Vision & Laser For Road Based Localisation

Ashley Napier
Lincoln College



Supervisor:
Professor Paul Newman

Robotics Research Group
Department of Engineering Science
University of Oxford

January 2014

"Don't cry because it's over, smile because it happened."

- Dr. Seuss, Alumnus Lincoln College

Ashley Napier
Lincoln College

Doctor of Philosophy
Michaelmas 2013

Vision & Laser For Road Vehicle Localisation

Abstract

This thesis presents novel solutions for two fundamental problems associated with autonomous road driving. The first is accurate and persistent localisation and the second is automatic extrinsic sensor calibration.

We start by describing a stereo Visual Odometry (VO) system, which forms the basis of later chapters. This sparse approach to ego-motion estimation leverages the efficacy and speed of the BRIEF descriptor to measure frame-to-frame correspondences and infer subsequent motion. The system is able to output locally metric trajectory estimates as demonstrated on many kilometres of data.

We then present a robust vision only localisation system based on a two-stage approach. Firstly we gather a representative survey in ideal weather and lighting conditions. We then leverage locally accurate VO trajectories to synthesise a high-resolution orthographic image strip of the road surface. This road image provides a highly descriptive and stable template against which to match subsequent traversals. During the second phase, localisation, we use the VO to provide high frequency pose updates, but correct for the drift inherent in all locally derived pose estimates with low frequency updates from a dense image matching technique. Here a live image stream is registered against synthesised views of the road image generated form the survey. We use an information theoretic measure, Mutual Information, to determine the alignment of live images and synthesised views. Using this measure we are able to successfully localise subsequent traversals of surveyed routes under even the most intense lighting changes expected in outdoor applications. We demonstrate our system localising in multiple environments with accuracy commensurate to that of an Inertial Navigation System.

Finally we present a technique for automatically determining the extrinsic calibration between a camera and Light Detection And Ranging (LIDAR) sensor in natural scenes. Rather than requiring a stationary platform as with prior art, we actually exploit platform motion allowing us to aggregate data and adopt a retrospective approach to calibration. Coupled with accurate timing this retrospective approach allows for sensors with non-overlapping fields of view to be calibrated as long as at some point the observed workspaces overlap. We then show how we can improve the accuracy of our calibration estimates by treating each single shot estimate as a noisy measurement and fusing them together using a recursive Bayes filter. We evaluate the calibration algorithm in multiple environments and demonstrate millimetre precision in translation and deci-degrees in rotation.

Statement of Authorship

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Ashley Napier, Lincoln College

Funding

The work reported in this thesis was funded by a Departmental Student Award

Acknowledgements

First and foremost I would like to thank my supervisor, Professor Paul Newman, for his unwavering support and idea filled discussions. Without his seemingly infinite optimism this thesis would never have materialised.

I would also like to thank every member of MRG who as a team have managed to achieve brilliant things. Working with MRG these last four years has been an honour and I am proud to have been part of it.

Equal thanks must go to my wife Charlotte, my son Isaac and my newly born daughter Adeline. Being greeted everyday with a smile and a hug makes all the hard work worth it[1].

[1] Although a full nights sleep might have come in handy.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Vision Based Localisation 1

1.3 Automatic Sensor Calibration 3

1.4 Contributions 4

1.5 Thesis Structure 5

1.6 Publications 6

1.7 Patents 7

2 Systems and Datasets **8**

2.1 Introduction 8

2.2 Layers of Abstraction 8

2.3 Low Level - Hardware 11

2.3.1 System Architecture 11

2.3.2 Sensors 12

2.3.3 Autonomous Platforms 17

2.3.4 Perception Platforms 18

2.3.5 Data Management 20

2.4 Middleware 21

2.4.1 Timing 22

2.5 High Level Algorithms 22

2.6 Data Sets 22

3 Computing Local Constraints - VO **24**

3.1 Introduction . . . 24

3.2 Overview . . . 26

3.3 Stereo Cameras . . . 28

3.3.1 Projective Camera Models . . . 28

3.3.2 Distortion Correction and Rectification . . . 32

3.3.3 Stereo Triangulation . . . 33

3.4 Data Association . . . 35

3.4.1 Feature Extraction and Description . . . 35

3.4.2 Stereo Matching . . . 38

3.4.3 Temporal Matching . . . 38

3.5 Motion Estimation . . . 39

3.5.1 Outlier Rejection . . . 39

3.5.2 Numerical Optimisation . . . 43

3.5.3 The State Vector . . . 46

3.5.4 Transform Derivative . . . 48

3.5.5 Sensor to Robot Transform Derivative . . . 51

3.5.6 Perspective Projection Derivative . . . 52

3.5.7 The Full Jacobian . . . 54

3.6 Evaluation . . . 57

3.6.1 Camera Placement . . . 57

3.6.2 Evaluation Against INS . . . 61

3.6.2.1 Begbroke Loops Dataset . . . 62

3.6.2.2 Kidlington Dataset . . . 65

3.6.2.3 Overall Results . . . 68

3.7 Conclusion . . . 69

4 Localisation Using Synthetic Road Imagery **70**

4.1 Introduction . . . 71

4.2 Motivation . . . 73

4.3 Background . . . 75
4.4 Exploiting Aerial Imagery . . . 76
4.4.1 Results . . . 78
4.5 The Survey . . . 80
4.5.1 Survey Trajectory . . . 80
4.5.2 Generating Local Orthographic Images . . . 82
4.6 Trajectory Representation . . . 86
4.7 Mutual Information for Image Alignment . . . 87
4.7.1 Demonstrating Lighting Invariance . . . 91
4.7.2 Sensitivity To Number of Bins in the Joint . . . 95
4.8 Optimisation . . . 97
4.8.1 The Objective Function . . . 97
4.8.2 Approximate $\mathbb{SE}2$ Pose Corrections . . . 97
4.8.3 Full $\mathbb{SE}3$ Pose Corrections . . . 99
4.8.4 Outlier Rejection . . . 100
4.9 Results . . . 101
4.9.1 Performance Evaluation . . . 101
4.9.1.1 Ground Truth . . . 101
4.9.1.2 Demonstrating Centimeter Precision . . . 103
4.9.1.3 Optimiser Convergence . . . 104
4.9.2 Approximate $\mathbb{SE}2$ Pose Corrections . . . 104
4.9.3 Full $\mathbb{SE}3$ Pose Corrections . . . 106
4.9.3.1 Begbroke Loops . . . 106
4.9.3.2 Begbroke New Road . . . 113
4.9.3.3 Kidlington - Limited Road Markings . . . 117
4.9.3.4 Woodstock - Severe Lighting Changes . . . 122
4.10 Limitations and Failure Modes . . . 127
4.10.1 Reliance on VO . . . 127
4.10.2 Road Surface Visibility . . . 127

4.10.3 Matching Failures . . . 128
4.11 Conclusions . . . 129

5 Camera Laser Calibration 131

5.1 Introduction . . . 131
5.2 Background . . . 132
5.3 Limitations of Stereo Depth Estimation . . . 135
5.3.1 Depth Estimation Uncertainty . . . 135
5.3.2 Dense Stereo . . . 137
5.4 Problem Formulation . . . 138
5.4.1 Generating A Swathe . . . 139
5.4.2 Timing . . . 140
5.4.3 Generating LIDAR Reflectance Imagery . . . 141
5.4.4 The Objective Function . . . 142
5.4.5 Optimisation . . . 143
5.5 Improving Accuracy With Estimate Fusion . . . 145
5.5.1 Recursive Bayes Filter . . . 145
5.6 Results . . . 147
5.6.1 LIDAR Reflectance Imagery . . . 148
5.6.2 Estimate Fusion . . . 150
5.6.3 Objective Function & Convergence . . . 153
5.7 Conclusions . . . 156

6 Conclusions 157

6.1 Summary of Contributions . . . 157
6.2 Future Work . . . 160
6.3 Concluding Remark . . . 163

A Google Protocol Buffer Example 164

List of Figures

1.1 Variation in the Visual Appearance of a Scene 2

1.2 LIDAR Reflectance Imagery 3

2.1 Layers of Abstraction 9

2.2 System Architecture 11

2.3 Sensor Housing - BumbleBee2, LadyBug2 and SICK LMS 12

2.4 LadyBug2 Panoramic Image 13

2.5 GPS With Differential Corrections 16

2.6 GPS Drift Caused by Interference 17

2.7 Autonomous Platforms 18

2.8 The Nabu Perception Suite 19

2.9 Wiring schematic for the Nabu perception suite 20

2.10 Example Images From Three Locations 23

3.1 Basic Principal Behind Stereo Visual Odometry 27

3.2 VO Pipeline 28

3.3 The Pinhole Camera Model 29

3.4 Image Distortion Correction 32

3.5 Stereo Triangulation 33

3.6 Computing Depth from Disparity 34

3.7 FAST Corners 37

3.8 Stereo and Temporal Matches 40

3.9 Sample and Consensus Methods Example 42

3.10 Trajectory Notation 47
3.11 RBA Jacobian 55
3.12 Graphical Model For Example VO Trajectory 55
3.13 Camera Placement Trials - Example Images 58
3.14 Computing Relative Pose Errors 61
3.15 VO Drift - Begbroke Data 63
3.16 Errors in VO Relative Trajectory against INS - Begbroke Data . . . 63
3.17 VO and INS Relative Trajectories - Begbroke Data 64
3.18 VO Drift - Kidlington 65
3.19 Disagreement in VO relative trajectory against INS - Kidlington Data 66
3.20 VO and INS Relative Trajectories - Kidlington Data 67

4.1 Overview of Using Synthetic Orthographic Images for Localisation . . 71
4.2 Feature Tracking Failure 74
4.3 System Overview Aerial Image Base Localisation 77
4.4 Overview of the coarse-to-fine road matching scheme 77
4.5 Aerial Image Matching Results 79
4.6 Aerial Image Errors Latitude and Longitude 80
4.7 Notation VO Trajectory 81
4.8 Synthetic Overhead Generation 82
4.9 Example Synthetic Overheads 85
4.10 Localisation Framework 86
4.11 View Alignment 88
4.12 Mutual Information - Venn Diagram 89
4.13 Mutual Information - Joint Distribution 90
4.14 Images With Artificial Shadowing 92
4.15 Mutual Information vs Sum of Square Differences 94
4.16 Registration Errors MI vs SSD 95
4.17 Number of Iterations MI vs SSD 96
4.18 Mutual Information Number Of Bins 96

4.19 VO Drift Against Number of Frames Traversed 98
4.20 The Bowler WildCat 102
4.21 Demonstrating centimeter accuracy 103
4.22 Trajectory Estimation Around Begbroke 105
4.23 Example Synthetic Matches 106
4.24 Variation of Mutual Information over the Begbroke Loop 108
4.25 Begbroke Loops Localised Trajectories 109
4.26 Begbroke Loops Cross Track Errors 110
4.27 Begbroke Loops Example Frames 111
4.28 Begbroke Loops Trajectory Corrections 112
4.29 Begbroke Loops Optimiser Convergence 113
4.30 Begbroke New Road Localised Trajectories 114
4.31 Begbroke New Road Cross Track Errors 115
4.32 Begbroke New Road Example Frames 116
4.33 Begbroke Loops Trajectory Corrections 117
4.34 Begbroke New Road Optimiser Convergence 117
4.35 Kidlington Localised Trajectory 118
4.36 Kidlington Cross Track Errors 119
4.37 Kidlington Example Frames 120
4.38 Kidlington Trajectory Corrections 121
4.39 Kidlington - Optimiser Convergence 121
4.40 Woodstock Localised Trajectory 123
4.41 Kidlington Cross Track Errors 123
4.42 Kidlington Example Frames 124
4.43 Woodstock Trajectory Corrections 125
4.44 Woodstock - Optimiser Convergence 125
4.45 Failure Modes 128

5.1 Push-Broom LIDAR with Non-Overlapping Field of View 133
5.2 Example Laser Swathe Woodstock 133

5.3 Transforming a probability density function 135
5.4 Stereo Range *pdf*s as Range Increases 137
5.5 Stereo Range Variance 137
5.6 Dense Stereo vs. Push-Broom LIDAR reconstruction 138
5.7 Sensor Trajectories 140
5.8 Laser Reflectance Imagery 141
5.9 Gradient Enhancement 144
5.10 Laser and Camera Configuration 147
5.11 Example Camera and LIDAR Images 149
5.12 Example Calibration Images 150
5.13 Calibration Precision - Box Plots 151
5.14 calibration Improvement With Recursive Bayes Filter 152
5.15 Assessing the Quality of a Calibration Estimate 154
5.16 Assessing the Quality of a Calibration Estimate More Frames 155

6.1 Example Localised Images Under Varied Conditions 158
6.2 An Example LIDAR Swathe With Camera Images Rendered At Their Capture Locations 159

A.1 An example Google Protocol Buffer definition file. This particular example if for an image grabbed from a camera. 165

List of Tables

3.1 VO Camera Placement Results 60

3.2 Disagreement in relative poses of VO against INS ground truth. . . . 68

4.1 Convergence statistics for Mutual Information vs Sum of Squares Difference 93

4.2 Statistics for Cross track errors over all datasets 126

4.3 Statistics for Mutual Information over all datasets 126

4.4 Statistics for corrections applied by our system over all data sets . . . 126

5.1 Calibration Precision 150

Chapter 1

Introduction

1.1 Motivation

Autonomous robotic systems are set to become ever more prevalent in society. In recent years we have seen robots sent to Mars [63], successfully drive around towns and cities [112] and most recently take part in simulated rescue operations [15]. In order for these systems to be useful they must possess many fundamental competencies and abilities. This thesis, with an eye on road vehicle autonomy, attempts to address two of these competencies; localisation and extrinsic sensor calibration.

1.2 Vision Based Localisation

Knowing where a vehicle is accurately and persistently is essential for self-driving cars. Having a measure of pose and position in the world allows us to plan and control subsequent motion.

A pervasive technology present in most modern vehicles is satellite global positioning (GPS). This technology, when operating in ideal conditions, can provide global pose estimates to centimetre accuracy. However, accuracy is significantly reduced in built up areas and can suffer from complete drop-outs. It is these limitations that motivate attempts to make localisation a self-contained process, with no reliance on external infrastructure.



Figure 1.1: In order to achieve reliable and persistent localisation, vision based algorithms must be versatile enough to cope with dramatic changes in visual appearance. These two sequences show how the appearance of a route varies under changing lighting conditions. The goal of this work is then to achieve full six degree of freedom pose estimation relative to a survey route in the presence of these variations.

In this thesis we attempt to answer the question, can we use vision alone to localise a road vehicle in different driving and environmental conditions? An approach widely adopted is the use of sparse features to match a survey route with a subsequent traversal [27]. This approach has been shown to be relatively brittle to changes in visual appearance such as those shown in Figure 1.1. In [12] they attempt to compensate for this brittleness by acquiring multiple experiences of the same route. Localisation is then achieved by matching against multiple experiences simultaneously. This however leads to large and complex databases of multiple experiences, here we ask if we can achieve the same robustness with a with a single survey?

We describe a possible solution to this problem, which exploits synthetic orthographic images of the road surface created from survey images. Views of these orthographic images are generated and aligned with live image streams using a dense matching procedure based on Mutual Information. We show that using only one survey we can successfully localise subsequent passes of the same route under the



Figure 1.2: An image captured by an off the shelf camera (left) and an image generated from LIDAR reflectance data (right). It can be seen that visually these images are striking similar.

extreme appearance changes expected in outdoor applications.

1.3 Automatic Sensor Calibration

Today robotic platforms are equipped with a plethora of sensors operating in different modalities. Increasingly algorithms running on these platforms attempt to simultaneously exploit multiple sensors, supplementing the short comings of individual sensing devices. An example of this is given in [5]; here they leverage the high geometric precision of LIDAR to extract planar surfaces. Visual texture is then projected onto this planar model from a spherical camera, generating a high fidelity 3D model. In [24] they adopt a similar approach but this time in outdoor environments.

These multi-modal approaches are only possible if the relative positions of sensors are known. In most cases sensors are configured such that the extrinsic calibration can be roughly measured or artificial targets are used. We ask whether it is instead possible to perform a target free automatic calibration, which could be recomputed on the fly?

We present a calibration technique, which automatically and in natural scenes estimates the relative position of a camera and a LIDAR (LIght Detection And Ranging) mounted such that the scanning plane normal is aligned with vehicle motion in what we term a Push-Broom configuration. Rather than requiring a

stationary platform we exploit vehicle motion to build up a 3D point cloud, which allows retrospective comparison of data. We are then able to exploit the fact that grey scale images captured with an off the shelf camera appear similar to those generated using LIDAR reflectance data, Figure 1.2. Dense registration between these images then allows us to estimate the extrinsic calibration between sensors.

1.4 Contributions

The main contributions of this thesis are:

- We present a method of accurately localising a road vehicle using only cameras and a prior survey. We describe how the local accuracy and speed of Visual Odometry (VO) can be complemented by low frequency updates from our Mutual Information based matching technique. The final system produces trajectory estimates commensurate with those of a high end Inertial Navigation System.

- We carry out extensive testing of our localisation algorithm in multiple environments and driving conditions. We demonstrate the suitability of Mutual Information for outdoor localisation and show the system successfully localising under extreme lighting changes and on poorly maintained road surfaces.

- We present a method of automatically calibrating a camera and 2D LIDAR in natural scenes. Vehicle motion is exploited and data aggregated to allow for sensor configurations with non-overlapping fields of view and on the fly calibration. The calibration routine is evaluated on 73 example scenes in different locations.

- We briefly describe work, which exploits aerial imagery to localise a road vehicle. This work inspired the bulk of the work presented in this thesis.

Secondary contributions include:

- We describe a sparse feature based Visual Odometry system. We investigate the effect of camera placement on trajectory estimation over many kilometers of data. We also evaluated the system against an INS with the camera mounted at the determined optimal position.

1.5 Thesis Structure

This thesis is structured as follows. Firstly in Chapter 2 we present an overview of the systems and datasets used in this work. This chapter provides the reader with the context in which the algorithms described in later chapters are evaluated and exploited.

In Chapter 3 we provide a background on locally derived trajectories and visual odometry systems. A reader with experience in stereo vision and ego-motion estimation may skip a number of sections in this chapter. However, the performance evaluation of the described VO system will aid in understanding later chapters.

The bulk of the novel work is presented in Chapters 4 and 5. In Chapter 4 we first give brief overview of preliminary work on a localisation algorithm which exploits freely available aerial imagery. This preliminary work inspired the main contribution of this chapter, a novel localisation algorithm based on a two-stage process, survey and localise. We then present extensive testing in multiple locations and driving conditions.

In Chapter 5 we present a method for automatically estimating the extrinsic calibration between a camera and push-broom LIDAR. We describe how vehicle motion is exploited enabling calibration of sensors with non-overlapping fields of view. Calibration results are then presented for 73 example scenes.

Finally in Chapter 6 we summarise the contributions made in this thesis and present ongoing and future work. Ongoing work of particular interest is the extension of Chapter 4 to arbitrary planar structure in the scene, such as buildings, rather than focusing exclusively on the road surface.

1.6 Publications

The work presented in Chapter 4 was published in the following papers

- Ashley Napier and Paul Newman. Generation and Exploitation of Local Orthographic Imagery for Road Vehicle Localisation. *In Proc. IEEE Intelligent Vehicles Symposium (IV). Madrid, Spain.* June 2012

- Ashley Napier, Gabe Sibley and Paul Newman. Real-Time Bounded-Error Pose Estimation for Road Vehicles Using Vision. *In 13th Intl. IEEE Conference on Intelligent Transportation Systems (ITSC). Funchal, Portugal, pp. 1141-1146.* September 2010

The calibration procedure presented in Chapter 5 and its application to multi sensor localisation was presented in

- Ashley Napier, Peter Corke and Paul Newman. Cross-Calibration of Push-Broom 2D LIDARs and Cameras In Natural Scenes. *In Proc. IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe, Germany.* May 2013

- Colin McManus, Winston Churchill, Ashley Napier, Ben Davis, and Paul Newman. Distraction Suppression for Vision-Based Pose Estimation at City Scales. *In Proc. IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe, Germany.* May 2013

Work not included in this thesis, but carried out by the author during the course of this degree was published in

- Ashley Napier, Sandria Gayadeen and Steven R. Duncan Fast Orbit Beam Stabilisation for a Synchrotron, *Control Applications (CCA), 2011 IEEE International Conference on , pp.1094-1099, 28-30* Sept. 2011

1.7 Patents

- Ashley Napier and Paul Newman. Localising A Vehicle Along A Route, *Pending - WO2013045935*

- Ashley Napier and Paul Newman. Extrinsic Calibration of Imaging Sensing Devices and 2D LIDARs Mounted on Transportable Apparatus, *Pending - GB1219850.3*

Chapter 2

Systems and Datasets

2.1 Introduction

In this chapter we discuss the hardware and software used in the Mobile Robotic Group (MRG). The basic design and operation of the robotic platforms will be described from the physical hardware to the highest-level algorithms. We will also describe the datasets collected and exploited in this thesis.

MRG uses several different robotic platforms, each designed for particular application domains, described in Section 2.3.3. The robots available to the group allow us to explore mobile robot autonomy; indoors and outdoors[1]. An understanding of these systems will aid the reader in understanding later chapters.

2.2 Layers of Abstraction

Development of algorithms for autonomous applications is a very complex affair. If all research projects were required to start from the fundamentals of reading data from sensors and implementing control systems for actuators, development would be slow and group efforts almost impossible. For this reason a lot of effort has been invested in developing fundamental cross platform software, which can simply be installed on new robotics platforms. This framework places components in one of

[1]For a full list of MRG's platforms please visit http://mrg.robots.ox.ac.uk

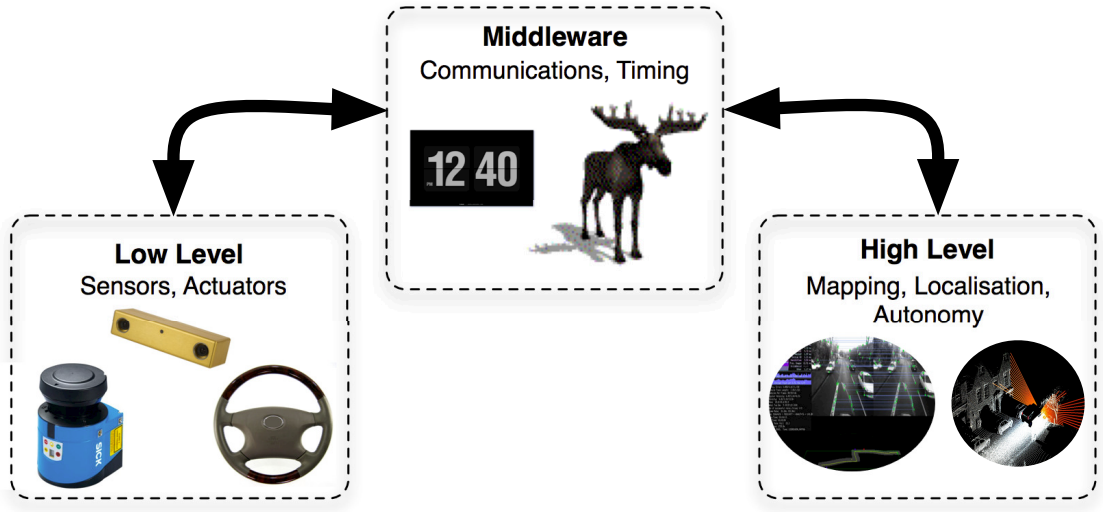


Figure 2.1: Layers of Abstraction. Software components of a robotic platform are placed into one of three levels of abstraction. At the lowest level are drivers and logging systems, which allow direct interaction with hardware. At the highest level are the mapping and localisation algorithms, middleware facilitates the communication between high level and low level systems.

three layers of abstraction, see fig 2.1. These layers allow us to, for example, grab an image from a camera without having to know anything about the communication protocols used by that particular camera.

Low Level - Hardware, Drivers & Logging

The physical hardware used in a robotic platform is the lowest level. For autonomous robots the hardware may consist of cameras, lasers, motors and actuators. All these devices have different interfaces and communicate over different standards. The function of the low level software is to interface with these varied devices and transform the raw data into a standard and usable form. The data from this level is then sent to the middleware where it can be consumed by other processes.

Another function of the low level software is to log the data recorded from hardware devices. As well as performing live experiments, it is advantageous to record data for later experimentation and performance evaluation.

Middleware

The middleware is essentially a hub for all processes to communicate. Drivers send standardised data to the middleware from which it can then consumed by a high level algorithms and visa versa.

Having this layer removes the need to include basic hardware code in the higher-level algorithms and allows systems to be more modular. For example if several systems consume images from a camera, it is advantageous for them to request images from middleware rather than directly from the camera. This abstraction away from the actual hardware means none of the algorithms have to include driver code and multiple processes are able to acquire images from the same camera via the middleware without issue.

Modern platforms consist of many devices, each with their own clock. In order to accurately know when things happen, it is essential that these distributed clocks be properly synchronised. An added benefit of having all data pass though middleware is the ability to perform clock synchronisation. Here all clocks can be calibrated against a master clock, achieving sub-millisecond synchronisation.

High Level - Mapping, Navigation, Object Detection

The highest level of the software stack is where the actual autonomy resides. The two lower levels provide a platform, which can be given instructions in a standard form, to for example, provide an image or move at a certain velocity. Without the high level algorithms a robot is little more than a remote controlled vehicle.

The high level algorithms perform mapping of the environment, localisation, object detection and planning among other things. Having these algorithms abstracted away from the physical hardware allows development, which is agnostic to the platform itself. Instead we are able to focus on the actual application or behaviour rather than the platform.

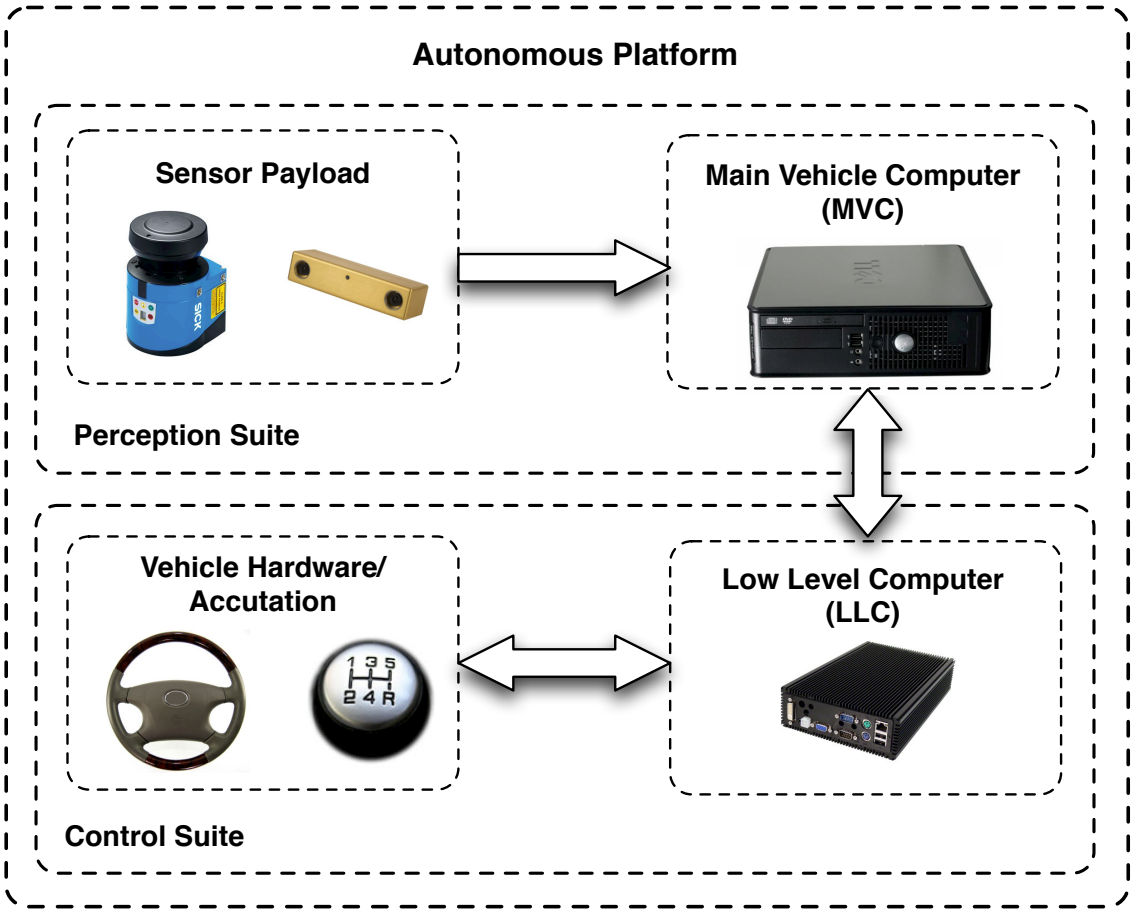


Figure 2.2: System Architecture. The hardware is arranged into two suites, the first a perception suite consisting of sensors and a computer able to run high-level algorithms. The second is a control suite, which is, responsible for controlling the engine, breaks, signals etc. and enables autonomous control.

2.3 Low Level - Hardware

The hardware like the software is modularized for simplicity and flexibility. MRG has several advanced platforms; here we will describe the platforms used to explore road autonomy.

2.3.1 System Architecture

Each platform can consist of two independent subsystems; a perception suite and a control suite, see fig 2.2. The separation of these two systems allows easy conversion of standard vehicles into fully autonomous vehicles and also allows standard vehicles to be used in a survey mode where no automatic control is required.

This separation is achieved by having two computers on each vehicle. The Main



a) BumbleBee2 b) Ladybug2 c) SICK LMS

Figure 2.3: Images of the three main sensors used in this work; a) BumbleBee2 – an off the shelf stereo pair, b) LadyBug2 – an off the shelf spherical camera consisting of six monocular cameras, c) SICK LMS – Laser range finder with $12mm$ accuracy up to $50m$.

Vehicle Computer (MVC) is a relatively powerful machine to which all sensors are connected and on which the high level algorithms run. The perception suite can run independently of the control suite allowing platforms such as the Nabu, described in Section 2.3.4. The Nabu is a small unit, which can be mounted on any car roof. It is able to perform all the high level localisation and mapping tasks without vehicle control, effectively making it a very quick and easy to use survey unit.

The Low Level Computer (LLC) is what enables actual vehicle control for fully autonomous systems. This is a real time computer to which all actuators are interfaced. This machine performs basic control of speed, steering, gears and indication. Messages can be passed between processes running on both computers via the middleware.

2.3.2 Sensors

It could be argued that the most important components of a robotic platform are the sensors. These are the devices with which the world is measured and subsequently perceived. For this reason we will now discuss the sensors used in this work independently of the platforms, which are all fitted with similar sensor configurations.



Figure 2.4: Example panoramic image captured using the LadyBug2

Stereo Vision - BumbleBee2

Stereo cameras consist of a pair of monocular cameras rigidly connected such that there is overlap in the images captured by each camera. The separation of the two cameras introduces a disparity, which can be used to compute the 3D geometry of a captured scene. In order to do this the internal characteristics of the camera must be known as well as the relative pose between them.

The Point Grey BumbleBee2 is an off-the-shelf stereo pair. The advantage of using a purpose built stereo pair as opposed to a custom built stereo rig is increased precision. Stereo camera units are manufactured such that the baseline is guaranteed to be within a certain tolerance, in this case sub-millimetre. The rigid construction of the units also means the image sensors cannot easily move relative to one and other. Another advantage is timing; the capture process is internally synchronised such that both left and right images are taken at the same time.

Here we use a model with a 67° field of view, capturing 512×368 resolution images at $20fps$ with a baseline of $12cm$. A global shutter reduces image distortion, which can be present with rolling shutter cameras [60]. The camera is low power, consuming $2.5W$ at $12v$ and uses a 6-pin IEEE 1394a interface. The camera also has 4 GPIO pin for external triggering if required.

Spherical Camera - LadyBug2

A spherical camera can be thought of as a collection of monocular cameras arranged such that the full horizontal and vertical fields of view are captured. Images from

the spherical camera can be used as single monocular images or stitched together to form a panorama or spherical image.

The 360° view is particularly useful for road vehicle autonomy as it enables an entire scene to be captured using a single sensor. In a single frame the road surface, building façades, and other vehicles are captured. Although multiple monocular cameras can be placed around a vehicle to provide an equivalent view, the LadyBug2 benefits from the same precision as the BumbleBee2 as it is manufactured as a single unit.

The Point Grey LadyBug2 is a 360° spherical camera. It consists of six 1024×768 pixel CCDs positioned such that five provide a full 360° horizontal view and the sixth is directed upwards. The camera is capable of running at up to $30fps$, however in general this is throttled to lower rates. Figure 2.4 shows an example stitched panoramic image of an urban scene. The total resolution of each frame is 4.7 Mega pixels.

Output data rates for the LadyBug2 can be as high as $80Mbs$. A typical lap of our test track at Begbroke can take approximately five minutes to complete and generate over $2GB$ of image data. Images collected from this sensor are therefore stored in their raw Bayer format [3] reducing the files sizes to a $1/3$. When required for processing images can be de-Bayered into full RGB images without any loss.

LIDAR - SICK LMS151

LIght Detection and Ranging (LIDAR) is an active sensing technology. Unlike cameras, which are passive, LIDAR does not rely on ambient light to record measurements. LIDAR units emit laser light and record returns. Distances can be computed using the time of flight and the reflectivity of the target object inferred from the attenuation of the original beam.

This active technology allows operation during both day and night, and unlike stereo, scene geometry is measured directly rather than inferred. LIDAR however, is not perfect and sufferers from effects such as mixed measurements. Here errors

are introduced due to the laser beam being cone shaped rather than an idealised line. As the distance from the sensor increases the area of the beam also increases leading to multiple returns. These multiple returns lead to spurious points in range measurements, there are however methods of detecting and removing these mixed measurements [101].

LIDAR sensors generally come in one of two flavours 2D or 3D. 3D sensors send beams out in all direction to create three-dimensional set of range measurements. A very popular example of a 3D LIDAR is the Velodyn HDL-64E, costing upwards of $70,000. The unit consists of 64 individual beams and generates full 360° horizontal and 40° vertical field of view scans at $10Hz$. 2D LIDARs scan in a single plane, effectively measuring a slice of the world. If these sensors undergo motion the individual slices can be stitched together to make a full 3D point cloud. Examples of 2D LIDAR used to make 3D sensors include mounting sensors on a nodding mechanism [38], a spinning plate [96] and a moving platform [73].

In this work we use the SICK LMS151, this is a 2D LIDAR with a 270° field of view. Scans are captured at $50Hz$ and consist of 512 measurements with an angular resolution of 0.5^o. The measurement error has an accuracy of of $\sigma = 12mm$ up to a range of $50m$. The LMS151 communicates over Ethernet using a TCP/IP interface.

INS with DGPS - OxTS

Inertial Navigation Systems (INS) provide high frequency pose updates in a global frame. These pose measurements are used as ground truth allowing quantitative performance analysis of other localisation systems.

The INS is a fusion of Inertial Measurement Unit (IMU) and Global Positioning System (GPS) measurements. Taken alone pose estimates derived from an IMU, accelerometers and gyroscopes, are prone to drift and require a good estimate of the direction of gravity. This drift makes poses derived from inertial measurements alone meaningless in a global frame after any length of time. GPS on the other hand offers globally consistent pose measurement, but is generally noisy, offering at best 10m

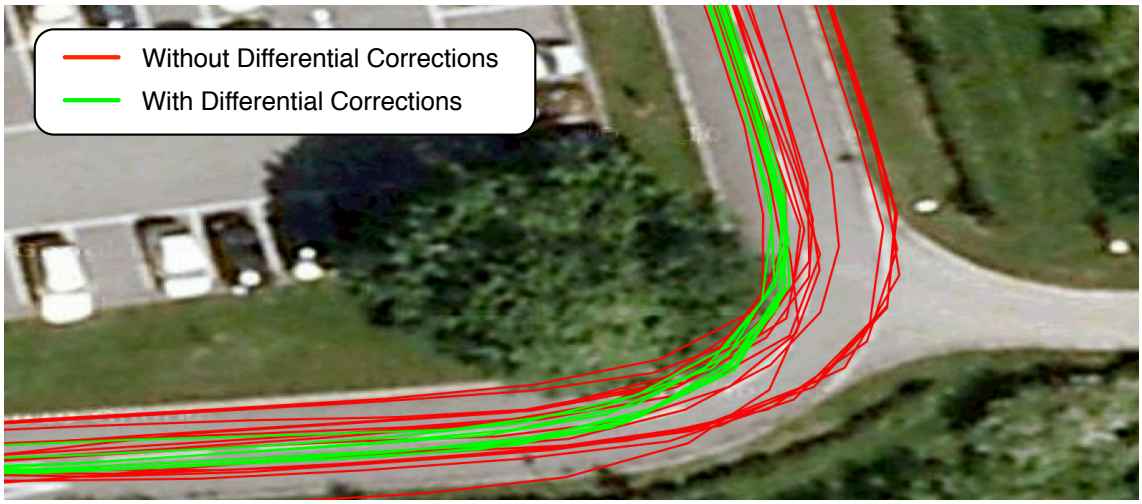


Figure 2.5: GPS With and Without Differential Corrections. The red paths show how without differential corrections the pose estimates can drift by several meters. The green paths show how this drift is removed when corrections are received from a DGPS base station.

accuracy. The fusion of these two sensor modalities allows the shortcomings of each to be compensated for. The end result is smooth high frequency pose estimates with inter-frame accuracy of better than $10cm$ but still subject to global drift associated with standard GPS, see Figure 2.5.

This drift can be corrected for using Differential GPS (DGPS). Here a stationary base station measures its position using GPS and estimates the error by exploiting the prior knowledge that it itself is stationary. This allows corrections to be made for atmospheric conditions that can cause GPS pose estimates to drift. These corrections are sent via a radio modem to the mobile INS. The improved accuracy of DGPS over GPS can be seen in Figure 2.5, this data was recorded over many months, and demonstrates how DGPS measurements more closely reflect the true vehicle poison in the left lane.

Even with DGPS trees, bridges and buildings can occlude signals from satellites and cause a loss in reception. Here, when GPS is not available, the vehicle pose is estimated using the IMU alone, which drifts quickly away from the true vehicle position, see Figure 2.6. Multipath interface is a large problem in built up urban environments. Here signals from satellites arrive at the GPS receiver directly from the satellite as well as via reflected paths from buildings. There have been many

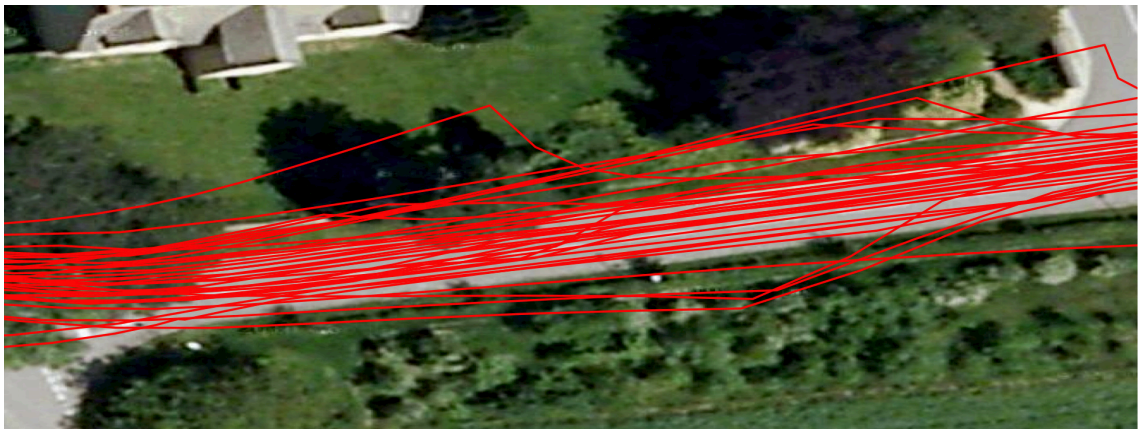


Figure 2.6: GPS drift caused by interference from overhanging foliage. It can be seen that as satellite reception is lost the pose estimates rapidly diverge away from the true vehicle position.

attempts to detect and correct for this, an example is given in [105].

In this work we use the dual-antenna OXTS RT3042 INS with DGPS corrections from a RT-Base GNSS Base Station. This unit provides full position and orientation measurements at $1kHz$ with an accuracy of $0.02m$ in position and $0.03degrees$ in rotation.

2.3.3 Autonomous Platforms

MRG has two main autonomous platforms, the WildCat and RobotCar, which are used for development and testing. The sensors described previously, among others can, be mouthed on these vehicles for quick and easy collection of data.

The Wildcat

The WildCat is a heavily modified Bowler WildCat, see Figure 2.7 left. This platform has been developed for rapid development and testing of algorithms. All around the vehicle and on the roof are aluminium mounting plates with M8 threaded holes. These mounting platforms allow rapid turnaround for testing new sensors as well as testing new sensor configuration.

The vehicle steering and breaking systems are fully actuated and can be controlled by navigation systems running on the MVC.



Figure 2.7: Two of MRG's fully autonomous road vehicles. Left - the Bowler Wildcat, Right - RobotCar, a converted Nissan LEAF.

RobotCar

RobotCar is a modified Nissan LEAF, an all-electric vehicle, see Figure 2.7 right. RobotCar, unlike the WildCat, is a vision of what autonomous vehicles might look like in the future. Flexibility has been sacrificed for finish as sensors are hidden away behind glass and under bumpers. RobotCar is platform used for testing more mature technologies and is also used as a survey vehicle.

Again all the vehicle systems are actuated and can be fully controlled by our navigation algorithms.

2.3.4 Perception Platforms

Many of the algorithms developed by the group such as localisation, obstacle detection and scene labelling don't require a fully autonomous platform for testing and development. The system architecture discussed in Section 2.3.1 allows us to easily build systems, which is capable of running all of these algorithms in complete isolation from any of the actuation systems. This led to the development of the Nabu Perception Suite.

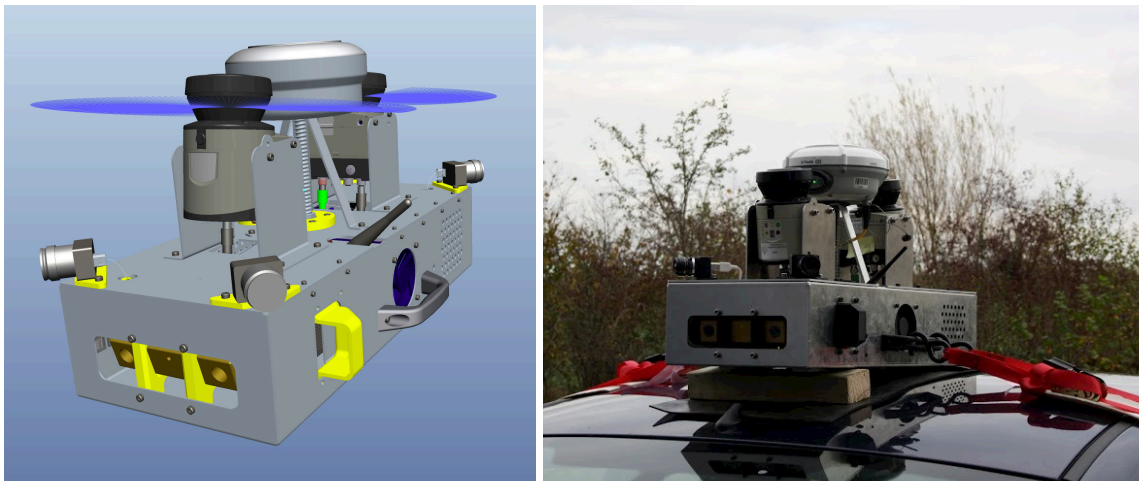


Figure 2.8: The Nabu perception suite. An independent unit equipped with a full complement of sensors and MVC. Left - A CAD model of the Nabu, Right - the Nabu mounted on a hire vehicle.

The Nabu

The Nabu is a self-contained unit equipped with a multitude of sensors and an internal computer. The unit is powered from $12v$ DC and can be controlled via wired Ethernet or a $Wi-Fi$ connection. The unit is designed such that it can be mounted on the roof of any vehicle, subsequently turning that vehicle into a survey platform. A CAD model and the actual unit mounted on vehicle can be seen in Figure 2.8. This platform allows the group much more freedom to collect data in environments where it may not be appropriate to use the WildCat or RobotCar.

The body of the Nabu is a custom designed piece of laser cut aluminum. The aluminum not only makes the unit robust but also, acts as a thermal sink dissipating heat generated by the sensors and MVC. In addition to this the unit has several fans and ventilation ports. The LIDAR mounts allow the pitch of the lasers to be adjusted as required. This allows the LIDARs to be used in a push broom or horizontal scanning orientations.

A schematic for the Nabu can be seen in figure 2.9. The unit takes a $12v$ power supply, which is stepped up to $24v$ for the LIDAR units. The MVC is based on an Intel $i7$ CPU, has $512MB$ DDR-RAM and a $512GB$ Solid State Drive for data storage. Sensors are connected to the MVC via one of Fire wire, USB or Ethernet

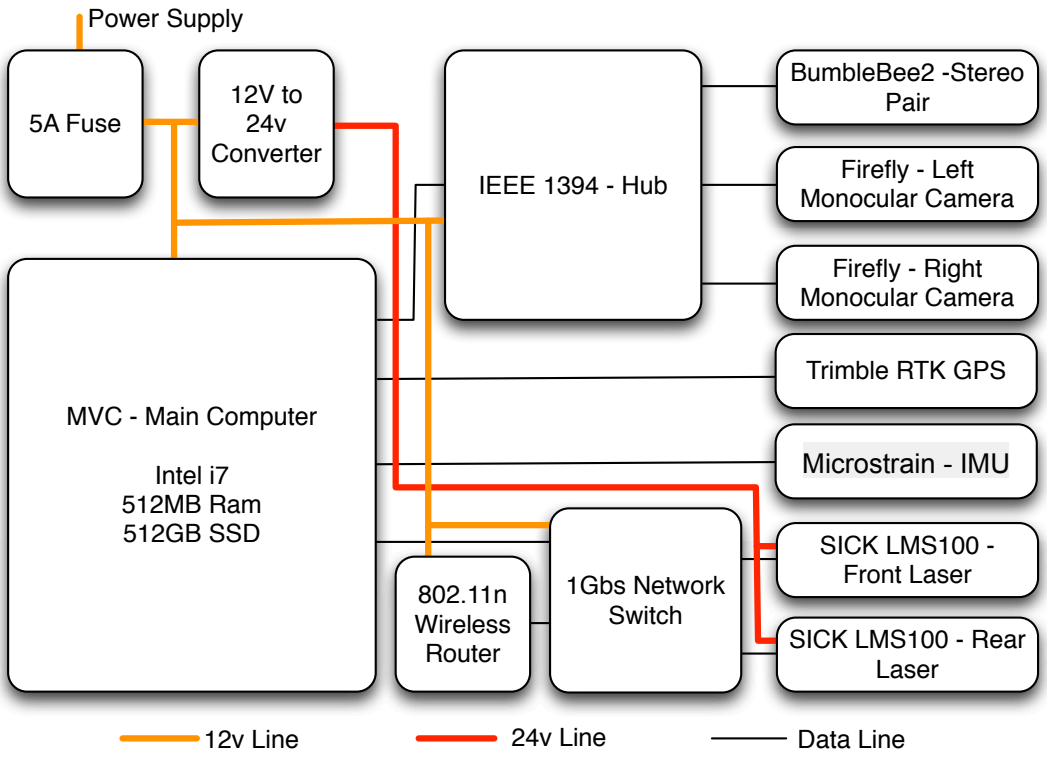


Figure 2.9: Wiring schematic for the Nabu perception suite. The unit requires a 12 volt power supply. Control of the unit is achieved via secure shell (SSH) connection over wired Ethernet or 802.11n wireless networking. It should be noted that all that is required to enable the Nabu to control a robotic platform is an Ethernet connection to a control suite.

interfaces and power is supplied as required.

2.3.5 Data Management

In order to run experiments and evaluate performance offline all data is logged. Traditionally data from sensors are stored in their own custom text files or in the case of cameras individual image files. This approach however does not scale well as a single dataset can contain, for example, thousands of images from many cameras. Managing this plurality of files individually is cumbersome and in many cases exceeds the limits placed on a OS's file system. Also if sensors output text files in their own format, parsers for each sensor must be written.

For these reasons data from each sensor is stored in a single file for each dataset, we refer to these files as monolithics. Google Protocol Buffers (GPBs) [33] are used to write and read to and from these files respectively, standardising the process for

all data.

GPBs are a language-neutral, platform-neutral serialisation and de-serialization tool. They allow structured data or messages to be defined using a `.proto` file. The fields within each message contain raw sensor data as well as meta data such as timestamps, source sensors and sensor specifications. Each field is defined as either required or optional and given a type. GPBs are backwards compatible making them robust to modifications of the `.proto` definition and do not require optional fields to be filled. See Appendix A for an example `.proto` file. Data access classes can be generated for c++, Java and Python by simply running the GPB compiler over the required `.proto` files.

2.4 Middleware

Middleware is what sits between the Low Level Hardware and the High Level algorithms. The middleware provides an opaque layer of abstraction to protect the High Level algorithms from the details of Low Level systems. All data shared between processes goes via the middleware.

There are several choices for middleware specifically written for robotic applications. The Robotic Operating System (ROS) [89] developed by Willow Garage is ever increasing in popularity. ROS not only aims to facilitate message passing between processes but also aims to be a complete operating system for robotic platforms. The operating system comes with a multitude of drivers and libraries to aid in the development of robotic applications and is open source.

The Mobile Robotics Group, however, uses the Mission Orientated Operating System (MOOS) [75]. MOOS is developed in house using C++ and is platform agnostic. It consists of a set of applications and libraries, which facilitate robust, high-speed communications between processes in a robotic system. While MOOS is much lighter weight than ROS it provides all of the functionality required by MRG.

2.4.1 Timing

Timing is critical in robotic systems; it is of utmost importance to know when things happen. Even the smallest timing error could place a vehicle in the wrong location, misjudge the position of a pedestrian or delay a break command. In order to ensure all timestamps from all processes are synchronised we use TicSync [39]. This algorithm is able to correct for offsets and skews in sensor and system clocks, enabling sub-millisecond precision. We will discuss this algorithm in greater detail in Section 5.4.2.

2.5 High Level Algorithms

the true autonomy lies in the high level algorithms. This layer concerns itself with more abstract problems such as mapping, localisation and object detection.

In terms of our application domain, autonomous vehicles, there are many core competencies we much achieve to enable safe and reliable operation. Arguably, the most important of these is 'knowing where you are'. This is a starting point from which all other actions can be assessed and carried out. We have several localisation systems, which are in various stages of development. In Chapter 4 we discuss one of the vision-based systems which exploits prior survey data.

