efficient handling of specific conditions simply by providing the relevant training samples. For example, radial distortion, fisheye projection and different types of simulated image noises can be added to the training samples to enable feature matching to handle them. The main problem with this approach is the large memory requirement and long training time. The experimental approach presented in Section 5.3.2.2 can be used to obtain optimal matching times and memory usage without a loss of accuracy. Furthermore, the use of peak probabilities greatly reduces the training time. Eventually, as memory density continues to increase, mobile devices will have no issue running such software in the near future.

The design of the ARTIST tracker allows the algorithms to be changed to meet new or additional requirements, while maintaining a focus on how they would affect the overall system performance. It is clear that multiple algorithms will be required for each system stage. This is particularly true for feature detection where blob and line detectors will be required to expand the types of surfaces that can be tracked. This may require a separate feature-matching algorithm or share the existing machine learning approach. It is expected that multiple algorithms, each optimal for a certain set of conditions, will be performed simultaneously. This is due to the user demand for AR tracking in all kinds of environments where no single algorithm is expected to function well. This is particularly true for feature detection. These algorithms can be executed in parallel or selected based on the conditions. Along with the increasing processing power, the system design will be particularly important in achieving good

## 5. COMPUTER VISION: HIGH PRECISION POSITIONING ON TEXTURED PLANAR SURFACES

tracker performance.
There are several issues with the current ARTIST CV tracker. The current tracker is limited to the tracking of planar surfaces. In order to track non-planar objects, changes are required for robust pose estimation and ESM. Pose recovery using five points to solve for the essential matrix (Stewénius et al., 2006) or three points and 3D map (Klein and Murray, 2007) can be used. For ESM, a model is required for warping the current image to match the reference image. One possible method is the parametric model proposed by Malis (2007), which allows for tracking of both rigid and deformable non-planar objects. However, this algorithm is slow due to the requirement to solve a large number of parameters. Another possible approach is the use of tri-focal transfer (Hartley and Zisserman, 2003), which allows for the generation of a novel view of the object to match the reference image using multiple images and the trifocal tensor. At the time of writing, the algorithms and codes are not fully optimized, and the tracker is not as efficient as the trackers developed by Wagner et al. (2008b). Furthermore, it is also susceptible to tracking failure due to camera motion blur. Finally, the addition of GPS and inertial components to CV tracking, enables the ARTIST hybrid tracker to continue tracking during motion blur and planar surfaces moving out of view, as shown in Chapter 6.2.3.

## ARTIST hybrid tracker: Experimental results

### 6.1 Introduction

Outdoor AR requires the tracking systems to operate under a wide range of environmental conditions and motions. As stated in Chapter 1, robustness, precision, low jitter and ease of use are the important requirements for satisfactory augmentation of a user's reality. The tracking systems would also be required to operate without any modifications to the environment. Consequently, these systems have to rely on the natural properties of the environment to perform the tracking. As no single tracking technology is applicable and robust in every condition, hybrid tracking systems are required. Two different hybrid trackers are presented here. The first is a loosely coupled system and the other is the tightly coupled Kalman filter based design described in Section 2.4.

### 6.1.1 Loosely Coupled Configuration

This chapter presents the research on the ARTIST hybrid tracking system for AR in outdoor urban environments Fong et al. (2009). It consists of CV, inertial and GPS tracking modules. The general scheme is to use the coarse Earth-Centered Earth-Fixed position, which is obtained from the standalone GPS receiver, and the orientation measured using the inertial and magnetic sensors to obtain an initial search set for the CV tracker. This setup can be described as a loosely coupled configuration, as the data exchange between the modules is confined only to $\boldsymbol{E C E F}$ positions and $\boldsymbol{N E D}$ orientations. The internal processing of each module is completely independent. Although a tightly coupled system is likely to provide better performance, the loosely coupled configuration is more flexible on hardware requirements. It allows ARTIST to run using
commodity hardware that does not expose low-level data, such as the GPS carrier phase measurements. This is particularly important for ARTIST to be hardware-platform independent for mobile outdoor AR, for example ported to a mobile phone that has camera, accelerometer, magnetometer and the GPS modules neatly packaged into a portable unit. This loose coupling is possible as the ARTIST CV tracking component is sufficiently robust and efficient, allowing the GPS and inertial to be utilized mainly for initialization and relocalization.

### 6.1.2 Distinctive Planar Surface for Outdoor AR

As presented in Chapter 5, the CV tracker is currently limited to the tracking of planar surfaces. However, as distinctive planar surfaces, such as building facades, road markings, signs and posters, are common in outdoor urban environments, ARTIST is usable in such areas. The use of distinctive planar surfaces avoids the need to perform 3D modeling of the outdoor environment for applications that do not need such models, while still providing planar surfaces for augmentation of 3D virtual objects. Therefore, these planar surfaces can be considered as natural markers in ARTIST, which can be used in a manner similar to the ARToolkit (2010) markers. This simplifies the development of outdoor AR applications for users who are familiar with the ARToolkit. Distinctive planar surfaces are also typically surfaces where users would require the augmentation of 3D virtual objects or annotations at precise locations. For large featureless locations, such as open fields, CV tracking is likely to fail. In this case, CV and the GPS are complementary. CV tracking functions well in dense urban areas with buildings that interfere with GPS tracking. While in large featureless areas, users are less able to dis-
cern the errors due to a lack of comparative features, and AR applications generally do not demand high accuracy. Therefore, the GPS tracking can substitute for CV tracking by operating as a sufficiently accurate position tracker in large featureless areas.