In addition to the above, other high level algorithms deal with problems such as object detection and path planning. Object detection is currently performed using active sensors such as the on-board lasers. However, algorithms are in development which use only passive cameras to detect obstacles, such as other vehicles, cyclists and pedestrians.

2.6 Data Sets

The platforms available to us make data collection a relatively low cost exercise. In this work we use data collected using these platforms at various locations around Oxfordshire.



Figure 2.10: Example images from three data sets used in this work.

During data collection all sensors listed in Section 2.3.2 are logged, generating many Gigabytes of data. In order to evaluate the algorithms presented in this thesis data were collected from multiple environments in various lighting conditions. Figure 2.10 shows example frames from datasets collected at Begbroke, Kidlington and Woodstock. The data used here cover over $7km$, consist of over $70,000$ images and many millions of laser measurements.

Our vanilla datasets are collected around our test site at Begbroke. They mostly cover well-maintained private roads, which allow us to set a performance benchmark. More challenging datasets were collected around Kidlington and Woodstock. The Kidlington data set covers suburban roads with a poorly maintained road surfaces and limited road markings. The Woodstock dataset is the most challenging, and contains sequences with extreme lighting changes, glare and oncoming traffic.

Chapter 3

Computing Local Constraints - VO

In this Chapter we describe and evaluate a state of the art Visual Odometry system. We start with an overview of the fundamentals of stereo vision and describe the underlying mathematics. We then investigate the effect of camera position on trajectory estimate accuracy and compare the optimal performance against an Inertial Navigation System.

3.1 Introduction

Estimating a vehicle's motion and pose via locally mounted sensors has been an extensively researched area. Traditional approaches have exploited sensors commonly mounted on mobile platforms such as wheel encoders, gyroscopes, accelerometers and depth sensors. The use of wheel encoders alone limits pose estimates to in plane motion (three degrees of freedom) and effects such as wheel slip and skidding lead to significant drift. Fuke [25] and Chung [11] fused dead reckoning from wheel encoders with an Inertial Measurement Unit (IMU) to account for these limitations and extend pose estimation to six degrees of freedom. Other approaches use range sensors such as LIDAR (Light Detection And Ranging) [17, 51]; here subsequent range scans are rigidly aligned to produce inter-scan pose estimates. This rigid alignment does not account for motion during the time taken to complete an individual scan, which in slower systems can be up to one second. This was addressed in [97] where they

achieved non-rigid point cloud registration by representing trajectories as splines.

An almost ubiquitous approach made popular in the last few decades by evermore inexpensive and powerful cameras and computing hardware is Visual Odometry (VO). The Stanford Cart 1980 [70] was the first platform to use cameras to estimate motion and scene structure. This four-wheeled vehicle used a stereo pair to triangulate the 3D position of objects and infer vehicle motion between consecutive images.

VO techniques fall largely into two categories; sparse and dense. Sparse techniques [70, 76] do not use the entirety of the captured image to estimate 3D structure and motion. Instead, to reduce computational cost they focus on small patches of image. Early approaches used Harris Corners [37] to extract key points and matched them between images using simple correlation based measures. The limited longevity and robustness of these simple matching techniques lead to the development of more complex and robust descriptors such as SIFT [62] and BRIEF [9], discussed in Section 3.4.1. More recent VO implementations such as [100, 26] exploit these more advanced descriptors, achieving improved performance.

Dense approaches attempt to exploit the whole image. These approaches are often more computationally expensive, but do not suffer from the complexities of feature tracking and matching associated with sparse approaches. Lovegrove [61] uses an automobile rear-parking camera to capture images of the road. Assuming a planar road surface the whole image is warped such that it is fronto-parallel and is then registered against the previous image. Here a simple sum of squares difference measure is used to measure image alignment over all image data. Other dense approaches using dense stereo [74] and more recently the Kinect structured light sensor [102] construct a 3D model to perform full image registration.

An issue associated with all motion estimates derived from local sensors without reference to prior data or infrastructure is accumulated drift. Various approaches have attempted to minimise drift. Thrun [108] attempted global consistency using an offline optimisation. Here data was post-processed in a large-scale optimisation. Instead of simple frame-to-frame motion the entire trajectory and 3D scene was

solved for in a joint optimisation. This approach limits the size of environment an autonomous platform can explore and carries significant computational cost. It is also questionable whether or not global consistency is even required if we have high accuracy local trajectory estimation. More recently the idea of adopting a sliding window of the most recent N poses [46, 100] has improved the local metric accuracy of trajectory estimates. Here rather than solving for the entire trajectory a subset of the latest N poses is used. These systems run in real-time and are less brittle than simple frame-to-frame VO, the latter approach is also referred to as relative bundle adjustment.

Other methods of improving the accuracy of VO include fusing camera data with an IMU [92, 104] and using multiple stereo cameras [80]. Here we have focused our attention on stereo VO, however trajectory estimation is also possible with a monocular camera. Parallel Tracking and Mapping (PTAM) [49] is a real-time system using a monocular camera, however the scale of trajectory estimates and structure is not recoverable. Often monocular VO systems are fused with an IMU in order to recover scale [50].

3.2 Overview

This chapter closely follows the system presented in [12]. We provide a high level overview of the VO algorithm and the underlying mathematics. A simplified representation of the fundamental principal behind sparse feature-based VO is represented in Figure 3.1. Given a sequence of stereo images, $\mathcal{F} = \{\mathcal{F}_k, \mathcal{F}_{k+1}, \mathcal{F}_{k+2}, \dots\}$, which we refer to as frames, we take measurements, $\mathbf{z}_{k,i} = [u, v]^T$, of 3D landmarks, $\mathbf{l}_i = [x, y, z]^T$, and match them between stereo images and frames. The underlying trajectory and scene geometry that this network of measurements describes is then inferred.

Figure 3.2 describes the VO pipeline in more detail. As new frames are captured the images are undistorted and rectified, as described in Section 3.3.2. Stereo measurements are then made, Section 3.3. Here we extract and describe measurements

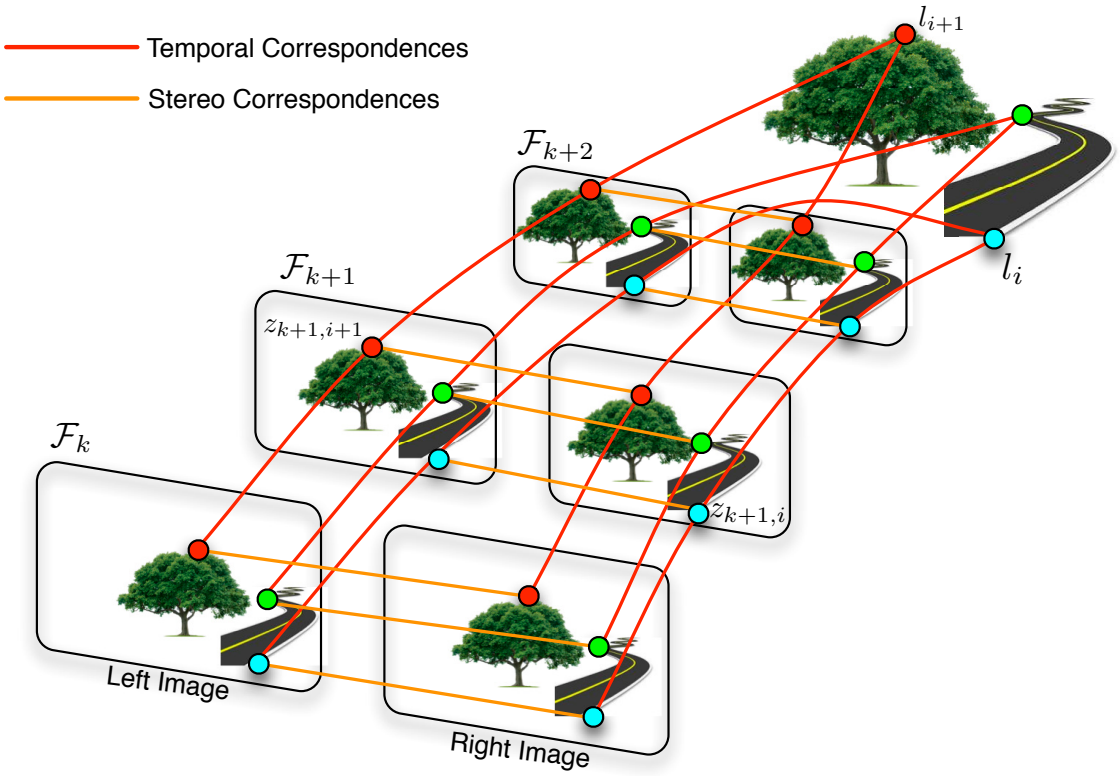


Figure 3.1: The basic principal of stereo VO, here measurements, $z_{k+1,i}$, of landmarks, l_i, are taken in multiple stereo pairs $\mathcal{F}_k, \mathcal{F}_{k+1}$ and $\mathcal{F}_{k+2}$. These measurements are matched between stereo image pairs and temporally between frames, denoted by orange and red lines respectively. Stereo geometry is then invoked to infer the motion of the camera and the position of the landmarks from this graph of observations.

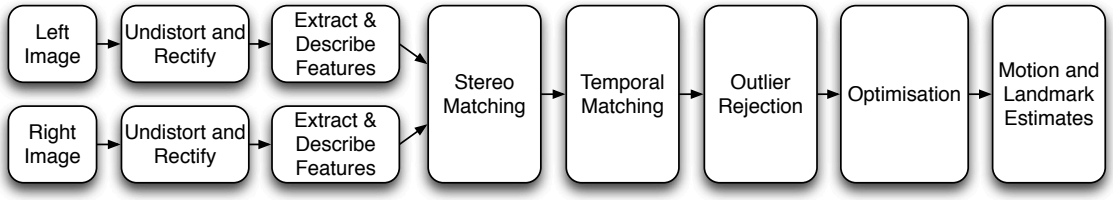


Figure 3.2: VO Pipeline

using FAST corners [91] and the BRIEF descriptor [9], see Section 3.4.1. Data association between stereo images and frames is then performed. This process also, referred to as matching, is covered in Section 3.4. As with all real systems data association is not perfect, since errors in this step can lead to significant errors in the final trajectory estimate. It is therefore essential that an outlier detection and removal step is included. Here we use MLESAC, a sample and consensus method [110] described in Section 3.5.1. The outlier rejection step provides us with a coarse estimate of scene geometry and motion, which is then refined using a numerical optimisation which includes all inlier data, as described in Section 3.5.2. Finally an estimate of vehicle motion in six degrees of freedom as well as sparse scene geometry is produced as output.

3.3 Stereo Cameras

3.3.1 Projective Camera Models

Cameras in general are passive sensors which provide a two dimensional representation of the world by capturing ambient light. In a more technical sense they project world points in $\mathbb{R}^3$ to points on the image plane in $\mathbb{R}^2$. There are many types of cameras with various projection models. For instance orthographic cameras capture rays perpendicular to the image plane such that all rays are parallel and all surfaces are imaged under an affine transformation. On the other hand ideal perspective and spherical projective cameras capture light such that each ray passes through the camera's optical centre intersecting the imaging plane or sphere respectively. This projective transformation is non-linear and results in objects further away from the

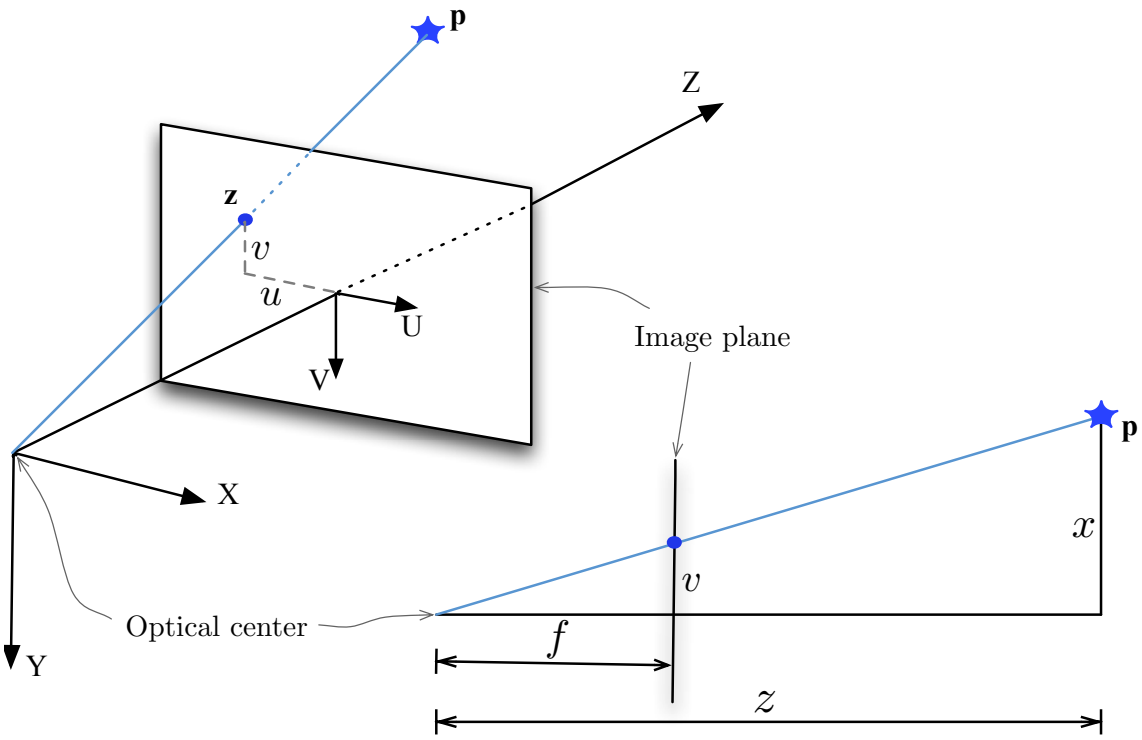


Figure 3.3: The pinhole camera model. Each $\mathbb{R}^3$ point $\mathbf{p}$ projects onto $\mathbb{R}^2$ pixel $\mathbf{z}$. The components of $\mathbf{z}$ can be computed using similar triangles as shown for u in the lower section of the figure.

camera appearing smaller and parallel lines out of the image plane converging to a vanishing point.

In this work we synthesise orthographic imagery but we only concern ourselves with perspective projective cameras. In practice images from real cameras suffer from many types of distortion, which are covered in Section 3.3.2. So assuming we have an ideal perspective projection camera, a standard pin hole camera model can describe the mapping from $\mathbb{R}^3$ to $\mathbb{R}^2$, shown in Fig 3.3.

First we define the camera frame with the origin at the optical centre of the camera. We adopt the standard computer vision coordinate frame convention with Z forward [40]. A point $\mathbf{p} = [x, y, z]^T$ in the camera frame is mapped to a pixel location $\mathbf{z} = [u, v]^T$ using similar triangles as can be seen in Fig 3.3. As $\mathbf{p}$ is in the camera's reference frame u and v can be computed independently for a known focal length f:

$$\frac{u}{f} = \frac{x}{z} \tag{3.1}$$

$$u = \frac{xf}{z} \tag{3.2}$$

Applying the same principal to y we can now computed both elements of $\mathbf{z}$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{xf}{z} \\ \frac{yf}{z} \end{bmatrix} \tag{3.3}$$

The projection shown is equation (3.3) is non-linear. We can however linearise this projection by adopting a homogenous coordinate system. This is achieved in both 3D and 2D by appending the vector $\mathbf{p}$ with 1 such that $\bar{\mathbf{p}} = [\mathbf{p^T}, \mathbf{1}]^\mathbf{T}$:

$$\begin{bmatrix} u' \\ v' \\ \lambda \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.4}$$

$$\mathbf{z}' = [\mathbf{K}|\mathbf{0}]\bar{\mathbf{p}} \tag{3.5}$$

It is important to note that $\mathbf{z}'$ in homogenous form is only up to scale. The non-homogeneous form can be recovered when required via the following:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{u'}{\lambda} \\ \frac{v'}{\lambda} \end{bmatrix} \tag{3.6}$$

The matrix $\mathbf{K}$ in equation (3.5) is the 3×3 matrix of camera intrinsics or camera calibration matrix. In reality the idealised configuration shown in Fig 3.3 does not apply as light is projected onto physical pixels rather than an idealised plane. To account for this $\mathbf{K}$ needs to be populated with more parameters to properly represent real cameras:

$$\mathbf{K} = \begin{bmatrix} f_u & \gamma & u_o \\ 0 & f_v & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

In most cases the origin of the image sensor or image plane does not align with the optical center of the camera. The point at which the Z axis intersects the image plane is called the principal point and can be accounted for with an offset $[u_o, v_o]^T$. Other imperfections in real cameras can include askew and non-square pixels. A skew is introduced when the U and V axes are not perfectly perpendicular. This is accounted for with the skew parameter γ . Deviation from perfectly square pixels on a CCD or equivalent sensor can be accounted for with independent focal lengths for each axis f_u and f_v.

In order to achieve high levels of accuracy $\mathbf{K}$ must be obtained for each camera individually. The need for this can be demonstrated by looking at $\mathbf{K}$ for two Point Grey BumbleeBee2 cameras[1] of the same model:

$$\mathbf{K_{BB2-12210179}} = \begin{bmatrix} 387 & 0 & 255 \\ 0 & 390 & 197 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{K_{BB2-7400627}} = \begin{bmatrix} 390 & 0 & 256 \\ 0 & 390 & 194 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Here we see a several pixel difference in three out of four of the parameters in $\mathbf{K}$ arising from the variability in the cameras manufacturing process. Often manufactures will provide the camera calibration upon purchase but if not there are a plethora of methods available to estimate $\boldsymbol{K}$ oneself. A popular method is described in [115, 47]. Here a plurality of images of a known structure such as a checkerboard are used to estimate the camera intrinsics online.

[1] In this work we refer to the BumbleBee2 model with; 65° Field of view, 512×368 image size, $20ms$ global shutter and f/2.0 (3.8 mm focal length) aperture.



Figure 3.4: These images show the distortion caused by lens systems (left) and the recovered image with the distortion corrected (right). Note that the distortion is more prominent in the peripheries of the image and makes the edges of the building appear bent.

3.3.2 Distortion Correction and Rectification

The pin hole camera model described in Section 3.3.1 in reality does not describe the behaviour of most cameras. In order for a camera to obey the pin hole model the aperture must be very small, limiting the amount of light able to reach the image sensor. Most modern cameras employ a system of lenses allowing larger apertures and a wider field to view. This allows them to achieve the shutter speeds and low light performance expected of modern cameras.

The major sources of distortion introduced by the use of lenses include radial and thin prism distortions [117]. These effects can be modelled and accounted for in same calibration process used in determining $\mathbf{K}$. The effect of these distortions can be seen in Fig 3.4. Once the mapping from the distorted image to the undistorted image is determined distortion correction can be implemented as a look-up table mapping pixels from the raw distorted image to an undistorted image which obeys the pin hole camera model.

When using a stereo pair or stereo camera the additional step of rectification is beneficial. Rectification aligns the left and right image such that a row of pixels in the left image is aligned with the same row in the right image. Another way of saying this is that for a point $\mathbf{p}$ projected into both images $v_l = v_r$ reducing the

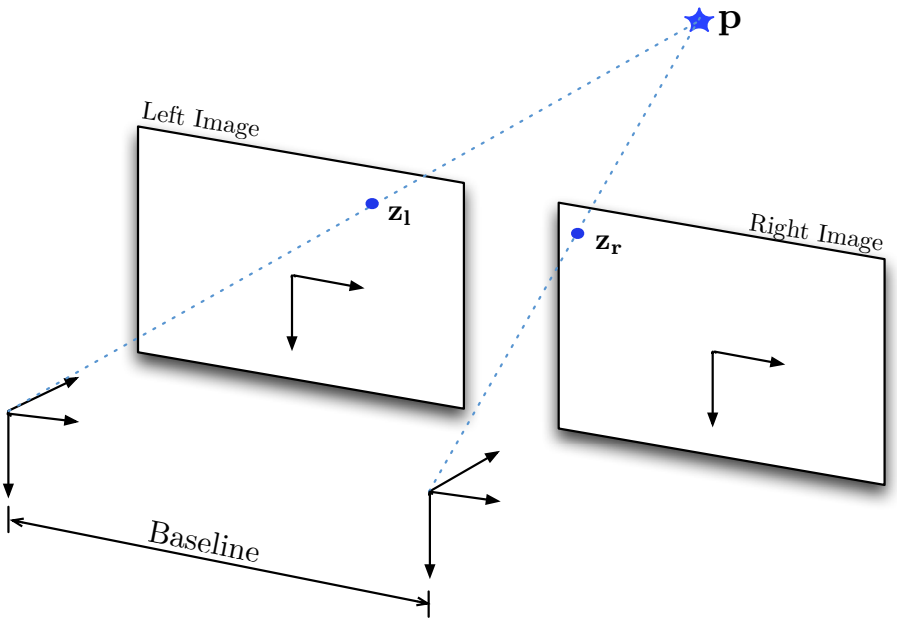


Figure 3.5: The stereo model for a pair of fronto-parallel pinhole cameras. The cameras are separated along the **X** axis by a known distance, b. The point **p** is projected into the left and right image at pixel locations $\mathbf{z_l}$ and $\mathbf{z_r}$ respectively.

search for stereo correspondences to line searches along rows of pixels.

3.3.3 Stereo Triangulation

A single image of a point **p** yields information about the ray on which the point lies, but without reference to further information its $3D$ position cannot be estimated. If however a second image of the point is taken from a known position relative to the first, **p′s** position can be triangulated. For the general case with an arbitrary pose between the two images the fundamental matrix **F** can be computed which relates pixel locations in the image pair [40].

$$\mathbf{x_r^T F x_l} = \mathbf{0} \tag{3.9}$$

This matrix maps a point in the left image to a line in the right image, $\mathbf{Fx_l}$. These lines are referred to as epipolar lines. In the case of specialist stereo pairs such as the Point Gray BumbleBee2 used in this work the left and right camera are separated along the X axis of the left camera frame. This separation is of a known distance b and is referred to as the baseline. This alignment results in points in the

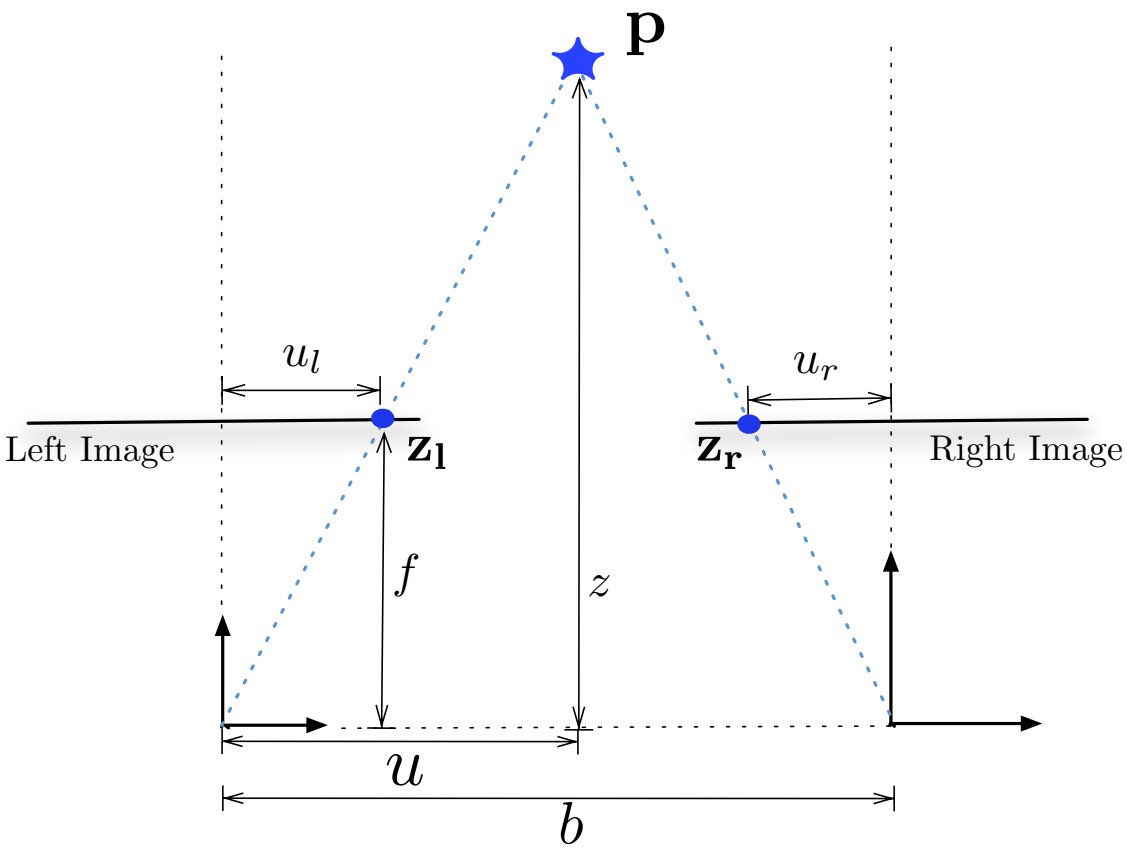


Figure 3.6: A cross section of the $Z - X$ plane of the stereo model shown in Fig 3.5. The point **p** is projected into both images which the same v but different u coordinates. This disparity can be used to triangulate **p** and estimate its $3D$ position.

left image mapping to lines parallel with the u axis in the right image, simplifying the search for stereo correspondences.

Fig 3.6 shows a cross section of a stereo pair along the $Z - X$ plane. The depth of the point **p** induces a disparity when it is projected into the left and right image. It is this disparity which allows us to compute the point's depth and recover its $3D$ position. Working in the frame of the left camera, using similar triangles for both the left and right images:

$$\frac{u_l}{f} = \frac{u}{z} \tag{3.10}$$

$$\frac{u_r}{f} = \frac{u+b}{z} \tag{3.11}$$

Substituting for u we get

$$z = \frac{bf}{u_l - u_r} = \frac{bf}{d}, \tag{3.12}$$

where d is the disparity. The depth of the point is then recovered, the x and y components can similarly be recovered by inverting the perspective projection:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{zu_l}{f} \\ \frac{zv_l}{f} \\ \frac{bf}{d} \end{bmatrix} \tag{3.13}$$

3.4 Data Association

The visual odometry system used in this work relies on the expeditious and accurate association of landmarks between stereo pairs and across temporal sequences. In this section, we will discuss how feature extraction and description are optimised to enable a real-time system.

3.4.1 Feature Extraction and Description

The main goal of feature extraction and description is to decompose an image into a set of small patches that are distinct and invariant under changes in view point. This reduces the problem of data association from a dense problem to a sparse problem. For applications such as matching stereo pairs and temporally adjacent image this approach has shown much success. However, we will argue in Chapter 4 that for applications such as matching between sequences a dense approach can be advantageous.

In some of the first implementations of Visual Odometry and SLAM [76, 16], affine regions in image sequences were detected using the Harris Corner detector [37]. There was no explicit description of the local image patch around the extracted points, but instead the Sum of Absolute Differences (SAD) or cross correlation was used for matching. The limited effectiveness of these methods under view point and

photometric change for comparing local image regions led to a surge of research in feature extraction and description. The move away from more raw similarity measures towards artificially contrived descriptors came to fruition with the Scale Invariant Feature Transform (SIFT) [62].

SIFT uses local image gradients at a selected scale to construct a feature vector describing an image patch. The use of gradients allows for successful association under more significant changes in illumination and spatial distortion. Mikolajczyk [68] demonstrated the performance improvement of the SIFT algorithm in comparison to previous methods.

The computational cost of SIFT lead to other algorithms such as the Speeded-Up Robust Features (SURF) [2]. Here various approximations are made to reduce computational cost and increase speed. One such speed up is achieved by using box filters to approximate the second order Gaussian derivatives when computing image gradients. Until relatively recently the use of these feature descriptors in real time systems has been prohibited by their associated cost. However, with hardware and implementation improvements several real time stereovision based VO and SLAM systems have now been presented [67, 26].

Other attempts to improve the robustness of descriptors include the more recent Illumination Robust Descriptor (DIRD) [54]. Here the descriptor is learnt from images of the same scene undergoing various lighting changes. A move away form the possibly myopic approach of using point features was presented by Hundelshausen [113]. Descriptor Nets (D-Nets) attempt to find associations between images using image strips rather than patches. Here a set of key points form a grid or net across the image. Pixel intensities across the length of the arcs are quantised into discrete tokens. These tokens are used for matching; this method has shown promise and robustness to extreme viewpoint changes. However, the current implementation can take several seconds to process a single frame.

The VO system used in this work uses Fast Corners (FAST) [91] in conjunction with Binary Robust Independent Elementary Features (BRIEF) [9]. The FAST

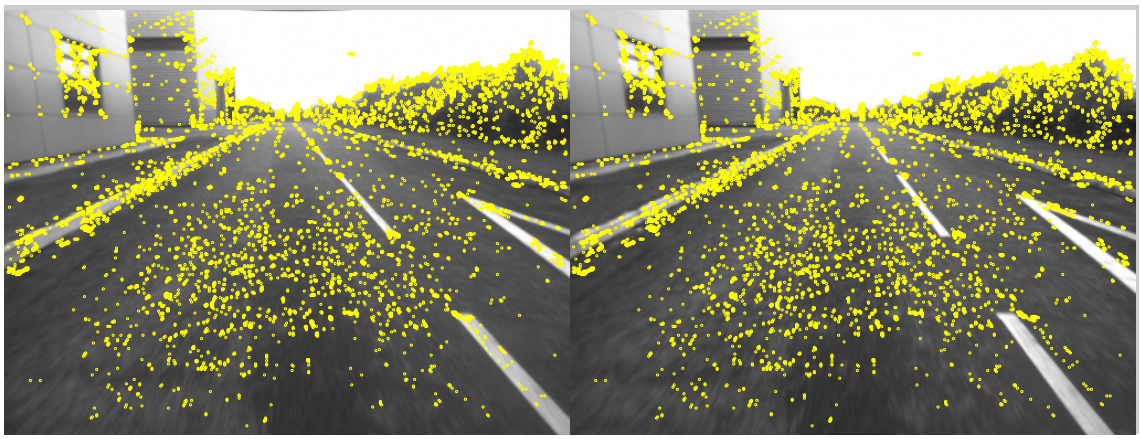


Figure 3.7: FAST Corners for a stereo pair consisting of a right and left image. The threshold determines how pronounced a candidate point has to be relative to its neighbours in order to be considered a corner. Here the threshold is low resulting in a large number of points.

detector computes the number of contiguous pixel around a fix radius from an image point which are lighter or darker than the said image point. If the number of pixels is greater than a threshold N, then that image point is said to belong to a corner. The BRIEF descriptor is computed using a fixed mask of binary potentials placed at detected FAST feature locations. In general sets of $16, 32$ or 64 binary potentials are used depending on the application and computational resources available. Each binary potential produces a 0 or 1 depending on which of the two pixels it is associated with has the largest intensity value. The overall descriptors are then vectors of binary numbers, which are very inexpensive to compare.

As FAST only provides discrete pixel locations for key points an extra step of sub pixel refinement is carried out using Efficient Second Order Minimisation (ESM) [66]. ESM is a dense image registration technique which we apply to small patches localised around FAST corners. The Sum of Square Distances (SSD) between all pixel values of the source and target patch is minimised to achieve sub-pixel registration. The ESM algorithm achieves second order convergence with the same computational cost as first order methods using an approximation of the Hessian.

3.4.2 Stereo Matching

Stereo matching is a prerequisite for triangulating the position of 3D points. As discussed in Section 3.3.2 the process of searching for stereo correspondences is simplified to a 1D search across rows of the image as both images are undistorted and rectified. Dense stereo matching [29] can be performed leading to a high fidelity 3D point cloud. However, for our purposes here we only require a sparse set of correspondences.

We use FAST corners to detect key points. Figure 3.7 shows FAST points overlaid on a stereo pair of a typical road scene. Here the threshold is low leading to may key points; in practice this number of points is not required so a subset can be selected or the threshold increased.

In order to find correspondences we take a key point in the left image and compute the Sum of Absolute Differences (SAD) for key points in the corresponding row in the right image. The highest score is taken as a match. If no key points exist in the row of the right image the key point in the left image is rejected. The use of SAD here rather than a descriptor based method is justified as the image pair is captured simultaneously under the same lighting conditions at a known relative pose.

3.4.3 Temporal Matching

In order to achieve a trajectory estimate from a sequence of stereo pairs observed landmarks must be matched between frames; this process is referred to as temporal matching. Temporal matching is more involved than stereo matching as the camera may have undergone general motion. A simple 1D line search is no longer sufficient as the pixel location of the landmark in the new frame may undergo arbitrary translation.

To avoid having to search across the entire image to find a correspondence various approaches have attempted to bound the search region using various assumptions and information gleaned from other sources. The most basic assumption is that between frames the change in position is small. Therefore, a predetermined region around the

pixel location of a landmark in a subsequent frame is searched for a match. This approach is sensitive to the size of the region as if in fact the inter-frame motion is large the system will be unable to find successful matches.

Another approach used by Tradiff [107] among others is to adopt a motion model. This putative estimate of the inter frame motion allows the 3D position of landmarks from the current frame to be projected into the subsequent images. This prediction defines a starting point for a search and associated region. Other systems use supplementary sensors such as gyroscopes and accelerometers as opposed to a motion model [1, 80]. With all these approaches, the trade-off between robustness and speed determines the size of the search region.

In the implementation used in this work, we take advantage of the low computational cost of the BREIF descriptor to search the entire image for correspondences. This ability to search the entire image without sacrificing real time performance removes the trade off between search region size and robustness. This exhaustive approach means system robustness is unaffected by the omission of a motion model or supplementary sensors.

3.5 Motion Estimation

3.5.1 Outlier Rejection

Even after taking care to find good correspondences between frames, mistakes are still made, leading to outliers. These can be a result of simple incorrect correspondences or dynamic objects such as cars and pedestrians, which move relative to the static world between frames. If these outliers were to be included in our numerical optimisation they could significantly perturb the final solution. It is therefore important to remove as many of them as possible.

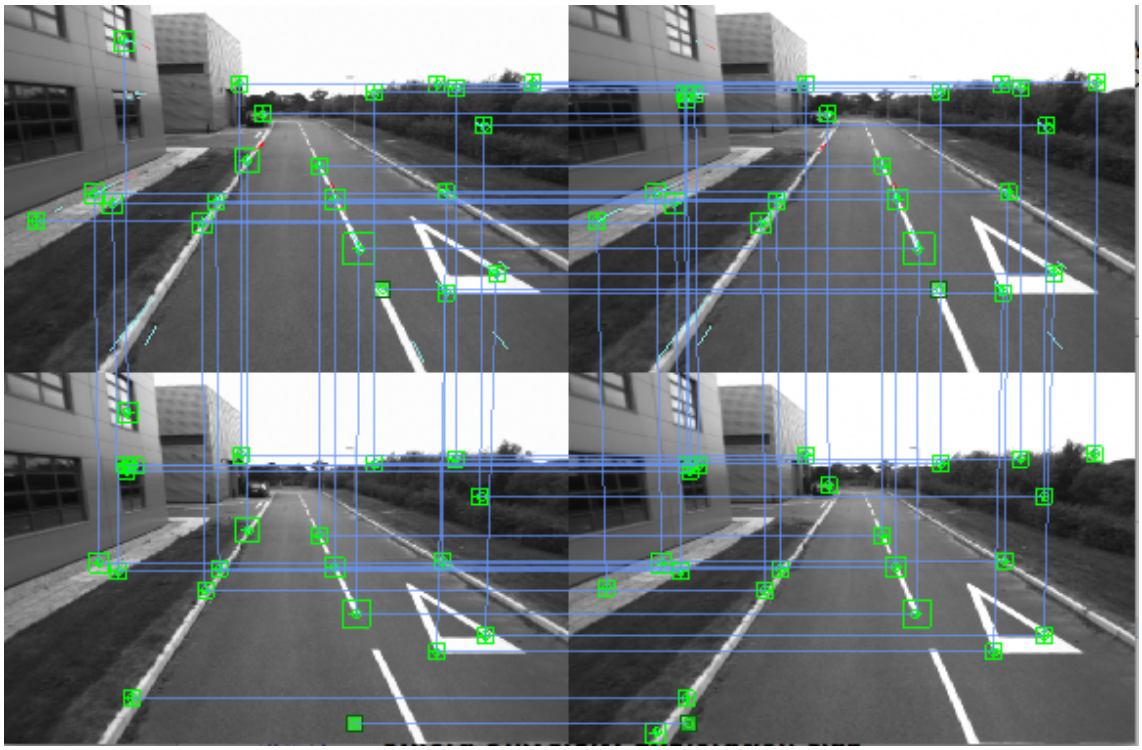


Figure 3.8: This figure shows successful stereo and temporal matches between a set of stereo pairs. Here horizontal and (almost) vertical lines represent successful stereo and temporal matches respectively.

Sample and Consensus Methods

Figure 3.9 shows a toy example of data with an underlying linear relationship heavily polluted with outliers. If we wish to fit a straight line to these data and naively include the outliers in a simple linear regression, the effect is clear. The final result is in error as the outliers have a disproportionate effect on the linear regression and pull the model away from the underlying trend.

RANdom Sample and Consensus (RANSAC) [22] was one of the first algorithms developed to effectively remove outliers before a numerical optimisation step. RANSAC recursively fits a simple model to a subset of the data and tests agreement with the remaining data. The models are scored using

$$C = \sum_i p(e_i) \tag{3.14}$$

$$p(e) = \begin{cases} 0 & e^2 < T^2 \\ T^2 & e^2 \leqslant T^2 \end{cases} \tag{3.15}$$

where e_i is the disagreement of the ith data point with the current model and T is a threshold, which in this case is determined by the position accuracy of correctly matched features in the input images. All outliers are given a score of T^2 or some constant and all inliers a score of 0. The model with the lowest score is selected and the outliers for that model rejected. A problem with the RANSAC method is choosing an appropriate threshold. Figure 3.9 shows how selecting a threshold, which is too large, can cause RANSAC to fail to identify the correct solution. The number of outliers included in the inliers set as T is increased causes this. No account of how well the inliers fit the model is taken in standard RANSAC. For no extra cost, as all errors are already computed, an M-Estimator [42] can be used to take account of the how well a model fits inliers.

$$p(e) = \begin{cases} e^2 & e^2 < T^2 \\ T^2 & e^2 \geqslant T^2 \end{cases} \tag{3.16}$$

Here inliers are given a weight proportional to their agreement with the model. This approach, coined MLESAC [110], has been shown to produce better estimates and is much more robust to threshold choice.

Other modifications to standard RANSAC include Pre-emptive RANSAC [78]. It was demonstrated in [77] that the general stopping criterion for random hypothesis based methods can be optimistic and the process of scoring every point against every model expensive. Pre-emptive RANSAC pre-computes a set of hypothesis models and iteratively scores them against randomly permuted data. The highest scoring models are discarded as the scoring progresses. Another modification by Jin [44] guided the hypotheses selection, reducing the number of hypotheses needed for convergence. Although these modified RANSAC methods potentially reduce the computational effort required for the outlier rejection step, it was felt that the added

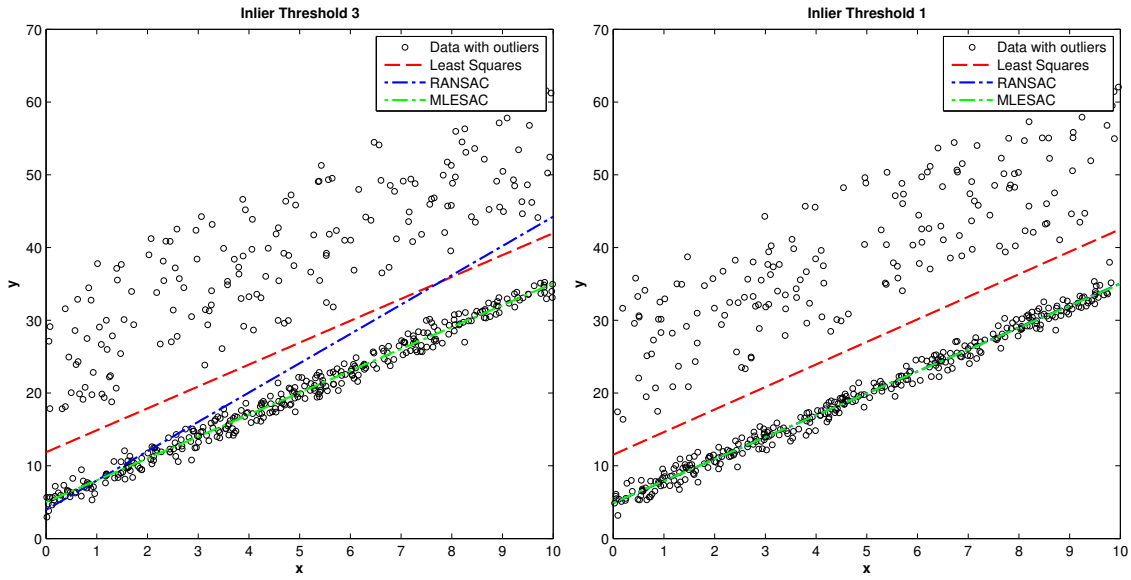


Figure 3.9: Data with a linear relationship polluted with outliers and Gaussian noise. Left shows the result of Least Squares, RANSAC and MLESAC with an inappropriately large threshold, right shows the same with a smaller threshold (please note that in the right plot the fit for RANSAC and MLESAC overlap). It can be seen the MLESAC is more robust to 'bad' threshold choices.

complexity did not justify the performance improvements for this work.

Model Selection

We require our sample and consensus method to generate a putative motion estimate based a set of stereo and temporal correspondences. It is advantageous that the model used is simple and cheap to compute as it may be invoked many times.

There are two main approaches to estimating motion between two frames using point correspondences. Early approaches such as Horn [41] estimated the transform between two frames by minimising the distance between three or more $3D$ point correspondences. These solutions can be particularly sensitive to errors in depth estimation from stereo correspondences.

More recently rather than $3D$ to $3D$ matching a $3D$ to $2D$ approach has been adopted. Here $3D$ points from the previous frame are projected into the current frame and the pixel re-projection error in image space is minimised. This is referred to as n point perspective pose estimation, PnP. The minimum value of n for this problem is three and there are many methods of quickly solving the $P3P$ problem

[36]. Solutions exist for larger values of n [8, 56, 71, 35] but in the case of random sample and consensus methods choosing the simplest method is advantageous.

In the system described here we use MLESAC with a P3P model to produce a putative estimate of motion between frames. This estimate is then refined using a numerical optimisation.

3.5.2 Numerical Optimisation

The outlier rejection step in Section 4.8.4 provides us with a set of inlier measurements $\mathbf{z}_i, i \in 1, 2, \ldots, M$ and a putative estimate of the inter-frame motion. This initial estimate however is only computed using three of the inliers with a P3P method. In order to hone this estimate we include all inliers in a numerical optimisation. We must first define a state $\mathbf{x}$, the details of which are discussed in Section 3.5.3, which represents the underlying system. We can then write the likelihood function as

$$\mathcal{L} = \prod_{i=0}^{M} p(\mathbf{z_i}|\mathbf{x}), \tag{3.17}$$

where $p(\mathbf{z}_i|\mathbf{x})$ is the probability of obtaining the measurement $\mathbf{z}_i$ given the state $\mathbf{x}$. The product of these probabilities over all $\mathbf{z}_i$ is then the likelihood of obtaining all measurements given the system state $\mathbf{x}$. If we were to maximise $\mathcal{L}$ with respect to $\mathbf{x}$ we obtain the Maximum Likelihood (ML) estimate $\hat{\mathbf{x}}_{\mathbf{ml}}$

$$\hat{\mathbf{x}}_{\mathbf{ml}} = \underset{\mathbf{x}}{argmax} \prod_{i=0}^{M} p(\mathbf{z_i}|\mathbf{x}). \tag{3.18}$$

If then we assume that after outlier rejection the measurements $\mathbf{z}_i$ are corrupted with Gaussian noise

$$\mathbf{z_i} = h_i(\mathbf{x}) + \omega_\mathbf{i}, \;\; \omega_\mathbf{i} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_i}), \tag{3.19}$$

where $h_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is an observation function which maps the state vector of length n to an uncorrupted measurement of length m, the observation function, which is also referred to as a predictor function, is equivalent to a perfect measurement

of a given state. ω_i is Gaussian noise with zero mean and a covariance matrix $\mathbf{R}_i$. The covariance matrix $\mathbf{R}_i$ is $m \times m$. It is possible for all measurements to have their own individual covariance matrix but in practice the same values are used for all measurements. The probability of a measurement $\mathbf{z}_i$ given the state $\mathbf{x}$ can now be written as

$$p(\mathbf{z_i}|\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m \, |\mathbf{R_i}|}} exp(\frac{1}{2}(\mathbf{z_i} - h_i(\mathbf{x}))^T \mathbf{R_i}^{-1}(\mathbf{z_i} - h_i(\mathbf{x}))) \tag{3.20}$$

The full likelihood function is now a product of exponentials. In order to simplify this we take the negative log of $\mathcal{L}$. This reduces the product to a sum of log terms. As the normalisation terms pre-multiplying the Gaussians are constant with respect to $\mathbf{x}$ they can be dropped and the Maximum Likelihood estimate of $\hat{\mathbf{x}}_{\mathbf{ML}}$ becomes a weighted least squares problem. Which is minimised when the difference between the measurements $\mathbf{z_i}$ and the measurement model $h_i(\mathbf{x})$ is minimised for all i:

$$\begin{aligned} \hat{\mathbf{x}}_{\mathbf{ml}} &= \underset{\mathbf{x}}{argmin}(-log(\mathcal{L})) \\ &= \underset{\mathbf{x}}{argmin} \sum_{i=0}^{M} (\tfrac{1}{2}(\mathbf{z_i} - h_i(\mathbf{x}))^T \mathbf{R_i}^{-1}(\mathbf{z_i} - h_i(\mathbf{x})). \end{aligned} \tag{3.21}$$

We can rewrite this as

$$\hat{\mathbf{x}}_{\mathbf{ml}} = \underset{\mathbf{x}}{argmin}\ F(\mathbf{x}) = \underset{\mathbf{x}}{argmin}\ \frac{1}{2}(\mathbf{z} - h(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{z} - h(\mathbf{x})), \tag{3.22}$$

where $\mathbf{z} = [\mathbf{z}_1^T, \mathbf{z}_2^T, \mathbf{z}_3^T, \ldots, \mathbf{z}_M^T]^T$, $h(\mathbf{x})$ is an observation function which generates predictions for all measurements in $\mathbf{z}$ and $R = diag\{R_1, R_2, R_3, \ldots, R_M\}$ a block diagonal matrix. Our objective function $F(\mathbf{x})$ now maps all measurement errors for a given $\mathbf{x}$ to a single scalar value.

In order to minimise our objective function and obtain our maximum likelihood estimate of $\mathbf{x}$ we use a Gauss-Newton (GN) gradient based method. The GN method is a simple method of minimising a non-linear objective function. If the objective function is differentiable and linear we could solve for the minima directly by finding

$\partial F(\mathbf{x})/\partial \mathbf{x} = 0$. However, in this case we cannot and must find a solution iteratively. We first take a first order Taylor series of the observation function, the aim is then to find a sequences of $\delta\mathbf{x}'s$ such that $F(\mathbf{x_o} + \delta\mathbf{x}) < F(\mathbf{x_o})$.

$$h(\mathbf{x_o} + \delta\mathbf{x}) = h(\mathbf{x_o}) + \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\delta\mathbf{x} = h(\mathbf{x_o}) + \mathbf{J}(\mathbf{x_o})\delta\mathbf{x} \tag{3.23}$$

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial h_1(\mathbf{x})}{\partial \mathbf{x_1}} & \cdots & \frac{\partial h_1(\mathbf{x})}{\partial \mathbf{x_n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m(\mathbf{x})}{\partial \mathbf{x_1}} & \cdots & \frac{\partial h_m(\mathbf{x})}{\partial \mathbf{x_n}} \end{bmatrix} \tag{3.24}$$

where $\mathbf{J}(\mathbf{x})$ is the Jacobian an $m \times n$ matrix, the derivation of which is discussed in Section 3.5.3. Substituting Eq 3.23 into $F(\mathbf{x})$ we get

$$F(\mathbf{x_o} + \delta\mathbf{x}) = \frac{1}{2}(\mathbf{z} - (h(\mathbf{x_o}) + \mathbf{J}(x_o)\delta\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{z} - (h(\mathbf{x_o}) + \mathbf{J}(x_o)\delta\mathbf{x}))^T. \tag{3.25}$$

Letting

$$\mathbf{e} = \mathbf{z} - h(\mathbf{x}), \tag{3.26}$$

we can now write the objective function in terms of $\mathbf{e}$, the disagreement between the predicted measurements and actual measurements;

$$F(\mathbf{x_o} + \delta\mathbf{x}) = \frac{1}{2}(\mathbf{e}^T\mathbf{R}^{-1}\mathbf{e} - 2\delta\mathbf{x}^T\mathbf{J}^T(\mathbf{x_o})\mathbf{R}^{-1}\mathbf{e} + \delta\mathbf{x}^T\mathbf{J}^T(\mathbf{x_o})\mathbf{R}^{-1}\mathbf{J}(\mathbf{x_o})\delta\mathbf{x}) \tag{3.27}$$

Taking the derivative with respect to $\delta\mathbf{x}$ and setting it to zero yields $\delta\mathbf{x}$ towards the minima.

$$\frac{\partial F(\mathbf{x_o} + \delta\mathbf{x})}{\partial \delta\mathbf{x}} = -\mathbf{J}^T(\mathbf{x_o})\mathbf{R}^{-1}\mathbf{e} + \mathbf{J}(\mathbf{x_o})^T\mathbf{R}^{-1}\mathbf{J}(\mathbf{x_o})\delta\mathbf{x} = 0 \tag{3.28}$$

$$\delta\mathbf{x} = (\mathbf{J}(\mathbf{x_o})\mathbf{R}^{-1}\mathbf{e} + \mathbf{J}^T\mathbf{R}^{-1}\mathbf{J}(\mathbf{x_o}))^{-1}\mathbf{J}(\mathbf{x_o})\mathbf{R}^{-1}\mathbf{e} \tag{3.29}$$

Iterative application of these steps $\mathbf{x}_{n+1} = \mathbf{x}_n + \delta\mathbf{x}$ refines the state such that the objective function is minimised. Iterations are carried out until a stopping criterion is reached. Stopping criterion can take many forms but commonly a minimum $\delta\mathbf{x}$ or $\delta F(\mathbf{x})$ will be predetermined.

3.5.3 The State Vector

The choice of state determines how complex, accurate and to what scale you can solve for. In traditional systems such as Graph-SLAM [108], which operate in a global frame, every pose and landmark position is optimised. The state vector in this case grows very quickly and small changes in one state variable must propagate across the entire state vector.

An alternative to the global approach is to use a relative framework. Here poses and landmarks are not contained in a common global frame but stored relative to one another. This effectively embeds the map in a manifold, which is locally Euclidean. The consequence of this is that over large scales only topological relations are conserved.

The full solution to this relative framework is Relative Bundle Adjustment (RBA), which is the optimal least-squares solution to the full Simultaneous Localisation & Mapping problem [111]. A complete RBA engine was presented by Sibley et al [99], which was capable of estimating camera trajectories over 100 kilometre scales.

Here we will present the state vector and the Jacobian for the full relative bundle adjustment problem and show how it simplifies for basic VO in Section 3.5.7. Figure 3.10 shows a simple trajectory segment. The state vector

$$\mathbf{x} = [\mathbf{l}, \mathbf{t}]^T \tag{3.30}$$

contains the position of the landmarks $\mathbf{l} = [\mathbf{l}_{0,1}, \ldots, \mathbf{l}_{i-2,n-1}, \mathbf{l}_{i,n}]$ and relative trans-

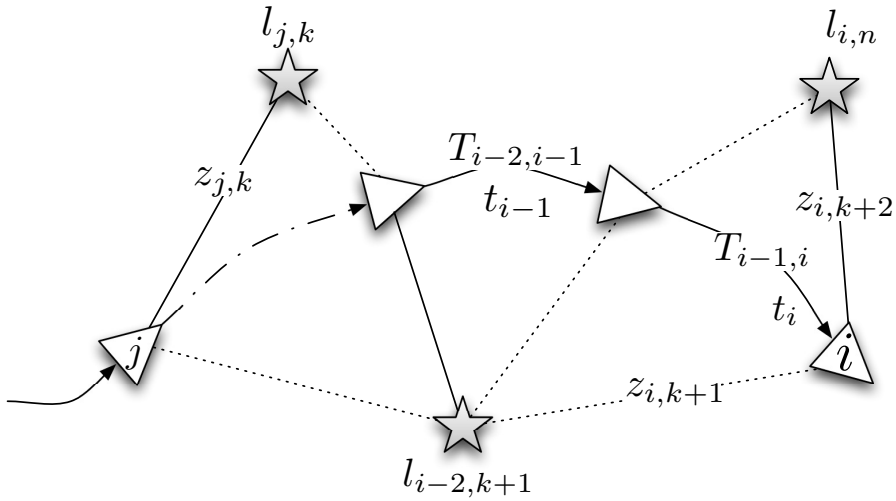


Figure 3.10: Notation for a simple trajectory: Vehicle poses are represented as triangles, landmarks as gray stars. Landmarks base frames are represented by solid lines and later observations as dotted lines. Each interpose transform is represented by an arrow.

forms between frames $\mathbf{t} = [\mathbf{t}_0, \ldots, \mathbf{t}_{i-2}, \mathbf{t}_{i-1}, \mathbf{t}_i]$. The landmark $\mathbf{l}_{j,k}$ is the $3D$ position of the kth point relative to the jth pose from which it was first observed. The relative poses $\mathbf{t_i} = [x, y, z, r, p, q]$ are a $6DoF$ Cartesian transform with a translation x, y, z and a rotation about each of the axes $r-roll$,$p-pitch$ and $q-yaw$. A measurement's $z_{i,k}$, is a measurement of the kth landmark from the ith pose.

The corresponding observation function for a measurement $\mathbf{z}_{i,k}$ is then:

$$h_{i,k}(\mathbf{x}) = \mathcal{K}(\mathcal{M}(\mathcal{T}^k_{i,j}(\mathbf{x}))). \tag{3.31}$$

The function is split into three main components. $\mathcal{T}^k_{i,j}(\mathbf{x})$ selects and transforms a landmark, $\mathbf{l}_{j,k}$ into the current robot frame i. $\mathcal{M}(\ldots)$ transforms the point from the robot frame to the computer vision frame of either the left or right camera and $\mathcal{K}(\ldots)$ is the perspective projection of the point into either the right or left image.

The Jacobian of the observation function can then be constructed from the individual components using the chain rule.

$$\frac{\partial h_{i,k}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \mathcal{K}}{\partial \mathcal{M}} \frac{\partial \mathcal{M}}{\partial \mathcal{T}^k_{i,j}} \frac{\partial \mathcal{T}^k_{i,j}}{\partial \mathbf{x}} \tag{3.32}$$

3.5.4 Transform Derivative

The transform derivative must be taken with respect to the relative transforms $\mathbf{t}$ as well as the positions of the landmarks $\mathbf{l}$. If from Fig 3.10 we consider $z_{i,k+1}$ the observation of $\mathbf{l}_{i-2,k+1}$ from the ith pose we must first transform the landmark into frame i. In order to do this we compose the two intermediate transforms $\mathbf{t}_{i-1}$ and $\mathbf{t}_i$. This is achieved by converting the $6D.o.F$ relative poses into 4×4 homogeneous transformation matrices.

$$\mathbf{t}_{i,i-1} = \mathbf{t}_i \oplus \mathbf{t}_{i-1} \tag{3.33}$$

This is achieved by converting the $6D.o.F$ relative poses into 4×4 homogeneous transformation matrices.

$$\mathbf{T}(\mathbf{t}_i) = \mathbf{T}_{i,i-1} = \begin{bmatrix} \mathbf{R}(\mathbf{t}_i) & \beta(\mathbf{t}_i) \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.34}$$

where $\beta(\mathbf{t}_i) = [x, y, z]^T$ is a function which extracts the linear translation components from the relative transform $\mathbf{t}_i$. $\mathbf{R}(\mathbf{t}_i)$ is a rotation matrix with $|\mathbf{R}(\mathbf{t}_i)| = 1$ constructed by successive rotations about each of the three axes.

$$\mathbf{R}(\mathbf{t}) = \mathbf{R}_\mathbf{z}(\mathbf{t})\mathbf{R}_\mathbf{y}(\mathbf{t})\mathbf{R}_\mathbf{x}(\mathbf{t}) \tag{3.35}$$

$$\mathbf{R}_z(\mathbf{t}_i) = \begin{bmatrix} c(q) & -s(q) & 0 \\ s(q) & c(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_\mathbf{y}(\mathbf{t}_i) = \begin{bmatrix} c(p) & 0 & s(p) \\ 0 & 1 & 0 \\ -s(p) & 0 & c(p) \end{bmatrix},$$
$$\mathbf{R}_x(\mathbf{t}_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(r) & -s(r) \\ 0 & s(r) & c(r) \end{bmatrix} \tag{3.36}$$

where $c(q) = cos(q)$ and $s(q) = sin(q)$. Relative transforms can then be composed by simple multiplication

$$\mathbf{T}_{i,i-2} = \mathbf{T}(t_i)\mathbf{T}(t_{i-1}). \tag{3.37}$$

For the more general case

$$\mathcal{T}^k_{i,j}(\mathbf{x}) = \mathbf{T}(\mathbf{t}_i)\mathbf{T}(\mathbf{t}_{i-1})\dots\mathbf{T}(\mathbf{t}_j)\mathbf{l}_{j,k} = \mathbf{T}_{i,j}\mathbf{l}_{j,k}. \tag{3.38}$$

So the transform component of the observation function for the kth landmark in frame j from frame i is a function of both the landmark position and all intermediate transforms between j and i. The derivative of $\mathcal{T}^k_{i,j}(\mathbf{x})$ with respect to one of the relative transforms $\mathbf{t}_j$ is computed by considering a delta transform $\mathbf{T}(\Delta\mathbf{t})$ where $\Delta\mathbf{t} = [\delta x, \delta y, \delta z, \delta r, \delta p, \delta q]^T$. This allows us to consider the transform $\mathbf{T}(\mathbf{t}_i)\mathbf{T}(\Delta\mathbf{t})$ holding $\mathbf{t}_i$ constant and evaluating $\frac{\partial \mathbf{T}(\mathbf{t}_j)\mathbf{T}(\Delta\mathbf{t})}{\partial \Delta\mathbf{t}}\Big|_{\Delta\mathbf{t}=\mathbf{0}}$. This formulation greatly simplifies the computation of the Jacobian as there is no need to compute sine and cosine elements of the transform matrix as they are always evaluated about zero.

$$\frac{\partial \mathcal{T}^k_{i,j}}{\partial \mathbf{x}} = \frac{\partial \mathcal{T}^k_{i,j}}{\partial \Delta\mathbf{t}} = \mathbf{T}_{i,i-1}\mathbf{T}_{i-1,i-2}\dots\left(\mathbf{T}_{j+1,j}\frac{\partial \mathbf{T}(\boldsymbol{\Delta}\mathbf{t})}{\partial \Delta\mathbf{t}}\right) \tag{3.39}$$

Taking the derivative of the $\mathbf{T}(\Delta\mathbf{t})$ and evaluating at zeros yields a 4×4 matrix for each component.

$$\frac{\partial \mathbf{T}(\mathbf{x})}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \frac{\partial \mathbf{T}(\mathbf{x})}{\partial y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\frac{\partial \mathbf{T}(\mathbf{x})}{\partial z} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \frac{\partial \mathbf{T}(\mathbf{x})}{\partial r} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.40}$$
$$\frac{\partial \mathbf{T}(\mathbf{x})}{\partial p} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \frac{\partial \mathbf{T}(\mathbf{x})}{\partial q} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

These six components combine to form a $4 \times 4 \times 6$ tensor.

$$\frac{\partial \mathbf{T}}{\partial \mathbf{t}} = [\frac{\partial \mathbf{T}}{\partial x}, \frac{\partial \mathbf{T}}{\partial y}, \frac{\partial \mathbf{T}}{\partial z}, \frac{\partial \mathbf{T}}{\partial r}, \frac{\partial \mathbf{T}}{\partial p}, \frac{\partial \mathbf{T}}{\partial y},] \tag{3.41}$$

Right multiplying by the landmark position $\mathbf{l}_{j,k}$ in homogenous form yields a 4×6 matrix.

$$\begin{aligned} \frac{\partial \mathbf{T}}{\partial \mathbf{t}} \mathbf{l}_{j,k} &= \begin{bmatrix} 1 & 0 & 0 & 0 & z_l & -y_l \\ 0 & 1 & 0 & -z_l & 0 & x_l \\ 0 & 0 & 1 & y_l & -x_l & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & [\mathbf{l}_{j,k}]^{\times} \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{3.42}$$

In the full RBA implementation the position of the landmarks are refined as well as the robot poses [100]. The derivative of $\mathcal{T}_{i,j}^{k}(\mathbf{x})$ relative to a landmark $\mathbf{l}_{j,k}$ is simply

$$\frac{\partial \mathcal{T}_{i,j}^{k}}{\partial \mathbf{l}} = \frac{\partial}{\partial \mathbf{l}} (\mathbf{T}_{i,j} \mathbf{l}_{j,k}) \tag{3.43}$$

$$\frac{\partial \mathcal{T}_{i,j}^{k}}{\partial \mathbf{l}} = \begin{bmatrix} R_{i,j} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{3.44}$$

yielding a 4×4 matrix. Both of these components are incorporated into the final Jacobian as will be shown in Section 3.5.7.

3.5.5 Sensor to Robot Transform Derivative

It is generally the case that the frame for which a pose is being estimated is not the frame of the sensor itself. To account for this we incorporate $\mathbf{T}_{s,r}$ into our observation function. This transforms a point $\mathbf{p}_r$ from the robot frame into the frame of the sensor. It should be noted that the transforms $\mathbf{T}_{s,r}$ are different from both the left and right sensors in the stereo pair as the two are separated by a baseline b. This does not, however, affect how we compute the Jacobian. $\mathbf{T}_{s,r}$ is often referred to as an extrinsic calibration and in generally only has to be computed once.

The second transform, $\mathbf{T}_{v,s}$, switches coordinate frames from the robotics convention of X-forward, Y-right and Z-down to the computer vision convention of Z-forward, X-right, and Y-down. This is implemented as a simple rotation.

$$\mathcal{M}(\mathcal{T}_{i,j}^{k}(\mathbf{x})) = \mathbf{T}_{v,s}\mathbf{T}_{s,r}\mathbf{p}_r \tag{3.45}$$

$$T_{v,s} = \begin{bmatrix} \mathbf{R}_{v,s} & 0 \\ \mathbf{0} & 1 \end{bmatrix}, T_{s,r} = \begin{bmatrix} \mathbf{R}_{s,r} & \mathbf{t}_{s,r} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.46}$$

where $\mathbf{R}_{s,r}$ is a rotation matrix, $\mathbf{t}_{s,r}$ is a translation and

$$\mathbf{R}_{v,s} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.47}$$

The derivative of this transform is then simply:

$$\frac{\partial \mathcal{M}}{\partial \mathcal{T}_{i,j}^{k}} = \frac{\partial \mathcal{M}}{\partial \mathcal{T}} = \begin{bmatrix} \mathbf{R}_{v,s}\mathbf{R}_{s,r} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{3.48}$$

Unlike other components of the Jacobian, the derivative of this transform is not dependant on the points $\mathbf{p}_r$ or the pose of the robot. It can simply be computed once from the rotation of coordinate conventions and the transform from the robot frame to the sensor frame.

3.5.6 Perspective Projection Derivative

The final component of the Jacobian is the perspective projection of a point in the computer vision frame of the camera, $\mathbf{p}_v$ onto the actual imaging plane. This is a non-linear $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ mapping and thus it is easer to consider it in two parts.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathcal{K}(\mathbf{p}_v) \tag{3.49}$$

becomes

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathcal{K}_2(\mathcal{K}_1(\mathbf{p}_v)) \tag{3.50}$$

$\mathcal{K}_1(\mathbf{p}_v)$ is simply the multiplication of the point by the camera intrinsics matrix $\mathbf{K}$. In order to isolate the components of the output of $\mathcal{K}_1$ the rows of $\mathbf{K}$ are written as $\mathbf{K}_n$ where n is the row number.

$$\mathcal{K}_1(\mathbf{p}_v) = \mathbf{K}\mathbf{p}_v = \begin{bmatrix} \mathbf{K}_1\mathbf{p}_v \\ \mathbf{K}_2\mathbf{p}_v \\ \mathbf{K}_3\mathbf{p}_v \end{bmatrix} \tag{3.51}$$

This function yields a 2D pixel location in homogeneous form. In order to recover the non-homogeneous form $[u, v]^T$the first two elements must be normalised by the third. This is performed by the function $\mathcal{K}_2$:

$$\mathcal{K}_2(\begin{bmatrix} \mathbf{K}_1\mathbf{p}_v \\ \mathbf{K}_2\mathbf{p}_v \\ \mathbf{K}_3\mathbf{p}_v \end{bmatrix}) = \begin{bmatrix} \mathbf{K}_1\mathbf{p}_v/\mathbf{K}_3\mathbf{p}_v \\ \mathbf{K}_2\mathbf{p}_v/\mathbf{K}_3\mathbf{p}_v \end{bmatrix} \tag{3.52}$$

The derivative of $\mathcal{K}_1$ is then simply the matrix of camera intrinsics

$$\frac{\partial \mathcal{K}_1(\mathbf{p}_v)}{\partial \mathbf{p}_v} = \mathbf{K} \tag{3.53}$$

and the derivative of $\mathcal{K}_2$ is a 2×3 matrix

$$\frac{\partial \mathcal{K}_2(\mathcal{K}_1(\mathbf{p}_v))}{\partial \mathcal{K}_1(\mathbf{p}_v)} = \begin{bmatrix} 1/\mathbf{K}_3\mathbf{p}_v & 0 & -\mathbf{K}_1\mathbf{p}_v/(\mathbf{K}_3\mathbf{p}_v)^2 \\ 0 & 1/\mathbf{K}_3\mathbf{p}_v & -\mathbf{K}_2\mathbf{p}_v/(\mathbf{K}_3\mathbf{p}_v)^2 \end{bmatrix}. \tag{3.54}$$

The complete derivative for the perspective projection is then the product of the the derivatives of $\mathcal{K}_1$ and $\mathcal{K}_2$

$$\frac{\partial \mathcal{K}(\mathbf{p}_v)}{\partial \mathbf{p}_v} = \begin{bmatrix} 1/\mathbf{K}_3\mathbf{p}_v & 0 & -\mathbf{K}_1\mathbf{p}_v/(\mathbf{K}_3\mathbf{p}_v)^2 \\ 0 & 1/\mathbf{K}_3\mathbf{p}_v & -\mathbf{K}_2\mathbf{p}_v/(\mathbf{K}_3\mathbf{p}_v)^2 \end{bmatrix} \mathbf{K}. \tag{3.55}$$

Substituting in the standard camera intrinsics matrix with a square pixel array

$$\mathbf{K} = \begin{bmatrix} f_u & 0 & u_o \\ 0 & f_v & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{3.56}$$

gives

$$\frac{\partial \mathcal{K}(\mathbf{p}_v)}{\partial \mathbf{p}_v} = \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \end{bmatrix} \tag{3.57}$$

It should be noted that this derivative is dependant on the input so has to be computed for every input point.

3.5.7 The Full Jacobian

The full Jacobian with pose and landmark refinement is expressed as

$$\begin{aligned} H(x) &= \frac{\partial h}{\partial x} \\ &= [\frac{\partial h}{\partial l}, \frac{\partial h}{\partial x_t}] \\ &= [H_l, H_t] \end{aligned} \tag{3.58}$$

where H_l is the Jacobian with respect to landmarks $\mathbf{l}$ and H_t with respect to the relative poses between frames $\mathbf{t}$. In the relative framework described here the Jacobian in Fig 3.11 describes the graphical model shown in Fig 3.12. If every landmark was observed from every pose the matrix would be dense however in reality this is not the case and the Jacobian becomes sparse. Taking $\partial h_{22}/\partial x$ as an example, this is the observation of landmark $\mathbf{l}_{0,2}$ from $\mathbf{p}_2$. As $\mathbf{l}_{0,2}$ is stored relative to $\mathbf{T}_0$ the observation is dependent on the relative poses $\mathbf{t}_1$ and $\mathbf{t}_2$ as well as the actual position of the landmark $\mathbf{l}_{0,2}$. This leads to three elements in that row of the Jacobian being populated $\partial h_{22}/\partial t_1$, $\partial h_{22}/\partial t_2$ and $\partial h_{22}/\partial l_{02}$.

In small work spaces where the size of the state vector is small then all landmarks and poses can be included in the optimisation such as in PTAM [49]. However, the computational cost is proportional to the number of poses and landmarks included in the optimisation. An approach taken by Konolige [46] was to reduce the size of the state by hanging sub maps from a skeleton trajectory. This allows for locally accurate sub maps as well as global consistency while limiting the size of the state vector. In outdoor environments when moving at urban traffic speeds the visual flow is much faster and parts of the world are only observed in passing. This results in landmarks only being observed in a small number of frames. Sibley [100] adopted a sliding window and showed that the effect of new observation falls off as $1/i$ as you move further back along the trajectory. This approach achieved constant time operation in normal operation. For loop closure events the window could be extended to include previous nodes in the graph.

For the simplest case of frame-to-frame Visual Odometry the state vector can

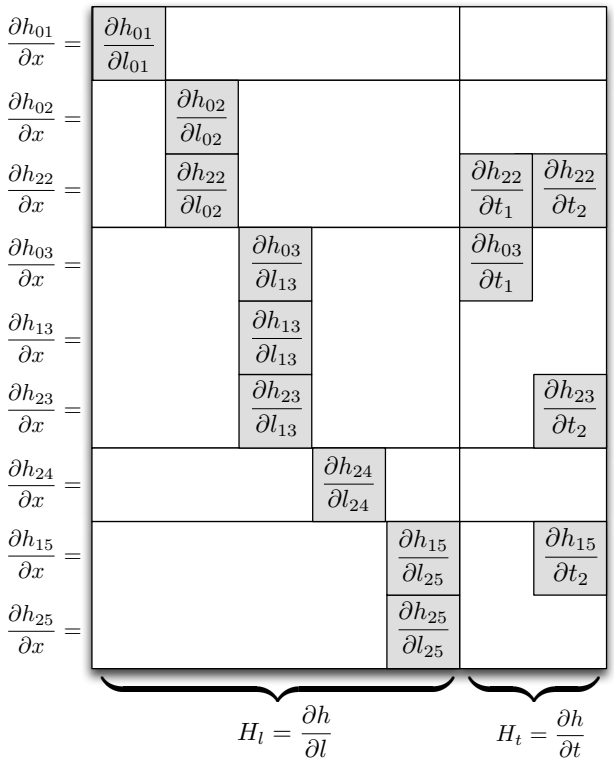


Figure 3.11: Example Relative bundle adjustment Jacobian structure for the sequence of observations in Figure 3.12

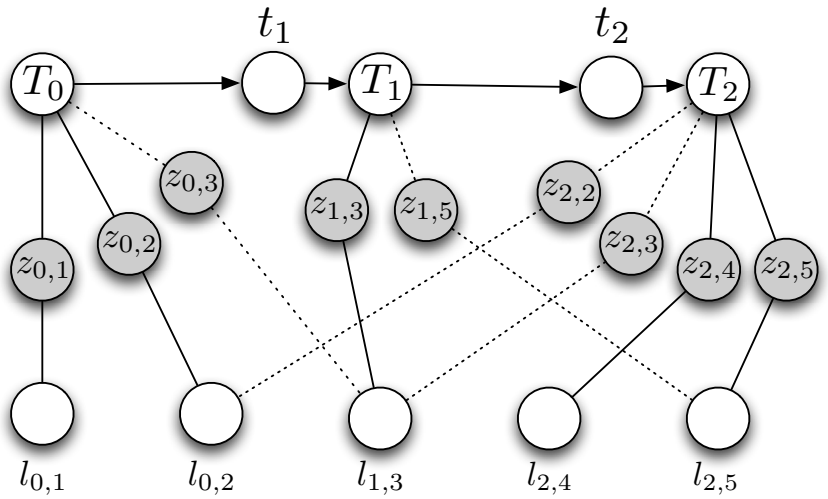


Figure 3.12: Graphical example for a sequence of several observations $z_{01}, ..., z_{25}$. There are three poses T_0, T_1, T_2, two relative transform estimates t_1, t_2 and five landmarks $l_{01}, ..., l_{2,5}$. The bold lines from poses to landmarks indicate base frames.

be reduced to the latest relative pose $\mathbf{t}_i^T$. Here we simply solve for the relative pose between the latest two frames without any bundle adjustment or landmark refinement. The state vector and Jacobian then reduce to

$$\mathbf{x} = [\mathbf{t}_i], \quad H(\mathbf{x}) = \frac{\partial h}{\partial \mathbf{t}}. \tag{3.59}$$

The implementation used allows us to control the complexity of the state vector from simple frame-to-frame VO to full bundle adjustment with landmark refinement by simply setting parameters in a configuration file.

3.6 Evaluation

3.6.1 Camera Placement

It is often the case in mobile robot design that the physical shape of the platform heavily restricts sensor placement. This often leads to cameras being mounted simply where they will fit. Road vehicles however, are much larger platforms allowing more freedom to place cameras around the vehicle. Very little work has been published on camera placement for road vehicle VO, however in [48] they attempted to find the optimal pitch of a stereo camera mounted on an UAV. Also, in [84] they investigated camera orientation for ground based exploration robots such as planetary rovers. Here in order to determine the optimal placement of a stereo camera used for road based VO we adopt an experimental approach.

We tested nine camera-mounting positions at various positions in and around the vehicle. A complete list and example images can be seen in Figure 3.13. We assess the precision of the VO trajectory estimates over a 700m loop of our Begbroke test site. For each mounting position we evaluate five laps of the track starting and finishing in the same position. The error of the trajectory estimates is therefore the delta pose between the initial and final pose for each lap.

Table 3.1 shows the errors statistics for all of the camera positions tested. It was found that a forward facing camera on either the bumper or the roof provided the highest precision trajectory estimates with less than $15m$ and 5 $degrees$ of error over the $700m$ course. The side-facing camera produced the worst performance, here the camera orientation was perpendicular to vehicle motion. Although we only offer the actually performance data, it is thought that the proximity to buildings lead to high visual flow in the side facing camera resulting in short lived features and poor motion estimates. It was also found that on average placing the camera inside the vehicle reduced the precision of motion estimates, this is thought to be caused by the distortion introduced by the glass and features tracked on the surface of the glass.

The camera mounting position chosen for further work presented in this thesis



Figure 3.13: Camera Trial Positions From top left to bottom right: Inside Windscreen Centre, Outside Front Bonnet, Outside Roof, Inside Rear Window Centre, Inside Side Facing, Inside Windscreen Centre, Inside Side Facing, Inside Windscreen Centre, Inside Windscreen Driver, Inside Windscreen Passenger and Outside Read Window.

is on the 'front bonnet'. Although the 'forwards facing roof' mounting position produced marginally better results, it is desirable for the work in Chapter 4 to acquire as much of the road surface as possible in the captured images.

	Mean Final Error (meters and degrees)						**Standard Deviation** (meters and degrees)					
Camera Position	x	y	z	*roll*	*pitch*	*yaw*	x	y	z	*roll*	*pitch*	*yaw*
Inside Windscreen Center	-8.631	-28.946	10.648	2.955	2.594	3.164	4.595	30.923	5.273	2.335	2.550	1.840
Outside Front Bonnet	**-1.837**	**12.020**	**-3.421**	**0.472**	**-2.629**	**0.307**	**1.088**	**0.979**	**4.145**	**0.917**	**1.172**	**0.639**
Outside Roof	**-3.934**	**3.382**	**-0.731**	**3.173**	**-0.377**	**-3.441**	**2.368**	**3.489**	**0.538**	**2.355**	**2.138**	**1.805**
Inside Rear Window Center	-22.279	-8.941	21.496	2.935	-6.524	15.462	9.994	27.326	6.817	2.389	3.503	3.460
Inside Side Facing	-74.798	6.911	-56.074	-26.008	-6.820	-64.975	35.327	27.223	25.570	114.28	24.203	75.928
Inside Windscreen Center	-26.63	-5.805	2.699	12.931	8.249	-24.071	15.454	40.326	10.166	1.861	3.585	35.283
Inside Windscreen Driver	-42.364	81.829	56.119	-3.013	11.540	-6.190	12.597	6.873	17.715	8.074	2.722	3.260
Inside Windscreen Passenger	7.359	39.900	-8.976	31.592	3.409	3.556	4.197	9.146	7.009	2.967	0.839	6.177
Outside Rear Window	-44.233	-28.841	42.200	13.561	-0.663	33.562	15.436	2.827	2.491	1.866	1.856	6.260

Table 3.1: A table showing the relative drift over a closed circuit around our Begbroke test track for various mounting positions. The mean and standard deviation is computed for the difference between the measured start points and end points. The best mounting position results are highlighted with bold text.

3.6.2 Evaluation Against INS

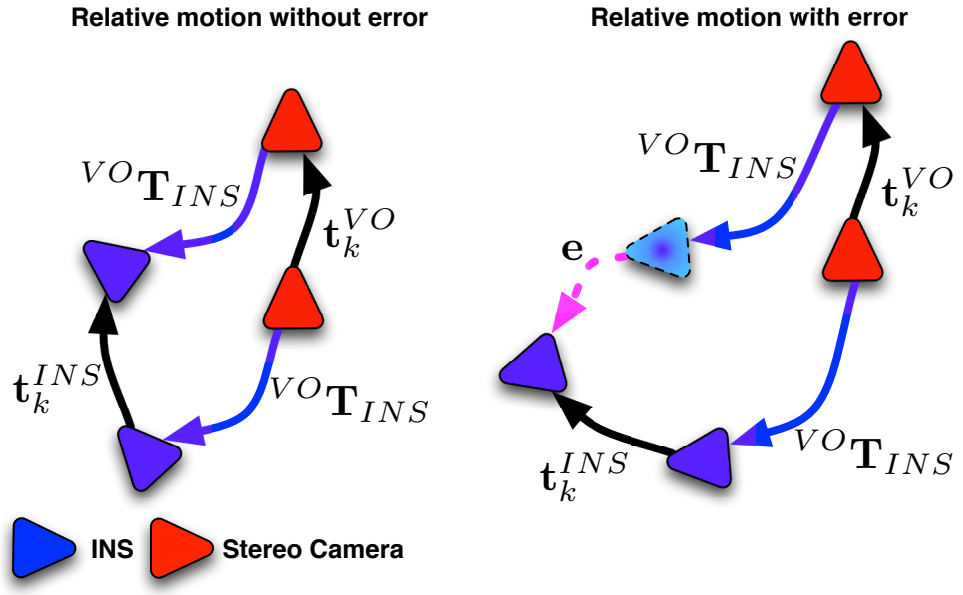


Figure 3.14: Computing relative pose error between the INS and VO. The inter-frame poses form the INS, $\mathbf{t}_k^{INS}$, are transformed into the Stereo Camera frame, $\mathbf{t}_k^{VO}$, using the extrinsic calibration $^{INS}\mathbf{T}_{VO}$. If there is no error in either of the measurements of relative pose this relative loop is closed (left), however if a measurement is in error the loop is violated and an error, $\mathbf{e}$, is introduced (right).

In addition to the camera placement testing we evaluate the VO system against our ground truth, the INS, for the chosen mounting position on the front bumper. We use the extrinsic calibration between the INS and stereo camera to transform the INS trajectory into the frame of the stereo camera, see Figure 3.14. This allows us to compute the inter-frame motion of the camera from both the VO and the INS. The error in the inter-frame motion acquired from the VO is then the delta pose between the INS and the VO relative poses, explicitly written as

$$\mathbf{e} = \ominus^{VO}\mathbf{T}_{INS} \ominus \mathbf{t}_k^{VO} \oplus^{VO} \mathbf{T}_{INS} \oplus \mathbf{t}_k^{INS}, \tag{3.60}$$

where $^{VO}\mathbf{T}_{INS}$ is the extrinsic calibration between the INS and stereo camera and $\mathbf{t}_k^{VO}$ and $\mathbf{t}_k^{INS}$ are the relative poses from the INS and VO respectively. Figure 3.14 shows how if the two pose sources agree the error would be zero and if they disagree the relative motion loop is violated and an error appears. It should be noted that we

re-sample the INS trajectory using the timestamps from the stereo frames and the calibration is assumed to be perfect.

3.6.2.1 Begbroke Loops Dataset

The Begbroke data set is $2.5km$ around our test site. Figure 3.15 shows the trajectory from both the INS and the VO registered into a global frame. The discrepancies between the inter-frame poses are of the order of millimetres and hundredths of degrees, see Figures 3.17 and 3.16. It should be noted that some of the outliers in Figure 3.16 are caused by the INS losing and re-establishing satellite lock. This process appears as spikes in the relative trajectory from the INS, seen in Figure 3.17.

The consistency of the VO with the INS, as shown in Figure 3.16, allows high metric accuracy for short, local trajectory segments (Section 4.8.4 deals with this in more detail). However, Figure 3.15 shows that even with this high local accuracy, for longer segments these small errors quickly compound leaving the global pose estimate from the VO only topologically correct. This further strengthens the case that without reference to prior data or a map open loop trajectory estimation is insufficient for reliable and accurate long term vehicle localisation.

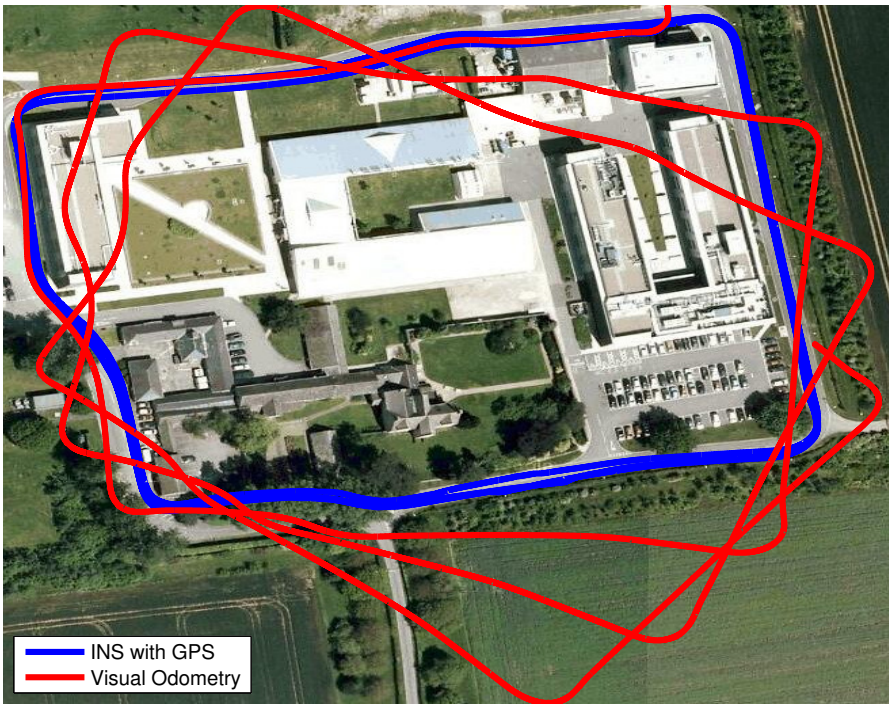


Figure 3.15: A 2.5km trajectory around our test site at Begbroke. The plot shows trajectory estimates from both the INS (blue) and the Stereo VO system (red). It can be seen that although the two trajectories are commensurate initially, the drift in the VO estimate makes it unsuitable for localisation as a stand alone system. Image courtesy of Google maps.

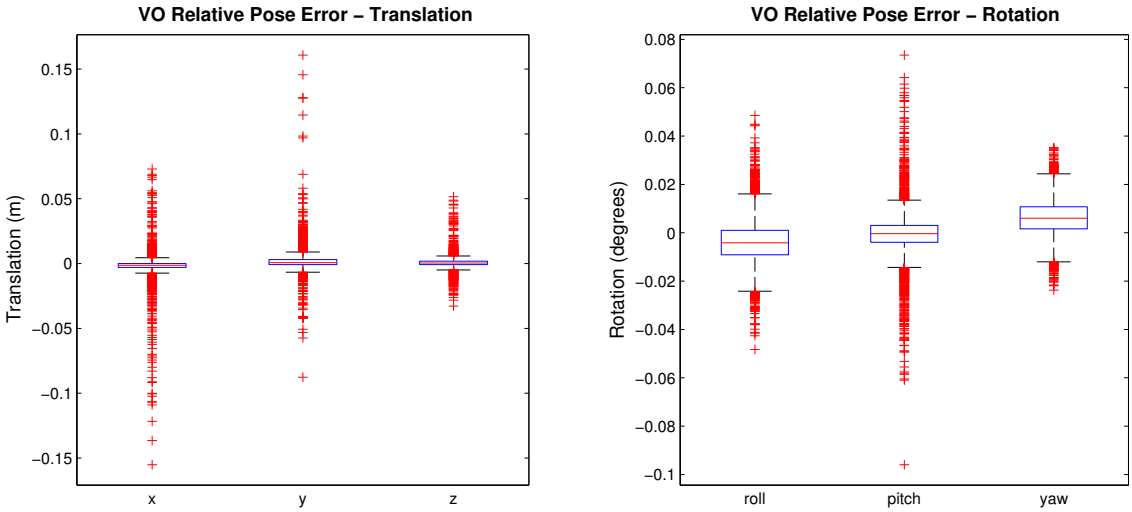


Figure 3.16: These box plots show the spread of the disagreement in the relative poses from the VO and INS for the Begbroke data set. The edges of the box denote the 25th and 75th percentiles, the whiskers denote the data not considered outliers, which are plotted individually. It can be seen that over 50% of the relative poses agree to the order of a few mm in translation and hundredth of a *degree* in rotation.

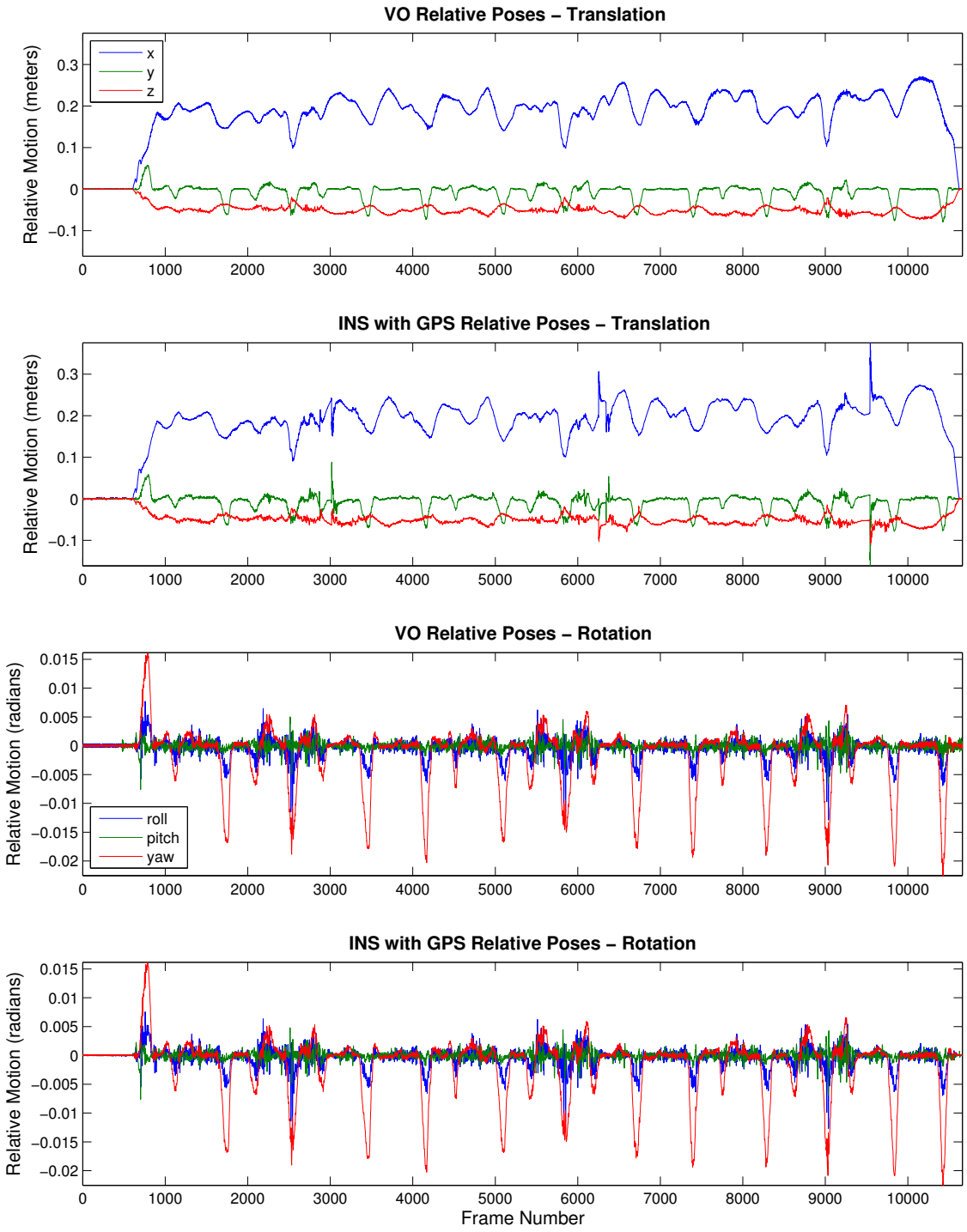


Figure 3.17: The relative trajectories from the VO and the INS with GPS for the Begbroke data set. The VO generates a relative trajectory with the commensurate smoothness to a high end INS system. Small errors in VO pose estimates however, cannot be corrected for without reference to prior data leading to drift. Spikes in the INS trajectory occur during periods with a reduced number of satellites and when GPS reception is lost. It should be noted that the INS poses have been transformed into the frame of the stereo camera for easy visual comparison.

3.6.2.2 Kidlington Dataset

The Kidlington data set covers a 300m stretch of suburban road with little visual texture on the road surface and buildings. Here we see commensurate performance with the Begbroke data set. However, the number of outliers for translation is significantly reduced as more satellites were visible to the INS.

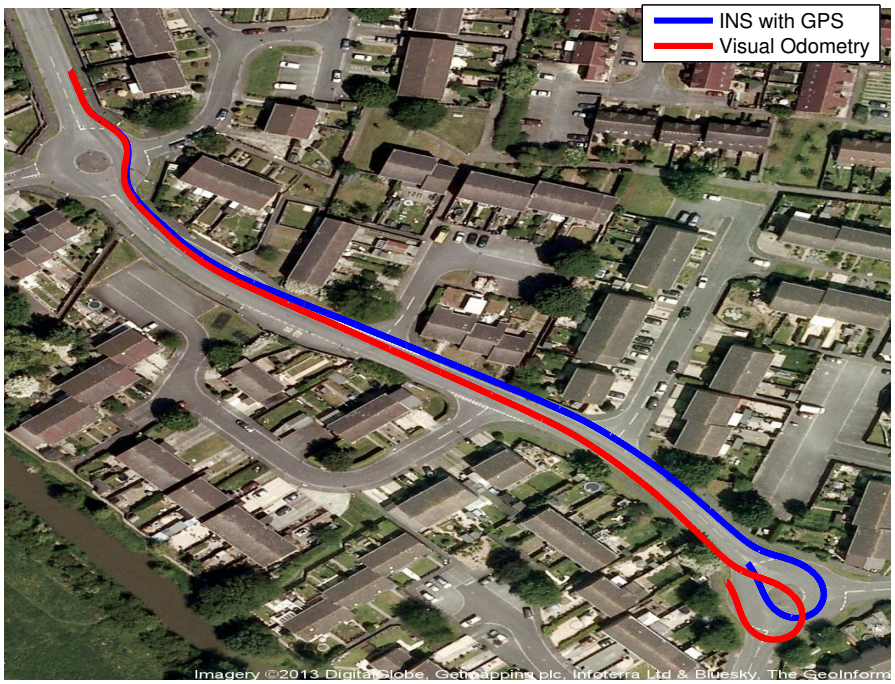


Figure 3.18: A 300 meter urban trajectory segment on urban roads showing trajectory estimates from both the INS (blue) and the Stereo VO system (red). It can be seen that although the two trajectories are commensurate, the drift in the VO estimate makes it unsuitable for localisation as a stand alone system. Image courtesy of Google maps.

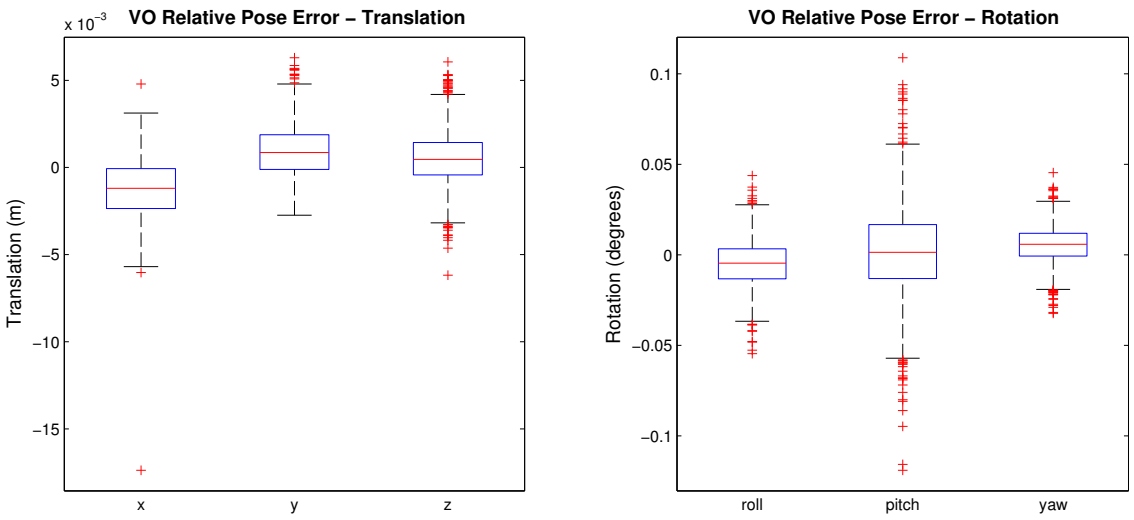


Figure 3.19: These box plots show the spread of the disagreement in the relative poses from the VO and INS for the Kidlington data set. It can be seen that 75% of the relative poses are are of the order of a few *mm* in translation and hundredth of a *degree* in translation.

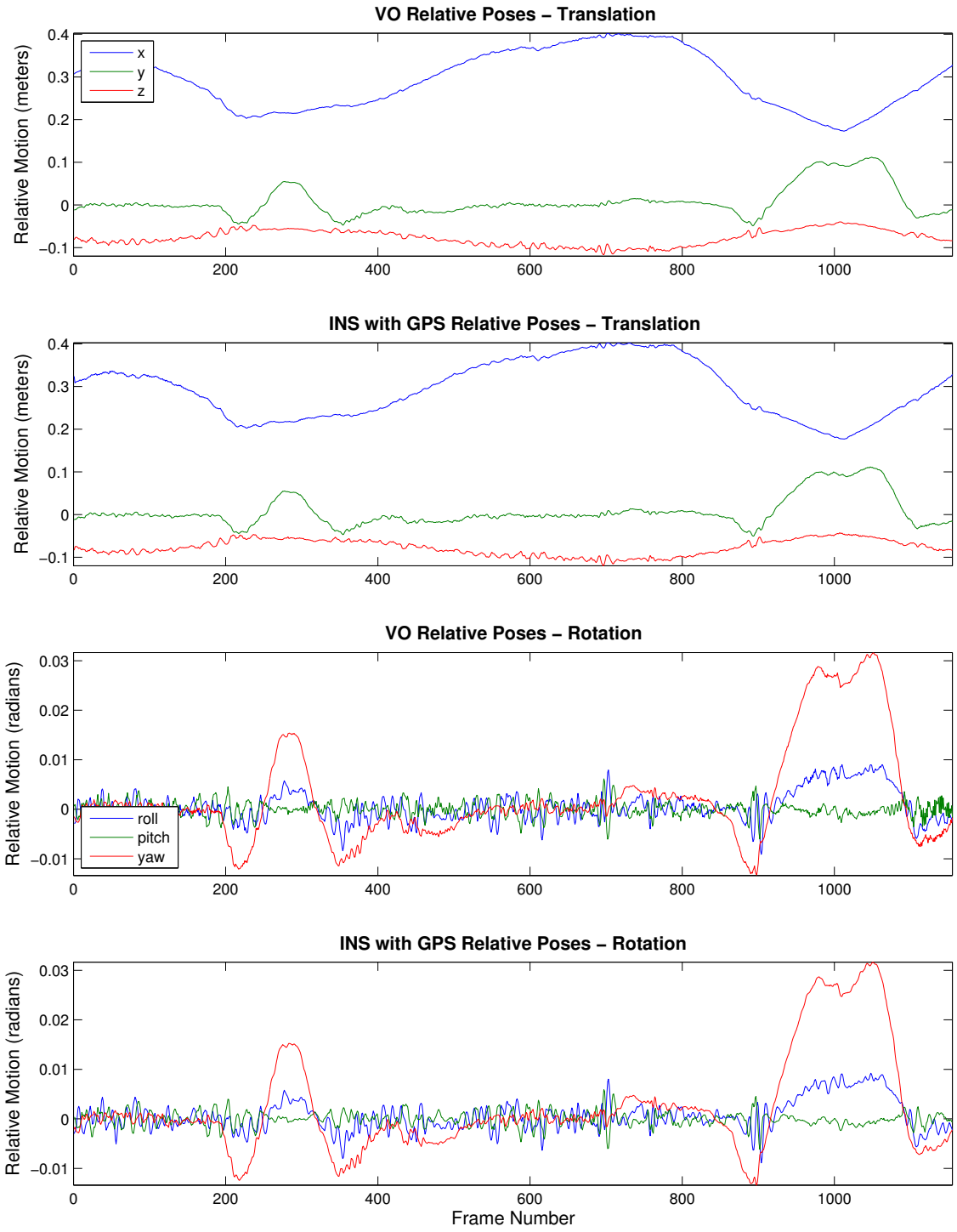


Figure 3.20: The relative trajectories from the VO and the INS with GPS for the Kidlington data set. The VO generates a relative trajectory with the commensurate smoothness to a high end INS system. Small errors in VO pose estimates however, cannot be corrected for without reference to prior data leading to drift. It should be noted that the INS poses have been transformed into the frame of the stereo camera for easy visual comparison.

		Translation (mm)			**Rotation** ($mDeg$)		
	Degree of Freedom	x	y	z	$roll$	$pitch$	yaw
Begbroke (2.5km)	Mean Error	-1.7	1.4	0.6	-3.7	-0.6	6.1
	Standard Deviation	6.7	5.8	3.5	8.5	7.8	7.0
Kidlington (300m)	Mean Error	-1.2	1.0	0.5	-4.6	1.4	5.2
	Standard Deviation	1.6	1.5	1.6	1.3	2.3	9.3

Table 3.2: Disagreement of relative poses from the VO system with our £10,000 INS with GPS unit. It can be seen that mm disagreement in translation and $mDegrees$ in rotation can lead to the drift shown in Figure 3.15 and 3.18

3.6.2.3 Overall Results

Overall performance can be seen in Table 3.2. The standard deviation of errors for Begrbroke is of the order of 5mm in translation and 8.5 milli-degrees in rotation, however outliers appear to dominate this. The outliers in the Begbroke data set appear to be caused by drop outs in GPS reception which were not present in the Kidlington data set. For the Kidlington data set, which contains fewer outliers, the standard deviations for translation are much smaller at 1.5mm. This equates to a 0.5% error for the distance traveled between frames. Although when to compared to the INS trajectories the performance of the VO system on these data sets is quite different, it appears that this difference is actually caused by variations in the performance of the INS we use as ground truth. The actual VO trajectories from both data sets appear to be smooth and consistent with the vehicle motion.

It should be noted that the errors in the relative VO pose estimates have non-zero mean. We suspect that rather than a statistical quirk this is instead a result of the inherent bias in stereo depth estimation. In Chapter 5 we derive the uncertainty associated with depth estimates derived from stereo triangulation. We show that the bias is introduced leads to persistent over estimation of the depth of $3D$ points. It is assumed this leads to the bias present in the VO derived pose estimates we see here.

3.7 Conclusion

In this chapter we describe a robust and versatile real time Stereo Visual Odometry system, which forms the foundation for the following Chapters. We first present an overview of the fundamentals of camera models and stereo vision. We then described how a Visual Odometry system robustly estimates the position of 3D landmarks and ego-motion from a sequence of undistorted and rectified stereo images.

The system exploits the high computational speed of the BREIF descriptor to allow the entirety of incoming images to be searched for temporal correspondences, a process not currently possible for real time applications with traditional features such as SIFT.

We then described the state vector and derive the full Jacobian. We also showed how the complexity of the state vector changes when, for example, we perform landmark refinement and or bundle adjustment.

We empirically determined the optimal camera mounting position for our road vehicle platforms. And we evaluated the optimal mounting position against a calibrated INS unit. The system was shown to produce trajectory estimates commensurate with a state of the art INS unit over small scales but suffers from the drift inherent in locally derived pose estimation over large scales.