### 6.1.3 Phases of System Operations

As ARTIST is highly CV centric, the ARTIST tracking system operates in a manner similar to CV tracking (Section 5.2.1) in four phases, namely, preparation, initialization, tracking and relocalization. In the preparation phase, the ESM reference image (Section 5.3.3.3) and the keypoint signatures (Section 5.3.2.1) of each planar surface to be tracked, as well as the GPS position and NED orientation of the camera, are obtained. As the RANSAC estimation of homography (Hartley and Zisserman, 2003) is used to detect the presence of a planar surface and the initial camera pose, the number of features detected is crucial. For RANSAC, the minimum number of inliers should be above 15 to ensure that the initial pose is geometrically consistent. As feature matching is imperfect, around 30 to 100 features are required for RANSAC to be effective. However, having more features can reduce the performance as there is an increased probability of similar features and mismatches, as well as an increased computational time for matching signatures. After the surfaces are selected, they act as large natural markers defining a plane for the augmentation of 3D objects, in a manner similar to ARToolkit. Authoring and interaction techniques developed for ARToolkit can be transferred to the current tracking system.

The initialization phase takes place when the hybrid tracking system is first started. The GPS position and the expected GPS error are used to define a circular region
encompassing the possible positions of the user. All planar patches with reference camera GPS positions within this region are tentatively included in the search set. The $\boldsymbol{N E D}$ orientation measured using the IMU is used to reduce the search set by eliminating surfaces where the $\boldsymbol{N E D}$ orientation of the surface normal is greater than $45^{\circ}$ from the current $\boldsymbol{N E D}$ orientation. Figure 6.1 illustrates the process of determining the search set. The signatures of all the features in the current frame are computed and matched against the features of the patches within the search set. As feature matching can be computationally intensive, only features from three surfaces randomly selected from the search set are matched in each frame. This is done in order to maintain a video rate of 16 frames per second (fps). The initial poses of the potential surfaces obtained using RANSAC are refined using ESM, and surfaces with an average pixel error below a pre-defined threshold of twenty are considered to have their pose accurately determined.


Figure 6.1: The selection of surfaces for feature matching based on the GPS position and the $\boldsymbol{N E D}$ orientation of the camera. (Selected surfaces are darkened)

After initialization, the detected surfaces are continuously tracked using ESM in the tracking phase. Feature matching is performed in the background to detect new

## 6. ARTIST HYBRID TRACKER: EXPERIMENTAL RESULTS

surfaces. However, as the precise camera pose is known, the surfaces that are near to the currently tracked surfaces are given higher priority during the selection for feature matching. Tracking failures are detected when the average pixel error exceeds the threshold of twenty, and the tracker goes into the relocalization phase. Recently lost surfaces are given the highest priority for feature matching. This priority reduces or decays with the elapsed time, as the probability of relocalizing the surface decreases with increasing time. GPS and inertial tracking are also continuously performed to speed up recovery from complete CV tracking failure. Figure 6.2 shows a summary of the tracking process.


Figure 6.2: Summary of the hybrid tracking system.

### 6.1.4 Tightly Coupled Configuration using Kalman Filter

In contrast, the tightly coupled configuration uses the GPS DSD position presented in Section 4.3 and the two Kalman filters described in Section 2.4. As such, the readings from all three sensors are incorporated whenever measurements are available. The
effectiveness of the design is tested by using the hybrid tracker in conditions where accurate Computer Vision (CV) measurements are available as the ground truth. By removal of CV tracking data for a period of time, the accuracy of the filtered GPS and IMU position and orientation can be compared against the ground truth.

### 6.2 Experimental Setup and Results

### 6.2.1 Experimental Setup

Both ARTIST configurations are implemented using a Macbook Pro with 2.4 GHz Core 2 Duo processor. The GPS position and carrier phase measurements are obtained using the U-Blox LEA-4T modulse with a ceramic patch antenna and the $\boldsymbol{N E D}$ orientation is obtained using InterSense Wireless InertiaCube. The video is captured using the Point Grey DragonFly camera with a 4 mm lens. The camera causes significant radio interference in the GPS band. Installing a conductive aluminum shield around the camera and mounting the GPS antenna 30 cm away mitigated the interference. The setup used is shown in Figure 6.3.

### 6.2.2 Experimental Results

### 6.2.2.1 CV Tracking in outdoor environment

This section shows the augmentation results for the various types of surfaces in the outdoor urban environment. The familiar Utah teapot is used to illustrate that the camera poses are determined accurately and tracking is successful under rotation and scale changes. Figure 6.4 shows examples of surfaces that are suitable for augmentation.

As ARTIST CV tracker is required to handle camera rotations and large-scale


Figure 6.3: The experimental setup consisting of the Dragonfly camera, InertiaCube and LEA-4T GPS module with antenna.


Figure 6.4: Examples of surfaces that can be augmented using ARTIST.
changes, it is tested under such motions for the case of the apartment block and road marking. The results of the augmentation are presented in Figure 6.5, and the results show that the surfaces are accurately registered.