Chapter 4

Localisation Using Synthetic Road Imagery

In this chapter we present a vision-based constant time, with respect to distance traveled, vehicle localisation system. We pose the problem as two separate processes, surveying and localisation. In the survey stage we synthesise high-resolution orthographic images of the road surface. This is achieved by exploiting the locally accurate pose estimates and 3D point features produced by the VO system described in Chapter 3. We use the trajectory of the camera and the 3D point features to extract the ground plane and project texture from the raw images onto locally metric orthographic images of the road surface. These orthographic images provide an excellent template to be matched against on subsequent traversals.

During the localisation stage a vehicle traverses a previously surveyed route. In order to localise this traversal against a survey, live images from a camera are aligned with synthesised views of the road surface. Unlike feature based approaches we use dense matching based on Mutual Information (MI). We evaluate the systems accuracy and robustness on over 7km (2km survey and 5km localisaiton) of real world data gathered using a BumbleBee2 stereo pair mounted on our Bowler WildCat platform, Figure 4.20.

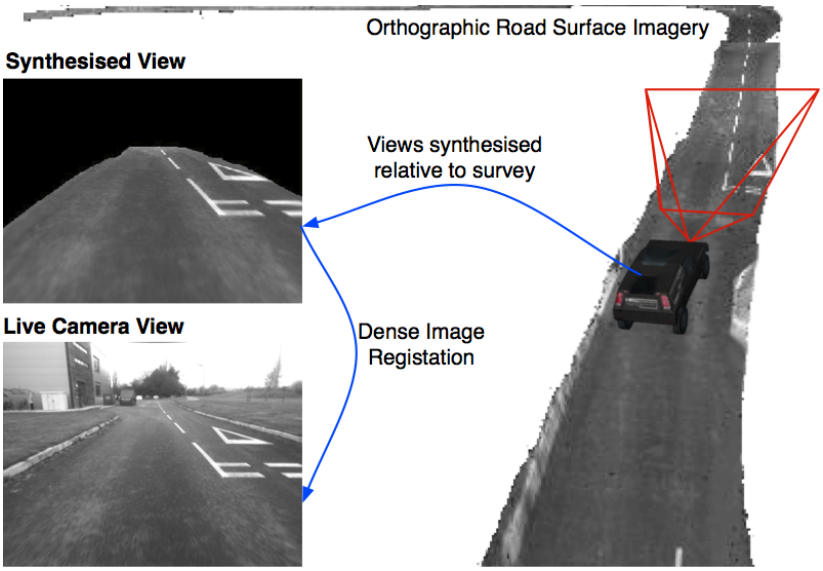


Figure 4.1: An overview of the basic idea behind the approach presented here. Given a live camera image, views of the orthographic road imagery or survey are synthesised and aligned using an information theoretic measure, Mutual Information. The Synthesised view with the 'best' alignment is taken to represent the vehicle pose relative to the survey.

4.1 Introduction

In order to achieve true road vehicle autonomy a fundamental competency is having an accurate and persistent measure of "where you are". Localisation and pose estimation derived from local sensors suffers from compounding errors. For example stereo Visual Odometry (VO) produces locally metric maps and trajectories [52]. However, when extending to larger scales without correction the metric precision is lost and maps and trajectories become only topologically correct. Small angular errors over the course of a few hundred meters will lead to pose estimates tens of meters in error. This makes localisation impossible without some sort of correction or reference to prior data.

Indoor robotics has enjoyed the luxury of infrastructure such as beacons and guide wires, which dramatically simplifies the problem. However, attempts to extend this infrastructure-based approach to larger scales has proved non-trivial [119]. Significant investment and an agreed framework between governments, car manufactures and other stakeholders are required. We therefore adopt an infrastructure free approach, requiring no modification to the existing road network.

For outdoor robotics, a popular recent approach is to use some combination of DGPS, inertial and 3D laser sensing coupled with a prior survey [10, 4, 58]. Here a survey vehicle equipped with an array of sensors including a 3D laser scanner, such as a Velodyne, builds a survey or map. This survey is then matched against live data enabling localisation on subsequent traversals. Here, with an eye on road vehicle localisation, we investigate how we might achieve commensurate precision using just vision.

The visual appearance of road surface is semi-permanent, changing very infrequently and under normal driving conditions can always been seen by cameras mounted on the vehicle. It is these qualities that make it a desirable region to focus on when attempting vision-based localisation. We adopt a dense image registration technique and attempt to use the gross appearance of the road surface as opposed to a more traditional sparse feature based approaches. During the survey stage we leverage our VO system to synthesise a continuous image strip of the road as seen from above, a synthetic orthographic image. We represent the orthographic imagery relative to the survey vehicle trajectory, which is not necessarily metrically correct over large scales. Our image generation is therefore not tied to a global frame, so does not require metric global consistency [99]. This lack of global consistency means we are unable to determine our position in a global sense of Northings and Eastings, but instead we estimate our position relative to a previous trajectory.

The synthetic orthographic images are generated at resolutions two orders of magnitude higher than the best available aerial images (5mm per pixel). Unlike previous work which relied on aerial imagery (Section 4.4) we are able to generate images where it is impossible to acquire aerial imagery, such as under bridges, trees and in man made structures such as multistory car parks. This method also has the advantage that only things which can be seen from a road vehicle's perspective are included in the survey, excluding rooftops and grassy fields, etc. The image strips need not be metrically correct over large scales (100m) but locally provide metric consistency to that of the VO system described in Chapter 3.

Subsequent traversals of surveyed routes by a vehicle can then be localised against the survey trajectory using these high resolution, high fidelity orthographic images. We operate in a relative frame work and as such only concern ourselves with the latest vehicle pose, resulting in a constant time algorithm. In contrast to many registration techniques we do not attempt a feature-based registration. Instead we seek the vehicle pose relative to the survey trajectory by maximising the Mutual Information between synthetic local orthographic images and the current view, Figure 4.1. We leverage the local accuracy of the VO system to produce high frequency pose estimates with low frequency corrections derived from our dense matching technique. This information theoretic based image alignment is shown to have superior robustness to lighting changes than simple correlation based approaches in Section 4.7.

In some sense what we propose here is a form of teach and repeat localisation, albeit one that negates the need for point features [27]. With this in hand global localisation is possible if the survey vehicle's trajectory has been post-processed-optimised into a single global frame [108]. This process of metric rectification is well studied [53, 30] but not necessary for accurate vehicle localisation relative to a survey.

In short, we present a vision based localisation system for road vehicles, which leverages the semi-permanent appearance of the road surface. We show that even under extreme appearance changes, a single survey coupled with a Mutual Information based image alignment measure is sufficient to localise subsequent traversals. Our method has no reliance on GPS and requires no additional infrastructure.

4.2 Motivation

One may ask why bother generating orthographic imagery? We have a VO system that extracts point features (in our case BRIEF features [9]) and saves them into a map (see Figure 4.7) why not localise using those?

Many point features are ephemeral, including vegetation, parked vehicles, and road speckle, which are suitable for frame-to-frame matching but unlikely to be seen

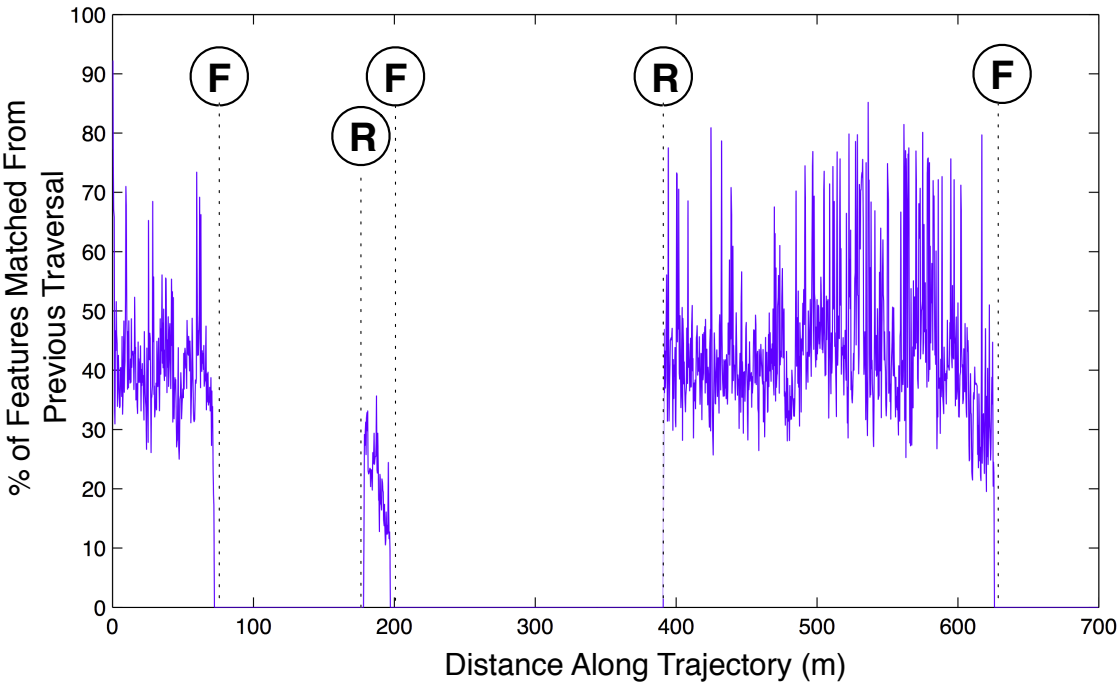


Figure 4.2: Shows the percentage of features matched against a previous trajectory of 700 meters. F indicates a failure i.e. not enough features matched to achieve localisation and R denotes a manual re-localisation. It can be seen that only $\sim 40\%$ of features are matched and the system fails 3 times over a 700m trajectory requiring manual intervention.

on subsequent traversals of the same route. The gross appearance of the road surface however, does not change for view point shifts experienced by a road vehicle. This is because the road surface is a largely planar, man-made structure. We successfully exploit both of these properties in this work.

Feature-based methods have been shown to be very sensitive to relatively small changes in view point and lighting conditions. This leads to a significant drop in matched features available for localisation, as made explicit in [27]. Figure 4.2 demonstrates how outdoor localisation using a feature based approach is not necessarily sufficient. Here we generate a point cloud of BRIEF features during a survey lap and attempted to localise a subsequent traversal using only these point features. Only 40% of the features are matched against the previous trajectory and after only $70m$ with a small deviation from the previous traversal there are not enough matched features for localisation, leading to irrecoverable failure. In the interests of further illustrating this point, two manual re-localisations were preformed

each of which eventually led to localisation failure.

As we will described in Section 4.3, there have been attempts to compensate for the brittleness of feature based approaches, by visiting places many times. These approaches have shown success in some regards, however they suffer from the complexities of managing databases full of features as well as the matching and tracking processes. Here we use a dense image alignment technique to avoid the need for point feature matching and tracking and attempt to glean enough information from one survey to enable robust localisation.

4.3 Background

The survey and localise framework is very similar to 'Teach and Repeat' and we shall discuss them interchangeably. Levinson et al [57] adopted an approach based on laser range finders. Here laser reflectance maps are built during a survey phase, which are then used for localisation. Their approach relied heavily on GPS and IMU and was based in a global frame. Ohno [79] and Tang [106] are some of the first attempts at visual teach and repeat. Here an indoor robotic platform first collects a sequence of images. Edges are then extracted and used to localise subsequent image sequences take from the same route. A more sophisticated approach was taken in [27], where point features are extracted and stored in sub-maps during a survey phase and then used for localisation. They evaluate their algorithm in desert like scenes and do not attempt to demonstrate robustness against scene or lighting changes.

In [12], a point feature based approach is used and an experience based framework is developed. This approach, coined Experience Base Navigation (EBN) does not use a privileged survey, but instead allows subsequent traversals to become surveys themselves if existing data is not sufficient for localisation. Figure 4.2 demonstrates that using a single survey is not sufficient and in-fact [12] showed that a route must be traversed several times before sufficient experience had been gain to enable for persistent localisation. Our system on the other hand can persistently localise the same route using a single survey with no need for multiple experiences. An issue,

which is not addressed in these feature-based approaches, is the ephemeral nature of the small patches associated with point features. In work such as [54] and [45] scenes are recorded repeatedly under varying conditions and used to learn what constitutes as a 'good' or permanent feature. An alternative to this approach is taken in [65] where prior 3D structure is used to determine static areas of the scene, such as buildings. It is shown that focusing attention of these static regions allows more consistent matching of point features and subsequently more accurate localisation.

Non-feature-based approaches include the dense approach in [13]. Here the alignment of an image from the survey and a subsequent traversal is measured using Mutual Information. There is no scene structure assumed and the pose of the vehicle is not explicitly estimated. Instead they simply attempt to roughly follow the original survey path. In this work we estimate the planar structure of the road ahead and use this in conjunction with MI for image alignment to explicitly extract the full $\mathbb{SE}3$ pose of the vehicle. In [103] a dense survey and localise approach is also used. Here scene structure acquired using a $3D$ LIDAR at survey time is used to project pixels between sequential images based on vehicle motion. This approach has shown promise, but requires lots of effort in post processing the LIDAR data to remove ephemeral objects such as parked cars and pedestrians etc. In the work presented here we not only don't require LIDAR data, but, as we focus on a know structure in the scene, the road surface, we implicitly mitigate the issue of dealing with ephemeral objects in the survey data, see Section 4.5.2.

4.4 Exploiting Aerial Imagery

Here we alert the reader to a slight jump as we briefly describe previous work, which inspired the main contributions in this Chapter.

Previous work [72] attempted to register a locally derived VO trajectory against aerial images used as prior information. This approach corrected for the drift inherent in VO systems, maintaining large scale metric accuracy of trajectory estimates.

Figure 4.3 shows a basic system overview. The only assumption we make about the

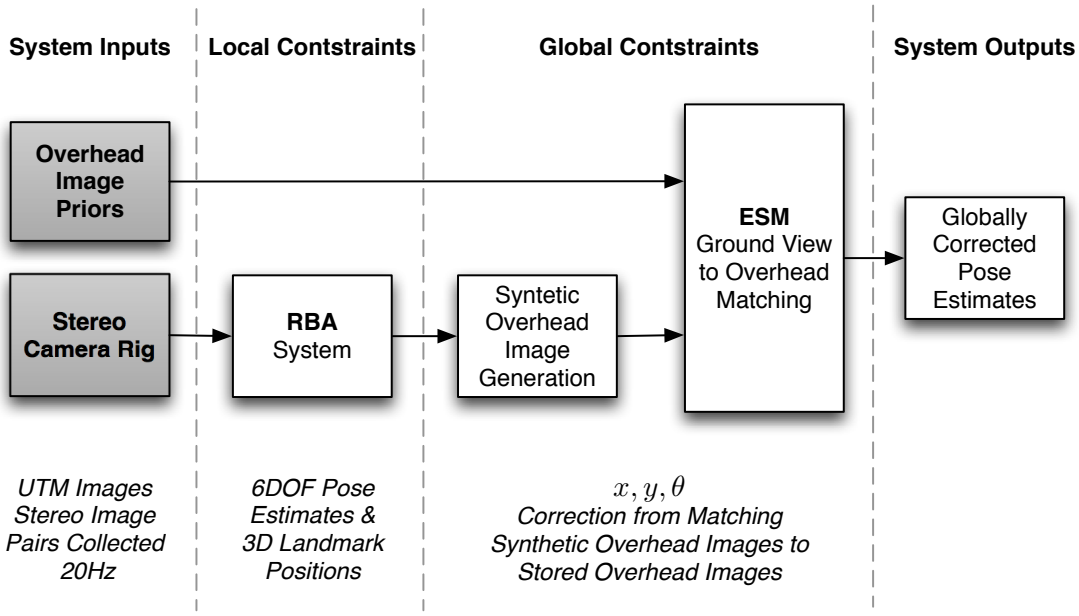


Figure 4.3: System overview of road vehicle localisation using aerial imagery. The ground view to overhead matching is elaborated on in Figure 4.4

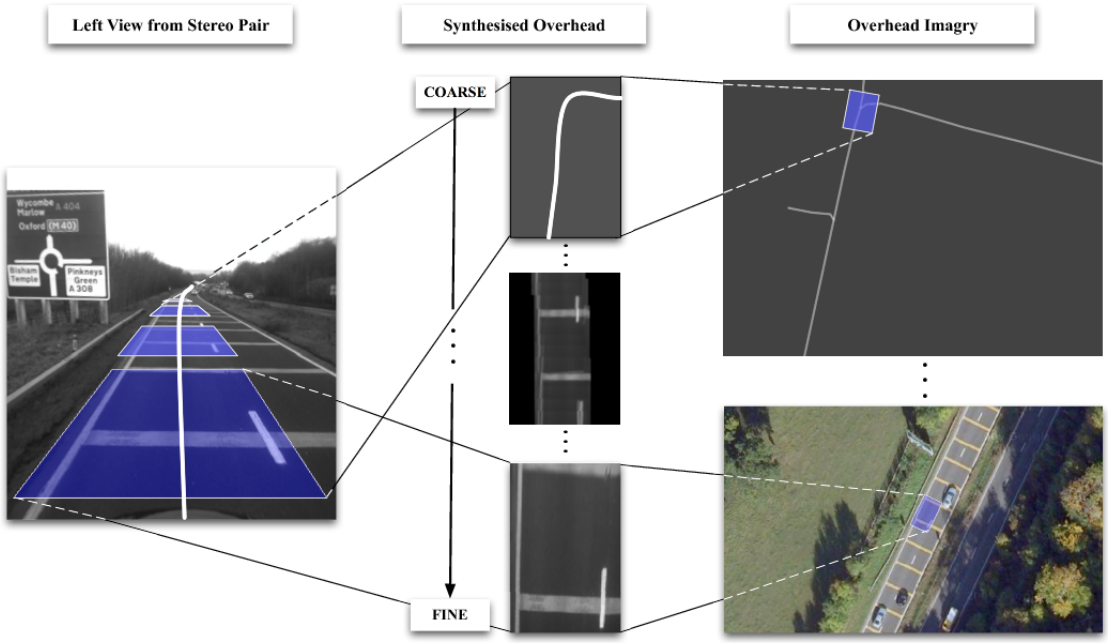


Figure 4.4: Overview of the coarse-to-fine road matching scheme. During nominal operation the vehicle is well localized in pixel coordinates in the overhead image. Using a coarse-to-fine search strategy, the minimum cost image-to-image alignment is found using efficient second-order minimization (ESM).

vehicle trajectory is that the vehicle is driven exclusively on the road network. This assumption is justified for our application domain, which is road vehicle autonomy. The inputs to the system are a sensor, the BumbleBee stereo pair, and aerial images registered against the Universal Transverse Mercator (UTM) [118] Coordinate system. The local trajectory is computed using VO. Synthetic Overhead Images are generated from the stereo imagery, discussed in Section 4.5.2, and then matched against the UTM images using an Efficient Second order Minimisation (ESM) [66]. A coarse-to-fine approach was adopted whereby progressively finer refinements are made to the pose estimates by matching images from the local stereo camera to aerial images. At the coarsest level we attempt to match the actual shape of the road network to the VO trajectory. We then progress to matching individual stereo frames to the UTM images at the finest level, see Figure 4.4. The acquired matches are then used as global constraints to relax the trajectory from the VO system onto the road network, correcting for any drift that might have occurred.

4.4.1 Results

This approach produced pose estimates commensurate with the performance of off-the-shelf GPS (10 meters) over kilometer scales. Figure 4.5 shows the raw VO trajectory and the corrected trajectory overlaid on the matched aerial image. It can be seen that the raw VO trajectory drifts significantly over the 700m trajectory while the corrected trajectory appears to stay locked onto the road network. Figure 4.6 shows the discrepancy between the estimated pose of the vehicle within the UTM registered aerial image and the off the shelf GPS available in this data set. Although

In practice the efficacy of aerial images for reliable and accurate road vehicles localisation is limited. Trees and bridges often occlude the road surface and image resolution is of the order of tens of centimeters per pixel. These factors make matching at the finer scales unreliable and limit the accuracy of the derived pose estimates. Although this approach showed promise, we will show how superior accuracy can be achieved through the replacement of aerial images with synthesised overhead images

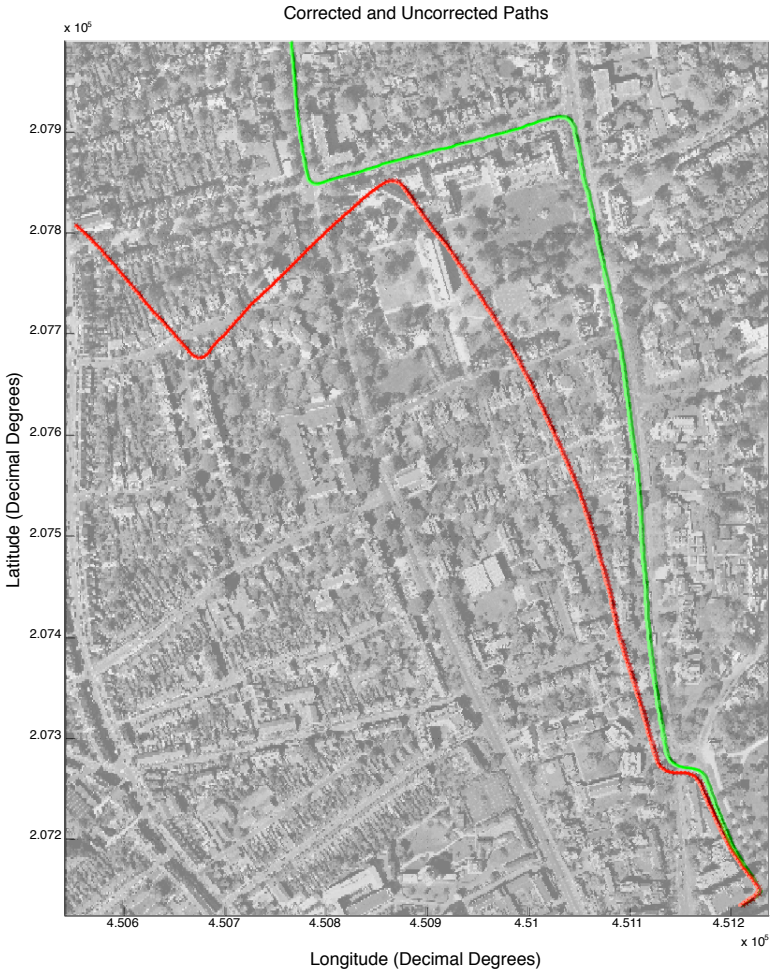


Figure 4.5: A 1.1km stretch of urban roads. Raw VO (red) and the trajectory corrected onto the road (green).

of the road surface. We refer an interested reader to [72] for more system details and results.

In similar work Pink et al [86, 87] extracted road markings from aerial and camera images to perform localisation. Another approach by Kummerle et al [90] extracted edges of buildings from aerial images for corrections using a 2D laser based system. More recently other approaches such as [7] and [23] use only the shape and topology of the road network to provide course GPS level accuracy. They make no effort to further refine the pose estimates.

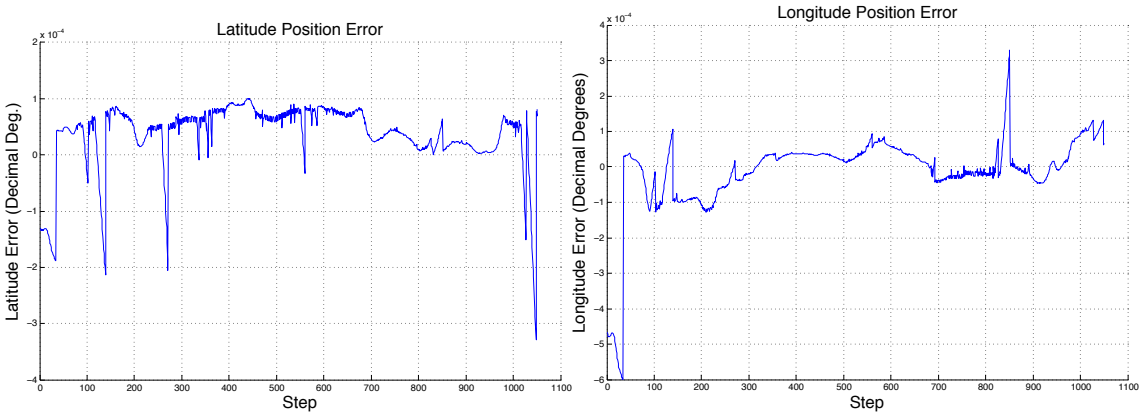


Figure 4.6: Error in Latitude (left) and Longitude (right) relative to GPS (with a 10m CEP). An error of 1e-4 deg. equates to ˜10m.

4.5 The Survey

The survey as referred to here consists of a privileged traversal of a route on which a vehicle is to be localised. As a survey is an infrequent event, care can be taken to ensure the images captured during the survey are in ideal weather and traffic conditions. This greatly simplifies the process of generating local orthographic images of the road surface as we don't have to concern ourselves with shadowing [83, 114] and vehicles occluding the road surface.

The process of generating the synthetic images of the road surface is effectively an image mosaicking problem [64]. However, rather than simply stitching images together into a panorama we assume a planar structure and project the texture onto a plane in 3D space. Previous work on structure from motion such as [85] uses a piecewise planar representation of the world and [20] uses a triangular mesh. Both papers attempt metric reconstructions of 3D structure from a video or image sequence. Here we are only interested in the road surface and limit our world model to a continuous strip of road imagery.

4.5.1 Survey Trajectory

Let us define the notation and terms on which we will rely. A vehicle trajectory $\mathcal{T}$, as generated by the VO, is a set of relative poses ${}^{i}x_{i+1}$ between consecutive vehicle poses X_i . The relative pose between X_i and X_{i+1} is written as ${}^{i}x_{i+1}$. In this way

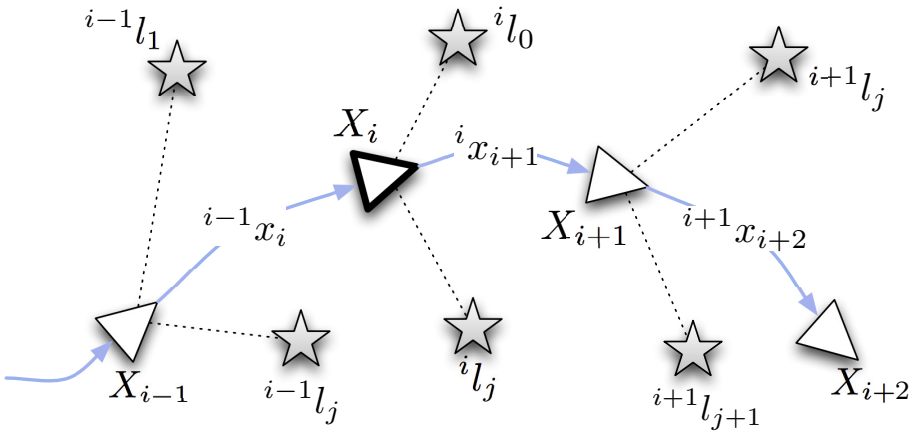


Figure 4.7: Notation for a simple VO trajectory and scene $\mathbf{M_i}$. Relative poses between frames are represented by arrows and lowercase $^{i-1}x_i$, Vehicle poses relative to X_i are represented by triangles and uppercase X_{i-1}. The location of the jth observed landmark from pose X_i are represented by stars $^{i}l_j$.

$\mathcal{T} = \{^0x_1, ^1x_2, \cdots, ^{N-1}x_N\}$ is a trajectory of N vehicle poses (see Figure 4.7). Note that in the interest of clarity we shall sometimes refer to X_i as a vehicle node rather than as a pose, the reason being that one can imagine $\mathcal{T}$ as a chain graph of nodes linked by relative poses. Each node in the graph has one or more 3D landmark features attached to it. The jth landmark attached to vehicle node i is denoted as $^{i}l_j$. We use the function $L(X_i)$ to represent the act of producing a local metric scene $\mathbf{M_i}$ around X_i which contains a set of local vehicle nodes and all attached landmarks in the frame of X_i (so X_i becomes the origin). In general $\mathbf{M_i}$ can be constructed from an arbitrary set of consecutive poses and landmarks but for the simple example in Figure 4.7

$$\mathbf{M_i} = \left\{ \begin{array}{c} X_{i-1}, X_i, X_{i+1}, X_{i+2} \\ ^{i-1}l_1, ^{i-1}l_j, .., ^{i}l_0, ^{i+1}l_{1+j}, ..., ^{i+2}l_1, ... \end{array} \right\} \tag{4.1}$$

$$X_{i+2} = X_i \oplus ^{i}x_{i+1} \oplus ^{i+1}x_{i+2} \tag{4.2}$$

$$X_{i-1} = X_i \ominus ^{i-1}x_i \tag{4.3}$$

where $\oplus$ and $\ominus$ represent the composition and inverse composition operators respec-

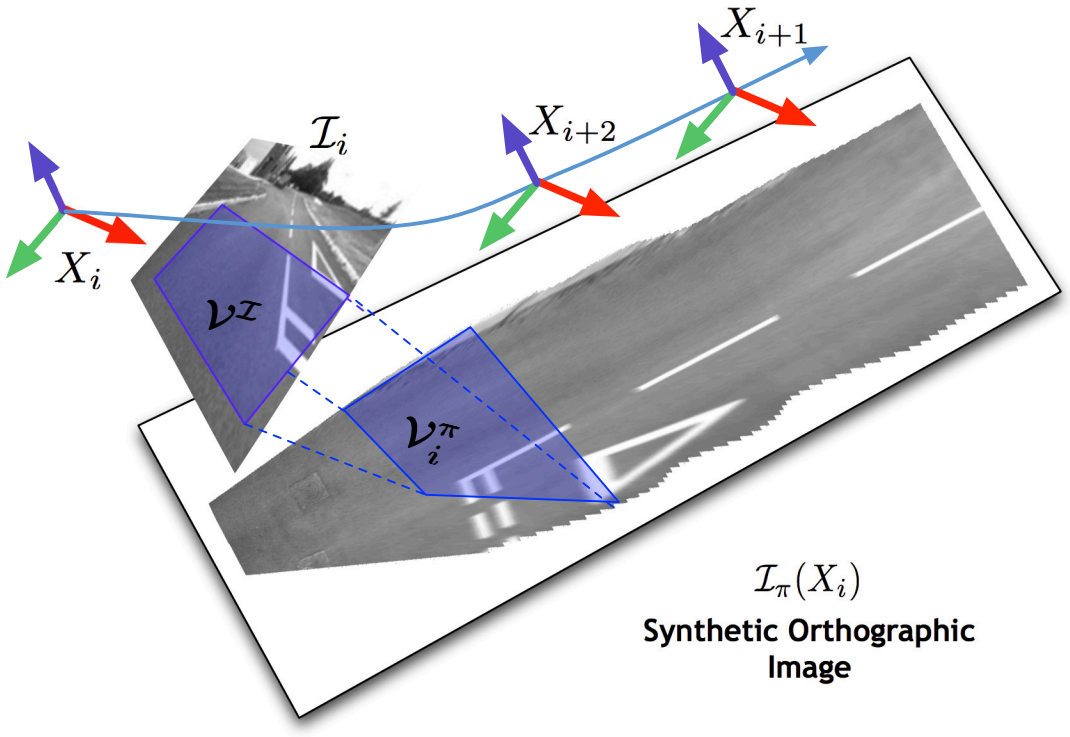


Figure 4.8: Shows the view $\mathcal{I}_i$ at pose X_i from the survey trajectory being projected onto the local ground plane to produce an orthographic image $\mathcal{I}_\pi(X_i)$. Road regions of interest $\boldsymbol{\mathcal{V}}^{\mathcal{I}}$ and $\boldsymbol{\mathcal{V}}_i^{\pi}$ are shown in blue.

tively, such that ${}^i x_{i+1} \ominus {}^i x_{i+1}$ is the identity transformation. In Section 4.5.2 we shall describe how $\mathbf{M_i}$ can be used to generate a synthetic local orthographic image around X_i.

4.5.2 Generating Local Orthographic Images

At run time we will use knowledge gleaned from a previously recorded excursion (survey) to ensure we stay localised relative to the prior survey. A fundamental competency on which we depend is the availability/generation of a synthetic orthographic image around a particular node X_i in the survey trajectory $\mathcal{T}$. We use π to indicate a variables association with the ground plane. The function $\mathcal{I}_\pi(X_i)$ denotes the generation of this image which utilises the local metric scene $\mathbf{M}_i$ defined above. $\mathcal{I}_\pi(X_i)$ can be computed at run time, but can also be precomputed. The process begins with the extraction of a ground plane using the landmarks ${}^i l_j$ in $\mathbf{M}_i$ and RANSAC [22] to solve.

$$\boldsymbol{b_i^\pi}.\hat{\boldsymbol{n}}_{\boldsymbol{i}}^{\boldsymbol{\pi}} = \boldsymbol{l}.\hat{\boldsymbol{n}}_{\boldsymbol{i}}^{\boldsymbol{\pi}} \tag{4.4}$$

where $\boldsymbol{b_i^\pi}$ is the base, $\hat{\boldsymbol{n}}_{\boldsymbol{i}}^{\boldsymbol{\pi}}$ the normal and $\boldsymbol{l}$ an arbitrary point on the plane. Note that in our experimental set up, the front bumper stereo camera (Figure 4.20) is orientated such that at least half of the view contains road surface, assuming we are traversing relatively smooth urban roads. This orientation ensures many landmark measurements correspond to points on the road surface aiding ground plane estimation. Based on the stereo camera's orientation a Region Of Interest (ROI) $\boldsymbol{\mathcal{V}^{\mathcal{I}}} = [\mathcal{V}_1^{\mathcal{I}}, ..., \mathcal{V}_J^{\mathcal{I}}]$ in the left camera image is set as a region likely to only contain road surface. Here $\mathcal{V}_j^{\mathcal{I}} = [u_j^{\mathcal{I}}, v_j^{\mathcal{I}}, 1]^T$ is a pixel location in homogeneous co-ordinates of the jth vertex of the ROI. The intersection of the rays associated with each $\mathcal{V}_j^{\mathcal{I}}$ and the local ground plane ($\boldsymbol{b_i^\pi}$, $\hat{\boldsymbol{n}}_{\boldsymbol{i}}^{\boldsymbol{\pi}}$) is calculated. $\boldsymbol{\mathcal{V}_i^\pi} = [\mathcal{V}_{i,1}^{\pi}, ..., \mathcal{V}_{i,J}^{\pi}]$ is then the ROI projected onto the local ground plane around pose X_i.

$$\mathcal{V}_{i,j}^{\pi} = \lambda_j K^{-1} \mathcal{V}_j^{\mathcal{I}} \tag{4.5}$$

$$\lambda_j = \frac{\boldsymbol{b_i^\pi}.\hat{\boldsymbol{n}}_{\boldsymbol{i}}^{\boldsymbol{\pi}}}{\left[K^{-1}\mathcal{V}_j^{\mathcal{I}}\right].\hat{\boldsymbol{n}}_{\boldsymbol{i}}^{\boldsymbol{\pi}}} \tag{4.6}$$

where K is the matrix of camera intrinsics, $K^{-1}\mathcal{V}_j^{\mathcal{I}}$ is the ray associated with $\mathcal{V}_j^{\mathcal{I}}$ and λ_j is the distance along $K^{-1}\mathcal{V}_j^{\mathcal{I}}$ to the intersection with the ground plane. As $\boldsymbol{\mathcal{V}^{\mathcal{I}}}$ and $\boldsymbol{\mathcal{V}_i^\pi}$ both lie on planes they define a projective homography $\mathcal{H}_i \in \mathbb{R}^{3\times3}$.

$$\boldsymbol{\mathcal{V}^{\mathcal{I}}} = \mathcal{H}_i \boldsymbol{\mathcal{V}_i^\pi} \tag{4.7}$$

$$\mathcal{H}_i = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & 1 \end{bmatrix} \tag{4.8}$$

As $\mathcal{H}_i$ is a projective homography or transformation it has eight degrees of freedom and must be defined by at least four point correspondences. $\mathcal{H}_i$ is found by rewriting

Equation 4.7 as a least squares formulation [40].

$$\begin{bmatrix} u_1^{\mathcal{I}} \\ v_1^{\mathcal{I}} \\ \vdots \\ u_N^{\mathcal{I}} \\ v_N^{\mathcal{I}} \end{bmatrix} = \begin{bmatrix} u_1^{\pi} & v_1^{\pi} & 1 & 0 & 0 & 0 & -u_1^{\pi}u_1^{\mathcal{I}} & -v_1^{\pi}u_1^{\mathcal{I}} \\ 0 & 0 & 0 & u_1^{\pi} & v_1^{\pi} & 1 & -u_1^{\pi}v_1^{\mathcal{I}} & -v_1^{\pi}v_1^{\mathcal{I}} \\ \vdots & & & & \ddots & & & \vdots \\ u_N^{\pi} & v_N^{\pi} & 1 & 0 & 0 & 0 & -u_N^{\pi}u_N^{\mathcal{I}} & -v_N^{\pi}u_N^{\mathcal{I}} \\ 0 & 0 & 0 & u_N^{\pi} & v_N^{\pi} & 1 & -u_N^{\pi}v_N^{\mathcal{I}} & -v_N^{\pi}v_N^{\mathcal{I}} \end{bmatrix} \begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{2,1} \\ h_{2,2} \\ h_{2,3} \\ h_{3,1} \\ h_{3,2} \end{bmatrix} \tag{4.9}$$

which for convenience we can write as

$$y = A\mathcal{H}_i^s, \tag{4.10}$$

where $\mathcal{H}_i^s = [h_{1,1}, h_{1,2}, ..., h_{3,2}]^T \in \mathbb{R}^8$ the pixel wise error between the original points and the projected points can the be written as

$$e = y - A\mathcal{H}_i^s \tag{4.11}$$

giving the least squares estimate of $\mathcal{H}_i^s$ as

$$\mathcal{H}_i^s = (A^T A)^{-1} Ay. \tag{4.12}$$

$\mathcal{H}_i$ is then used to project the texture in the survey images ROI taken at X_i into an orthographic image which we call $\mathcal{I}_\pi(X_i)$. We do this for all poses in the scene $\mathbf{M}_i$. The camera frame rate of $20Hz$ coupled with the survey vehicle's velocity of approx. 20kph leads to adequate overlap between consecutive $\boldsymbol{\mathcal{V}_i^\pi}$ (road regions of interest projected onto the orthographic image $\mathcal{I}_\pi(X_i)$). This presents an opportunity to combine ROIs for consecutive poses in $\mathbf{M}_i$ by taking the mean of intensity values. This generates an image of length defined by the poses in $\mathbf{M}_i$ of the road surface as

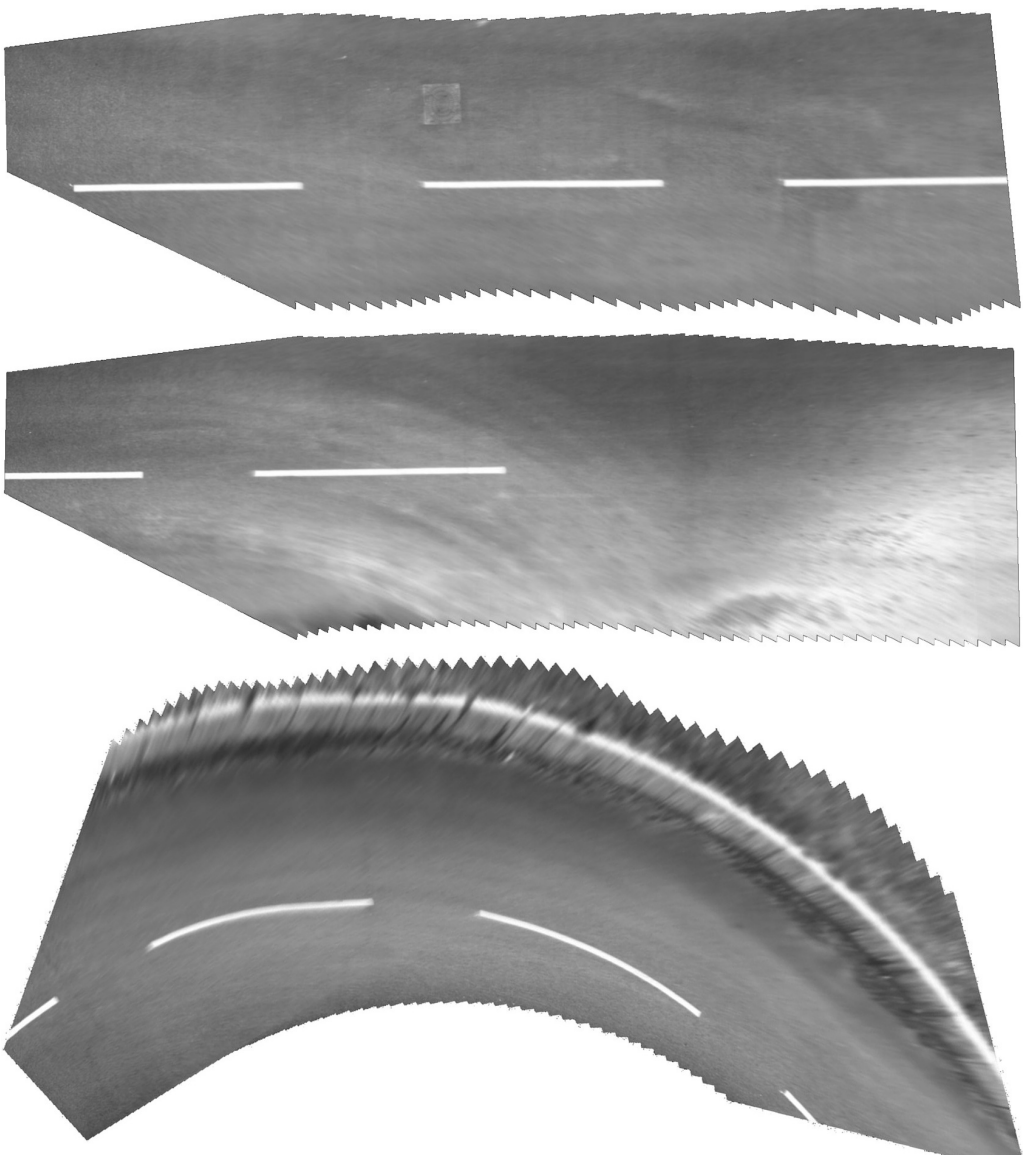


Figure 4.9: 10 meter segments of the synthetic local orthographic images generated from the 700m test route. The top two are from straight segments and the third is part of a 90° corner.

seen from overhead in the vicinity of X_i (Figure 4.8). Here we use a resolution of 5mm per pixel in $\mathcal{I}_\pi(X_i)$ (Note that this resolution was chosen as a balance between accuracy and storage requirements, approx. 10MB/km). Examples of images $\mathcal{I}_\pi(X_i)$ associated with our test route are shown in Figure 4.9. The image alignment accuracy is of the order $10^{-3}m$ and the images contain a high level of detail.

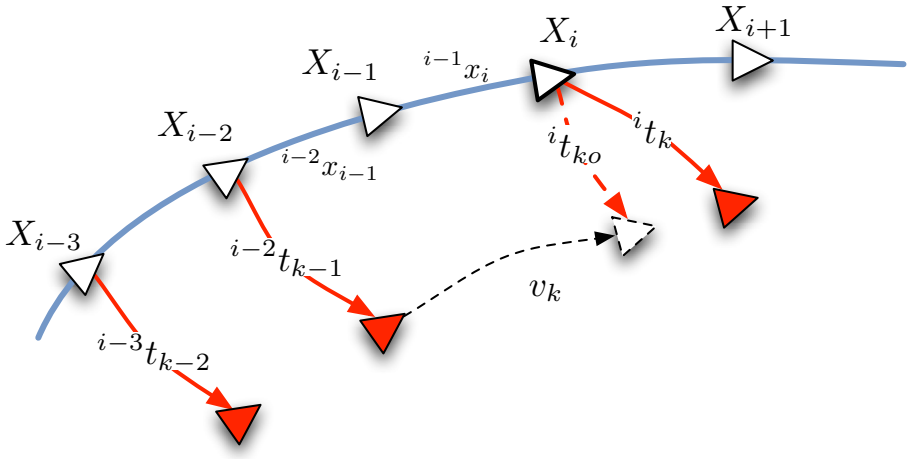


Figure 4.10: Localisation Framework - The survey trajectory is shown in blue with vehicle nodes/poses X_i. A subsequent traversal of the route localised against the survey trajectory is shown in red ${}^{i}t_k$, the current relative VO pose v_k and subsequent corresponding seed for localisation ${}^{i}t_{ko}$ are indicated by dashed lines.

4.6 Trajectory Representation

Consider now an image $\mathcal{I}_k$ acquired on a subsequent traversal of a surveyed route at time k in the vicinity of X_i. The pose of the vehicle can now be represented relative to X_i with ${}^{i}t_k$ which is the relative pose between X_i and the location of the vehicle at time k (See Figure. 4.10). If required the global pose of the vehicle X_k is then simply

$$X_k = X_i \oplus {}^{i}t_k. \tag{4.13}$$

At run time our Stereo VO system provides a continual stream of estimates of the relative pose between camera frames v_k. In the absence of any other knowledge we could use this to infer our trajectory in open loop. If however we can leverage the synthetic orthographic images to correct relative poses from the VO we would be in a position to track our motion (stay localised) relative to the survey trajectory. Furthermore, if as a new stereo pair is presented to us, we could use v_k to seed a guess for the relative pose ${}^{i}t_{ko}$ between the new camera frame and the survey trajectory, for the example in Figure 4.10 ${}^{i}t_{ko}$ would be expressed as.

$${}^{i}t_{ko} = \ominus {}^{i-1}x_i \ominus {}^{i-2}x_{i-1} \oplus {}^{i-2}t_{k-1} \oplus v_k. \tag{4.14}$$

Of course as we move we need to track our location relative to sequential poses in the trajectory - X_i will change as we move. However, this is a trivial data association problem; we can simply predict the transition to a new reference pose using v_k as indicated by our VO system. This trajectory representation makes the algorithm constant time as only the latest frame is used for localisation. The goal of this work is then to develop a way to hone this initial estimate ${}^{i}t_{ko}$ and we shall do so by comparing the live view $\mathcal{I}_k$ and that predicted by a hypothosised view of the synthetic orthographic image $\mathcal{I}_\pi(X_i)$ at ${}^{i}t_k$. We shall now pose the task of finding ${}^{i}t_k$ as an optimisation problem.

4.7 Mutual Information for Image Alignment

In order to exploit the synthetic overhead imagery to localise a vehicle we must develop a way of measuring image alignment. If we knew ${}^{i}t_k$ perfectly then the projection $proj(\mathcal{I}_\pi(X_i),{}^{i}t_k)$ (hypothesised view) of the road lying in $\mathcal{I}_\pi(X_i)$ into the live view $\mathcal{I}_k$ would align. Conversely, if the pose is in error, the two views will not align and should have a significant reduction in the image alignment measure used. The optimisation therefore finds a relative pose ${}^{i}\hat{t}_k$ which maximises the alignment of the live image $\mathcal{I}_k$ and a synthetic view of the overhead imagery $proj(\mathcal{I}_\pi(X_i),{}^{i}t_k)$. Figure 4.11 shows an example live image taken from the left image sensor of the stereo pair. The left and right images show synthetic views taken from a pose in error and the true pose respectively. It can be seen that there is a significant misalignment between the live view and the synthetic view when the pose is in error.

Rather than use a feature-based approach for image alignment we use a dense method. Using a dense method exploits all of the pixels in the image and avoids a lot of the data association complexities associated with feature-based approaches. We adopt the information theoretic measure of Mutual Information (MI) rather than a simple correlation-based approach to measure image alignment. We shall show in Section 4.7.1 that MI is more robust to lighting changes and occlusions than a simple Mahalanobis distance image similarity measure.

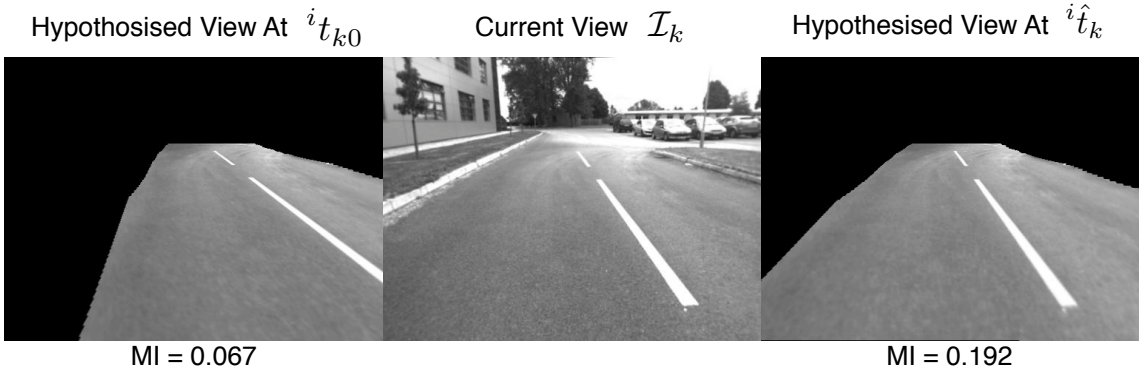


Figure 4.11: Example of view alignment between the live view and hypothosised views and their corresponding values for mutual information. Left shows the hypothosised view at ${}^{i}t_{ko}$ provided by v_k, Middle shows the live image $\mathcal{I}_k$ and Right shows the hypothosised view at the corrected pose ${}^{i}\hat{t}_k$.

Mutual Information was first defined by Shannon [95] and has a history of use in medical image registration [116, 88]. Its lighting invariance has also encouraged its use in outdoor robotics applications such as work from Dame et al [13, 14].