Figure 6.5: Augmentation in the presence of camera rotation and large scale changes.

### 6.2.2.2 Campus walkthrough using Loosely Coupled Configuration

To demonstrate the effectiveness of loosely coupled ARTIST configuration, seven sites with suitable planar surfaces around the building where the author's laboratory is located, are chosen as the augmentation sites. For each of the seven sites, the reference image, keypoint signatures, GPS position and $\boldsymbol{N E D}$ orientation of the camera are obtained. This is followed by a full walkthrough, where the author visited the seven sites and attempted to augment a virtual object at each site. Figure 6.6 shows the walkthrough route plotted onto the satellite photograph of the building using Google

## 6. ARTIST HYBRID TRACKER: EXPERIMENTAL RESULTS

Earth. Although the application does not specify the geographical accuracy of its satellite and aerial photographs, Figure 6.6 shows that the recorded GPS positions coincide well with the actual locations where the reference images were taken. The images of the augmentation at each of the seven sites are shown in Figure 6.7.


Figure 6.6: The GPS positions of the seven augmentation sites, where the orientation is the direction from the smiling mouth towards the eyes of the icon.

### 6.2.3 Kalman filter results

### 6.2.3.1 Position Filter

The position filter is tested by using the CV tracking result as the reference, and simulating CV tracking failure by removal of data. Figure 6.8 shows the reference CV tracking data for a small planar surface in the outdoor environment. The planar surface consists of a piece of paper with printed graphics, mounted on a stand. This allows for

(a) Site 1

(c) Site 3

(e) Site 5

(b) Site 2

(d) Site 4

(f) Site 6

(g) Site 7

Figure 6.7: Augmentation at the seven selected sites.

## 6. ARTIST HYBRID TRACKER: EXPERIMENTAL RESULTS

a controlled setup, where parameters such as the position and orientation of the planar surface in $\boldsymbol{N E D}$ frame can be accurately determined. In this case, the planar surface is placed so its planar normal is aligned along the North-South axis, with a slight tilt upwards. This alignment of the planar surface is not strictly required as the angular difference between its orientation and the orientation reported by the IMU is constant. However, knowing the orientation of the planar surface in the $\boldsymbol{N E D}$ frame is helpful for debugging coordinate frame inconsistencies, especially when manufacturer documentation is inadequate. The difference quaternion representing the angular difference will be constant and, if available, equal to the known value when the coordinate frames are correctly configured. For this experiment, the position reported by CV is relative to the planar surface in the camera frame of reference, while the GPS DSD method outputs position relative to the starting position in the $\boldsymbol{N E D}$ frame. Therefore, CV position is converted to match the GPS DSD position in $\boldsymbol{N E D}$ frame, by subtracting the initial CV position and multiplying the difference quaternion. The use of a smaller planar surface gives higher accuracy for CV tracking. Correspondingly, the range of motion is smaller $(-0.15 \mathrm{~m}$ to 0.15 m$)$ and thus provides a more difficult test for GPS positioning. Figure 6.9 shows the corresponding motion measured by the GPS method developed in this research. It can be observed that the GPS DSD method performs well during motion, but is noisy when stationary.

CV tracking failure is simulated by removing the tracking data between $\mathrm{t}=4.5 \mathrm{sec}$ and $\mathrm{t}=6.2 \mathrm{sec}$, as well as between $\mathrm{t}=16.0 \mathrm{sec}$ and $\mathrm{t}=17.7 \mathrm{sec}$. The position tracked using the position filter described in Section 2.4, using only the IMU and GPS data.

Figure 6.10 shows the altered CV tracking data and the error between the position given by the position filter and the unaltered reference CV position. The errors plot in the bottom row of Figure 6.10 show that in both instances of failure, the position filter has a error of up to 1 cm for $x$ and $y$-axes, and up to 5 mm for the $z$-axis, which had larger motions of up to 4 cm . This result is observed in repeated experiments, where the GPS DSD method performs better when there is a relatively large change in position in each GPS time step. The addition of quasi-static detection reduces the errors for axis with little motions to a limited extend. As these results are obtained with low cost GPS modules, it is expected that the results will be much improved for modules with lower noise in carrier phase measurements. In terms of augmentation, there is increased position jitter during the simulated CV failures, but the virtual object does not drift drastically.

### 6.2.3.2 Orientation Filter

For the test on the orientation filter, the camera is moved with rotations of up to 60 degrees, so that the planar surface is out of view. Figure 6.11 shows the variation of the four elements of the orientation quaternion output by the orientation filter. As accurate CV tracking data is not available when the planar surface is out of view, the augmentation result is examined visually. Figure 6.12 shows the video frames depicting the different phases of tracker operation. In both rotations, there are initial transient position and orientation errors when CV tracking fails. However, the frames showing the augmentation just before CV tracking is recovered, demonstrates that the errors did not increase significantly during the period with large orientation changes. The


Figure 6.8: Reference computer vision tracking data for tightly coupled tracker in the camera frame, relative to the starting position. The data has been scaled so that the distance is in metres.


Figure 6.9: The GPS Differential Single Difference tracking data corresponding to the motion shown in Figure 6.8. The Down-axis motion has significant noise.


Figure 6.10: The comparison of performance of position filter tracking against the reference computer vision data. The top row shows the altered computer vision position, where data is removed between $t=4.5 \mathrm{sec}$ and $\mathrm{t}=6.2 \mathrm{sec}$, as well as between $\mathrm{t}=16.0 \sec$ and $\mathrm{t}=17.7 \mathrm{sec}$ (highlighted with ellipses for z -axis motion). The bottom row shows the error between the output of the position filter and the reference.
cause of this transient error is unclear, even after efforts to tune the filter and ensuring that each tracker component is accurate. It is likely due to a tracker behavior that is unknown and unaccounted for, or simply due to data collection errors.

Although the output from both the position and orientation filters, without CV updates, are not sufficiently accurate for augmentation at close distances, the errors will be sufficiently low for augmentation at longer distance. Furthermore, the video results show that during CV tracking failures lasting for a small number of frames, accurate flicker free augmentation can be maintained. Therefore, the tightly coupled configuration is robust in conditions with fast camera motions that can cause camera blur. Higher quality GPS carrier phase measurements will be required for longer periods of computer tracking failure.


Figure 6.11: Plot of the variation of the four elements of the orientation quaternion during the orientation filter test. The blue line is the scalar element, while the other three lines are the vector elements. There are two large rotations in this test.

### 6.3 Concluding Remarks

The design and experimental results of two ARTIST configurations are presented. ARTIST integrates the GPS, inertial and CV tracking systems, where their complementary properties are combined to achieve the robust, accurate and jitter-free augmentation. As the robustness of the CV tracking algorithms improves, it is expected that similar hybrid trackers will be more CV-centric. This is because the camera is a sensor that can provide a large amount of information about the environment at high resolution. However, it is improbable that the CV algorithms can be scaled to cover the large areas of an outdoor environment. Therefore, GPS and inertial systems will be required to support and enhance CV-based tracking. The experimental results presented in Section 6.2.2 show that CV-based tracking is becoming increasingly viable for tracking in an uncontrolled environment. It is also well-suited for integration with the GPS and inertial sensing.

(c) $t=8.35 \mathrm{sec}($ error after 15 frames)
(d) $t=14.2 \mathrm{sec}$ (error after large rotation, before CV recovery)

(e) $t=23.0 \mathrm{sec}$ (error before second large (f) $t=26.5 \mathrm{sec}$ (error after rotation, before rotation)

Figure 6.12: The errors in augmentation during computer vision tracking failure.
The initial errors did not increase significantly during large rotations.

## 6. ARTIST HYBRID TRACKER: EXPERIMENTAL RESULTS

The main limitation of both ARTIST configuration at present is that only planar surfaces with rich features can be augmented. Several improvements are required to allow for more types of surfaces to be augmented. First is a new transformation model for ESM, such as the trifocal tensor transfer, for handling non-planar surfaces. Second is a new robust pose estimation technique for non-planar surfaces. Third is a new robust geometrical consistency check other than RANSAC, such as the hashing method used by Lowe (2004), which requires as few as three features. This allows surfaces with fewer features to be used for augmentation. More efficient feature signature matching will enable more surfaces to be tested for each frame, allowing for faster initialization and tolerance to larger GPS positioning errors. Compared to previous hybrid trackers, the CV component in this research has improved robustness and provided a more general framework for various improvements to be added.

Conclusion

## 7. CONCLUSION

This thesis has presented the research into the problem of wide area, unassisted tracking for high precision 3D AR applications. The main motivation is to move AR out of the laboratory environment, so that AR can be used in mobile environments, such as the outdoor areas, homes and work places. Furthermore, marker-based trackers, such as ARToolkit are avoided, as markers are not applicable to mobile AR and typically decrease the usability of AR applications. Therefore, this research takes a multidisciplinary approach towards solving the research question, through investigating three different but complementary tracking systems, namely CV, inertial measurement and the GPS. Research and development into each of the three systems has resulted in the following advances and contributions in mobile AR tracking. They are:

1. CV tracking with highly accurate 3D augmentation and good robustness against illumination changes, partial occlusion and extreme object poses.
2. An efficient system design for the CV tracker, and improvements to keypoint signatures and ESM, which enable the tracker to be sufficiently fast for accurate augmentation of three objects at 16 fps .
3. Calibration methods for tri-axial accelerometers and gyroscopic systems that are completely independent of external equipment. This allows end-users to perform calibration on-site. This is particularly important for gyroscope calibration.
4. Differential Single Difference (DSD) of GPS carrier phase measurements, which allows for a new method of GPS positioning. It is suitable for AR positioning with an accuracy of 10 cm , while avoiding the resolution of integer ambiguity.
5. Outdoor hybrid tracking system with a flexible, loosely coupled configuration. It demonstrates that CV tracking is viable as the main tracking mechanism, while the GPS and inertial measurements are used to enable CV tracking to operate over a larger area.
6. Tightly coupled outdoor hybrid tracking system using Kalman filters. It demonstrates that simple filters are effective for combining available sensor readings. In particular, it shows that GPS DSD can substitute for accurate CV tracking for a short period of time, but with increased jitter.