The MI between two images $\mathcal{I}$ and $\mathcal{I}^*$, intuitively can be thought of as the information shared between the two images. Figure 4.12 shows a Venn Diagram of the information contained within two images. If the two images were identical the Mutual Information would be equal to the information contained within each of the individual images, as the difference in the images increase the MI decreases.

$$MI(\mathcal{I}, \mathcal{I}^*) = H(\mathcal{I}) + H(\mathcal{I}^*) - H(\mathcal{I}, \mathcal{I}^*) \tag{4.15}$$

The MI is obtained by evaluating the Shannon entropy of the images individually $H(\mathcal{I})$ and $H(\mathcal{I}^*)$, and then evaluating the joint entropy $H(\mathcal{I}, \mathcal{I}^*)$. The entropy of a single image is a measure of how much information is contained within the image,

$$H(\mathcal{I}) = -\sum_{n=0}^{N} p_{\mathcal{I}}(n) log(p_{\mathcal{I}}(n)) \tag{4.16}$$

where $p_{\mathcal{I}}(n)$ is the probability of a pixel in image $\mathcal{I}$ having intensity n. An image can therefore be thought of as a random variable with each pixel location x having a distribution defined by, $p_{\mathcal{I}}(n) = p(I(x) = n)$ for $n \in [0, N]$, where N is the maximum intensity (in our case 255, as we are using 8-bit gray scale images). $H(\mathcal{I})$ and $H(\mathcal{I}^*)$

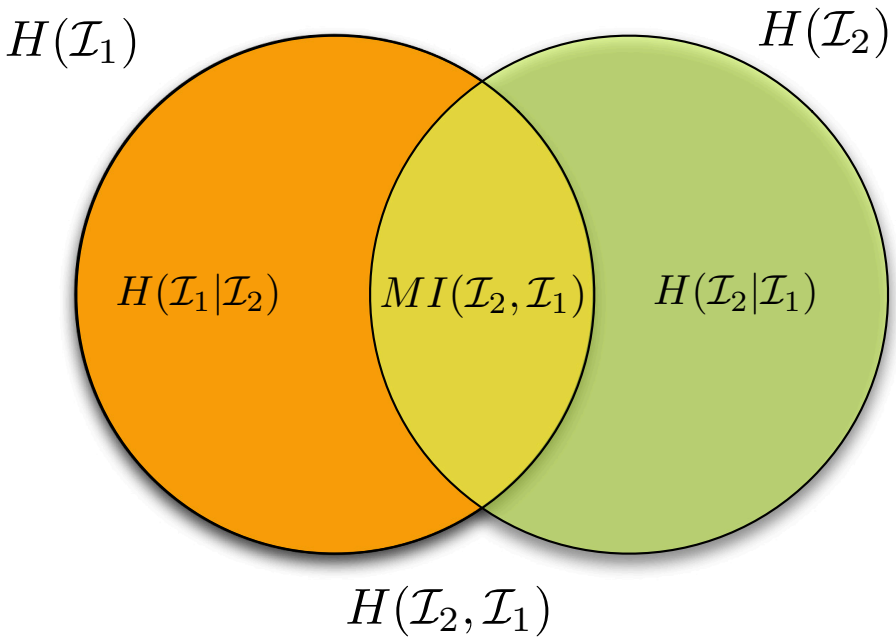


Figure 4.12: Mutual Information - Venn Diagram

alone describe the distribution of intensities within the two images. These two distributions can describe how similar two images are [32] but by themselves cannot tell us anything about the actual alignment. It is the joint entropy, which allows MI to be used as a measure of image alignment as well as similarity and is defined by

$$H(\mathcal{I},\mathcal{I}^*) = -\sum_{n=0}^{N}\sum_{m=0}^{N} p_{\mathcal{I}\mathcal{I}^*}(n,m) log(p_{\mathcal{I}\mathcal{I}^*}(n,m)) \tag{4.17}$$

where $p_{\mathcal{I}\mathcal{I}^*}(n,m) = p(\mathcal{I}(x) = n, \mathcal{I}^*(x) = m)$ is the joint probability of intensity co-occurrences in both images. The analytical form of Mutual Information can then be written as

$$MI(\mathcal{I},\mathcal{I}^*) = \sum_{n=0}^{N}\sum_{m=0}^{N} p_{\mathcal{I}\mathcal{I}^*}(n,m) log(\frac{p_{\mathcal{I}\mathcal{I}^*}(n,m)}{p_{\mathcal{I}}(n) p_{\mathcal{I}^*}(m)}) \tag{4.18}$$

Figure 4.13 shows two simplified images consisting of nine pixels each. Here each of the nine pixels can take one of three color values red, green or blue. It should be noted that the $H(\mathcal{I})$ and $H(\mathcal{I}^*)$ are the same for all example images shown here. Row A) shows the ideal case where two images are identical. Here the joint distribution is diagonal as at any pixel location we seen blue in $\mathcal{I}$ we also see blue in $\mathcal{I}^*$ and the

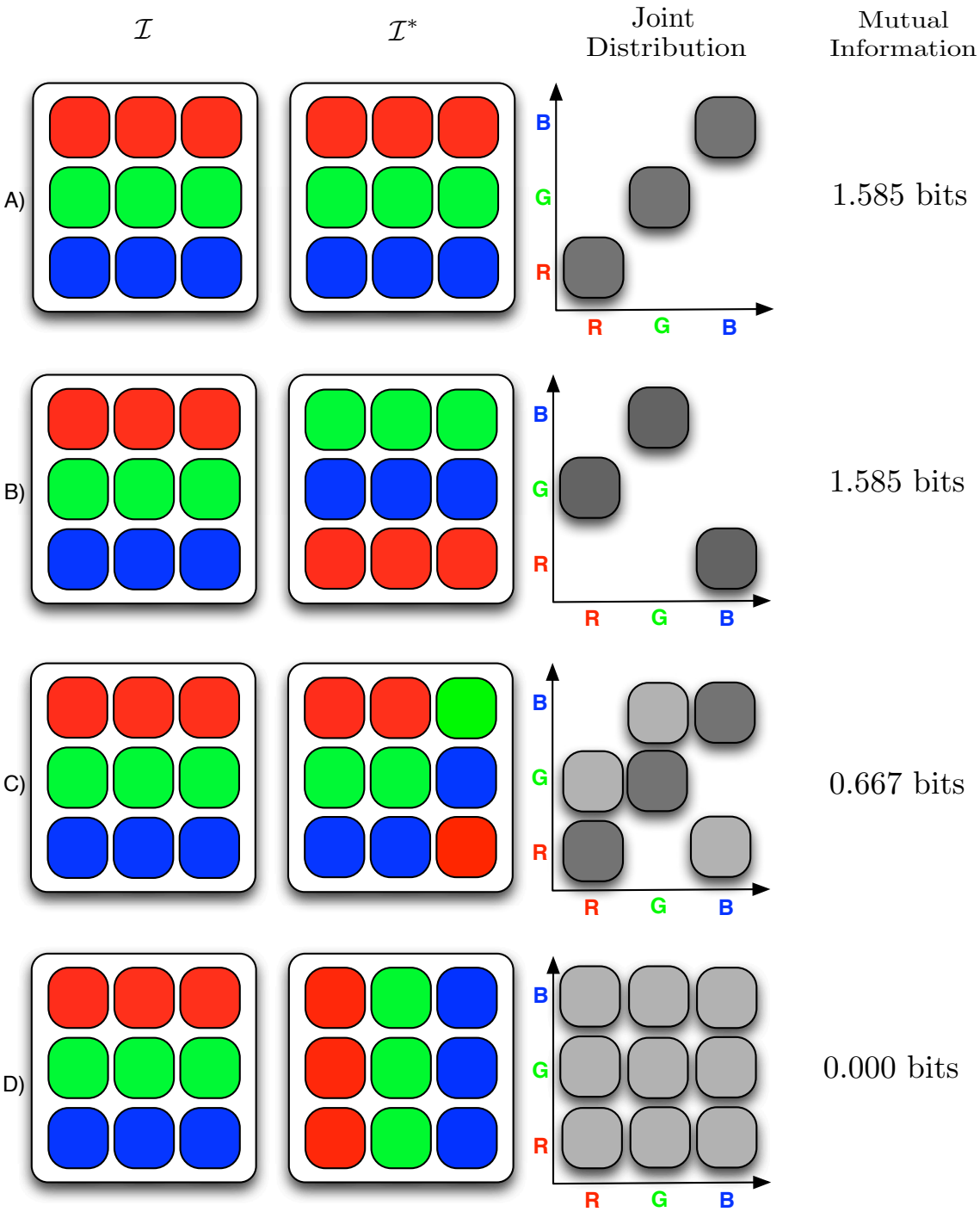


Figure 4.13: A simplified demonstration of how the joint distribution of pixel intensities varies as the two input images change.

same for each of the other colors. This results in the maximum amount of MI. In B) we have permuted all of the colors so where we see blue in $\mathcal{I}$ we see red in $\mathcal{I}^*$ and so on. This permutation of colors does not affect the MI for the image pair as the occurrence of colors is still correlated. In real images this permutation of colors could be caused by shadowing or lighting changes. Row C) shows an example where color in one region of the image are permuted, here there is a drop in MI. Finally, in row D) we have two images with the same overall distributions of colors but no correlation resulting in zero MI.

The Mutual Information between two images is therefore a measure of the co-occurrence of colors or intensities between the images. It is this that gives the measure its lighting invariance as a permutation of colors between images can still result in the maximum possible amount of MI, which is not the case for other measures such as the Sum of Square pixel differences.

4.7.1 Demonstrating Lighting Invariance

In order to evaluate the lighting invariance of Mutual Information we artificially shadowed 1000 images taken using out WildCat platform. Figure 4.14 shows a selection of original images paired along side their artificially shadowed counterparts . In order to apply the shadowing we simply subtract a constant from all of the pixels in the right side of the image. By artificially shadowing images we are then able to register them against themselves knowing ground truth exactly.

Figure 4.15 shows the behavior of Mutual Information and Sum of Squares Differences (SSD) between an original image and its shadowed counterpart. First we evaluate MI and SSD for the original image against itself and change the alignment of the images in both u and v. We then did the same for the original and the shadowed versions. It can be clearly seen that the shadowing has little effect on the MI but leads to significant warping of the SSD.

Figure 4.16 shows the results from 1000 image registration attempts. Here we introduced a random misalignment of between 10 and 20 pixels between the original



Figure 4.14: Example images before and after the artificial shadowing is applied. Here the right half of the image is darkened by 30 points.

	Final Error (pixels)				Number Of Iterations	
	< 0.1	< 1	< 5	> 5	Mean	Std
Mutual Information	97.8%	98.0%	98.1%	1.9%	48.0	4.91
Sum of Squares	0.1%	11.2%	60.0%	40.0%	54.2	7.48

Table 4.1: Convergence statistics for Mutual Information vs Sum of Squares Difference

image and its shadowed counterpart, which corresponds to the expected $10cm$ to$20cm$ camera translation for our camera set up and VO accuracy. A simple optimisation was then used to try and recover this misalignment. Table 4.1 shows that when using MI the image alignment is recovered to within one pixel in 98.0% of cases, however with SSD this was only 11.2%. An interesting result of using SSD for image alignment in the presence of shadowing is also shown in Figure 4.16. The peak for SSD at $(-15, -15)$ shows that the objective function has a tendency to try and push the shadow out of the active region in order to minimise the absolute pixel differences. This was not the case when using Mutual Information. Figure 4.17 shows that when using MI the number of iterations required by the optimiser to align the images is reduced when compared to SSD.

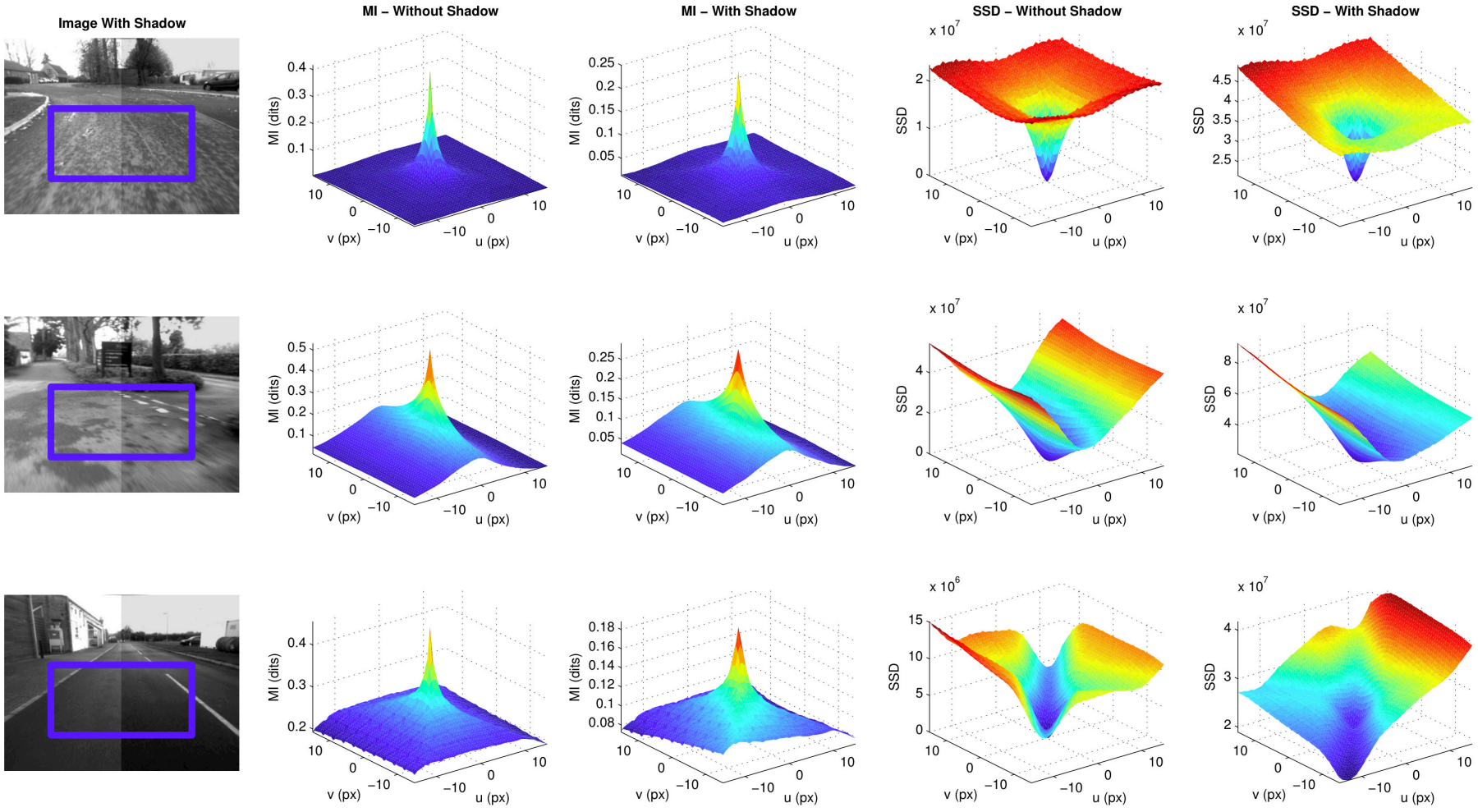


Figure 4.15: The behavior of Mutual Information and Sum of Squares Distance under lighting changes. The first column shows the original image with a shadow cast over the right half; the blue box indicates the region of the image used for evaluation. The second and third columns show the variation in MI without and with shadowing applied respectively. The fourth and fifth columns shows the same for SSD.

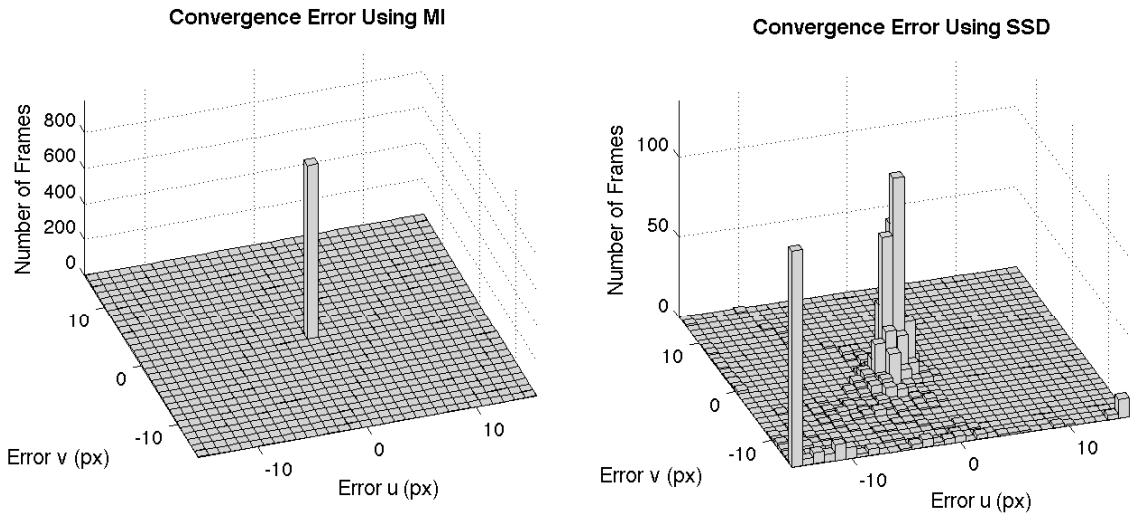


Figure 4.16: The final image registration errors for 1000 frames with artificial shadowing applied, MI (left) and SSD (right). The peak at (-15,-15) for the SSD based alignment measure suggests that the shadow dominates. We suspect this causes the optimiser to diverge as the shadow is pushed out of the active image region.

4.7.2 Sensitivity To Number of Bins in the Joint

An implementation detail when using MI is the width of the bins used to compute the joint distribution of pixel intensities. It was found empirically that evaluating MI over all possible pixel intensity values had little advantage over histogramming intensities into wider bins. Figure 4.18 shows how the Mutual Information changes for various numbers of bins. The more bins used the sharper the peak when the images are correctly aligned. When using a large number of bins the peak is more pronounced due to the fact that a small change in intensity for a particular pixel location will lead to a change in the intensity bin which that pixel location is associated with. Conversely if the number of bins is small the pixel intensity value has to change more significantly for the associated bin to change and consequently the Mutual Information to change. Quantising the intensity values into 32 bins was found to provide a suitable trade-off between the smoothness of the cost function and the sharpness of the peak.

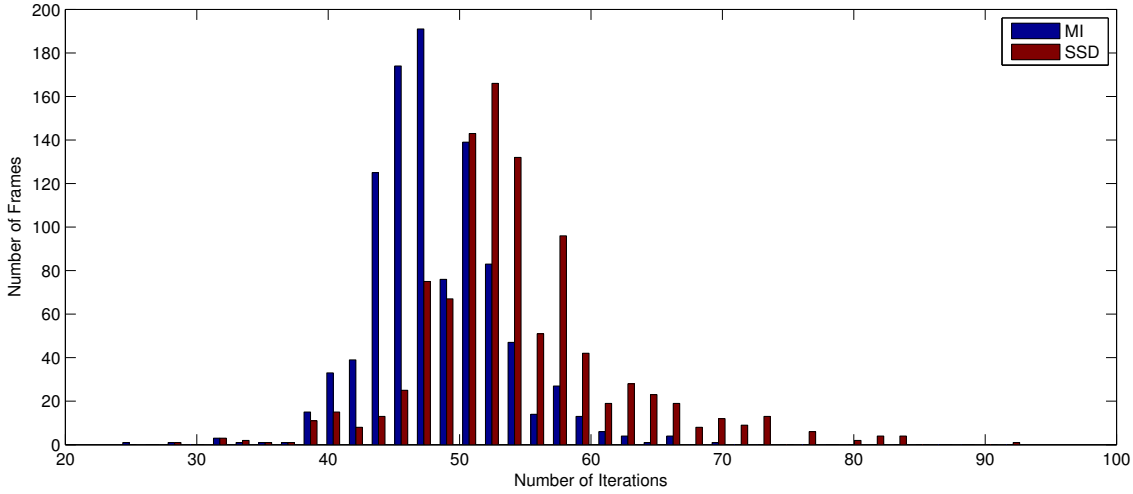


Figure 4.17: The number of iterations taken for image registration form MI and SSD.

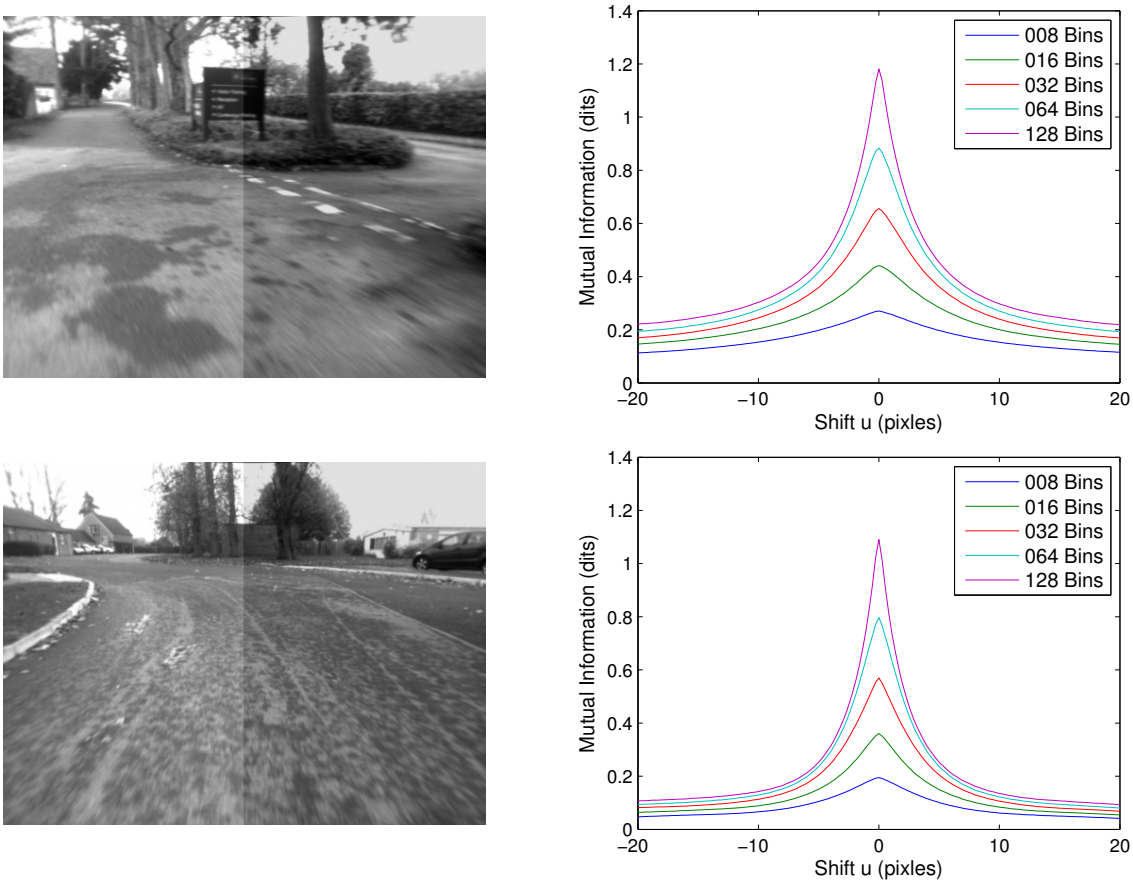


Figure 4.18: Effect of changing the number of bins when computing Mutual Information

4.8 Optimisation

4.8.1 The Objective Function

Now given a live image $\mathcal{I}_k$ and a putative motion estimate v_k we are in a position to hypothesise views of the synthetic orthographic road images $proj(\mathcal{I}_\pi(X_i),^i t_k)$ and measure their alignment using Mutual Information. The problem of estimating our current pose relative to a pose X_i in the survey trajectory $\mathcal{T}$ then reduces to

$$^i\hat{t}_k = \underset{^i t_k}{argmax} \left\{ MI(\mathcal{I}_k, proj(\mathcal{I}_\pi(X_i),^i t_k)) \right\} \tag{4.19}$$

As the Stereo VO has high metric accuracy over small distances $\sim 10m$, the deviation from the initial position estimates $^i t_{ko}$ are relatively small, of the order of centimeters. Two approaches have been implemented for estimating $^i t_k$ the first is an approximate $\mathbb{SE}2$ and the second a full $\mathbb{SE}3$ pose correction.

4.8.2 Approximate $\mathbb{SE}2$ Pose Corrections

Our application domain is road vehicles so in our first approach to reduce complexity and increase speed we confine $^i t_k$ to in road plane motion, reducing our search space to $\mathbb{SE}2$. We still however maintain the $\mathbb{SE}3$ pose information for rolling and pitching during cornering and accelerations respectively. Rather than solve Equation 4.19 iteratively by, for example, using non-linear Gauss Newton methods, we take advantage of the small search radius and use a histogram filter to evaluate an approximation to Equation. 4.19. The histogram filter as used in [58] evaluates the objective function over a predefined grid. The size of this grid is set using statistics on the drift of the VO relative to the INS. Figure 4.19 shows the mean drift with the error bars indicating the standard deviation. The drift statistics were calculated by taking all the sets of ten consecutive poses from the Begbroke trajectory data. The drift against the INS was then computed from frame one to frame ten for each of the sets. The mean and standard deviation of these drift measurements were then calculated using standard techniques.

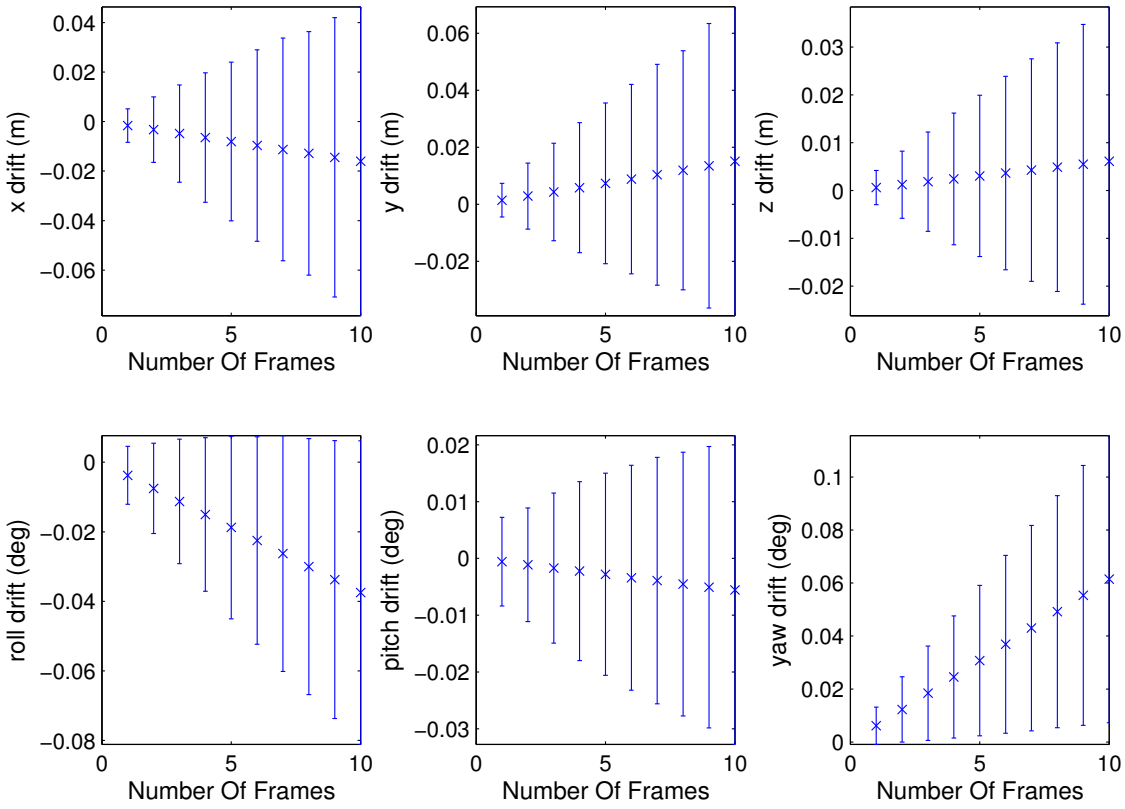


Figure 4.19: VO drift against the number of frames traversed. These graphs show the mean drift of the VO against the INS as the number of frames traversed increases. The error bars indicate one standard deviation assuming a Gaussian distribution. This data is collected from 3.5km of data around our Begbroke test site.

The local accuracy of the VO allows us to define a relatively small and dense grid. Although other search based approaches such as Particle filters [34] are available, this method avoids issues such as a particle occurring near to but actually missing the correct solution. The histogram filter allows us to explicitly choose how precise we want our solution to be and the size of the search region.

The in plane motion approximation has the consequence of reducing the sensitivity of the matching step to high frequency image content, such as fine texture on the tarmac. Very small errors in pitch, roll or height cause misalignment which stops the matching process leveraging this fine detail. As we generally operate in urban environments where the vast majority of roads have distinct road markings this was deemed to be an acceptable trade off for speed and simplicity.

4.8.3 Full $\mathbb{SE}3$ Pose Corrections

Solving for all six degrees of freedom with no in plane motion constraint allows the matching step to leverage much finer detail image content such as texture in the tarmac, as we will show in Section 4.9. Solving for the full $\mathbb{SE}3$ pose corrections will allow the localisation method to be used in less urban environments and should reduce the amount of time in open loop (Section 4.8.4). The need for the consensus stage, as described in Section 4.8.4, may also be reduced as the matching process should cope with a wider variety of road surfaces.

Here we use a line search optimisation technique [31]. Our aim is to maximise the objective function defined in equation 4.19 with respect to the pose ${}^{i}t_{k}$. In order to explain the optimisation procedure we shall switch notation and define our objective function as $F(x) \in \mathbb{R}^1$, our variable as $x \in \mathbb{R}^6$ and our target pose as $\hat{x}$. Using the pose seed from the VO, x_o, we approximate the gradient using a central difference method, $\tilde{G}(x)$. This defines a search direction which we traverse along to find a maximum.

$$x_{n+1} = x_n + \alpha_n \tilde{G}(x_n) \tag{4.20}$$

$$F(x_{n+1}) > F(x_n) \tag{4.21}$$

where $\alpha_n \in \mathbb{R}^1$ is a scalar representing a step length found by stepping along the line $\tilde{G}(x)$, subject to Equation 4.21. Computing the gradient and searching along it to find α_n constitutes one iteration of the optimiser. The process is repeated until one or more of the following criterion is met;

- Mimimum step size - α_n
- Minimum change in the objective function - $\Delta F(x)$
- Minimum magnitude of the gradient - $\left|\tilde{G}(x_n)\right|$

The objective function itself should be agnostic to the optimisation algorithm. However, the line search approach described here proved to be sufficient.

With the current MATLAB implementation, solving for $\mathbb{SE}3$ pose corrections runs at approximately $1Hz$ on a single core. As we operate in a relative framework and only use the latest frame at each step, the algorithm is constant time. The process of computing $\tilde{G}(x)$ and searching along the gradient is highly parallelisable and when running on multiple cores the system can provide corrections at approximately $5Hz$. There is also preliminary work on an OpenCL implementation which has demonstrated two orders of magnitude speed up. With this limitation on the speed of the system, we provide corrections every fifth frame. Figure 4.19 shows the average and standard deviation of VO drift against the INS from one to ten frames. It can be seen that at five frames the standard deviation in pose is $\sim 4cm$ and ~ 0.06 degrees. This level of drift is acceptable between corrections as it is equivalent to the accuracy of the INS when operating under optimal conditions.

4.8.4 Outlier Rejection

The localisation process can fall into local minima if there is insufficient 'texture' in the images, for example if there are no road markings. This can lead to errors in our estimations of ${}^{i}\hat{t}_k$ which has the effect of pulling the trajectory off course. In order to avoid this, we first we compute the difference between the optimiser output ${}^{i}\hat{t}_k$ and the initialisation from the VO ${}^{i}t_{ko}$.

$$e = {}^{i}\hat{t}_k \ominus {}^{i}t_{ko} \tag{4.22}$$

We define a threshold on e based on the drift statistics of the VO. As it would be undesirable to invoke the outlier rejection unnecessarily we set this threshold for a 1 in 1000 event. The standard deviation of VO drift over five frames is $4cm$ in translation and 0.7^o in rotation, see Figure 4.19. Assuming a Gaussian distribution, a 1 in 1000 event occurs at 3.29σ which gives us a threshold of $\sim 15cm$ and 0.21^o. If e is greater than this threshold we invoke the right camera, perform localisation on

$\mathcal{I}_{k,right}$ and check for consensus. If the pose estimates from both $\mathcal{I}_{k,right}$ and $\mathcal{I}_k$ are commensurate we adopt the pose correction into the trajectory. If the pose estimates do not agree we ignore the match and simply adopt ${}^{i}t_{ko}$ into the trajectory. We repeat this until matching can be reestablished. In essence when the image matching step fails the system falls back to raw VO and runs open loop.

4.9 Results

We evaluate the performance of our localisation system over four data sets with varying levels of difficulty. The four test sites expose the algorithm to conditions ranging from vanilla loops of our private test site at Begbroke to more challenging conditions such as extreme lighting changes and unmarked sub-urban roads. The data sets cover more than 7km and consist of more than 60,000 images.

The data was collected using our autonomous vehicle platform the 'WildCat' (see Figure 4.20). This modified Bowler WildCat is equipped with a multitude of sensors and 32 processor cores available for computation. For the results presented here we use the front mounted Point Gray Bumblebee2, running at 640×480 at $20fps$. The stereo VO runs at frame rate on a single core.

4.9.1 Performance Evaluation

4.9.1.1 Ground Truth

As discussed in Section 2.3.2, we use an Inertial Navigation System (INS) with Differential Corrections (DGPS) to obtain a measure of the vehicle's ground truth trajectory. In order to evaluate the accuracy of our localisation system we compute the Cross Track Error (CTE) between trajectories from the survey and subsequent traversals produced by our system and the INS. If the INS produced a perfect measure of vehicle pose the CTE would effectively be the difference in road position between two traversals. The level agreement between the CTE computed from the output of our localisation system and the INS should therefore give us a measure of our

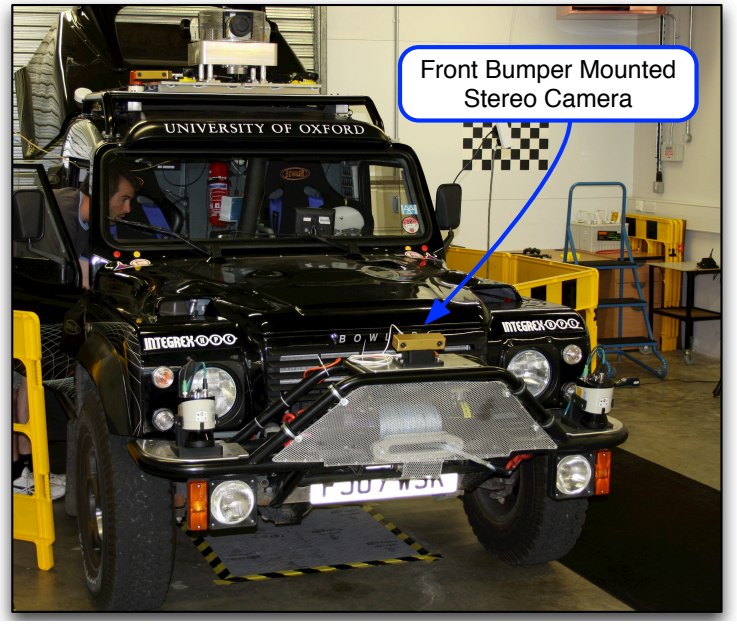


Figure 4.20: The Bowler Wildcat is one of MRG's robotics platforms. It is a 4x4 all terrain vehicle supporting state of the art computing and sensors, including Stereo Vision used in this work.

system's accuracy. In reality the accuracy of the INS can vary from two centimeters to several meters. The CTE therefore bakes in errors from the INS as well as the difference in road position, limiting our measure of system accuracy to that of the INS.

Even with this limitation the CTE is still a useful measure of localisation accuracy to the order of meters. The CTE can also offer a sanity check on the localisation output as the difference in road position is bound by the driving lane width. Therefore, if the CTE between a survey and a subsequent traversal is greater than even half the lane width then we can infer that the localisation has most likely failed.

As a further sanity check on the performance of our system, we compare the distributions of the corrections made by our system to the disparity between the VO and the INS. Although these two quantities are not directly comparable they should be of a similar magnitude and distribution.

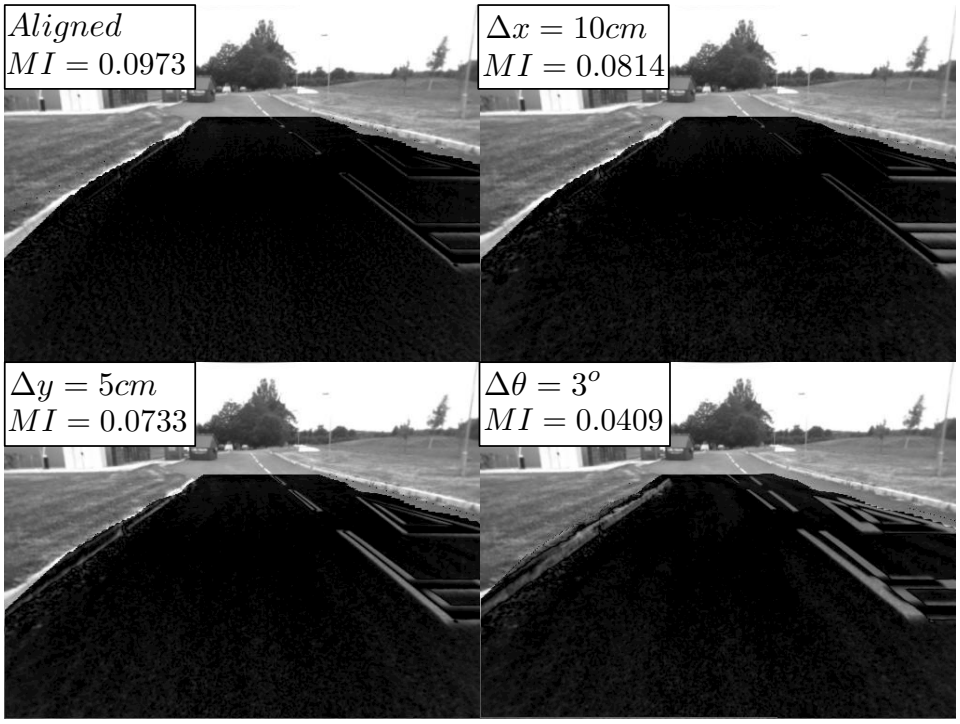


Figure 4.21: Top left shows the difference in image intensities between a driver view $\mathcal{I}_k$ correctly localised relative to the survey trajectory and the hypothosised view $proj(\mathcal{I}_\pi(X_i), {}^i t_k)$ at ${}^i t_k$ (Note white pixels correspond to larger differences in intensity). Top right to bottom right show the effect of artificially perturbing the pose by $\Delta x = 10cm, \Delta y = 5cm, \Delta\theta = 5^0$ respectively. It can be seen there is a significant misalignment and corresponding drop in Mutual Information.

4.9.1.2 Demonstrating Centimeter Precision

As we do not have centimeter accurate ground truth we qualitatively demonstrate the high precision of the localisation system by comparing localised images $\mathcal{I}_k$ and the corresponding artificially perturbed hypothosised views $proj(\mathcal{I}_\pi(X_i), {}^i t_k \oplus \epsilon)$, where ϵ is a perturbation of the order of centimeters. Figure 4.21 shows that by artificially perturbing the vehicle pose from its true value we can see significant image misalignment and a corresponding drop in MI. For each data set we select several example frames to demonstrate the alignment of the live images $\mathcal{I}_k$ with the synthesised images. The alignment of these images suggests that the system is localising to centimeter accuracy.

4.9.1.3 Optimiser Convergence

We demonstrate the convexity of our Mutual Information based objective function for all datasets used. We select several frames from each dataset and plot the objective function around the estimated pose for all six degrees of freedom. It is not possible to plot a six-degree of freedom objective function so each dimension is plotted individually.

We demonstrate the convergence behavior of the optimiser by plotting histograms of the number of iterations taken for each frame.

4.9.2 Approximate $\mathbb{SE}2$ Pose Corrections

We present a condensed set of results for the $\mathbb{SE}2$ pose corrections. Figure 4.22.(a) shows the $650m$ survey trajectory (blue) overlaid with a subsequent traversal localised against the survey in (green). It should be noted that although the subsequent traversal is commensurate with the survey trajectory they should not align perfectly. This is because the survey vehicle may have driven at different positions on the road. The pose distance from the survey trajectory (CTE) should therefore vary, but be bounded as shown in Figure 4.22.(b). Figure 4.23 shows the alignment between live images and the synthetic images at various points along the trajectory.

The system when using $\mathbb{SE}2$ pose corrections was able to successfully localise against the survey for approximately 70% of frames. In Figure 4.22.(a) we indicate where localisation fails, as per Section 4.8.4, with a red cross. In these cases the matches are rejected and the system falls back to using raw VO until a frame can be successfully re-localised. These failures tend to be in areas where there is only fine texture on the road surface and an absence of road markings, see example images in Figure 4.22.(a). Small errors in the degrees of freedom (z,$roll$, $pitch$) not being accounted for in the $\mathbb{SE}2$ approximation stop the optimiser leveraging these fine details and the matching process fails. We will show in Section 4.9.3 how accounting for all 6 degrees of freedom allows the system to leverage even the fine details on the road surface and successfully localise.

The system using the $\mathbb{SE}2$ approximation successfully completed a loop of Begbroke without any manual re-localisation. It is however, undesirable for a road vehicle's trajectory to be uncorrected for several meters where the drift could be enough to put the vehicle on a pavement. We therefore do not present any further results for this method and move onto evaluating the full $\mathbb{SE}3$ solution.

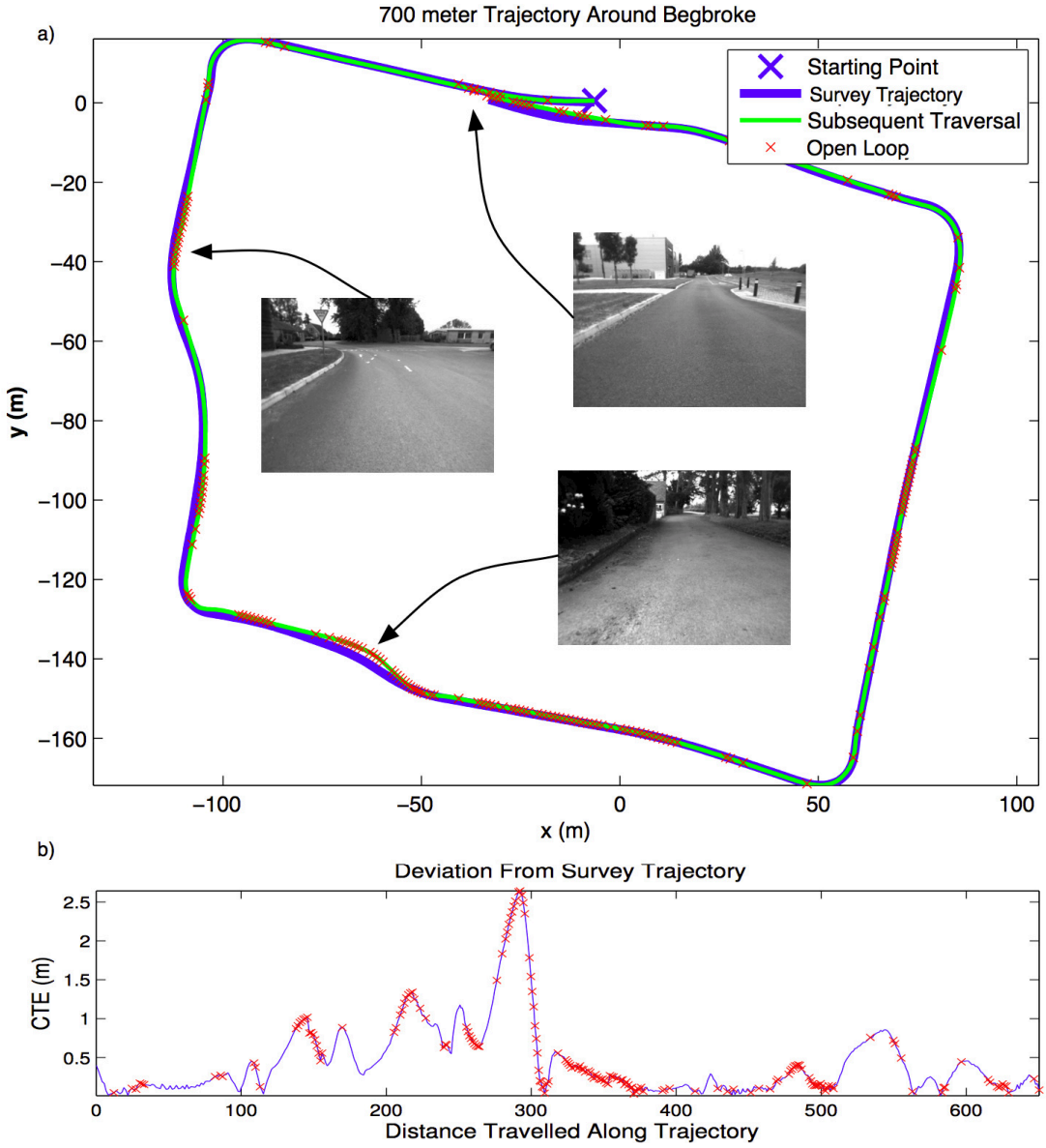
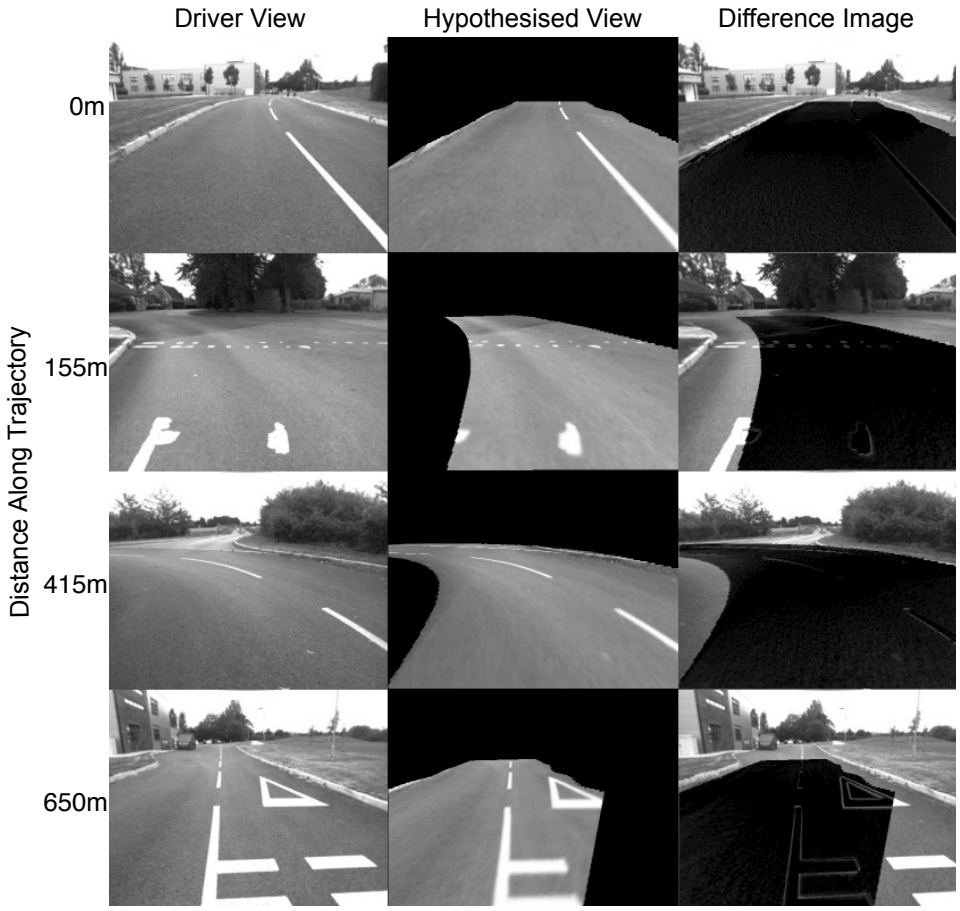


Figure 4.22: (a) Route around the Begbroke site, approximately 650m loop approximated $\mathbb{SE}2$ corrections, (b) localised pose distances from survey trajectory. The red crosses indicate frames where the localisation has failed and the system is running in open loop using the raw VO poses. It can be seen at around 300m there is a significant stretch open loop trajectory and the CTE increases to almost 2.5m, this is then corrected for when a successful match is established.



4.23

Figure 4.23: Live images $\mathcal{I}_k$ and associated hypothosised views $proj(\mathcal{I}_\pi(X_i), {}^i\hat{t}_k)$ along the trajectory shown in Figure 4.22. The image differences demonstrate cm level alignment at various points along the 650m trajectory with reference to Figure 4.21.

4.9.3 Full $\mathbb{SE}3$ Pose Corrections

Here we present a more comprehensive evaluation for the full $\mathbb{SE}3$ solution. Each dataset is analysed separately with overall statistics given on page 126 in Tables 4.2, 4.3 and 4.4.

4.9.3.1 Begbroke Loops

The Begbroke Loops data set consists of multiple loops of our field center. The trajectory is approximately 650m long and covers various types of road surface. Figure 4.24 shows the variation in the appearance of the road surface as well as the variation in Mutual Information over the route. It can be seen that in areas where

the road is marked the Mutual Information is highest and in areas where there is less distinct visual texture the Mutual Information is reduced.

The system successfully localised over 3km worth of data with only one failure 380m into the fifth traversal (discussed in Section 4.10.3). Solving for the full $\mathbb{SE}3$ pose allowed the system to leverage fine detail on the road surface and successfully localise in areas where the $\mathbb{SE}2$ approximation failed. Figure 4.25 shows the localised trajectories relative to the survey route. It can be seen that the trajectories are commensurate and the drift associated with raw VO is not apparent. The actual CTE is shown in Figure 4.26 it can be seen that the CTE reported by our system is significantly less than the INS. This suggests that our system is reporting pose more accurately than the INS as care was taken to drive the same route. It should be noted that the vehicle did not successfully receive DGPS corrections over the course of this data set. As such, the ground truth suffers from the drift associated with standard GPS. Figure 4.27 shows five example images chosen from areas with high disagreement between our system and the INS. The alignment of these example images with their localised synthetic views further suggests that our system is reporting the vehicle pose to a higher degree of accuracy than the INS. The example frames cover the various types of road surface contained within the data set. It can be seen that the objective function (third row) has a distinct peak around the optimum and has a convergence basin of more than 30cm and 20 degrees. Figure 4.28 shows the corrections applied to the raw VO by our system are bounded within 10cm and 0.1 degree, well within the convergence basin. The figure also shows that the corrections are commensurate with the disparity between the raw VO and the INS. Although these two quantities are not directly comparable it is encouraging that they are of a similar magnitude and distribution.

Figure 4.27 (fourth row) shows the behavior of the optimiser for the example frames chosen and Figure 4.29 shows the number of iterations taken for convergence over all frames. Over 60% of frames converged within three iterations with less than 5% requiring more than 10 iterations.