### 7.1 Analysis

It is evident from prior work and in this research that multiple tracking methods will be required for wide area AR tracking. The combination of CV, inertial measurement and the GPS is emerging as a promising approach towards accurate high-fidelity 3D augmentation in all kinds of mobile environments. Accurate 3D augmented in all kinds of mobile environment is significantly harder than navigation as the precision required for 3D augmentation is much higher. This is because the human user is able to discern minute errors in registration. It is expected that this level of accuracy will be derived from advances in CV tracking. However, it is improbable that CV tracking will be scalable and robust in every environment where AR is expected to be used. This is particularly true for outdoor environments. Therefore, systems will require GPS and inertial measurement that are not as accurate, but more robust than CV tracking.

The recent improvements to the speed, accuracy and robustness of CV tracking, as

## 7. CONCLUSION

demonstrated here with ARTIST, and systems, such as PTAM and RT based tracking, will push hybrid trackers towards being more CV-centric. Early systems, such as the Touring Machine, were more GPS and inertial based. At the beginning of this research, it was expected that CV tracking will not be robust or sufficiently fast, and the primary means to achieve the research goals was to improve the accuracy of inertial measurement and the GPS. However, the inherent error characteristics of both the inertial sensors and the GPS, limits the accuracy. Further improvements will require major advances in the design, manufacture and material properties of MEMS inertial sensors, as well as upgrades in signal transmission and processing of the GPS. However, the greatest limitation is that both inertial measurement and the GPS do not provide real time information about the environment that the user is in. Therefore, even if inertial measurement and the GPS become extremely accurate, they can only augment using known models of the environment and will not be able to react automatically to real time changes.

The high resolution image sensor will remain as the main sensor for AR, as it is able to provide real-time information about the environment. From a biological perspective, vision is the most common and successful sensory mechanism that most advanced organisms use for survival in a dynamic environment. For example, it enables them to look for food, avoid becoming a prey and react to changes. Furthermore, the development of AR was in part due to the inability to simulate all aspects of our experience in the real world, particularly the haptic properties. By allowing the real world to be part of the simulation, AR is used as a way to overcome the shortcomings
of VR. This is turn requires the system to detect where the user is and what object he is holding and manipulating, in order to insert the virtual elements in a consistent manner. Therefore, the usability and naturalness of AR will largely depend on how well CV can be used to extract information about the real world.

### 7.2 Recommendation for Future Research

Research into various tracking methods reveals that much work will be required to achieve accurate augmentation for all kinds of environment and under all kinds of conditions. Although, the ARTIST CV tracker is efficient and robust, it is limited to planar surfaces with a good number of point features. Therefore, one area will be the tracking of non-planar and/or deformable surfaces. The other area is the use of more types of features, such as blobs and lines. This will increase the types of objects and locations where ARTIST can augment. As there is existing research on resolving these issues, the main goal here is to be able to perform them in real time. This will entail modification of existing algorithms or invention of new ones. The systems perspective presented in Section 5.2 .1 becomes particularly crucial as multiple algorithms are either combined or executed in parallel. Both the algorithms and the system as a whole have to be improved in tandem for CV tracking to be effective.

Another area is making these algorithms sufficiently efficient to enable them to be executed on mobile phones. Although mobile phones do not provide an immersive experience, its mobility and wide availability serve as an excellent platform to popularize the use of AR. Therefore, there will be greater need for even more efficient CV algo-

## 7. CONCLUSION

rithms. The work by Wagner et al. (2008b) can already be used for suitable planar surfaces found in everyday life. While recent demonstration of PTAM (Klein and Murray, 2007) on the Apple iPhone, extends the applicability to non-planar surroundings. With development of user interfaces for authoring content, the mobile phone can be one of the first consumer friendly platform for using AR. Finally, the tight integration of CV, inertial and the GPS can be further investigated. For the current tightly coupled configuration, the differential GPS doppler measurement can be used for measuring the velocity. This was not done in this research due to the low accuracy of doppler measurements from the low cost units. The accuracy of DSD position tracking over large distance can be investigate, it was not done due to lack of survey equipment for accurate positioning over large distances. Another future work is to use the precise positions and orientations from CV tracking to improve the accuracy of the inertial and GPS tracking. Similarly, the low level information from the inertial and GPS tracking can improve the robustness and efficiency of the CV algorithms. Although the looselycoupled configuration is more flexible and has been proven to work, the interchange of low level data and the integration of the three tracking processes in novel ways will bring about more efficient and robust tracking systems. Ultimately, it is hoped that all these systems will be integrated into a single low-powered chip, so that AR can be used by everyone in a natural and effective fashion for improving their quality of living.

## 8

## Publications

W. T. Fong, S. K. Ong, and A. Y. C. Nee. Methods for in-field user calibration of an inertial measurement unit without external equipment. Journal of Measurement Science and Technology (JMST), 19(8):085202(11pp), 2008.

W.T. Fong, S.K. Ong, and A. Y. C. Nee. A differential GPS carrier phase technique for precision outdoor AR tracking. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2008), pages 25-28, Cambridge, 15-18 Sep 2008.

[^0]8. PUBLICATIONS

## References

Arinc. ICD-GPS-200, IRN-200C-004. Arinc Research Corporation, 2000.