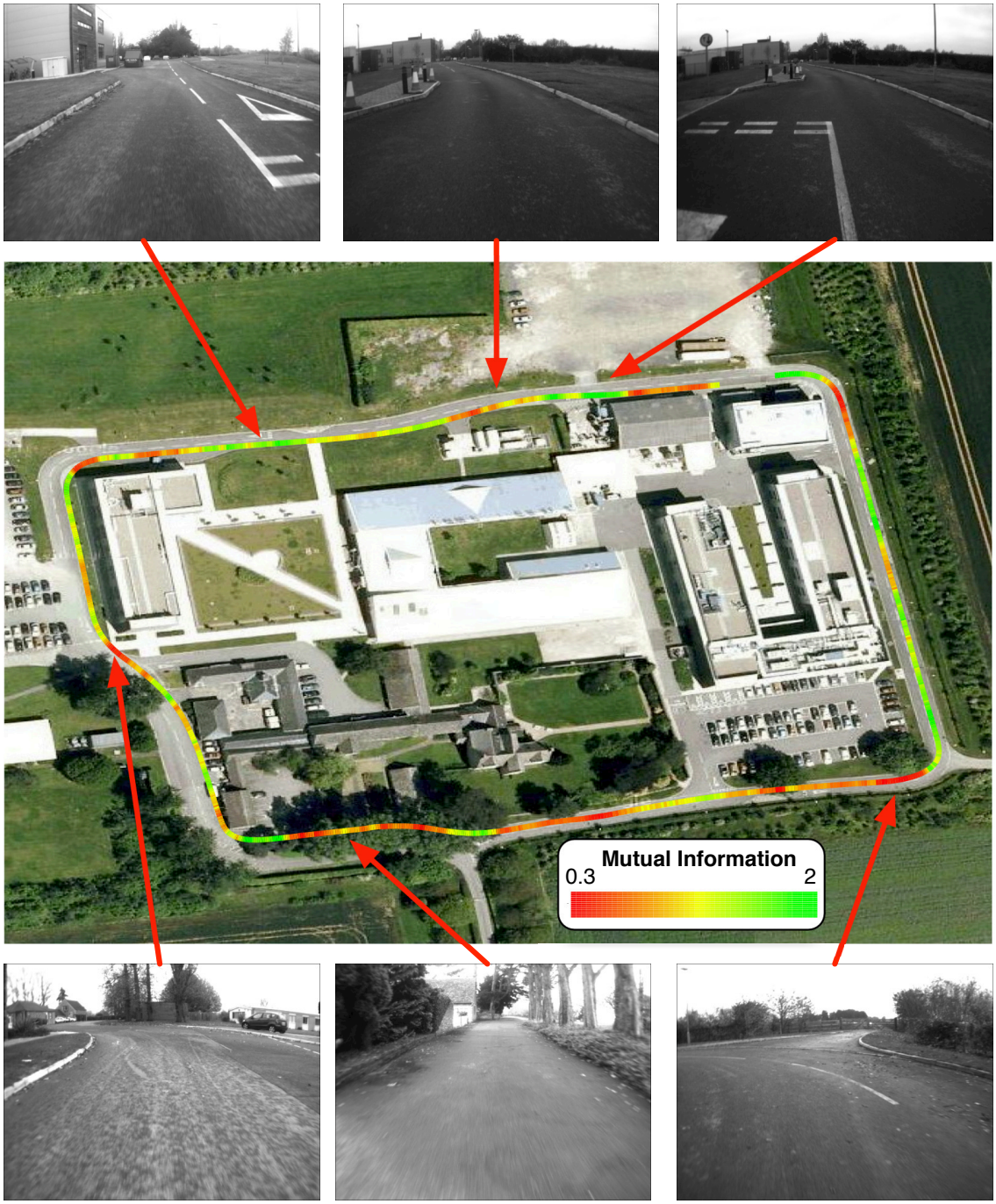


Figure 4.24: Mutual Information over the Begbroke Loop. The heat map shows how the Mutual Information between the synthesised road images and the live stream varies over the trajectory. The example frames show where there is little texture on the road surface the Mutual Information is low and where there more texture and road markings the Mutual Information is higher.

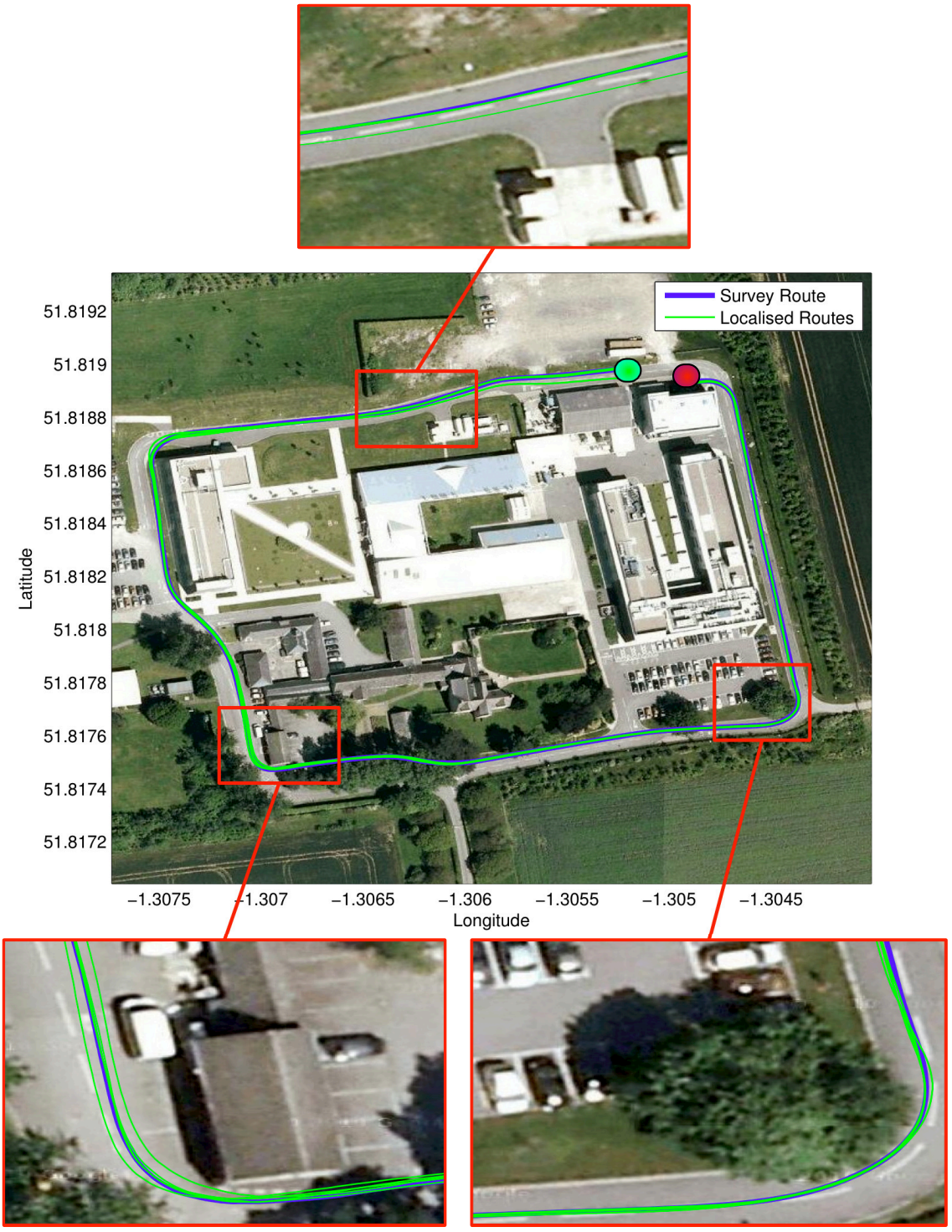


Figure 4.25: Five laps (3km) of the Begbroke loop (green) localised against a single survey. The central image shows the survey trajectory in blue with the start and end points indicated with a green and red dot respectively. The peripheral images show magnified segments of the trajectories.

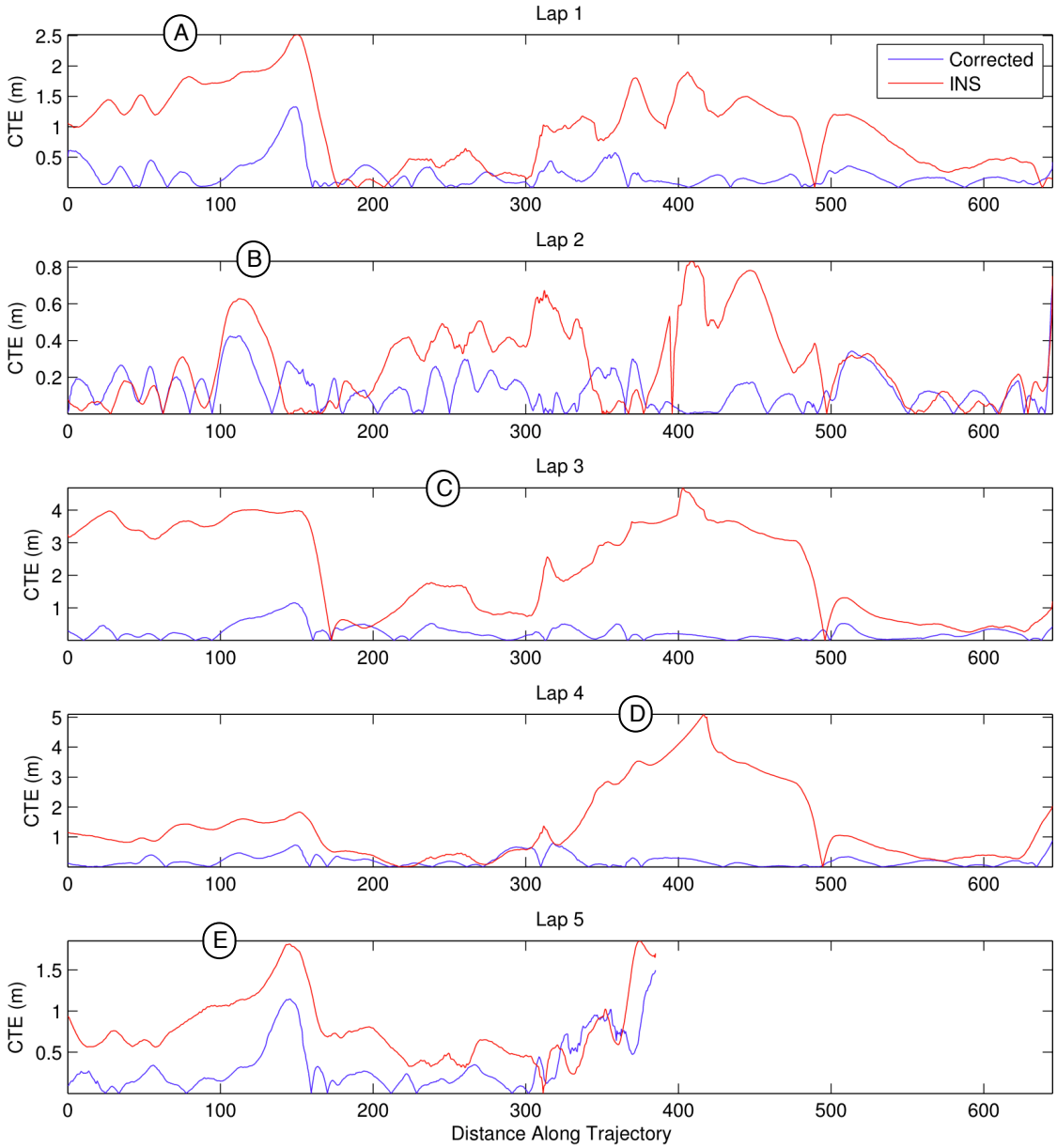


Figure 4.26: The cross track error (CTE) between the survey trajectory and five subsequent traversals. The CTE is shown for the INS (red) and our localisation system (blue). It can be seen that our system reports a significantly lower cross track error than the INS (DGPS reception was poor). Four full laps were successfully localised with a failure on the fifth due to insufficient road texture. Example frames for the labeled positions A,B,C,D and E can be seen in Figure 4.27. The point of failure is discussed in Section 4.10.3

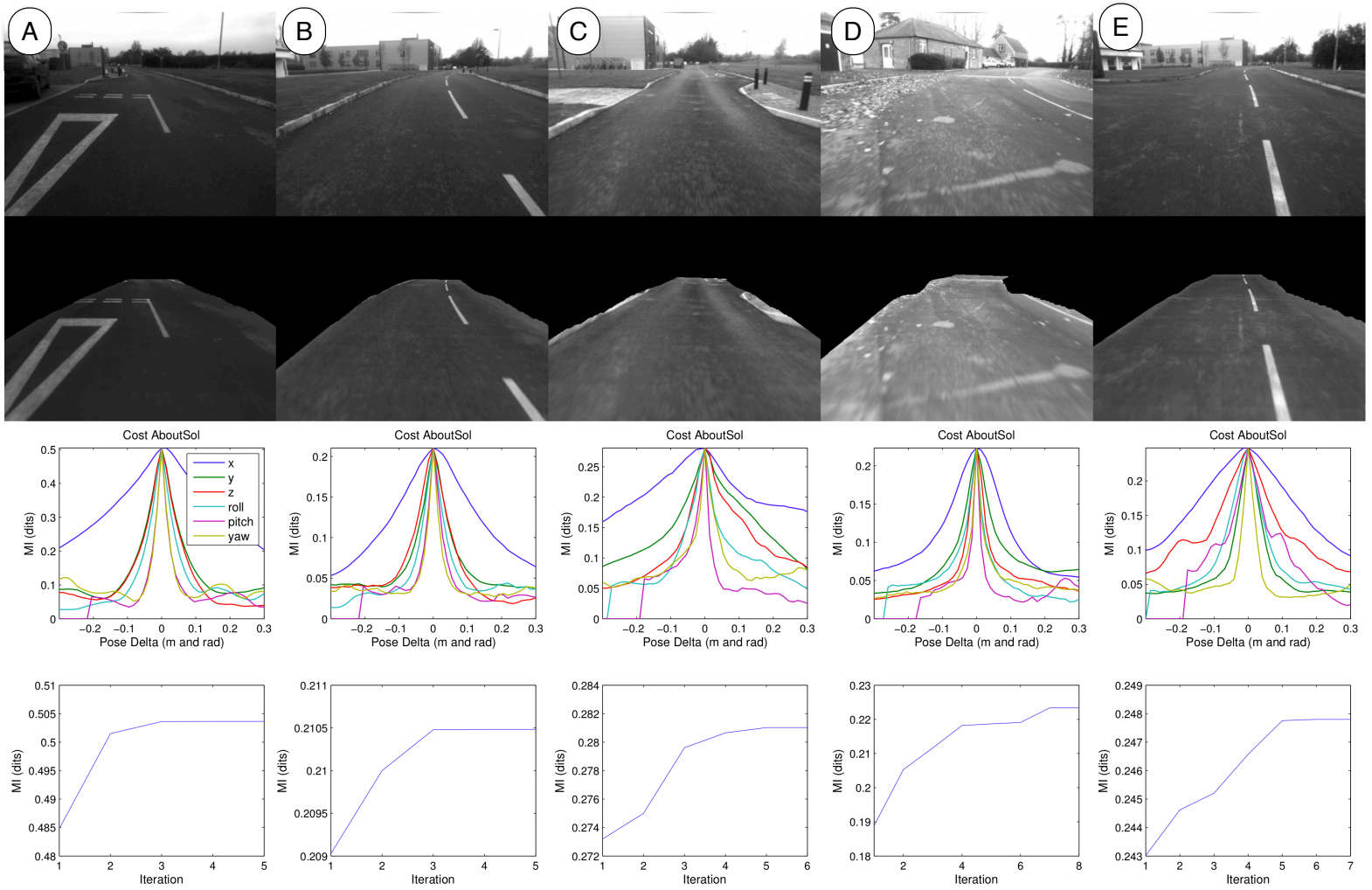


Figure 4.27: Example frames from the labeled positions in Figure 4.26. The top row shows the live images and the second shows the synthetic road images from the localisation output. The Mutual Information about the localised pose is show in the third row and the fourth row shows how the Mutual Information is increased for each iteration of the optimiser.

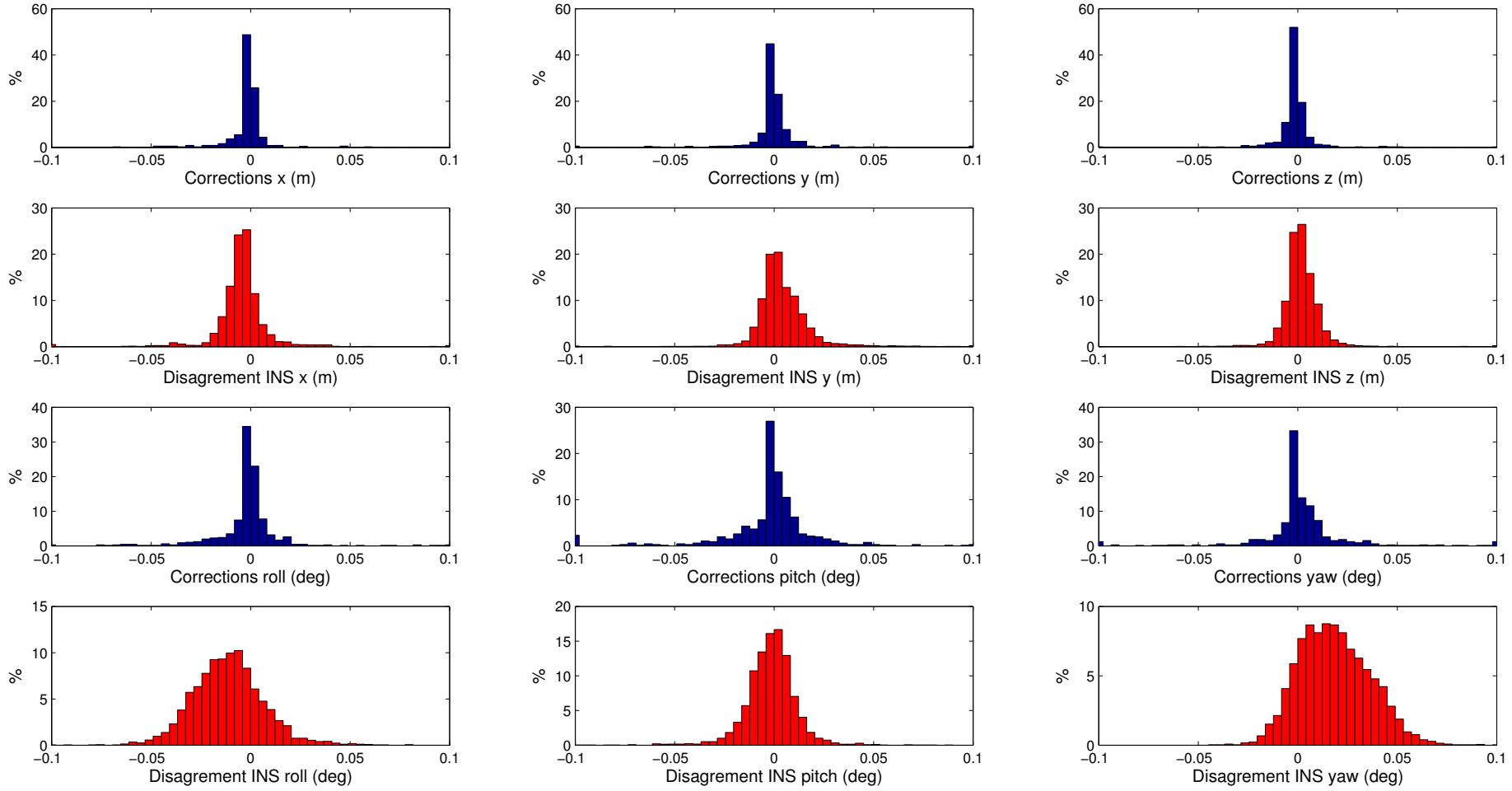


Figure 4.28: Histograms of the corrections applied to the VO trajectory by our system (blue) compared to the disparity between the raw VO and the INS system (red) for the Begbroke Loops data set.

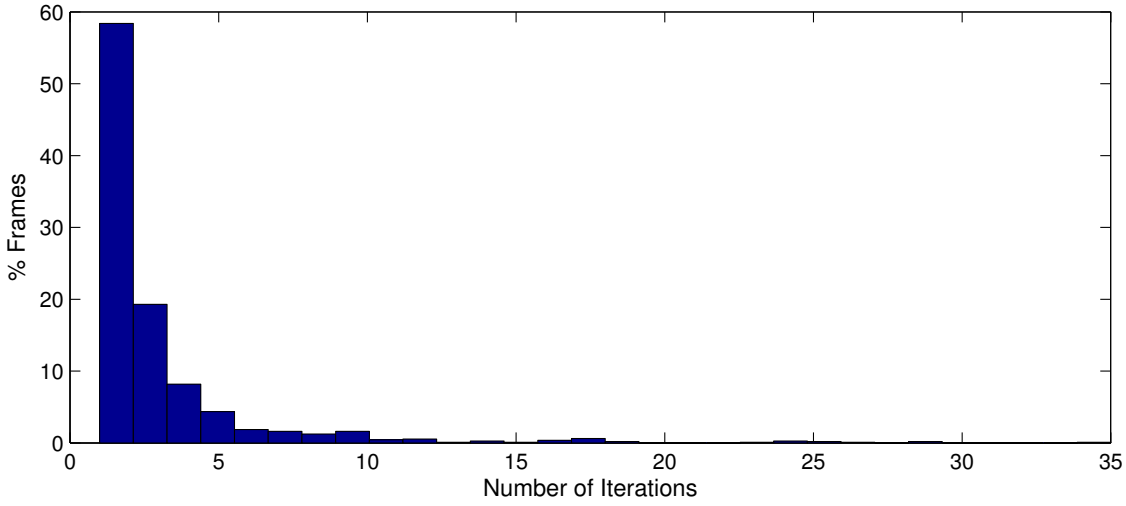


Figure 4.29: A histogram of the number of iterations required for the optimiser to converge on the Begbroke Loops data set. It can be seen that approximately 60% of frames converged within three iterations.

4.9.3.2 Begbroke New Road

The Begbroke New Road dataset consists of seven $300m$ runs of a site access road at our field centre. This data demonstrates the system's ability to operate on new roads where there is very little texture on the road surface other than road markings. The dataset starts with a sharp 90 degrees turn out of a junction, followed by a traversal of the main access road. It should be noted that as this is a new road no aerial imagery is available for this dataset.

Figure 4.30 shows the localised trajectory against the six subsequent traversals. Here we have included the raw VO to show how even slight initial misalignments can quickly compound when sharp turns are made. Figure 4.31 shows the CTE over the trajectories; here the INS was receiving DGPS corrections leading to tighter agreement of the two systems. Example frames from areas of high disagreement are shown in Figure 4.32 again our system appears to be producing *cm* aligned synthetic images. Figure 4.33 shows the corrections made to the VO and Figure 4.34 shows the behaviour of the optimiser.

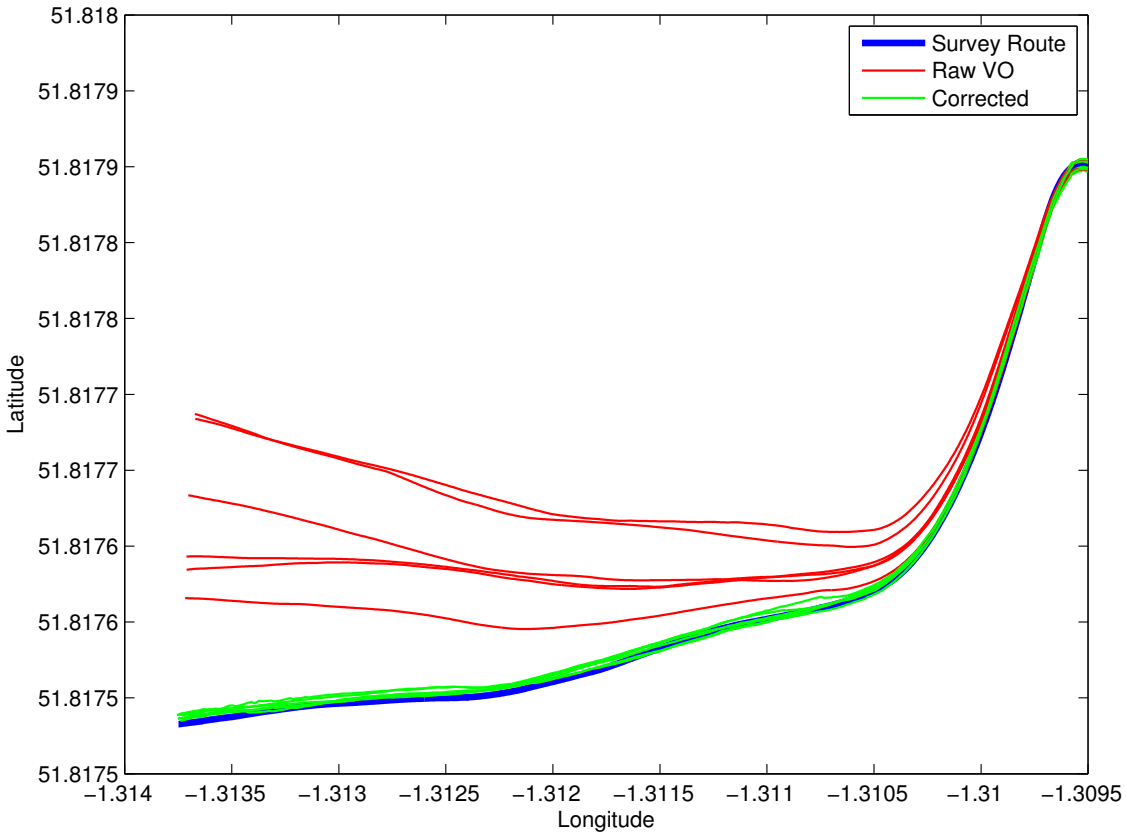


Figure 4.30: Localised Trajectories for Begroke New Road segments. Six runs of the Begbroke New Road (1.8km) localised against a single survey. The survey route (green) is plotted against the output of our system (green) and raw VO (red). Note this road was built in 2012 so no overhead imagery is available

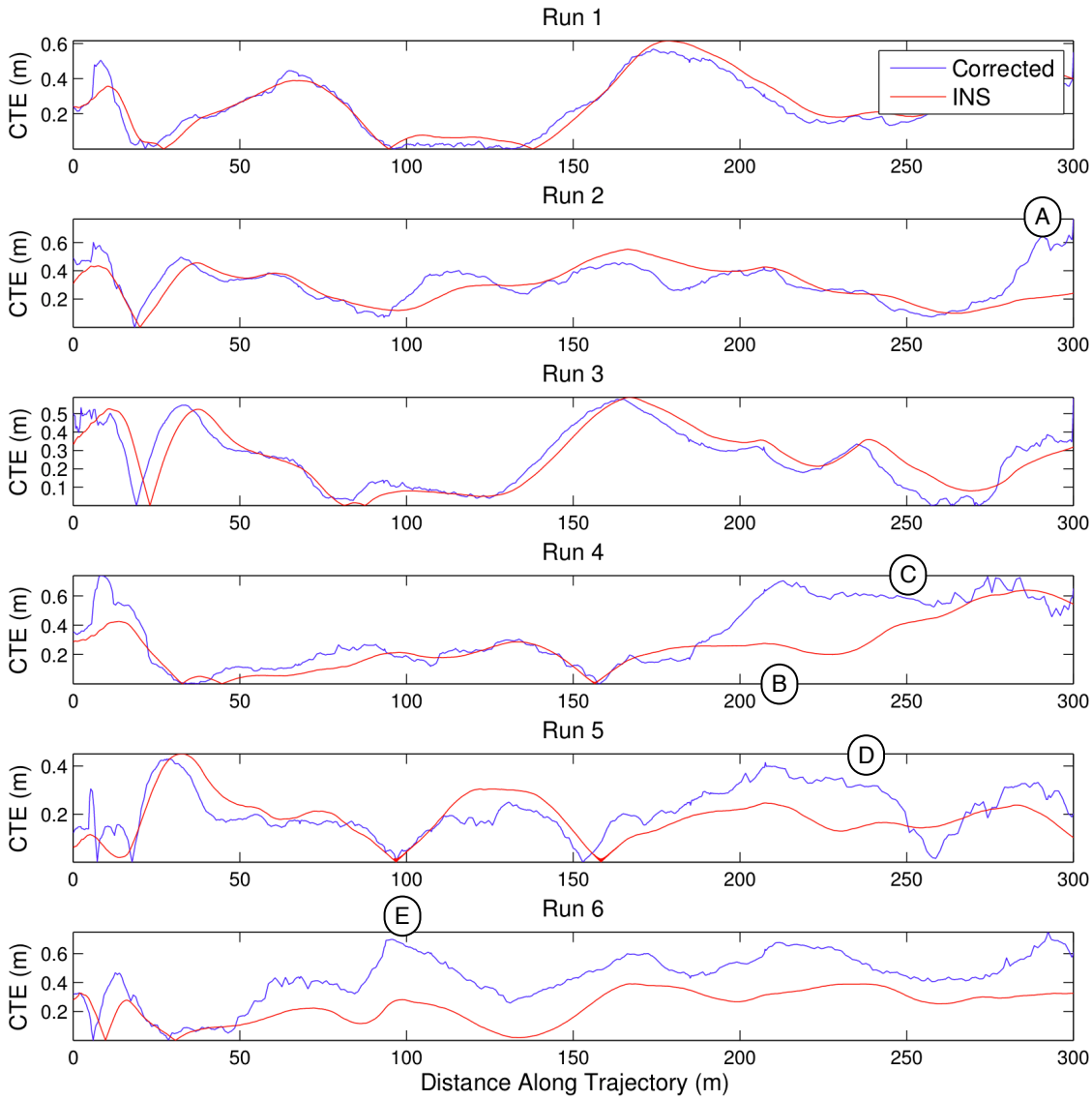


Figure 4.31: The cross track error (CTE) between the survey trajectory and six subsequent traversals. The CTE is shown for the INS (red) and our localisation system (blue). It can be seen that our system reports a CTE commensurate with the INS. Example frames for the labeled positions A,B,C,D and E can be seen in Figure 4.32.

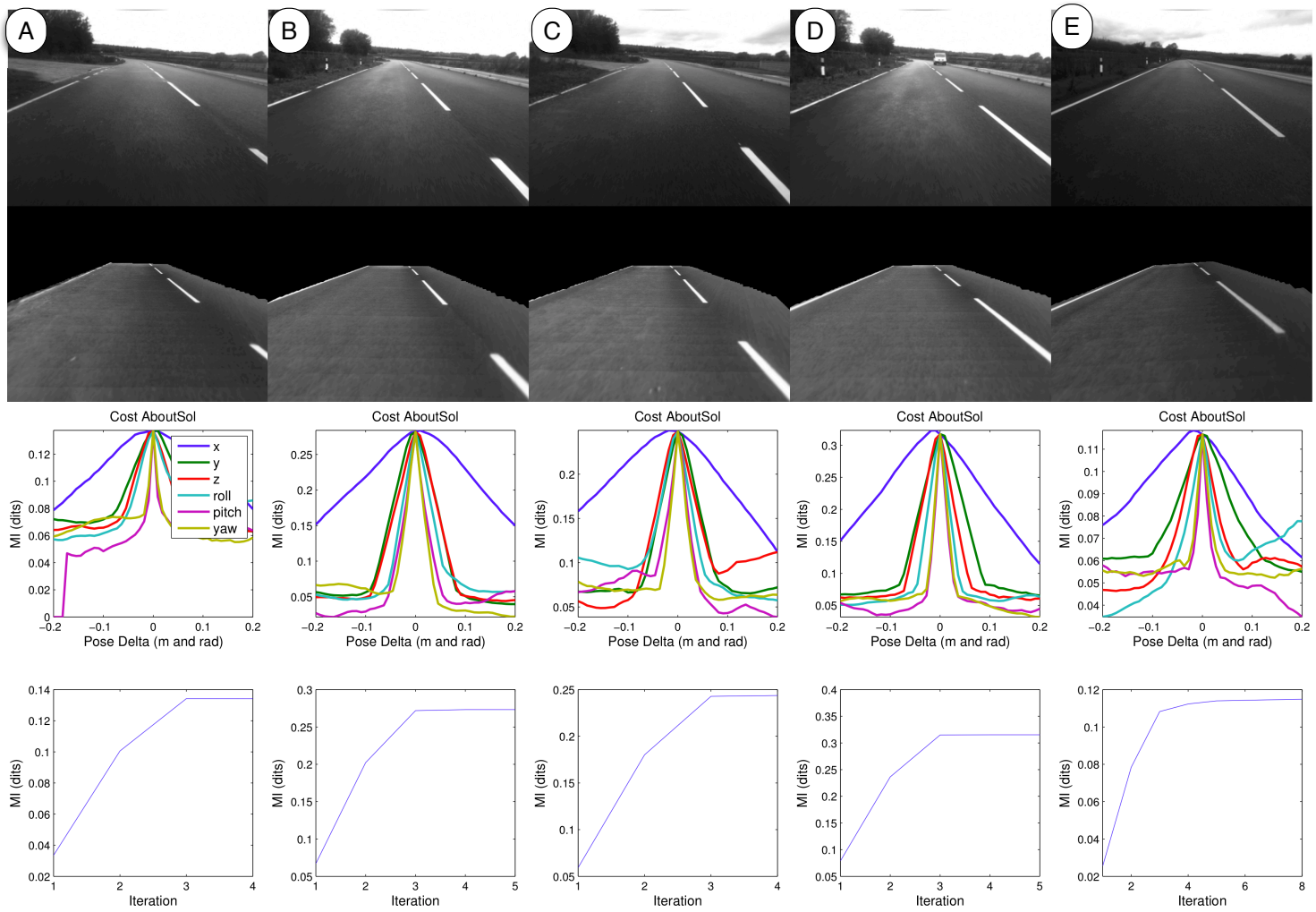


Figure 4.32: Example frames from the labeled positions in Figure 4.31. The top row shows the live images and the second shows the synthetic road images from the localisation output. The Mutual Information about the localised pose is show in the third row and the fourth row shows how the Mutual Information is increased for each iteration of the optimiser.

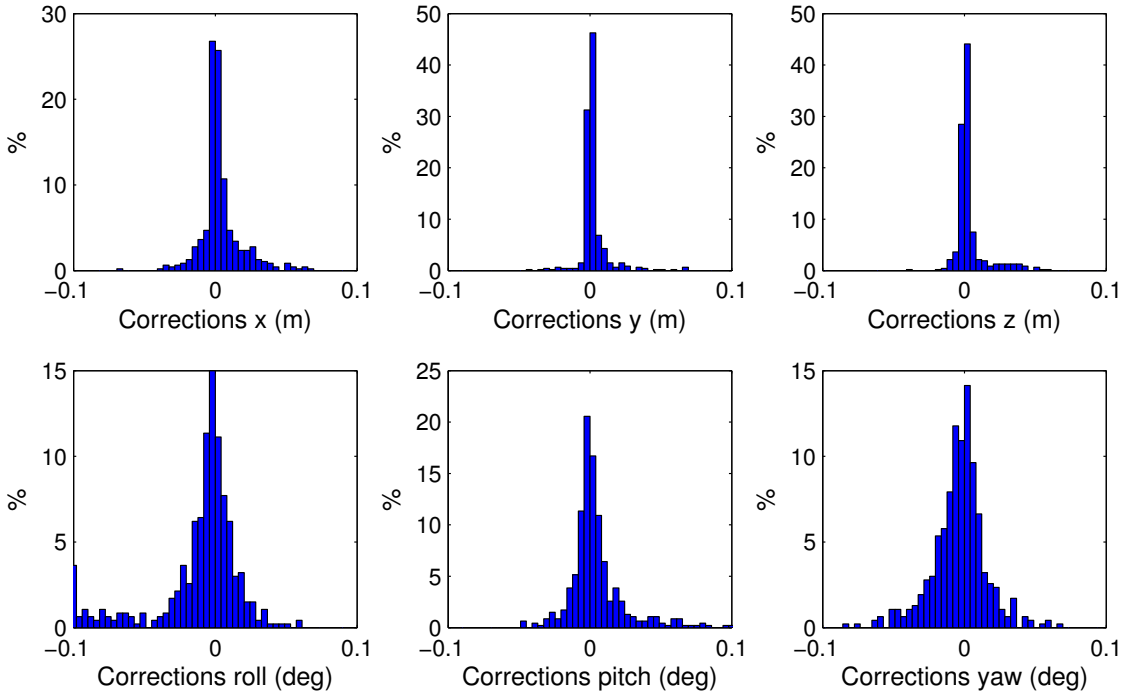


Figure 4.33: Histograms of the corrections applied to the VO trajectory by our system for the Begbroke New Road data set. Note we have omitted the INS disparity as it is unchanged from Figure 4.28.

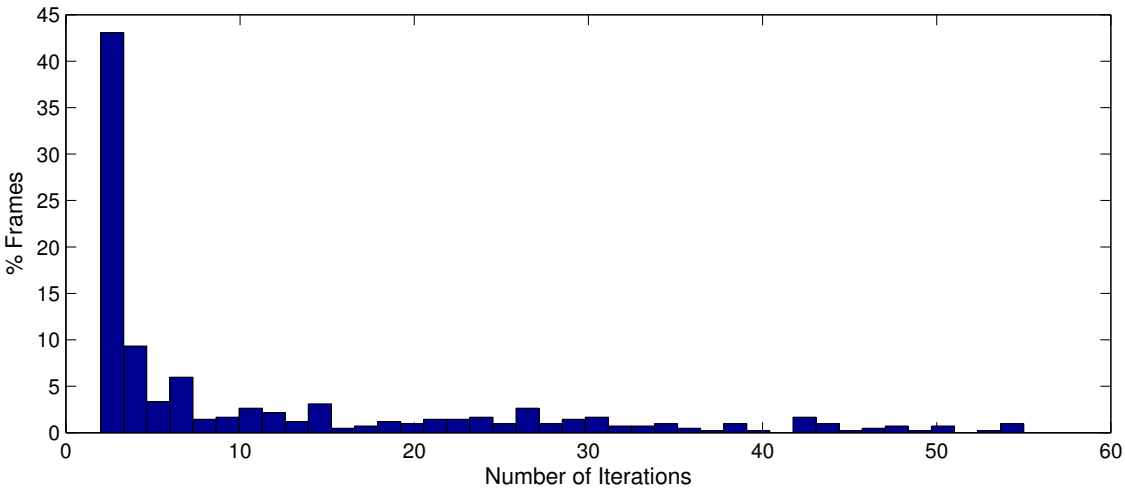


Figure 4.34: A histogram of the number of iterations required for the optimiser to converge on the Begbroke New Road data set. It can be seen that approximately 60% of frames converged within three iterations.

4.9.3.3 Kidlington - Limited Road Markings

The Kidlington dataset consists of a 255m run of a more challenging type of suburban road surface. Over the course of the trajectory there are very limited and badly faded

road markings. Here the system is only able to exploit the texture in the tarmac and other objects such as drains and manholes to localise.

Figure 4.35 shows the trajectory overlaid on an aerial image and Figure 4.36 shows the cross track error. Again we show example frames and their associated synthetic road image and objective function in Figure 4.37. It can be seen that in areas where there is little texture on the road surface the peak in the objective function is maintained, however the sharpness is increased and the overall objective functions smoothness is reduced. Even with this reduced convergence basin the system was able to successfully localise over the entire 255m trajectory.

Figure 4.38 shows the corrections made to the VO and Figure 4.39 shows the behaviour of the optimiser.

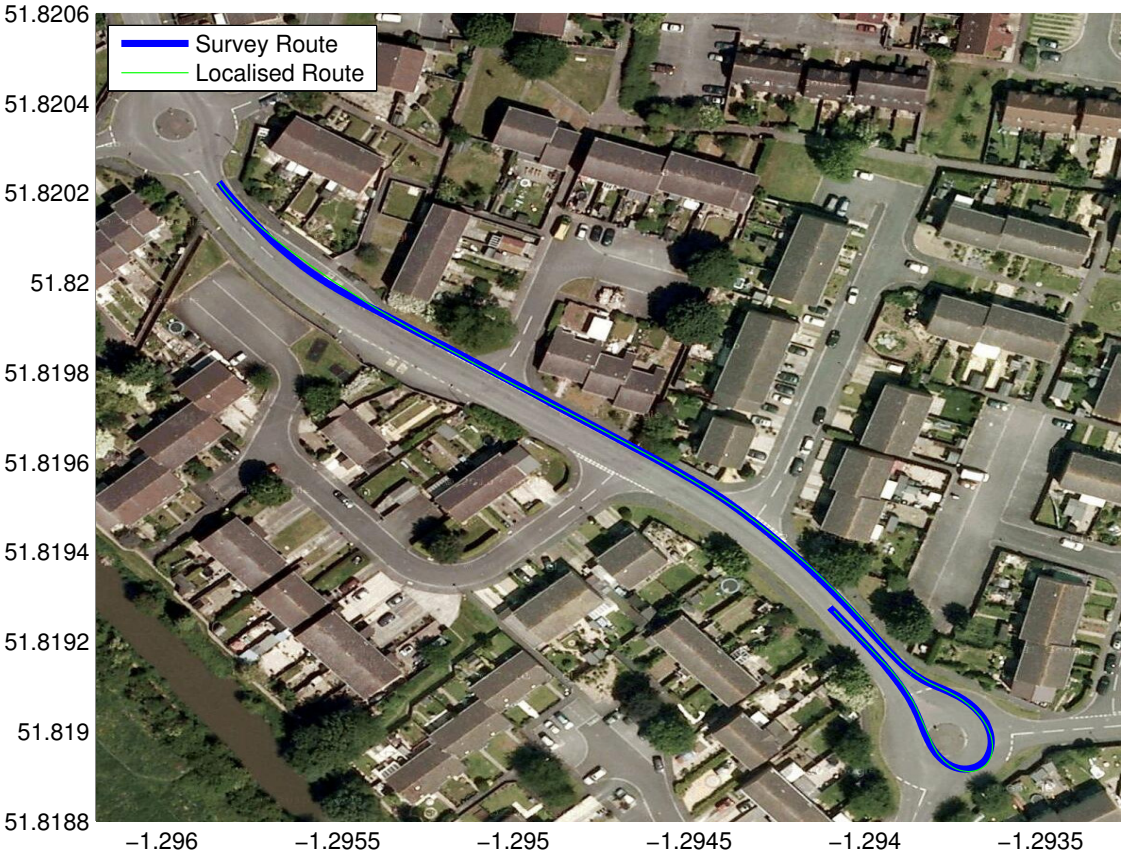


Figure 4.35: Localised Trajectories for Kiglington trajectory. A 255m run without road markings localised against a survey. The survey route (green) is plotted against the output of our system (green).

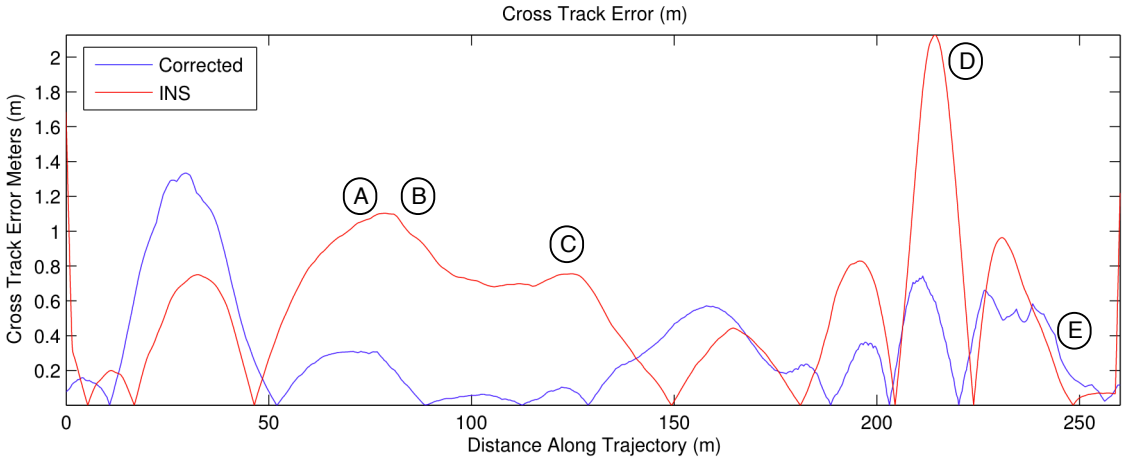


Figure 4.36: The cross track error (CTE) between the survey trajectory and a subsequent traversal. The CTE is shown for the INS (red) and our localisation system (blue). It can be seen that our system reports a CTE commensurate with the INS. Example frames for the labeled positions A,B,C,D and E can be seen in Figure 4.37.

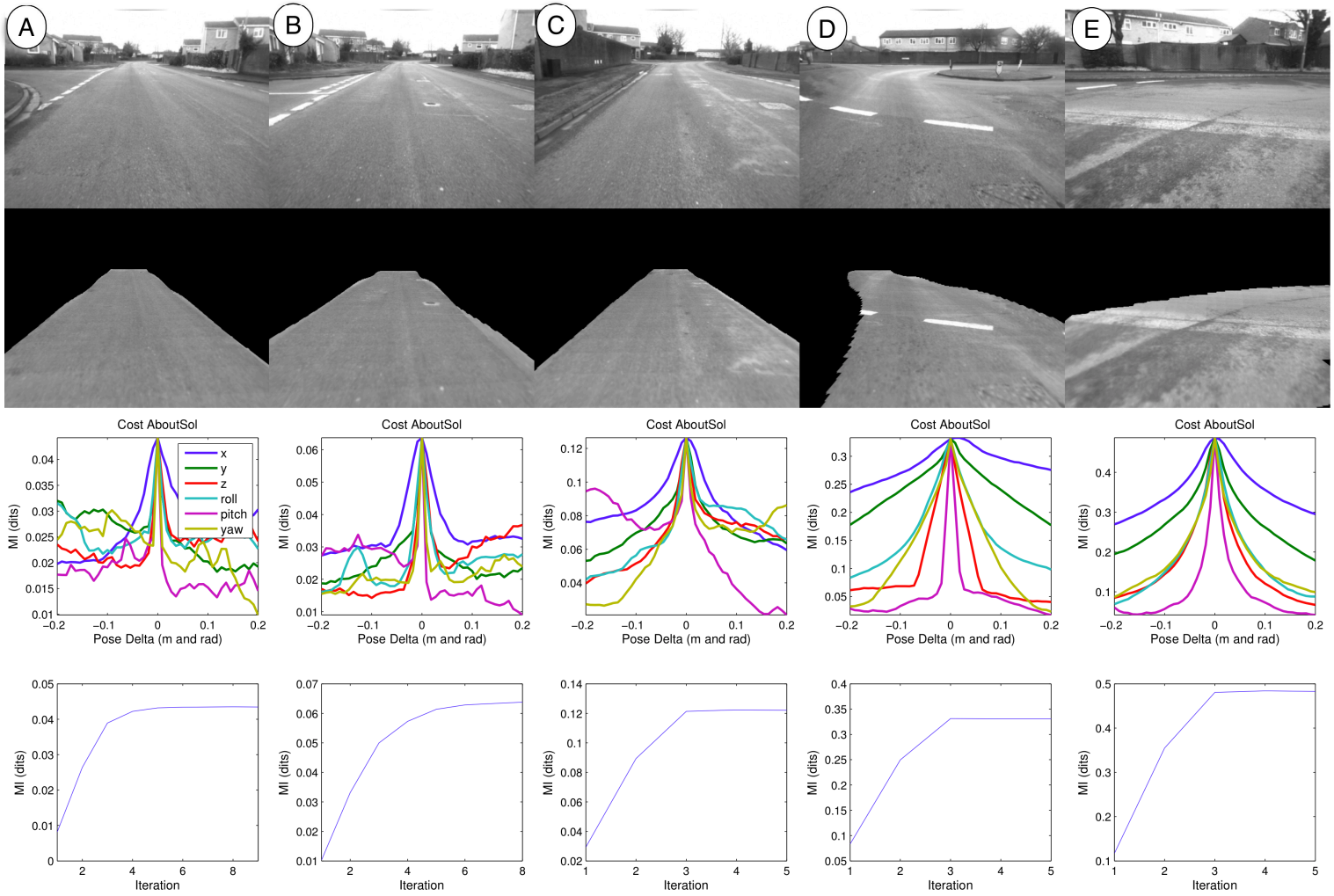


Figure 4.37: Example frames from the labeled positions in Figure 4.36. The top row shows the live images and the second shows the synthetic road images from the localisation output. The Mutual Information about the localised pose is show in the third row and the fourth row shows how the Mutual Information is increased for each iteration of the optimiser.

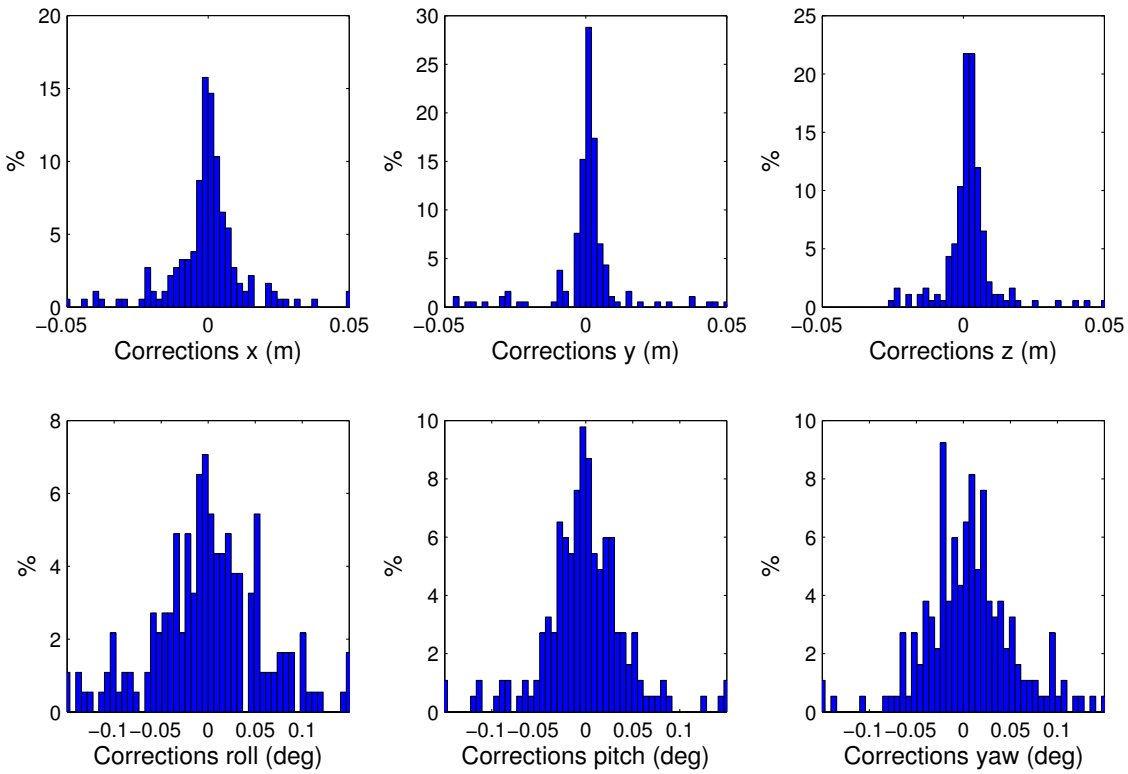


Figure 4.38: Histograms of the corrections applied to the VO trajectory by our system for the Kidlington data set. Note we have omitted the INS disparity as it is unchanged from Figure 4.28.