ARToolkit. Available at: http://www.hitl.washington.edu/artoolkit/ [last access: Febuary 2010], 2010.
R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. IEEE Computer Graphics and Applications (CGEA), 21(6):34-47, 2001.
R. Azuma. Tracking requirements for augmented reality. Communications of the ACM, 36(7):50-51, 1993.
R. Azuma. A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 6(4):355-385, 1997.
B. Barshan and H.F. Durrant-Whyte. Inertial navigation systems for mobile robots. IEEE Transactions on Robotics and Automation, 11(3):328-342, 1995.
S. Benhimane and E. Malis. Homography-based 2D visual tracking and servoing. International Journal of Robotic Research (IJRR), 26(7):661-676, 2007.

## REFERENCES

D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. 3D User Interfaces: Theory and Practice. Addison-Wesley Professional, Boston, 2004.
M. Calonder, V. Lepetit, and P. Fua. Keypoint signatures for fast learning and recognition. In Proceedings of the 10th European Conference on Computer Vision (ECCV08), pages 58-71, Marseille, France, 12-18 Oct 2008.
X. Chang, C. Paige, and L. Yin. Code and carrier phase based short baseline GPS positioning: computational aspects. GPS Solutions, 9(1):72-83, 2005.
X. Chang, X. Yang, and T. Zhou. MLAMBDA: a modified LAMBDA method for integer least-squares estimation. Journal of Geodesy, 79(9):552-565, 2005.
J. W. S. Chong, S. K. Ong, and A. Y. C. Nee. Methodologies for immersive robot programming in an augmented reality environment. The International Journal of Virtual Reality, 6(1):69-79, 2007.
R. Cosentino, D. Diggle, M. Uijt de Haag, C. Hegarty, D. Milbert, and J. Nagle. Differential gps. In G. E. Kaplan, editor, Understanding Global Positioning System Principles and Application. Artech House, Norwood, 2006.
P. David, D. DeMenthon, R. Duraiswami, and H. Samet. SoftPOSIT: simultaneous pose and correspondence determination. International Journal of Computer Vision (IJCV), 59(3):259-284, 2004.
A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-Time single camera

SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 29(6):1052-1067, 2007.
M. El-Diasty, A. El-Rabbany, and S. Pagiatakis. Temperature variation effects on stochastic characteristics for low-cost MEMS-based inertial sensor error. Journal of Measurement Science and Technology (JMST), 18(11):3321-3328, 2007.
C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In Proceedings of Photogrammetric Computer Vision (PCV 2006), pages 266-271, Bonn, 20-22 Sep 2006.
J. Farrel and Barth. The global positioning system and inertial navigation. McGrawHill, New York, 1999.
O. Faugeras. Three-Dimensional Computer Vision. The MIT Press, Cambridge, Massachusetts, 1993.
S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In Proceedings of the 1st IEEE International Symposium Wearable Computers (ISWC97), page 74, Cambridge, Massachusetts, 13-14 Oct 1997.
W. T. Fong, S. K. Ong, and A. Y. C. Nee. Methods for in-field user calibration of an inertial measurement unit without external equipment. Journal of Measurement Science and Technology (JMST), 19(8):085202(11pp), 2008.
W.T. Fong, S.K. Ong, and A. Y. C. Nee. A differential GPS carrier phase technique

## REFERENCES

for precision outdoor AR tracking. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2008), pages 25-28, Cambridge, 15-18 Sep 2008.
W. T. Fong, S. K. Ong, and A. Y. C. Nee. Computer vision centric hybrid tracking for augmented reality in outdoor urban environments. In Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI2009), pages 185-190, Yokohama, Japan, 14-15 Dec 2009.
E. Foxlin and L. Naimark. Miniaturization, calibration \& accuracy evaluation of a hybrid Self-Tracker. In Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2003), pages 151-160, Tokyo, 7-19 Oct 2003.
E. Foxlin, M. Harrington, and Y. Altshuler. Miniature six-DOF inertial system for tracking HMDs. In Proceedings of Helmet- and Head-Mounted Displays III, pages 214-228, Orlando, 13-17 Apr 1998.
E. Foxlin. Motion tracking requirements and technologies. In K. Stanney, editor, Handbook of Virtual Environments: Design, Implementation, and Applications, pages 163-210. Lawrence Erlbaum Associates, New Jersey, 2002.
M. Grewal, V. Henderson, and R. Miyasako. Application of kalman filtering to the calibration and alignment of inertial navigation systems. IEEE Transactions on Automatic Control, 36(1):3-13, 1991.
M. Grewal, L. Weill, and A. Andrews. Global positioning systems, inertial navigation, and integration. John Wiley and Sons, New York, 2001.
R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, Cambridge, 2003.
R. Hayward, D. Gebre-Egziabher, and J. Powell. GPS-based attitude for aircraft. In Proceedings of International Conference on Integrated Navigation Systems, pages 8594, St. Petersburg, May 1998.
S. Hilla. The extended standard product 3 orbit format (SP3-c). Available at: http://www.ngs.noaa.gov/orbits/sp3c.txt [last access: Febuary 2010], Feb 2007.
B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. Global Positioning System: Theory and Practice. Springer, New York, 5th edition, 2004.
J. How, N. Pohlman, and C. Park. GPS estimation algorithms for precise velocity, slip and race-track position measurements. In Proceedings of SAE 2002 Motorsports Conference., Indianapolis, 2-5 Dec 2002.