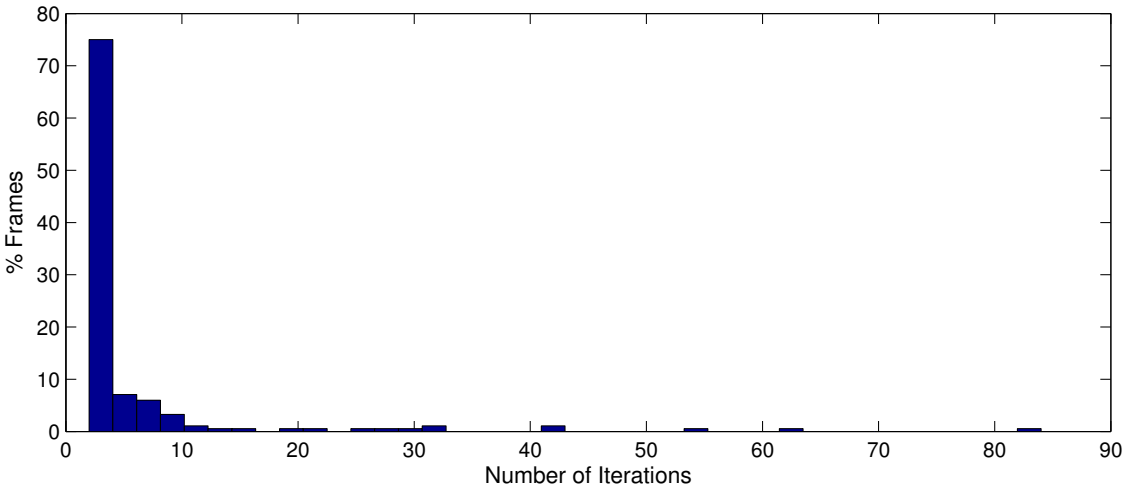


Figure 4.39: A histogram of the number of iterations required for the optimiser to converge on the Kidlington data set. It can be seen that approximately 70% of frames converged within five iterations.

4.9.3.4 Woodstock - Severe Lighting Changes

The Woodstock dataset consists of a 500m run of urban roads with extreme lighting variations and oncoming traffic. The survey route was taken under optimal lighting conditions, minimising shadows and lighting artefacts. The subsequent traversal was captured during a period with high shadowing and glare with oncoming traffic.

Figure 4.40 shows the trajectory overlaid on an aerial image and Figure 4.41 shows the cross track error. Again we show example frames and their associated synthetic road image and objective function in Figure 4.42. The example frames show how the system behaves with shadows cast on the road surface, oncoming traffic as well as glare. It can be seen from frames A,B and C that the shadowing has little effect on the objective function. However, the combination of glare, oncoming traffic and shadowing causes local minima to develop in D. Even under these circumstances the system was able to successfully localise.

Figure 4.43 shows the corrections made to the VO and Figure 4.44 shows the number of iterations taken by the optimiser.



Figure 4.40: Localised Trajectories for Woodstock trajectory. A 500m run under extreme lighting changes localised against a survey. The survey route (green) is plotted against the output of our system (green).

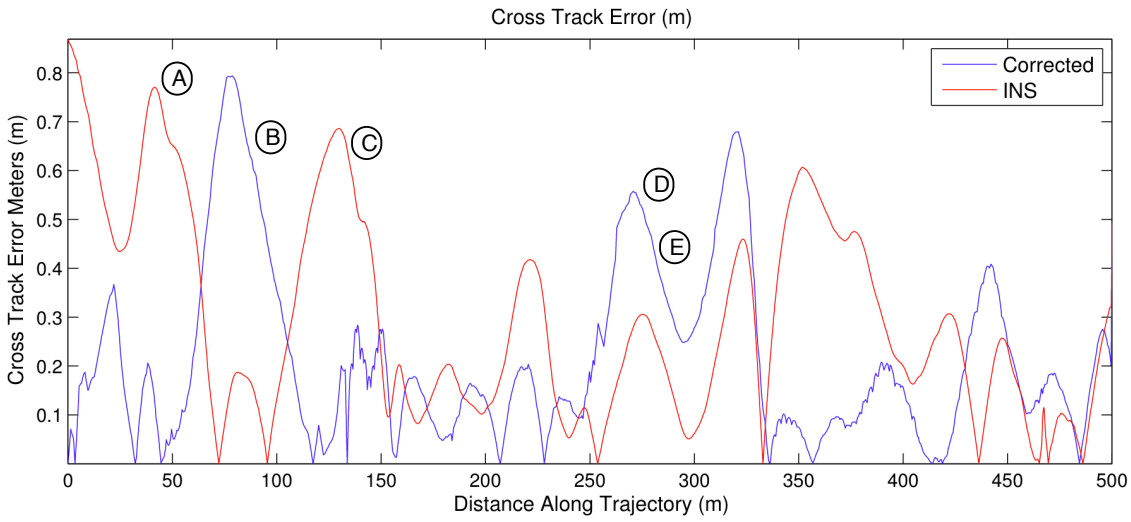


Figure 4.41: The cross track error (CTE) between the survey trajectory and a subsequent traversal. The CTE is shown for the INS (red) and our localisation system (blue). It can be seen that our system reports a CTE commensurate with the INS. Example frames for the labeled positions A,B,C,D and E can be seen in Figure 4.42.

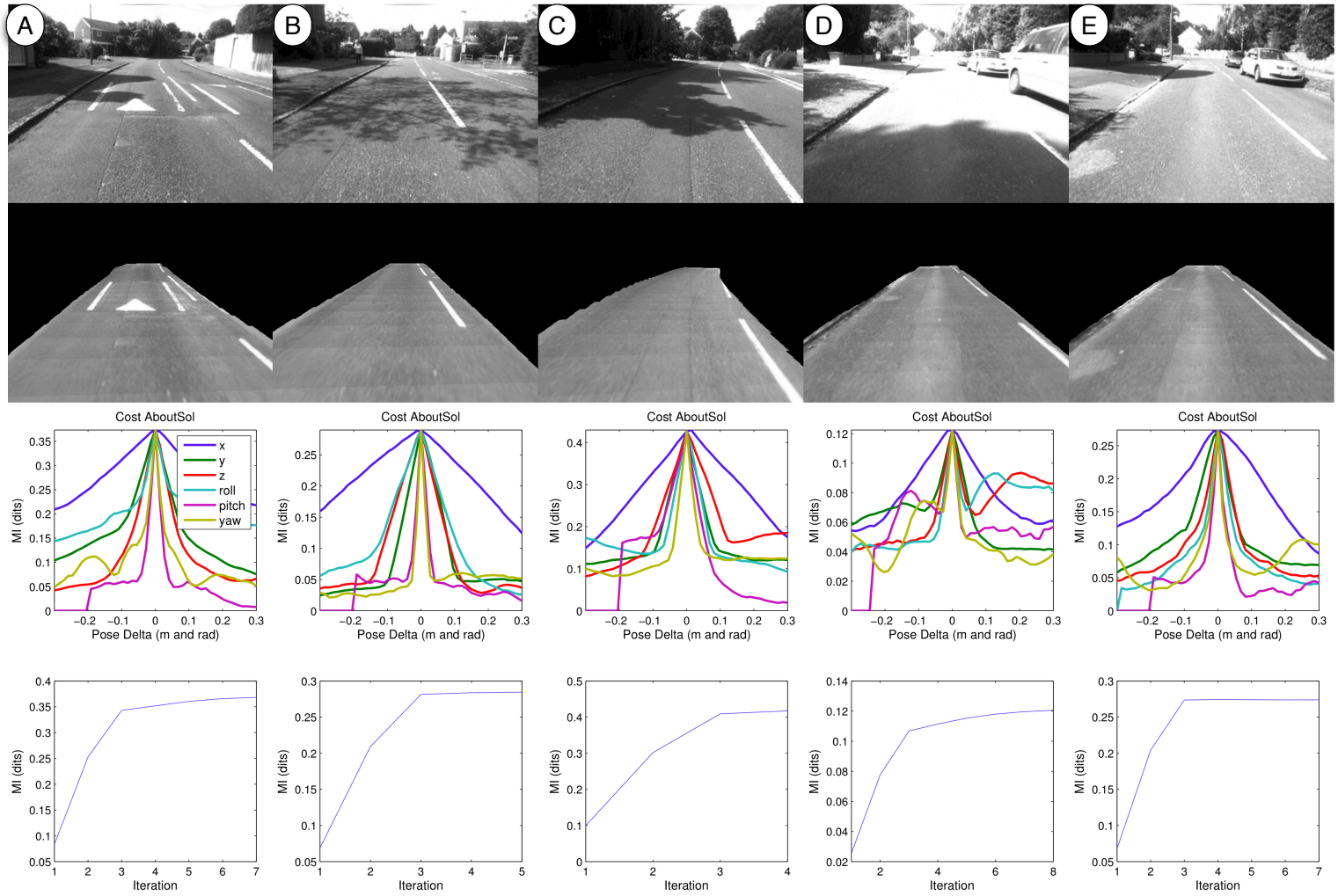


Figure 4.42: Example frames from the labelled positions in Figure 4.41. The top row shows the live images and the second shows the synthetic road images from the localisation output. The Mutual Information about the localised pose is show in the third row and the fourth row shows how the Mutual Information is increased for each iteration of the optimiser.

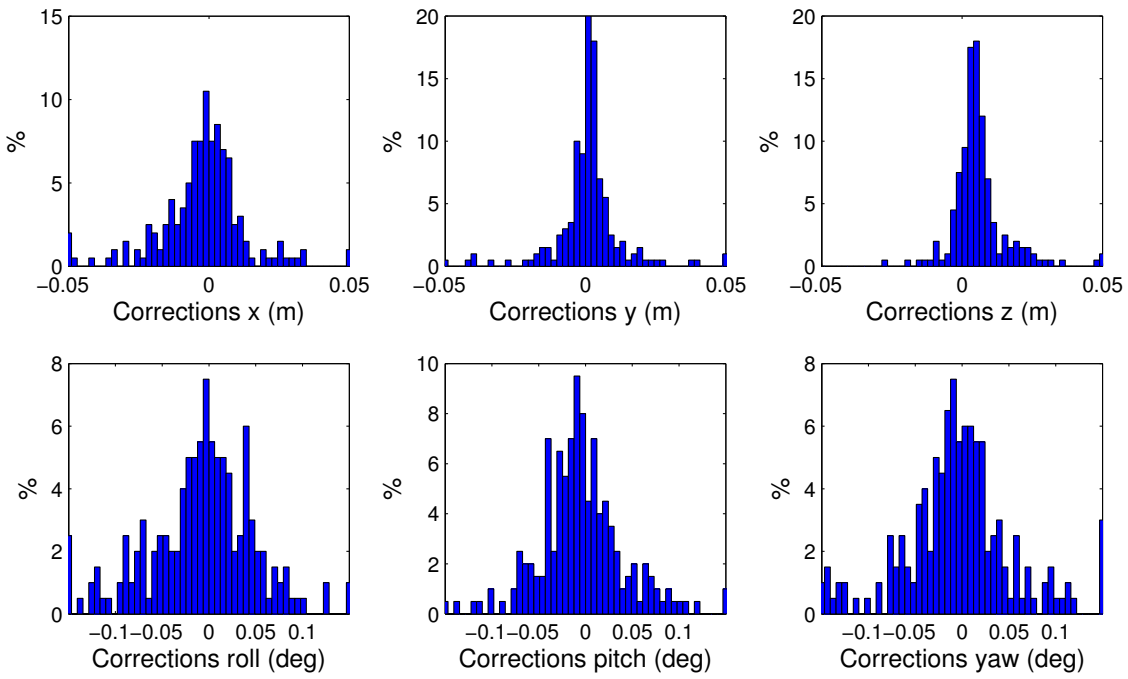


Figure 4.43: Histograms of the corrections applied to the VO trajectory by our system for the Woodstock data set. Note we have omitted the INS disparity as it is unchanged from Figure 4.28.

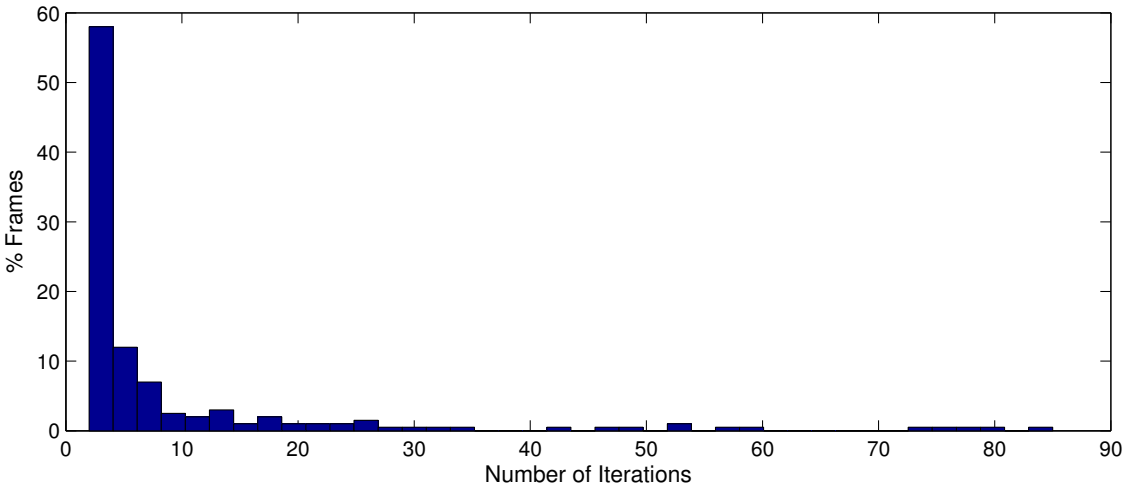


Figure 4.44: A histogram of the number of iterations required for the optimiser to converge on the Woodstock data set. It can be seen that approximately 60% of frames converged within five iterations.

	Distance Traveled (km)	Cross Track Error		Cross Track Error INS	
		Mean (m)	Std (m)	Mean (m)	Std (m)
Begbroke Laps	3.0	0.334	0.317	0.791	0.399
Begbroke New Road	1.8	0.309	0.128	0.296	0.126
Kidlington	0.3	0.329	0.309	0.568	0.408
Woodstock	0.5	0.217	0.182	0.294	0.207

Table 4.2: Statistics for Cross track errors over all datasets

	Mutual Information (dits)				Mean Number of Iterations
	Mean	Std	Maximum	Minimum	
Begbroke Laps	0.304	0.127	0.558	0.037	3.22
Begbroke New Road	0.139	0.069	0.331	0.031	10.61
Kidlington	0.166	0.102	0.511	0.019	5.86
Woodstock	0.291	0.1265	0.593	0.030	9.58

Table 4.3: Statistics for Mutual Information over all datasets

	Translation Corrections (mean (std))			Rotation Corrections (mean (std))		
	x (mm)	y(mm)	z(mm)	roll (mdeg)	pitch (mdeg)	yaw (mdeg)
Begbroke Laps	−1.5(12.9)	0.3(19.3)	−0.7(13.7)	−1.7(18.7)	−4.1(35.6)	1.8(29.5)
Begbroke New Road	4.0(49.5)	5.4(33.8)	3.8(32.1)	5.0(96.3)	2.0(80.5)	−13.9(86.7)
Kidlington	−1.0(28.8)	0.6(24.8)	4.4(18.4)	3.1(132.3)	−2.6(100.2)	14.4(113.9)
Woodstock	−4.5(32.3)	3.1(31.9)	12.4(22.5)	−15.2(138.4)	−8.8(97.4)	−5.2(126.9)

Table 4.4: Statistics for corrections applied by our system over all data sets

4.10 Limitations and Failure Modes

4.10.1 Reliance on VO

We have demonstrated our system's ability to operate in conditions varying from ideal to extreme lighting changes. However, the system is dependant on relative pose estimates from the VO to build a survey and seed the optimisation on subsequent traversals. If the VO fails completely then so does our system. This being said the VO system has shown that it is able to provide relative pose estimates in very challenging conditions.

This limitation is not a particular problem when generating a survey as this is a one time event where great care can be taken in choosing optimal lighting and driving conditions e.g. dry and over cast. However, with subsequent traversals it may be possible to use other pose sources such as wheel odometry or a basic accelerometer plus gyro system. This is a possible direction for future work in this area.

4.10.2 Road Surface Visibility

The system has a reliance on being able to observe the road surface immediately ahead of the vehicle. We have demonstrated that our system is robust to oncoming vehicles obscuring the road surface in Section 4.9.3.4. However, if a vehicle was to completely obscure the road surface ahead of the camera the system may fail. The obvious solution to this problem is to maintain a pre-specified driving distance when using this system. This, however, may not be desirable as one of the goals of autonomous vehicles is to increase the traffic density [21].

In Section 6.2 we discuss some preliminary work which attempts to not only exploit the road surface for localisation, but also attempts to use building façades. This approach looks up as well as down, potentially limiting the risk of surfaces used for localisation being obscured.

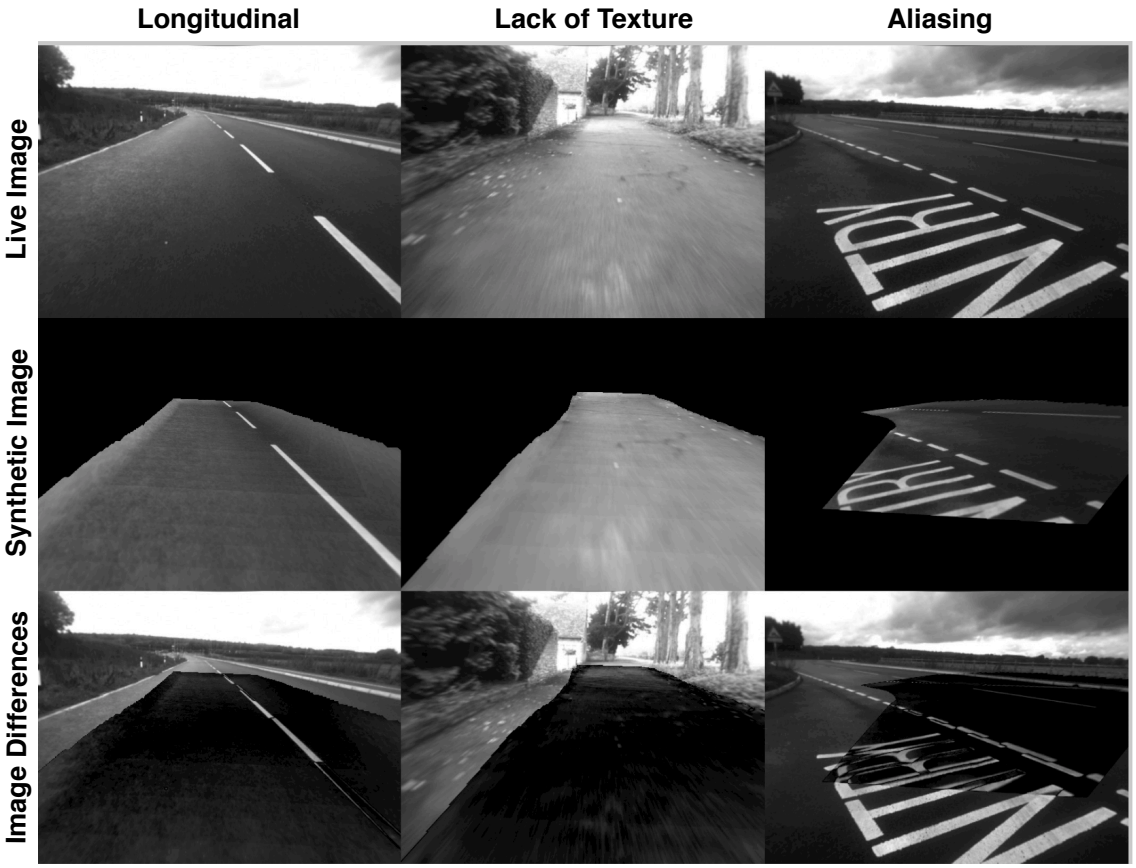


Figure 4.45: Failure Modes

4.10.3 Matching Failures

Figure 4.45 shows some of the potential failure modes of the view matching process used. The most likely failure mode to be encountered is a lack of texture on the road surface. The lack of texture example in Figure 4.45 shows the point of failure on the fifth lap in Figure 4.26. Although the system was able to localise in this area on 4 out of 5 occasions it did fail once. Here there is almost no distinctive texture on the road surface and the motion of the vehicle and limited light due to overhanging tress has resulted in image blur. These conditions resulted in a non-convex objective function and localisation failed.

The two other failure modes are caused by aliasing and relate to the regular and repeating structure of road markings. The longitudinal tracking failure seen in Figure 4.45 occurs when a jump in the VO trajectory causes the localisation system to fall into a local minima. Here successive road markings overlap, causing an initial

decrease in Mutual Information if a move towards the true vehicle position is made. Although longitudinal tracking is in error, lateral tracking is maintained.

A second aliasing example shown in Figure 4.45 is a similar case where the regular structure of the text painted on the road has caused the optimiser to fall into a local minima. This example is taken from an incorrect initialisation at the start of a traversal in Figure 4.31 lap 4. The system successfully recovered once the text was passed.

4.11 Conclusions

This chapter presents a methodology for generating and exploiting synthetic local orthographic images to achieve centimeter precision road vehicle localisation without any infrastructure or work space modification. The method improves accuracy by an order of magnitude on a previous method using off the shelf aerial images [72]. We use Stereo VO to generate synthetic orthographic images from a survey vehicle, which are far superior in terms of resolution and fidelity to available aerial images. Our approach also allows us to generate orthographic images in areas unavailable to aerial photography such as under bridges, trees and covered areas. These images provide a high fidelity and stable template for view matching as unlike feature-based systems we use the gross appearance of the road surface ahead of the vehicle. Our approach avoids all the tracking and data association required by feature-based approaches.

We demonstrate the robustness of Mutual Information as a measure of image alignment compared to Sum of Squares Differences. Showing that not only is the measure lighting invariant, but it reduces the number of iterations required to perform image registration.

We evaluated our localisation system on over 5km of data consisting of more than 60,000 images. We demonstrate that one survey is sufficient for localisation on multiple types of road surface and under extreme lighting changes. The system using only a single stereo pair has produced pose estimates commensurate with a state of the art INS system. We have also qualitatively demonstrated, by showing

image alignment, that the accuracy of the system is not only commensurate with the INS but potentially superior. However, to confirm this we would require a better estimate of ground truth.

Chapter 5

Camera Laser Calibration

In this Chapter we present a novel target-less extrinsic calibration technique which determines the $\mathbb{SE}3$ pose between a camera and LIDAR mounted in a push-broom orientation. Rather than requiring a stationary platform, as with previous calibration procedures, we actually exploit vehicle motion. As the platform traverses through an environment we build up at swathe of $3D$ LIDAR data using a VO derived trajectory estimate and a putative extrinsic calibration. Synthetic LIDAR reflectance imagery is then generated and dense registration between live camera images and synthetic views allows us to recover the extrinsic calibration. We develop a robust objective function and perform Bayesian fusion of individual estimates to improve final calibration estimates. We evaluate our algorithm on 73 scenes and demonstrate calibration precision to better than 5mm and 0.4 degrees.

5.1 Introduction

Two of the most prevalent sensors in robotics, especially in the transport domain, are cameras and LIDAR (Light Detection And Ranging). Increasingly these sensors are being used to supplement each other, an examples of which is Google Street View, where LIDARs are used in conjunction with an omnidirectional camera to display planes in the captured images [19]. A multi-modal approach to localisation was also presented in [103] where a monocular camera is localised using scene geometry

provided by a 3D LIDAR sensor. In order to achieve this projection of LIDAR scans into the images captured by a camera, the extrinsic calibration between the various sensors must be known to a high degree of accuracy [55].

We pose the calibration as a view matching problem and require no explicit calibration targets or human intervention, as has been required in most prior art [120, 82, 28]. Instead we explicitly exploit the fact that scenes in laser light look similar to scenes as recorded by an off-the-shelf camera. We synthesise images from LIDAR reflectance values based on the putative calibration between sensors and measure how well they align with camera images, Figure 5.8. Rather than use a feature-based approach to measure alignment we exploit the gross appearance of the scene using a robust metric in the form of a gradient based Sum of Squares objective function. The calibration giving maximal alignment is then accepted to be the best estimate of the camera LIDAR calibration.

In contrast to prior art, our approach does not require both sensors to have overlapping fields of view. Instead, we exploit the motion of the vehicle to retrospectively compare LIDAR data with camera data, Figure 5.1. This is achieved through the generation of a swathe of LIDAR data built up as the LIDAR mounted in a push-broom configuration traverses through the workspace, Figure 5.2. Therefore, our method can be used as long as the motion of the vehicle causes eventual overlap of the observed workspaces. This feature of the calibration is particularly useful in transport as often sensors can be mounted all over the vehicle with non-overlapping fields of view.

5.2 Background

Here we concern ourselves only with the extrinsic calibration of multiple sensors. We assume that the intrinsic calibrations for both the LIDAR and camera are known. A large body of work exists on intrinsic calibration for both cameras [117, 121] and LIDAR [96, 59, 18] sensors.

The extrinsic calibration of sensors can then be performed in many different

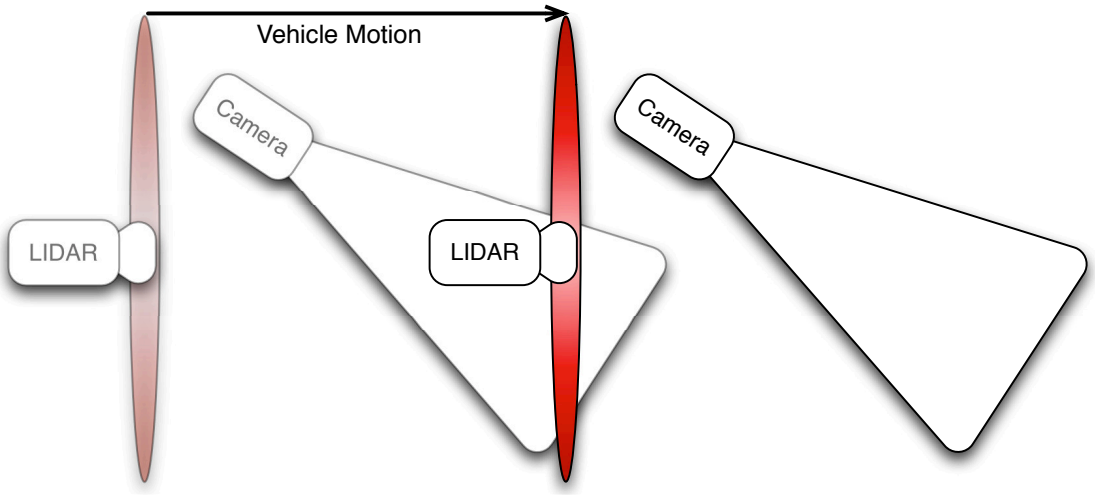


Figure 5.1: LIDAR push-broom orientation and non-instantaneously overlapping fields of view. Only as the platform moves do the LIDAR data become visible in previous camera images making a stationary calibration impossible.



Figure 5.2: An example of a swathe which can be thought of as a 3D point cloud built up as a push-broom 2D LIDAR sweeps through an environment with vehicle motion. The red lines illustrate the latest scan from the 2D LIDAR, which is mounted on the front of our Bowler WildCat research platform. This swathe was constructed using calibration estimates obtained from the proposed method relative to a stereo camera providing a trajectory derived from visual odometry.

ways. The simplest approach is to physically measure the position of sensors relative to each other. This approach, however, proves to be more complicated than first thought as sensors are generally housed in casings which do not allow for accurate measurement of the sensing element itself.

Another approach is to place calibration targets into the workspace which are simultaneously in the field of view of both sensors. Calibration is performed by aligning features on the calibration targets observed by both sensors. A method of extrinsic calibration of a camera with a 2D range finder using a checkerboard pattern was presented in [120]. Checkerboards are again used in [81] and [28] to cross-calibrate 3D LIDAR with a camera requiring varying amounts of user guided pre-processing.

Robust long-term autonomy, however, requires a continuous assessment of calibration accuracy, which makes the use of calibration targets impractical. The sensors considered here are often included in safety critical systems so the ability to test the validity of calibrations or recalibrate after bumps, knocks and vibrations is critical to reliable operation. A method of target-less extrinsic calibration, which required the user to specify several point correspondences between 3D range data and a camera image was presented in [93]. Levinson 2012 [43] reduced the need for user intervention by examining edges in images produced by an omnidirectional camera. These edges are then assumed to correspond to range discontinuities in 3D LIDAR scans. Our approach is most closely related to a method described in [82], where reflectance values from a static 3D LIDAR are registered against captured images using an objective function based on Mutual Information.

As far as we are aware our approach is the first to present automatic calibration of a camera and 2D LIDAR in natural scenes without explicit targets placed into the workspace or other user intervention. This is also the only method not requiring sensors to be mounted such that they have overlapping fields of view.

Finally we show how by treating individual calibration estimates as noisy measurements we are able to fuse estimates, improving calibration precision.

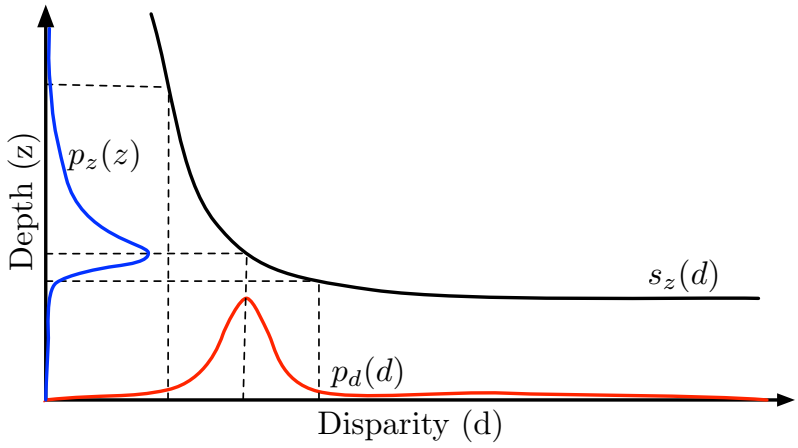


Figure 5.3: Here we show how the Gaussian distribution describing the uncertainty in disparity measurements is transformed to depth measurement uncertainty via the stereo triangulation process described in Equation 5.1, $s_z(d)$.

5.3 Limitations of Stereo Depth Estimation

Many modern autonomy applications require a 3D representation of the environment in which a vehicle is traveling. Although cameras offer a high fidelity measurement of the visual texture of a scene, the underlying geometry is not explicitly measured and must be inferred. In Section 3.3 we described how a stereo camera can be used to infer scene geometry and here we will discuss its limitations.

5.3.1 Depth Estimation Uncertainty

Sibley [98] showed that the accuracy of feature measurement in images is approximately Gaussian. Recalling the stereo triangulation process in Section 3.3.3, the inferred position of a 3D point is written as

$$s(\mathbf{z}) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{u_l b}{d} \\ \frac{v_l b}{d} \\ \frac{bf}{d} \end{bmatrix}, \tag{5.1}$$

where $\mathbf{z} = [u_l, v_l, u_r, v_r]^T$ is a feature measurement in the left and right camera images, b is the stereo pair baseline, f is the focal length and $d = u_l - u_r$ is the disparity. If we take the Gaussian probability density function (pdf) of the disparity $p_d(d)$ and

transform it via equation 5.1 using the standard result illustrated in Figure 5.3, we obtain

$$p_z(z) = p_d(d)\left|\frac{\partial d}{\partial z}\right| = p_d[s_z^{-1}(z)]\left|\frac{\partial s_z^{-1}(z)}{\partial z}\right|, \tag{5.2}$$

where $|\cdot|$ denotes the absolute value of the Jacobian, and $s_z^{-1}(z) = {}^{bf}/_{z}$. We acquire $p_z(z)$, the *pdf* of the stereo depth or range estimation, z, as

$$p_z(z) = \frac{bf}{\sigma_d z^2\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(bf\frac{\frac{1}{z}-\frac{1}{\mu_z}}{\sigma_d}\right)^2\right), \tag{5.3}$$

where μ_z is the mean depth and σ_d is the disparity variance. Figure 5.4 shows how with a constant disparity uncertainty, $\sigma_d = 0.5\,pixels$, the original Gaussian *pdf* is stretched and skewed as the distance to a measured object increases. Although the mean and variance of $p_z(z)$ does not appear to have an analytical solution we can compute the variance, σ_z, numerically. Figure 5.5 shows how the range uncertainty, σ_z, increases as range increases.

In outdoor scenes we are often dealing with distances much greater that $10m$, but with the $12cm$ baseline of the BumbleBee2 we can expect to see an uncertainty of $1m$ this range. In order to improve accuracy at larger ranges an ever-increasing baseline or focal length is required. With a baseline of $1m$, which in reality is the maximum achievable when the stereo pair is to be mounted on a car, the uncertainty at $20m$ is $\sigma_z = 30cm$. When comparing this to the accuracy of the SICK LMS 151 (Section 2.3.2) which has a constant $\sigma_z = 12mm$ up to $50m$. It is clear that measuring scene geometry with active sensors such as this can be favorable.

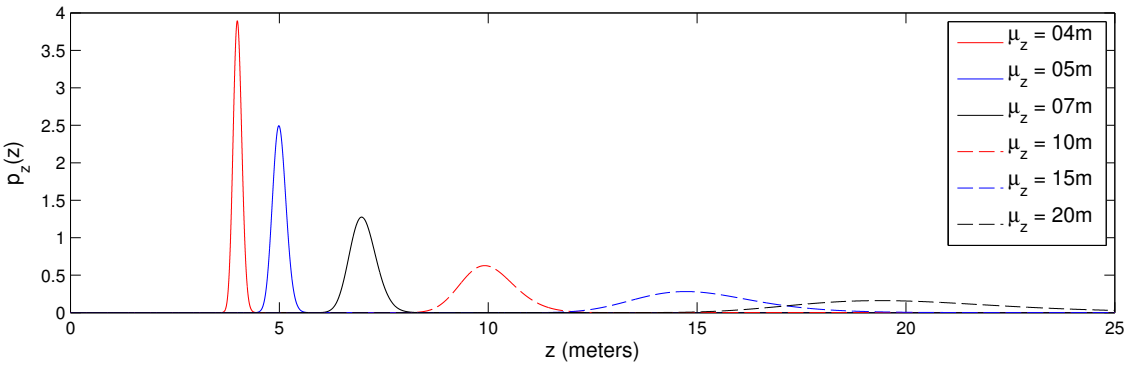


Figure 5.4: $p_z(z)$ for z uncertainty as as μ_z increases. This data is for the BumbleBee2 stereo pair described in Section 2.3.2. Here $f = 390$, $b = 0.12m$ and $\sigma_d = 0.5\,pixels$.

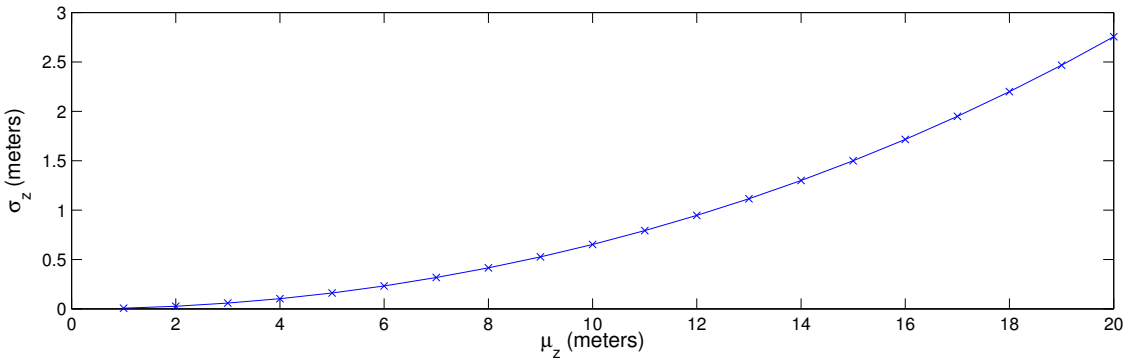


Figure 5.5: This figure shows the variance on range measurements, σ_z as range increases. It can be seen that with the BumbleBee stereo pair used here the range uncertainty is up to $1m$ at ranges of $11m$.

5.3.2 Dense Stereo

Figure 5.6 further illustrates how fusing output from two sensors rather than relying on one we can achieve a better model of a scene's 3D geometry. The dense stereo reconstruction in Figure 5.6 was computed using LIBELAS [29]. LIBELAS is a dense stereo library which generates a colored $3D$ point cloud from a calibrated pair of stereo images. Dense correspondence between the left and right images are found using a correlation based measure and used to project every pixel into $3D$ space.

Point clouds from 10 stereo pairs over 20m are stitched together using a VO trajectory. It can be seen that when viewed from close to the original camera perspective the point clouds look reasonably accurate. However, when we move away from the original camera view we can see that the walls, which are distant

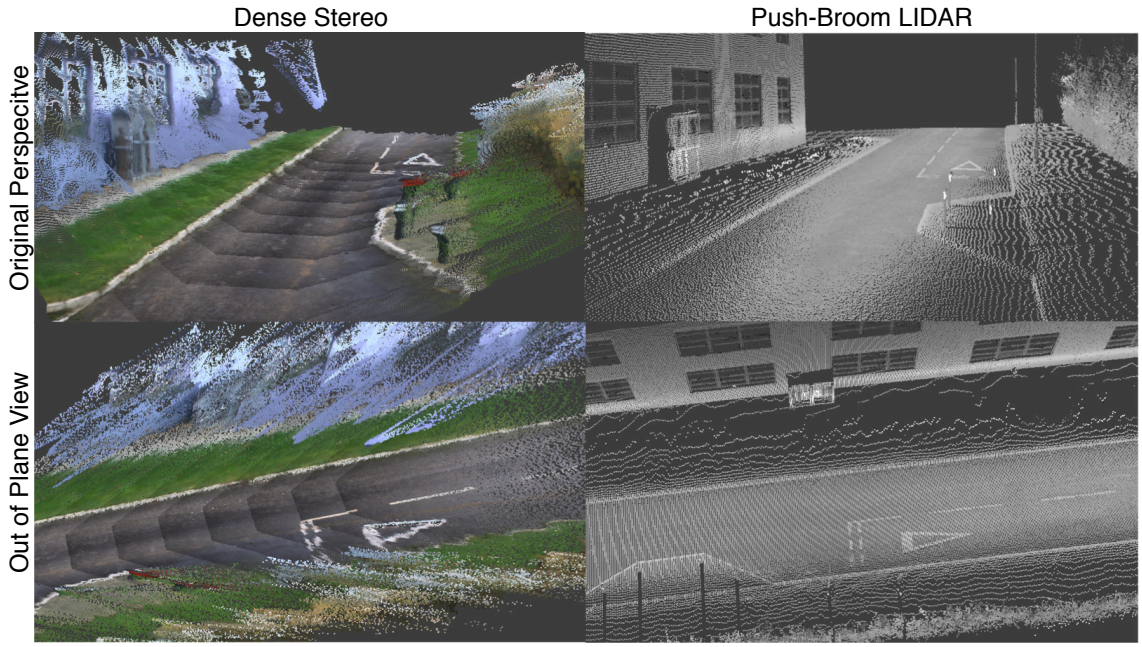


Figure 5.6: Examples of 3D reconstruction using dense stereo (left) and push-broom LIDAR (right). It can be seen that close to the original camera perspective the dense stereo reconstruction appears accurate. However, when the view is moved out of plane it can be seen that the dense stereo reconstruction is torn and distorted.

from the camera, contain tears and are significantly warped. This tearing in not caused exclusively by the range uncertainty, and can be influenced by regions without significant visual texture. It should be noted that the road surface, which is relatively close to the camera, appears to be well estimated.

In contrast the point cloud generated by the push-broom LIDAR, who's scans are stitched together using the same stereo VO trajectory, is much crisper. Unlike dense stereo the LIDAR explicitly measures range with a much higher accuracy and is able to cope with "visually smooth" surfaces. Having an extrinsic calibration between a stereo camera and push-broom LIDAR allows us to generate these point clouds effortlessly in almost any environment.

5.4 Problem Formulation

In order to create a swathe with a 2D push-broom LIDAR it must first undergo motion through its environment. In the general case, we construct a swathe using a base trajectory estimate, $X^b(t)$, obtained using an INS or, in our case, visual odometry

and the putative calibration ${}^{b}T_{l}$ between the base trajectory and the LIDAR, l. The swathe is then projected into the camera using the current calibration between the camera c and base trajectory ${}^{b}T_{c}$, Figure 5.7. In the specific case investigated here, poses are derived from VO such that the base trajectory $X^{b}(t)$, is the same as the camera trajectory $X^{c}(t)$. This simplifies the problem to finding the single extrinsic calibration ${}^{c}T_{l}$ as opposed to ${}^{b}T_{c}$ and ${}^{b}T_{l}$. An interpolated LIDAR reflectance image is then generated, Figure 5.8. We use an edge-based, weighted SSD (Sum of Squares Distance) objective function to measure the alignment of an image captured by the camera and the LIDAR reflectance image. A simple iterative optimisation is used to search over the $\mathbb{SE}3$ pose, which defines the extrinsic calibration and maximises the alignment of the camera image and the generated LIDAR reflectance image. It is assumed that the best estimate of the extrinsic calibration achieves the best alignment of the two images.

5.4.1 Generating A Swathe

To generate a metrically correct swathe from the push-broom LIDAR requires accurate knowledge of the sensor's motion. In the general case shown in Figure 5.7, a base trajectory $X^{b}(t)$ consists of full $\mathbb{SE}3$ poses, $(x, y, z, roll, pitch, yaw)$, as a function of time. $X^{b}(t)$ can be derived from a multitude of sensors including Inertial Navigation Systems (INS) and Visual Odometry (VO), as long as the trajectory is metrically accurate over the scale of the swathe. The poses of the LIDAR and camera are given relative to this base trajectory. In order to ensure millisecond timing accuracy we use the TicSync library to synchronise the clocks of the sensors to the main computer, see Section 5.4.2.

Let the ith LIDAR scan be recorded at time t_i and consist of a set of points, $\mathbf{x_i}$, and a set of corresponding reflectance values, $\mathbf{R_i}$, such that laser point j in this scan, $x_{ij} = [x_j, y_j]^T$, is associated with reflectance value $R_{i,j}$. We currently make the approximation that all points j within the ith scan are captured at the same time, however in reality each scan takes $20ms$. As the data used for calibration were

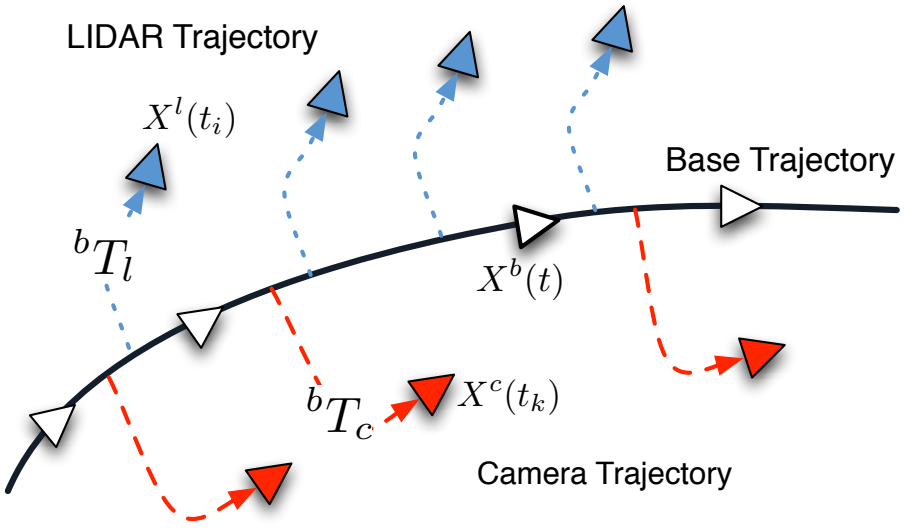


Figure 5.7: Shows the base trajectory $X^b(t)$, the camera $X^c(t_k)$ and LIDAR $X^l(t_i)$ trajectories relative to it. It should be noted that this is the general case. In the results presented here $X^b(t)$ is generated using visual odometry so the camera trajectory and base trajectory are one and the same, reducing the dimensionality of the search space from twelve degrees of freedom to six.

collected at purposefully slow speeds this approximation has a negligible effect. We first compute the pose of the LIDAR $X^l(t_i)$ based on the current putative extrinsic calibration ${}^bT_l$ and the base trajectory as

$$X^l(t_i) = X^b(t_i) \oplus {}^bT_l, \tag{5.4}$$

where $\oplus$ denotes a composition operator. Each scan can then be projected into a local 3D scene $\mathbf{P_i}$ creating a swathe of laser data as

$$\mathbf{P_i} = X^l(t_i) \oplus \mathbf{x_i}. \tag{5.5}$$

We can now generate a swathe as a function of the extrinsic calibration between the sensor base trajectory and the LIDAR. An example is shown in Figure 5.2.

5.4.2 Timing

Accurate timing is of significant importance when using multiple sensors. Each sensor has its own internal clock, as do the logging computer or computers. Timing errors



a) b) c)

Figure 5.8: a) An interpolated laser reflectance image at the estimated extrinsic calibration $\mathcal{I}_k^l({}^cT_l)$ created from LIDAR reflectance values projected into the camera. b) The actual camera image $\mathcal{I}_k^c$. c) An interpolated laser image with incorrect calibration $\mathcal{I}_k^l({}^cT_{0l})$. This scene was observed during a parking maneuver.

between LIDAR and camera data streams, for example, could lead to LIDAR data being associated with images many milliseconds in the past. This would introduce an extra degree of freedom to the calibration, which would have to be solved for. Fortunately TicSync [39] is a stand-alone library, which is able to take care of clock synchronisation automatically.

TicSync learns the mapping between distributed clocks, by simply observing timestamps. A probabilistic approach is used to estimate clock skew and offsets. In addition to this an upper-bound estimate on the error is produced. Sub-millisecond timing is achieved and synchronisation occurs within a few seconds, unlike Network Timing Protocol (NTP) [69] which can take several hours.

5.4.3 Generating LIDAR Reflectance Imagery

The next stage is generating LIDAR reflectance images as viewed from the pose of the camera c capturing an image $\mathcal{I}_k^c$ at time t_k. First, the swathe $\mathbf{P_i}$ is transformed into the camera's frame of reference using the current estimate of the extrinsic calibration between the base trajectory and the camera, ${}^bT_c$. The pose of the camera $X^c(t_k)$ at time t_k is then written as

$$X^c(t_k) = X^b(t_k) \oplus {}^bT_c \tag{5.6}$$

The swathe is then transformed into the camera's frame and projected into the

camera using the camera's intrinsics, K, which are assumed to be known. Thus,

$$\mathbf{p_{i,k}} = proj(\ominus X^c(t_k) \oplus \mathbf{P_i}, K) \tag{5.7}$$

gives us the pixel locations of the swathe points $\mathbf{p_{i,k}}$ in the camera image $\mathcal{I}_k^c$. At this point we could use the individual LIDAR reflectance values $\mathbf{R_{i,k}}$ and compare their reflectivity to the pixel intensities $\mathcal{I}_k^c(\mathbf{p_{i,k}})$. However, the density of the points is extremely variable due to foreshortening effects as points at larger ranges from the camera map to smaller areas within the image. We therefore use cubic interpolation to sample the intensities $\mathbf{R_{i,k}}$ at pixel locations $\mathbf{p_{i,k}}$ over the same grid as the pixels in $\mathcal{I}_k^c$. This generates a laser reflectance image $\mathcal{I}_k^l({}^bT_c, {}^bT_l)$ as a function of the extrinsic calibration, an example of which can be seen in Figure 5.8. From here we will assume that the base trajectory $X^b(t)$ is derived from stereo Visual Odometry, see Chapter 3. This simplifies the extrinsic calibration as the base frame is equivalent to the camera frame reducing ${}^bT_c$ to the identity and, in turn, the search space from twelve degrees of freedom to six. The laser reflectance image then becomes a function only of ${}^bT_l$, which is equivalent to ${}^cT_l$, the extrinsic calibration between the LIDAR and camera.

5.4.4 The Objective Function

At this point we have the ability to take a single camera image $\mathcal{I}_k^c$ and generate, given data from a 2D LIDAR and knowledge of the platform trajectory, a corresponding laser reflectance image $\mathcal{I}_k^l({}^cT_l)$ based on a putative extrinsic calibration between the two sensors. We now seek a metric which accurately reflects the quality of the alignment between the two images. This task is made difficult by non-linearities in the reflectance data [94]. Reflectance returns are not only effected by the reflective properties of the surface but also to other effects such as the angle of incidence of the incoming beam. These non-linearities render basic correlation measures such as standard SSD and Mutual Information ineffective for this application. It was found empirically that taking a smoothed gradient image was far more stable. Further, patch-based normalisation is applied whereby local variations in gradient are

normalised to be consistent across the whole image or at least between corresponding patches in $\mathcal{I}_k^c$ and $\mathcal{I}_k^l({}^cT_l)$. Applying patch based normalisation enhances local image gradients and avoids very strong edges completely dominating the objective function, see Figure 5.9. The pixel values from both images are then weighted by $w_{\mathcal{I}_k^c}$ the inverse of the distance transform of the reflectance measurement $\mathbf{p_{i,k}}$ over the image grid, giving extra weight to areas with a higher sampling density. The objective function can thus be expressed as

$$O({}^cT_l) = \sum_{\mathcal{I}_k^c} w_{\mathcal{I}_k^c} \left\| Q(\mathcal{I}_k^c) - Q(\mathcal{I}_k^l({}^cT_l)) \right\|_2 , \tag{5.8}$$

where $\sum_{\mathcal{I}_k^c}$ denotes the sum over all pixels in the image pair $\mathcal{I}_k^c$ and $\mathcal{I}_k^l({}^cT_l)$ and $Q(\bullet)$ denotes a function which performs Gaussian smoothing before taking the magnitude gradient image and performing patch based normalisation. In the results presented here we use a Gaussian kernel of 25x25 pixels with a variance of 6.5 and a patch size of 20x20 pixels for the patch based normalisation procedure.