IGS. IGS products. Available from http://igscb.jpl.nasa.gov/components/prods.html [last access: Febuary 2010], 2010.
C. Jekeli. Inertial navigation systems with geodetic applications. Walter de Gruyter, Berlin, 2000.
B. Jiang, U. Neumann, and S. You. A robust hybrid tracking system for outdoor

## REFERENCES

augmented reality. In Proceedings of IEEE Virtual Reality 2004 (VR2004), pages 3-10, Chicago, 27-31 Mar 2004.
R. Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering, 83(Series D):35-45, 1960.
E. Kaplan and C. Hegarty. Understanding GPS: Principles and Applications. Artech House, Norwood, 2nd edition, 2005.
A. Kim and M.F. Golnaraghi. Initial calibration of an inertial measurement unit using an optical position tracking system. In Proceedings of Position Location and Navigation Symposium (PLANS2004), pages 96-101, Monterey, 26-29 Apr 2004.
S. Kim, S. DiVerdi, J. S. Chang, T. Kang, R. Iltis, and T. Hllerer. Implicit 3D modeling and tracking for anywhere augmentation. In Proceedings of ACM symposium on Virtual Reality Software and Technology (VRST2007), pages 19-28, Newport Beach, California, 5-7 Nov 2007.
G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 6th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2007), pages 225-234, Nara, Japan, 13-16 Nov 2007.
G. Klein and D. Murray. Compositing for small cameras. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008), pages 57-60, Cambridge, 15-18 Sep 2008.
G. Klein and D. Murray. Improving the agility of Keyframe-Based SLAM. In Pro-
ceedings of the 10th European Conference on Computer Vision (ECCV08), pages 802-815, Marseille, 12-18 Oct 2008.
D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A hybrid and linear registration method utilizing inclination constraint. In Proceedings of the 4 th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2005), pages 140-149, Vienna, 5-8 Oct 2005.

J Lalonde, A. Efros, and S. Narasimhan. Estimating natural illumination from a single outdoor image. In Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV2009), Kyoto, Japan, 29 Sep - 2 Oct 2009.
A. Leick. Global Positioning System Satellite Surveying. John Wiley, New Jersey, 3rd edition, 2003.
V. Lepetit and P. Fua. Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 28(9):1465-1479, 2006.
X. Liu, C. Tiberius, and K. de Jong. Modeling of differential single difference receiver clock bias for precise positioning. GPS Solutions, 7(4):209-221, 2004.
J. Lötters, J. Schipper, P. David, W. Olthuis, and P. Bergveld. Procedure for in-use calibration of triaxial accelerometers in medical applications. Sensors and Actuators, A 68:221-228, 1998.

## REFERENCES

D. Lowe. Distinctive image features from Scale-Invariant keypoints. International Journal of Computer Vision (IJCV), 60(2):91-110, 2004.
E. Malis and M. Vargas. Deeper understanding of the homography decomposition for vision-based control. Technical report, INRIA, France, 2007.
E. Malis. An efficient unified approach to direct visual tracking of rigid and deformable surfaces. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2007), pages 2729-2734, San Diego, 29 Oct - 2 Nov 2007.
F. Markley. Attitude error representations for Kalman filtering. Journal of Guidance Control and Dynamics, 26(2):311-317, 2003.
P. Maybeck. Stochastic models, estimation and control, volume 1. Academic Press, New York, 1979.
K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 27(10):16151630, 2005.
P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. SPIE 2351-34 (Proceedings of telemanipulator and telepresence technologies, pages 282-292, 1994.
A. Moore and R. Neilan. The international GPS service tracking network: enabling diverse studies and projects through international cooperation. Journal of Geodynamics, 40(4-5):461-469, 2005.
L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable Vision-Inertial Self-Tracker. In Proceedings of the 1st International Symposium on Mixed and Augmented Reality (ISMAR2002), pages 27-36, Darmstadt, 30 Sep - 1 Oct 2002.
X. Niu and N. El-Sheimy. Development of a low-cost MEMS IMU/GPS navigation system for land vehicles using auxiliary velocity updates in the body frame. In Proceedings of the Institute Of Navigation Global Navigation Satellite Systems (GNSS), pages 2003-2012, Long Beach, California, 13-16 Sep 2005.
S.K. Ong and A.Y.C. Nee. Virtual Reality and Augmented Reality Applications in Manufacturing. Springer, New York, 2004.

OpenCV. Available from http://opencv.willowgarage.com/wiki/ [last access: Febuary 2010], 2010.
M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR 2007), pages 1-8, Minneapolis, 17-22 Jun 2007.
Y. Park, V. Lepetit, and W. Woo. Multiple 3D object tracking for augmented reality. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2008), pages 117-120, 15-18 Sep 2008.
A. Peternier, F. Vexo, and D. Thalmann. Wearable mixed reality system in less than

## REFERENCES

1 pound. In Proceedings of 12th Eurographics Symposium on Virtual Environments (EGVE2006), pages 35-44, Lisbon, 8-10 May 2006.
J. Pilet, V. Lepetit, and P. Fua. Retexturing in the presence of complex illumination and occlusions. In Proceedings of the 6th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2007), pages 249-258, Nara, Japan, 13-16 Nov 2007.
J. Pilet, V. Lepetit, and P. Fua. Fast Non-Rigid surface detection, registration and realistic augmentation. International Journal of Computer Vision (IJCV), 76(2):109122, 2008.
W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, New York, 2nd edition, 1992.
G. Reitmayr and T. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In Proceedings of the 5th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2006), pages 109-118, Santa Barbara, 22-25 Oct 2006.
M. Ribo, P. Lang, H. Ganster, M. Brandner, C. Stock, and A. Pinz. Hybrid tracking for outdoor augmented reality applications. IEEE Computer Graphics and Applications (CGधA), 22(6):54-63, 2002.
E. Rosten and T. Drummond. Machine learning for High-Speed corner detection. In

Proceedings of 9th European Conference on Computer Vision (ECCV2006), pages 430-443, Graz, 7-13 May 2006.
A. Sabatini. A wavelet-based bootstrap method applied to inertial sensor stochastic error modelling using the allan variance. Journal of Measurement Science and Technology (JMST), 17(11):2980-2988, 2006.
P. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. Journal of Guidance, Control and Dynamics, 21:19-28, 1998.
P. Savage. Strapdown inertial navigation integration algorithm design part 2:velocity and position algorithms. Journal of Guidance, Control and Dynamics, 21:208-221, 1998.
A. Saxena, G. Gupta, V. Gerasimov, and S. Ourselin. In use parameter estimation of inertial sensors by detecting multilevel quasi-static states. Lecture Notes in Computer Science, 3684:595-601, 2005.
J. Shi and C. Tomasi. Good features to track. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, Seattle, 21-23 Jun 1994.
G. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR2007), pages 1-6, Minneapolis, 2007.

I Skog and P. Händel. Calibration of a MEMS inertial measurement unit. In Proceedings of XVII IMEKO World Congress, Rio de Janeiro, 17-22 Sep 2006.

## REFERENCES

H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. ISPRS International Journal of Photogrammetry and Remote Sensing, 60(4):284-294, 2006.
Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy. A new multi-position calibration method for MEMS inertial navigation systems. Journal of Measurement Science and Technology (JMST), 18(7):1897-1907, 2007.
D. Titterton and J. Weston. Strapdown Inertial Navigation Technology. Peter Peregrinus Ltd. on behalf of Institution of Electrical Engineers, London, 1997.
V. Vlahakis, N. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne, and L. Almeida. Archeoguide: An augmented reality guide for archaeological sites. IEEE Computer Graphics and Applications (CGBA), 22(5):5260, 2002.
D. Wagner, T. Langlotz, and D. Schmalstieg. Robust and unobtrusive marker tracking on mobile phones. In Proceedings of 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2008), pages 121-124, Cambridge, 15-18 Sep 2008.
D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In Proceedings of 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR2008), pages 125-134, Cambridge, 15-18 Sep 2008.
G. Welch and G. Bishop. SCAAT: incremental tracking with incomplete information. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques SIGGRAPH 1997, pages 333-344, Los Angeles, California, 3-8 Aug 1997.
G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. IEEE Computer Graphics and Applications (CGEA), 22(6):24-38, 2002.
G. Xu. GPS: Theory, Algorithms and Applications. Springer, Berlin, 2nd edition, 2007.
Y. Yokokohji, Y. Sugawara, and T. Yoshikawa. Accurate image overlay on video SeeThrough HMDs using vision and accelerometers. In Proceedings of the IEEE Virtual Reality (VR2000), pages 247-254, New Jersey, 18-22 Mar 2000.
S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In Proceedings of IEEE Virtual Reality (VR2001), pages 71-78, Yokohama, 13-17 Mar 2001.
S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In Proceedings of the IEEE Virtual Reality (VR1999), pages 260267, Houston, 13-17 Mar 1999.
F. Zhou, H. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008), pages 193-202, Cambridge, 15-18 Sep 2008.

## REFERENCES

J. Zhu. Conversion of earth-centered earth-fixed coordinates to geodetic coordinates. IEEE Transactions on Aerospace and Electronic Systems, 30(3):957-961, 1994.

Appendix A
Results videos

## A. RESULTS VIDEOS



## Figure5.17.mov

Augmentation of a teapot using ESM onto a planar surface with illumination interferences and extreme object pose.


## Figure5.19.mov

Augmentation of teapots onto three objects.


Figure5.20.mov
Augmentation on a surface with rich and varied patterns.


## Figure5.21.mov

Tracking of a high contrast surface with large changes in scale.


## Figure5.19a.mov

A variation of the Figure5.19.mov video, where three different objects are augmented with a teapot, torus and cone. The objects are also moving more randomly and rapidly.

Figure5.22.mov
Augmentation on a high gloss surface.


## Figure4.11.mov

Augmentation using the proposed Differential GPS tracker and IMU (The checker board is used to indicate the drift).

## Figure6.4.mov

Augmentation on the side of an apartment block.

## Figure6.7.mov

Augmentation at the seven selected sites. The video is a continuous walkthrough where uninteresting portions have been sped up.

## Figure6.8.mov

Hybrid tracking with simulated computer vision tracking failures.

## Figure6.12.mov

Hybrid tracking with large rotations.


[^0]:    W. T. Fong, S. K. Ong, and A. Y. C. Nee. Computer vision centric hybrid tracking for augmented reality in outdoor urban environments. In Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI2009), pages 185-190, Yokohama, Japan, 14-15 Dec 2009.