5.4.5 Optimisation

The objective function provides a pronounced narrow convergence basin around the correct solution, see figure 5.15. We therefore use a simple iterative optimisation to find a solution. As this calibration is not required to be a real time application (it can be run as a background process on a vehicle), high speed is not a priority. Therefore, starting from an initial estimate ${}^cT_{lo}$, the search for a minimum is conducted along each degree of freedom individually, updating the estimate of ${}^cT_l$ as it proceeds. For the results presented in here we used a range of 30cm and 10 degrees with a resolution of ˜1mm and 0.02 degrees. While this brute force optimisation method was found to work well in our experiments our approach is agnostic to the exact optimisation method used. The estimate for a calibration for a particular image at time k is explicitly written as

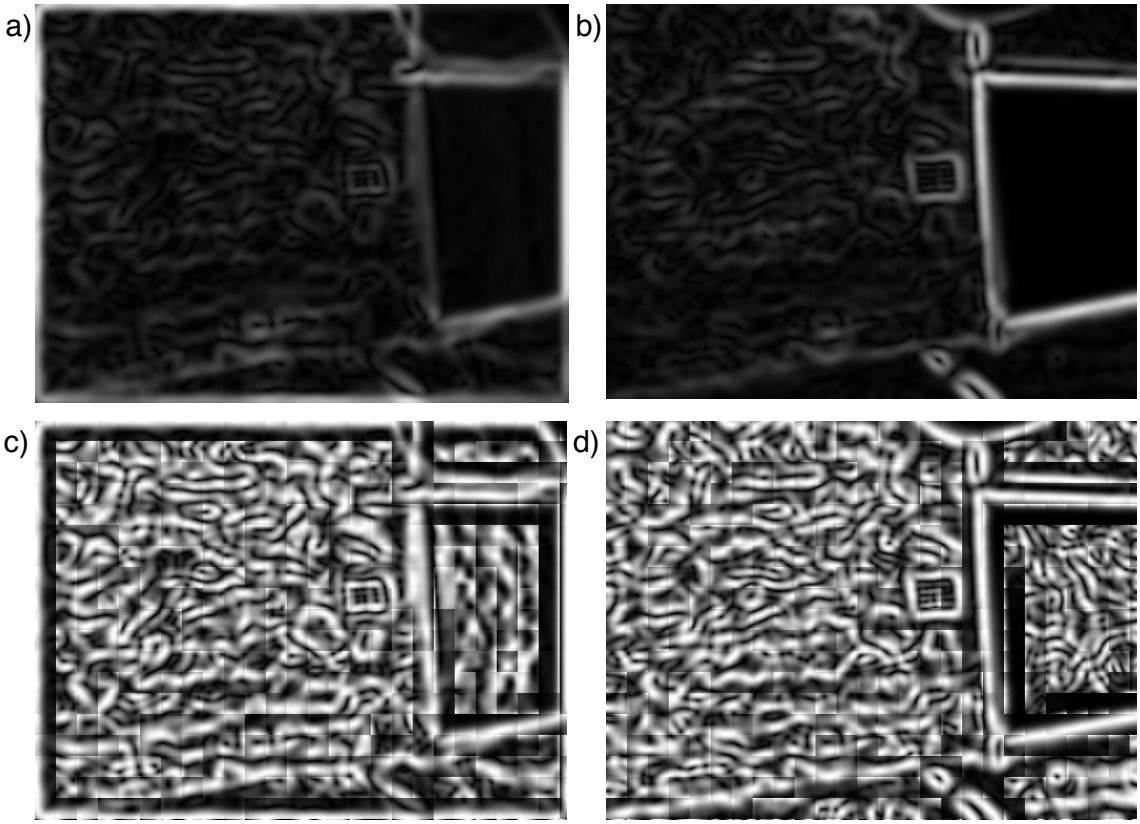


Figure 5.9: The smoothed edge image from the laser a) and the corresponding camera image b) from Figure 5.8. c) and d) are the same gradient images after the patch based normalisation procedure respectively. Note how the details are emphasised by the patch-based normalisation leveraging details that would have been drowned out by the few dominant edges if the sum of squares distance of a) and b) were computed directly.

$$ {}^{c}\bar{T}_l = \underset{{}^{c}T_l}{argmin} \sum_{\mathcal{I}_k^c} w_{\mathcal{I}_k^c} \left\| Q(\mathcal{I}_k^c) - Q(\mathcal{I}_k^l({}^{c}T_l)) \right\|_2 \tag{5.9} $$

5.5 Improving Accuracy With Estimate Fusion

We can now obtain, at any time, an estimate for the calibration between the LIDAR and camera. What remains to be asked is

1. are all scenes equally useful in supporting the cross model calibration?
2. how might we fuse multiple calibration estimates?

The answer to these questions are one and the same. Looking at Figure 5.15 we gain the intuition that minima at the bottom of a sharp trench are more informative than those at the bottom of a rough, broad bowl. By flipping the cost function and fitting a Gaussian $G({}^{c}\bar{T}_l, \sigma^2)$ to the the peaks (which were minima) we can obtain a likelihood function $\mathcal{L}({}^{c}T_l)$. This is the Laplace approximation and results in a "measurement" model parameterised by $\sigma^{2.}$.

We are now in a position to fuse a sequence of noisy measurements of the latent state ${}^{c}T_l$. Such a sequence ${}^{c}\mathcal{T}_l = ({}^{c}\bar{T}_{l1}, {}^{c}\bar{T}_{l2}, {}^{c}\bar{T}_{l3}..., {}^{c}\bar{T}_{lN},)$ with associated variances $\sum = (\sigma_1^2, \sigma_2^2, \sigma_3^2, ..., \sigma_N^2,)$ is fused via a recursive Bayes filter, allowing us to sequentially update our calibration as new parts of the workspace are entered, without the need for an expensive joint optimisation over N frames.

5.5.1 Recursive Bayes Filter

The recursive Bayes filter allows us to update our estimate of the latent state ${}^{c}T_l$ as new measurements are taken. Here for clarity we will revert back to standard notation. We start with Bayes rule, which is written as

$$ p(x|z) = \frac{p(z|x)p(x)}{p(z)}, \tag{5.10} $$

where z is a measurement and x is the underlying state. $p(z|x)$ is the probability density function of a measurement given the underlying state x, $p(x)$ is a prior, $p(z)$ is a normalisation term and finally $p(x|z)$ is the posterior *pdf*. For multiple measurements of the latent state

$$p(x|z_1, \ldots, z_n) = \frac{p(z_n|x, z_1, \ldots, z_{n-1})p(x|z_1, \ldots, z_{n-1})}{p(z_n|z_1, \ldots, z)}. \tag{5.11}$$

Using the Markov assumption [109] that a new measurement z_n is independent of previous measurements $z_1, \ldots, z_{n-1}$

$$\mathcal{L}(x|z_1, \ldots, z_n) = \mathcal{L}(z_n|x)\mathcal{L}(x|z_1, \ldots, z_{n-1}). \tag{5.12}$$

Here we have dropped the normalisation term and switched to Likelihood functions $\mathcal{L}(\bullet)$. Starting with the first measurement z_1

$$\mathcal{L}(x|z_1) = \mathcal{L}(z_1|x)\mathcal{L}(x), \tag{5.13}$$

and then the second z_2

$$\mathcal{L}(x|z_1, z_2) = \mathcal{L}(z_2|x)\mathcal{L}(z_1|x)\mathcal{L}(x), \tag{5.14}$$

the recursive Bayes update is then written generally as

$$\mathcal{L}(x|z_1, \ldots, z_n) = \mathcal{L}(x) \prod_{i=1,\ldots,n} \mathcal{L}(z_i|x). \tag{5.15}$$

For Gaussian *pdf*s, as used here, the update of the mean, $\bar{x}_n$, and variance, $\sigma^2_{x,n}$, of the posterior is explicitly written as

$$\bar{x}_n = \frac{\bar{z}_n\sigma^2_{x,n-1} + \bar{x}_{n-1}\sigma^2_n}{\sigma^2_{x,n-1} + \sigma^2_n} \quad \sigma^2_{x,n} = \frac{\sigma^2_n\sigma^2_{x,n-1}}{\sigma^2_{x,n-1} + \sigma^2_n} \tag{5.16}$$

This process of treating each optimisation as a noisy measurement significantly reduces the standard deviation of the calibration result when compared to the

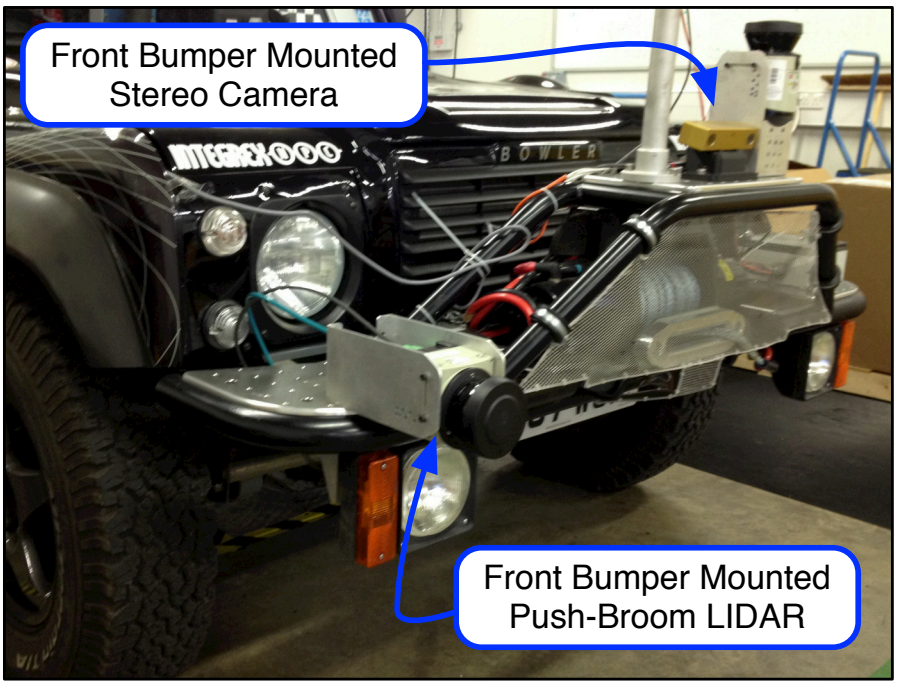


Figure 5.10: The WildCat platform used in our experiments is a fully equipped autonomous vehicle. It has on-board computing and multiple sensors including a SICK 2D LIDAR and a PointGrey BumbleBee stereo camera. The platform enables fast and reliable collection of survey quality data used in this work. It should also be noted that the 2D LIDAR and stereo camera do not have overlapping fields of view when stationary, meaning a retrospective calibration technique as presented here is required for data-based extrinsic calibration.

deviation of the raw estimates from individual frames, as we will show in Section 5.6.

5.6 Results

We used our Wildcat platform with a Point Grey BumbleBee2 stereo camera and SICK LMS-100 2D LIDAR, see Figure 5.10, to evaluate the performance of the proposed algorithm. The results shown here were obtained using several locations around our field center at Begbroke. Data was purposely collected during slow speed maneuvers, which produces maximum overlap of LIDAR and camera data. Approximately 70 meters worth of maneuvering data were collected. The images used for calibration were spaced at approximately one meter throughout the data set. In total 73 different images taken from the left camera of the stereo pair were used to validate the calibration estimates.

Swathes of approximately 300 LIDAR scans, equaling over 140,000 measurements, were used for each individual calibration estimate. Trajectory estimates were provided by our stereo visual odometry (VO) system Chapter 3 . Obtaining ground truth for trajectory estimation is always problematic, however the VO system has demonstrated high local metric accuracy. Detailed statistics are given again in Chapter 3.

5.6.1 LIDAR Reflectance Imagery

A key assumption in our approach is that the scenes appear similar in both camera and synthetic LIDAR imagery. Figure 5.11 shows six examples of camera images with their associated LIDAR reflectance images. It can be seen that the LIDAR reflectance imagery captures fine detail in the scene and operates in the presence of stationary vehicles. In some of the frames tearing is observed, where objects in the foreground occlude the background. The inverse distance transform down weights these areas and we are still able to acquire calibration estimates.



Figure 5.11: Example camera images alongside their synthesised LIDAR reflectance counterparts.



Figure 5.12: A set of 10 randomly selected frames used in a fused calibration estimate. It can be seen that the frames are acquired at one meter intervals within different scenes

5.6.2 Estimate Fusion

In order to estimate the performance and repeatability of the proposed algorithm we first evaluated the calibration on 73 individual frames. The results for this experiment can be seen in Figure 5.13 and Table 5.1. The standard deviation of the estimates — which is important as we are unable to accurately measure ground truth — for individual frames is of the order of a couple of centimeters and degrees, which is akin to the expected level of accuracy a hand measured calibration would yield. In order to test any improvement achieved by fusion of individual frames we performed twenty N choose M trials with N being the 73 frames and $M = 10$. For each fused estimate ten random frames were chosen and then fused in sequence using the Recursive Bayes filter. An example set of ten frames can be seen in Figure 5.12. The effect of the fusion stage can be seen in Figure 5.13 and Table 5.1 with the standard deviations decreasing by almost an order of magnitude.

Figure 5.14 shows how the standard deviation of calibration estimates decreases

Standard Deviation	***Translation (mm)***			***Rotation (degrees)***		
	x	y	z	roll	pitch	yaw
Individual Results	28	21	15	1.4	1.4	1.5
Fused Results	4.5	5.2	4.6	0.38	0.39	0.44

Table 5.1: Table shows the standard deviation of the cross-calibration estimate over 73 frames from several scenes, ten of which are shown in Figure 5.12. The effect of fusing estimates using the Bayes filter can be clearly seen with almost an order of magnitude reduction in the standard deviation of the calibration estimates.

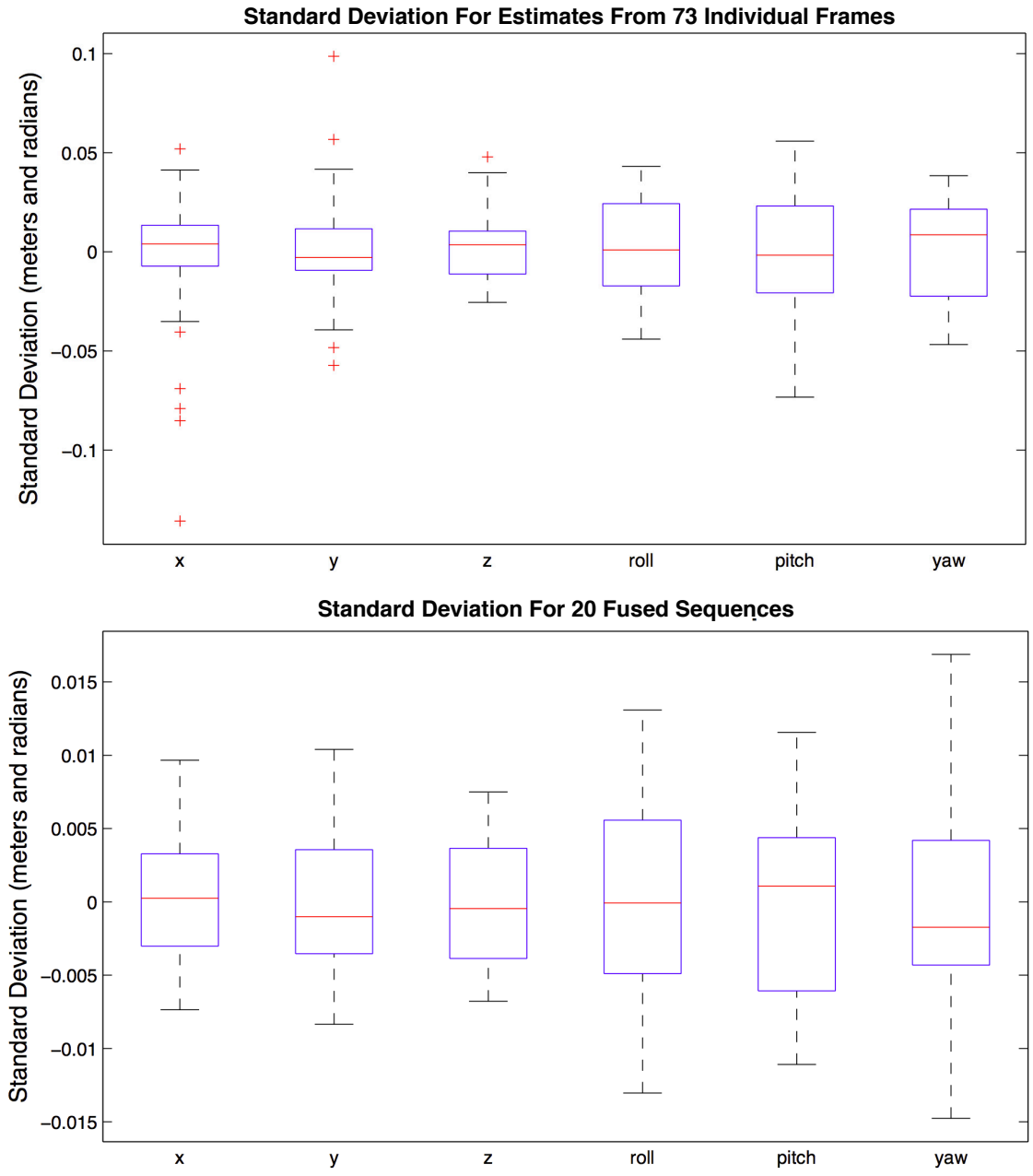


Figure 5.13: Box plots of the standard deviation of the extrinsic calibration estimate in each of the six degrees of freedom. The top plot shows the standard deviation of optimisation results from single frames while the bottom shows the deviation after fusing multiple frames using the Laplace approximation and a recursive Bayes filter. Twenty fused results were generated by selecting sequences of ten random calibration estimates from the possible 73 single frames in a M pick N trial. It can be clearly seen that after fusion the standard deviation is significantly reduced, see Table 5.1 for details, and outliers are effectively ignored. Boxes extend from the 25th to 75th percentiles and whiskers extend to the extrema of the data not considered to be outliers, which are denoted by red crosses. Note the different vertical scales.

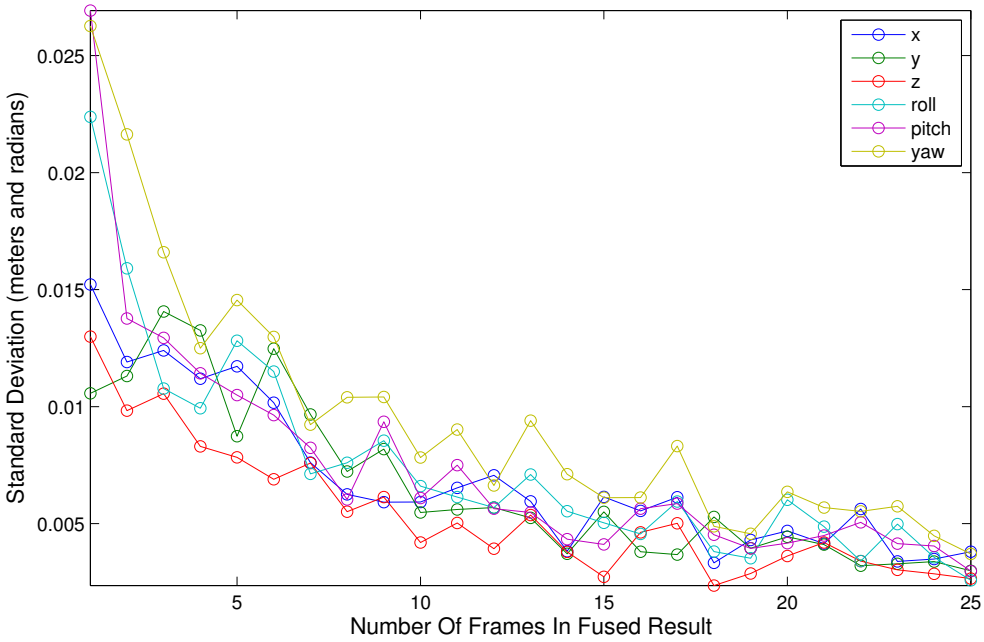


Figure 5.14: A plot showing how the standard deviation of the calibration estimate reduces as more frames are encountered and the calibration estimate is updated. This data represents twenty sequences of 25 frames randomly selected from a possible 73 frames.

as more individual estimates are fused together. The effect appears to saturate after approximately ten frames.

Figure 5.15 shows how frames with ill-conditioned and noisy calibration estimates, see Figure 5.15(b), are automatically assigned a higher σ^2, as per the process in Section 5.5. Estimates with more distinct troughs around the minima, see Figure 5.15(a), conversely are assigned a lower σ^2 and as expected contribute more information to the final calibration estimate after fusion. This is illustrated by the bar charts in Figure 5.15, which plots the inverse variance, $1/\sigma^2$, which effectively weight the individual estimates in the Bayes filter. Here the frame in Figure 5.15(a) is given over 50% more weight that the frame in Figure 5.15(b).

Given that the stereo camera and 2D LIDAR do not have instantaneously overlapping fields of view this calibration would not be possible with any of the other techniques reported in the literature.

5.6.3 Objective Function & Convergence

Figure 5.15 and 5.16 show the objective function plotted about the estimated extrinsic calibration for four example frames. It can be seen that there is a distinct concave basin in a) and c) but less so in b) and d). However, away from the true solution the objective function is non-concave, which justifies the choice of the simple search based optimisation over a traditional gradient-based method. It is possible that the final calibration could be improved by a final gradient descent type optimisation once the search has found a coarse solution. This has not been investigated in this work. It can also be seen that some degrees of freedom are more sensitive than others, this is thought to be due to the geometry of the typical scene and sensor placement. Note this is handled naturally by our filtering.

The objective functions in Figure 5.15 and 5.16 provide some idea of the convergence basin of the calibration technique. Here we see there is a distinct minimum within a range of $15cm$ and 5 degrees either side of the true calibration. As this is a dense approach the viewing frustum of laser data must be representative of the camera images. Therefor a relatively 'good' initialisation is required and an initialisation from an arbitrary guess would most likely fail to converge.

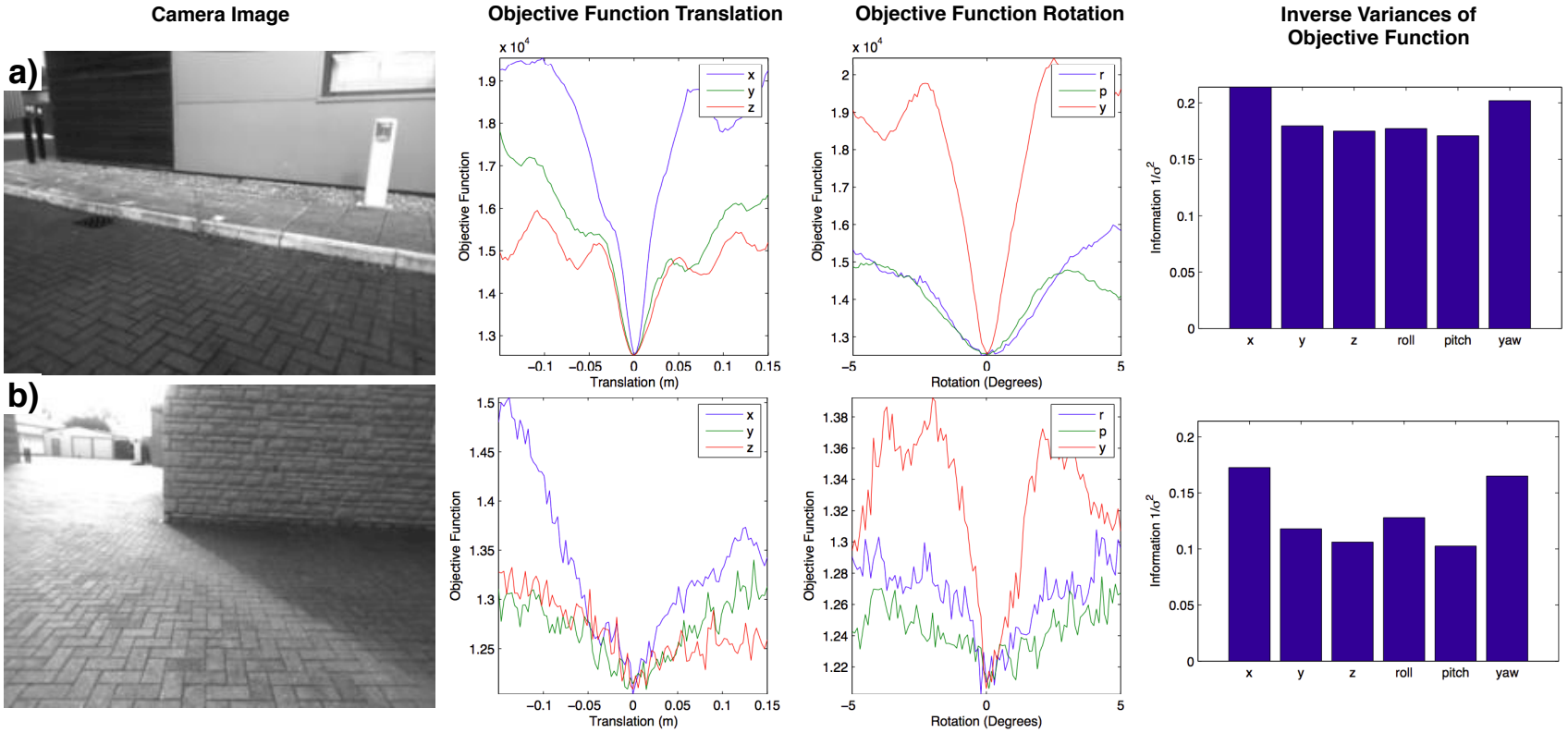


Figure 5.15: The objective function plotted about the estimated calibration for two example scenes. a) shows an example of how the objective function is concave around the solution and has a clear global minima. Away from the minima the objective function can be non concave, justifying the use of the grid search approach, improving the basin of convergence and reducing the required accuracy of the initialization. However, b) shows how certain scenes can produce noisy and ill conditioned objective functions around the solution. The bar charts of inverse variance of the estimate derived from the shape of the objective function at the minima demonstrate how less pronounced minima yield less information and are given less weight in the Bayes filter.

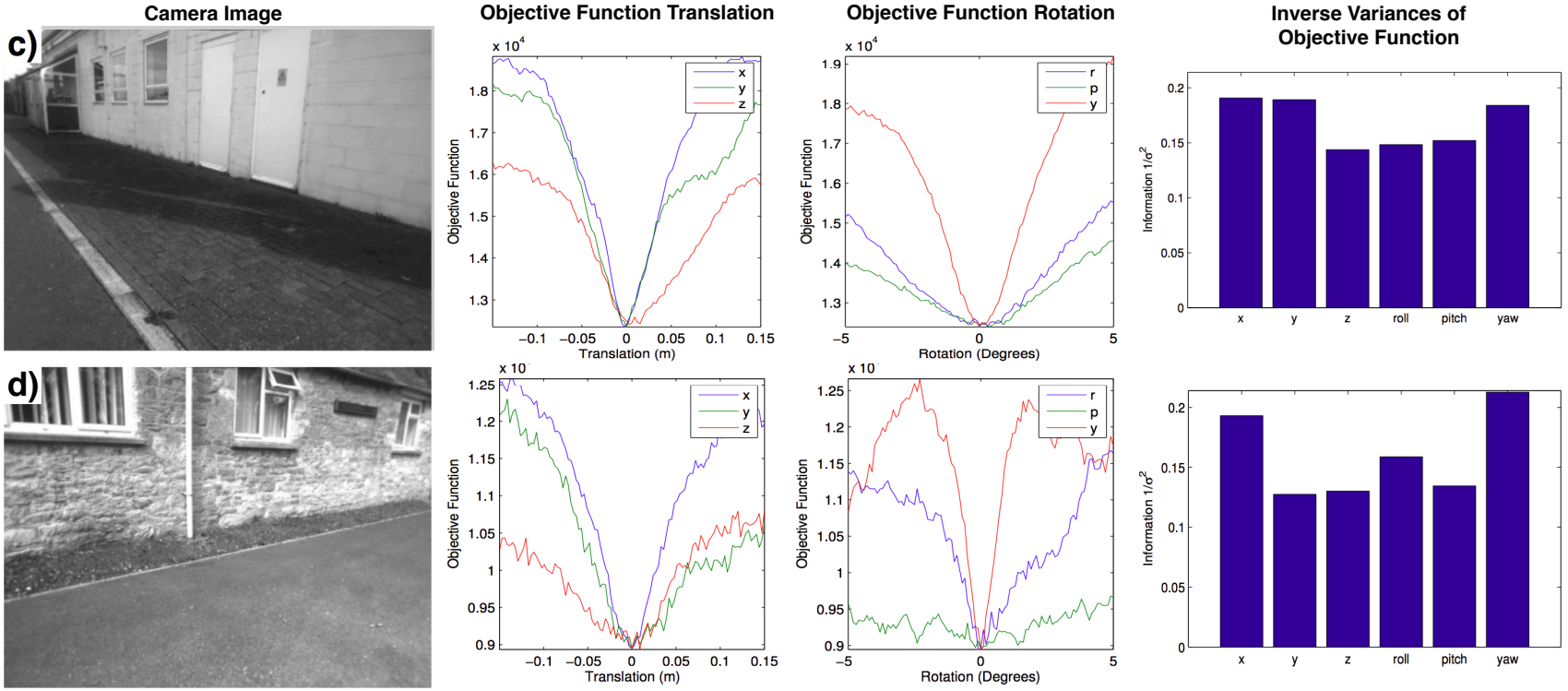


Figure 5.16: The objective function plotted about the estimated calibration for two additional example scenes. c) shows an example of how the objective function is concave around the solution and has a clear global minima. This example has a relatively shallow basin meaning the global minima is clear, but the actual information content is lower. However, d) shows another noisy objective functions around the solution. It should be noted that in d) the objective function for yaw is a very pronounced peak. This peak results in a large weighting on the yaw estimate in the recursive Bayes filter.

5.7 Conclusions

We have presented an automatic calibration procedure for a camera and a 2D LIDAR under general motion. This is a method which can be used in natural scenes without the need for targets, enabling on the fly calibration. The method also does not require sensors to be mounted such that they have overlapping views. Unlike other approaches we exploit the motion of the vehicle using a trajectory estimate to build a swathe of LIDAR data to generate laser reflectance imagery for comparison to images captured by the camera. We have adopted a robust correlation measure that is invariant to non-linearities in the reflectance returns and camera images. Furthermore we have demonstrated the calibration leveraging exposure to multiple scenes and fusing multiple one-shot calibrations yielding accuracies of millimeters in translation and deci-degrees in rotation. This is done in a way which is sympathetic to the utility of the scene structure and appearance in constraining the extrinsic calibration parameters.

Chapter 6

Conclusions

In this Chapter well will summarise the work described in this thesis. We will also present avenues of ongoing and future work.

6.1 Summary of Contributions

Chapter 2 described the platforms used in this work. We gave a high level overview of our hardware and software architecture, which enables road vehicle autonomy research. We also presented data sets used in this work and the range of environments and conditions used for testing our algorithms.

Chapter 3 described a stereo visual odometry system used to estimate ego-motion. This VO system forms a foundational competency for future chapters and allows the estimation of locally accurate vehicle trajectories. We first presented background on methods of using locally mounted sensors to estimate motion. We then described approaches using vision and their various merits. The system described uses a sparse feature based approach and exploits the high speed of the BRIEF descriptor to allow for large inter-frame motion. A sample and consensus outlier detection scheme is used to remove data association errors. The Jacobian for full bundle adjustment is derived and we discuss how its structure changes from simple frame-to-frame motion estimation to multi-frame pose and landmark refinement.

We then presented experimental evaluation of the VO algorithm on 45 laps of our

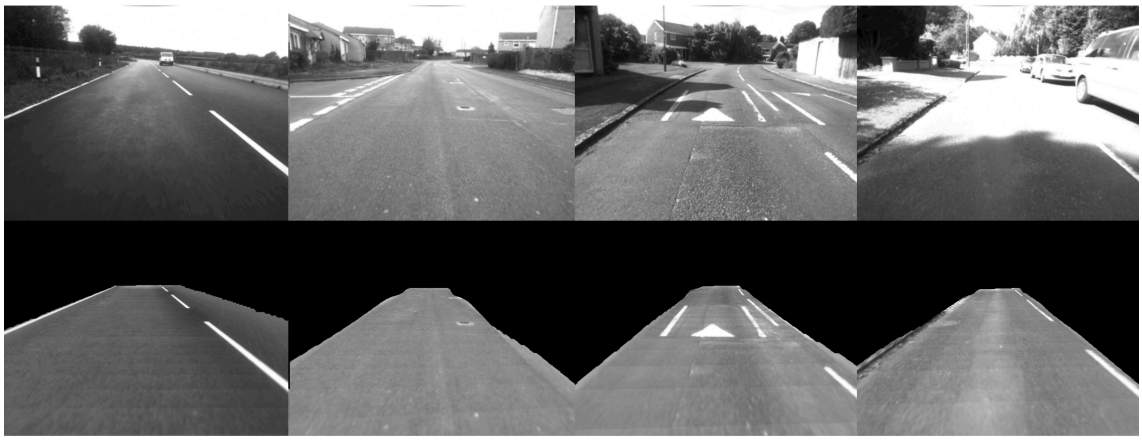


Figure 6.1: Example localisation results ranging from vanilla matching to extreme lighting changes. The live images (top) and their successfully localised synthetic counterparts (bottom)

test track at Begbroke. We evaluated nine mounting positions (five laps for each) around the vehicle in order to determine the optimal camera placement for road vehicle ego-motion estimation. We show that external forward facing is optimal for the algorithm described here. Finally we provided a detailed analysis of the VO system against an INS for the optimal camera position. We show the inter-frame ego motion is millimeter accurate, but suffers from the inherent drift associated with all dead-reckoning techniques.

Chapter 4 first presented a method of localising a vehicle using a camera and a UTM registered aerial imagery. The limitations of this approach were discussed and we then presented a vision-based localisation technique for road vehicles, which exploits a high-fidelity model of the road surface. We briefly covered initial work using aerial imagery to localise a road vehicle, which inspired this chapter. Initial results and limitations are explained and motivation for the chapters main contribution is given. The main system consists of two phases, survey and localise. For the first we presented a method of synthesising orthographic images of the road surface using the output of the VO system in Chapter 3. These images are generated at arbitrary resolutions, which are much higher than available aerial imagery. The images provide an excellent template against which to match subsequent traversals. We demonstrated the brittleness of sparse feature-based approaches and show how

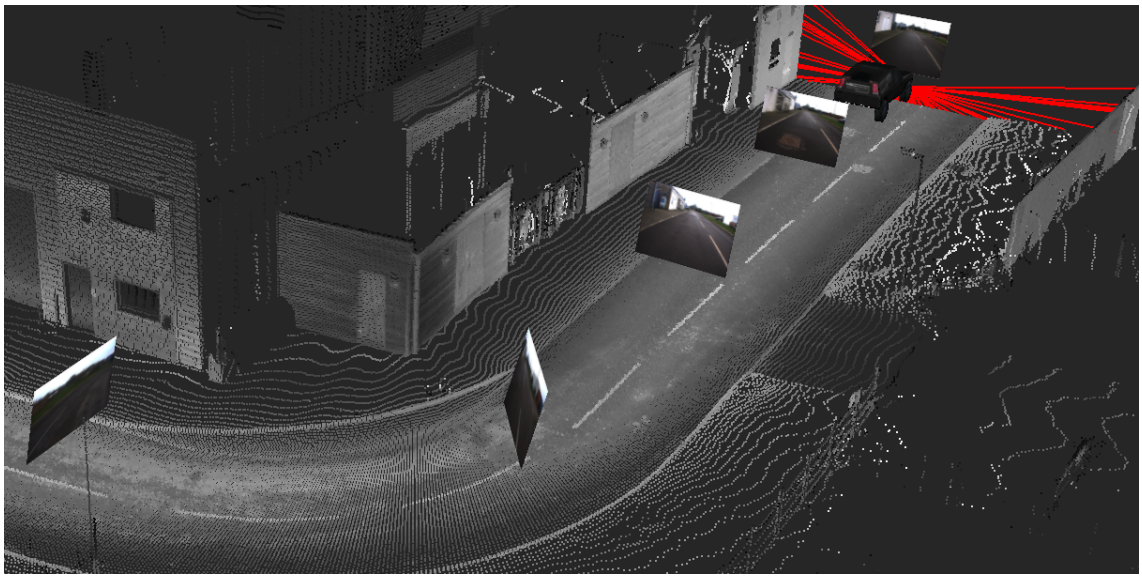


Figure 6.2: A 3D LIDAR swathe constructed using VO and the extrinsic calibration estimated by the calibration routine presented in Chapter 5. We are able to render camera images in their actual capture location within the point cloud. The red lines emanating from the vehicle indicate the latest incoming LIDAR scan.

localisation is improved using our dense approach. Localisation is framed as a view-matching problem whereby live images from the camera are aligned with synthesised views of the road surface captured in a survey. We exploit the local metric accuracy of the VO system to provide high frequency pose estimates with low frequency corrections provided by our dense matching technique. We use Mutual Information as a measure of image alignment, which was shown to be robust to varying lighting conditions and scene change.

We presented experimental results for over $7km$ of data; $2km$ survey and $5km$ localisation. The data covers four locations around Oxfordshire and contains various types of road surface and lighting conditions. We show that even with poorly marked roads and extreme lighting changes the system is able to localise with commensurate precision to that of a state of the art INS system, Figure 6.1. We then discussed failure modes and in Section 6.2 we discussed initial work on extending the approach to multiple planes.

Chapter 5 we again exploited locally derived ego motion, but now for the purpose of extrinsic sensor calibration. Here an automatic, target-less calibration procedure was presented for push broom LIDAR and cameras. Vehicle motion is exploited to generate

3D laser point clouds using a putative calibration estimate. This retrospective approach allows for non-overlapping fields of view and on the fly calibration, not possible with prior art. The reflectance data from these point clouds are then used to synthesise laser reflectance imagery, which is aligned with camera images using a dense edge based sum of squares method. This edge based method proved robust in the presence of non-linearities in the reflectance data which rendered standard similarity measures ineffective. A search-based optimisation then determines the highest value of this alignment measure, and what is assumed to be the correct calibration. We then showed how not all calibration estimates are equally useful and use the Laplace approximation to treat each one as a noisy measurement of the underlying state. These noisy measurements are then fused in a recursive Bayes filter to improve the calibration estimates accuracy.

We presented results from the calibration procedures in 73 different scenes from various locations around our Begbroke site. Before fusion the calibrations estimates offer centimeter precision, which is improved to millimeter precision after Bayesian fusion. Knowing accurately where a camera and LIDAR are relative to one another allows us to freely construct point clouds using VO and actually place images at their capture locations, see Figure 6.2.

6.2 Future Work

- **Extension To Multiple Planes.** Most recently we have attempted to extend our approach in Chapter 4 to multiple planes. The idea behind this approach is to opportunistically take advantage of any planar surfaces visible to cameras mounted on a vehicle. Looking up as well as down promises to increase the robustness of localisation when the road surface contains insufficient visual texture. This approach will also allow the system to leverage the appearance of building facades and traffic signs, potentially removing the need to maintain road visibility in dense traffic, as required in Chapter 4.

We have successfully exploited the calibration presented in Chapter 5 to extract scene geometry from LIDAR data, overcoming the limitations of stereo-vision. Large planar regions are segmented by assessing the local planarity of laser points [6] and then clustering is performed. These planar regions are then associated with visual texture from panoramic images captured using the LadyBug2 (See Section 2.3.2).

Localisation is then attempted using multiple planes in addition to the road plane. Unfortunately due to time restrictions this work was not developed to a stage where we could confidently present it as a contribution in this thesis.

- **Determining the Efficacy of Planar Surfaces.** The current implementation is only able to exploit texture on the road surface for localisation. By extending our approach to multiple planes, the system will be able to use building façades and road signs, etc. An interesting question we are then able to ask is, which surfaces are the most useful for localisation and can we place emphasis on these regions to improve localisation accuracy?

- **GPU Implementation.** The current MATLAB implantation is able to run on a single core at approximately $1Hz$ and at $5Hz$ on multiple cores. These limitations on speed lead to us using the VO for high frequency local pose estimation with low frequency updates provided by our dense matching technique. Preliminary GPU implementations have shown a speed up of several orders of magnitude. With this increase in speed, would it be possible to perform dense matching at frame rate and remove the reliance on stereo VO?

- **Experience Based Navigation.** Here we have shown that one survey is sufficient for localising a vehicle under extreme changes in visual appearance. However, failures still occur, such as those when there is limited visual texture on road surface. Could localisation in these difficult areas be improved by adopting a multiple experiences framework such as [12]?

- **Extension To Larger Scales.** Although we have tested the localisation

algorithm presented here on many kilometres of data in multiple environments and conditions. In order to take the algorithm further towards possible commercial exploitation, evaluation on many thousands of kilometres should be carried out.

- **Dynamic Calibration Updates.** We have shown how an automatic camera LIDAR calibration can be performed in natural scenes. This allows for on-the-fly calibration and monitoring of sensor configurations. Future investigations could include changing sensor configurations over the course of a data collection run. This would enable testing of the algorithms ability to detect these changes and recalibrate. As this work used mission critical sensors on the WildCat platform we were unable to carry out these experiments.

- **Extension to Many Sensors.** We present a pair wise sensor calibration here. However, robotic platforms are equipped with many sensors. As such, solving for single pairs at a time may not produce the overall best results. It is expected that extending the approach to a joint optimisation over multiple sensors will prove beneficial.

6.3 Concluding Remark

Autonomous road vehicles are set to become a ubiquitous technology in the coming decades. The main contributions of this thesis are a new infrastructure independent localisation technique and a target-less extrinsic sensor calibration. Both of these functions form fundamental competences required for true road vehicle autonomy, we hope this work will prove useful and be adopted by the research community.

Appendix A

Google Protocol Buffer Example

```
package Mrg.Datatypes.Protobuf.Image;

message PbImage {
        // Enum Specifying The Pixel Type.
        enum PixelType {
                RGB_U8_PIXEL  = 0;
                ...
                INVALID_PIXEL = 1000;    }

        // Enum Specifying The Pixel Channel Data Type.
        enum ChannelDepth {
                DEPTH_UINT_8    = 10;
                ...
                DEPTH_FLOAT_64 = 60;    }

        // Timestamps
        required Int64 Timestamp = 5;
        optional Int64 Host_Timestamp = 10;

        // Source Camera ID
        optional String Source_Camera = 6;
        optional Uint32 Source_Camera_Sensor = 20;
        optional Uint32 Camera_Grab_Index = 8;

        // Pixel Type.
        required PixelType Pixel_Type = 2;

        // Pixel Channel Data Type.
        optional ChannelDepth Channel_Depth = 40;

        required bytes pixel_data = 1;
        required Uint32 Height = 3;
        required Uint32 Width  = 4;
```

Figure A.1: An example Google Protocol Buffer definition file. This particular example if for an image grabbed from a camera.

Bibliography

[1] Khaled S Ali, C Anthony Vanelli, Jeffrey J Biesiadecki, Mark W Maimone, Yang Cheng, A Miguel San Martin, and James W Alexander. Attitude and position estimation on the mars exploration rovers. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 20–27. IEEE, 2005.

[2] H Bay, A Ess, T Tuytelaars, and L Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[3] Bryce E Bayer. Color imaging array, July 1976.

[4] R Behringer, W Travis, R Daily, D Bevly, W Kubinger, W Herzner, and V Fehlberg. RASCAL - an autonomous ground vehicle for desert driving in the DARPA Grand Challenge 2005. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 644–649, 2005.

[5] Peter Biber, Henrik Andreasson, Tom Duckett, and Andreas Schilling. 3d modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. In *Intelligent Robots and Systems.(IROS). Proceedings. IEEE/RSJ International Conference on*, volume 4, pages 3430–3435. IEEE, 2004.

[6] M. Bosse and R. Zlot. Continuous 3d scan-matching with a spinning 2d laser. In *IEEE International Conference on Robotics and Automation (ICRA).*, pages 4312–4319. IEEE, 2009.

[7] Marcus A. Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Conference on Computer Vision and PatternRecognition (CVPR)*, 2013.

[8] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A general solution to the P4P problem for camera with unknown focal length. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[9] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010.

[10] Q Chen, U Ozguner, and K Redmill. Ohio State University at the 2004 DARPA Grand Challenge: developing a completely autonomous vehicle. *Intelligent Systems, IEEE*, 19(5):8–11, 2004.

[11] Hakyoung Chung, Lauro Ojeda, and Johann Borenstein. Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope. *Robotics and Automation, IEEE Transactions on*, 17(1):80–84, 2001.

[12] Winston Churchill. *Experience Based Navigation: Theroy, Practice and Implementation*. PhD thesis, University of Oxford, 2012.

[13] Amaury Dame and Eric Marchand. A new information theoretic approach for appearance-based navigation of non-holonomic vehicle. *IEEE Int. Conf. on Robotics and Automation, ICRA'11*, pages 1–6, February 2011.

[14] Amaury Dame and Eric Marchand. Mutual Information-Based Visual Servoing. *Robotics, IEEE Transactions on*, (99):1–12, 2011.

[15] DARPA. Robotics challenge. *http://www.theroboticschallenge.org*, 2013.

[16] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007.

[17] Albert Diosi and Lindsay Kleeman. Fast laser scan matching using polar coordinates. *The International Journal of Robotics Research*, 26(10):1125–1153, 2007.

[18] Hang Dong, Sean Anderson, and Timothy D Barfoot. Two-axis scanning lidar geometric calibration using intensity imagery and distortion mapping. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3672–3678. IEEE, 2013.

[19] Dragomir Anguelov et al. Google street view: Capturing the world at street level. *Computer*, 43:32–38, 2010.

[20] Olivier D Faugeras and Francis Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(03):485–508, 1988.

[21] Pedro Fernandes and Urbano Nunes. Platooning of autonomous vehicles with intervehicle communications in sumo traffic simulator. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1313–1318. IEEE, 2010.

[22] M Fischler and R Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[23] Georgios Floros, Benito van der Zander, and Bastian Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[24] Christian Frueh, Siddharth Jain, and Avideh Zakhor. Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. *International Journal of Computer Vision*, 61(2):159–184, 2005.

[25] Yasutaka Fuke and Eric Krotkov. Dead reckoning for a lunar rover on uneven terrain. In *Robotics and Automation. Proceedings. IEEE International Conference on*, volume 1, pages 411–416. IEEE, 1996.

[26] Paul Furgale and Tim Barfoot. Stereo mapping and localization for long-range path following on rough terrain. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4410–4416. IEEE, 2010.

[27] Paul Furgale and Timothy D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010.

[28] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. *IEEE Int. Conf. on Robotics and Automation*, 2012.

[29] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010.

[30] Wolfram Burgard Giorgio Grisetti, Cyrill Stachniss. Non-linear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10:428–439, 2009.

[31] Roland Glowinski. *Lectures on Numerical Methods for Non-Linear Variational Problems*. Springer, 2008.

[32] Jacob Goldberger, Shiri Gordon, and Hayit Greenspan. An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 487–493. IEEE, 2003.

[33] Google. Protocol buffers. *http://code.google.com/apis/protocolbuffers*, 2010.

[34] Neil Gordon, B Ristic, and S Arulampalam. Beyond the Kalman filter: Particle filters for tracking applications. *Artech House*, 685, 2004.

[35] Yang Guo and Xin-He Xu. An analytic solution for the P5P problem with an uncalibrated camera. *Chinese Journal Of Computers-Chinese Edition-*, 30(7):1195, 2007.

[36] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.

[37] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[38] Alastair Harrison and Paul Newman. High quality 3d laser ranging under general vehicle motion. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 7–12. IEEE, 2008.

[39] Alastair Harrison and Paul Newman. Ticsync: Knowing when things happened. *Proc. IEEE International Conference on Robotics and Automation*, 2011.

[40] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, April 2004.

[41] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988.

[42] Peter J Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985.

[43] Levinson Jesse and Sebastian Thrun. Automatic calibration of cameras and lasers in arbitrary scenes. *International Symposium on Experimental Robotics*, 2012.

[44] Hailin Jin, P. Favaro, and S. Saotto. Real-time 3d motion and structure of point features: a front-end system for vision-based control and interaction. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 778–779 vol.2, 2000.

[45] Edward Johns and Guang-Zhong Yang. Dynamic scene models for incremental, long-term, appearance-based localisation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[46] Konolige. K and Agrawal. M. FrameSLAM: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24:1066–1077, 2008.

[47] Abdallah Kassir and Thierry Peynot. Reliable automatic camera-laser calibration. In *Australasian Conference on Robotics and Automation*, 2010.

[48] Jonathan Kelly and Gaurav S Sukhatme. An experimental study of aerial stereo visual odometry. In *Proc. Symp. Intelligent Autonomous Vehicles, Toulouse, France*, 2007.

[49] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[50] Laurent Kneip, Margarita Chli, and Roland Siegwart. Robust real-time visual odometry with a single camera and an imu. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2011.

[51] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable SLAM system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

[52] K. Konolige, M. Agrawal, and Joan Solà. Large scale visual odometry for rough terrain. In *Proc. International Symposium on Research in Robotics (ISRR)*, November 2007.

[53] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A General Framework for Graph Optimization. *Proceedings of the International Conference on Robotics & Automation (ICRA) 2011*, pages 1–7, February 2011.

[54] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013.

[55] Quoc V. Le and Andrew Y. Ng. Joint calibration of multiple sensors. *Intelligent Robots and Systems*, 2009.

[56] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnp: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[57] J Levinson, M Montemerlo, and S Thrun. Map-based precision vehicle localization in urban environments. *Proceedings of the Robotics: Science and Systems*, 2010.

[58] Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4372–4378. IEEE, 2010.

[59] Jesse Levinson and Sebastian Thrun. Unsupervised calibration for multi-beam lasers. In *Experimental Robotics*, pages 179–193. Springer, 2014.

[60] Mingyang Li, Byung Hyung Kim, and Anastasios I Mourikis. Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera.

[61] Steven Lovegrove, Andrew J Davison, and Javier Ibanez-Guzmán. Accurate visual odometry from a rear parking camera. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 788–793. IEEE, 2011.

[62] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[63] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.

[64] Philip F McLauchlan and Allan Jaenicke. Image mosaicing using sequential bundle adjustment. *Image and Vision Computing*, 20(9):751–759, 2002.

[65] Colin McManus, Winston Churchill, Ashley Napier, Ben Davis, and Paul Newman. Distraction suppression for vision-based pose estimation at city scales. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[66] C. Mei. Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *Robotics, IEEE Transactions on*, 24(6):1352 – 1364, 2008.

[67] Christopher Mei, Gabe Sibley, Mark Cummins, Paul Newman, and Ian Reid. A constant time efficient stereo SLAM system. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, September 2009.

[68] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[69] David L Mills. Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on*, 39(10):1482–1493, 1991.

[70] Hans Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Computer Science Department, Stanford University, March 1980.

[71] Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Accurate non-iterative O(n) solution to the PnP problem. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[72] A Napier, G Sibley, and P Newman. Real-time bounded-error pose estimation for road vehicles using vision. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1141–1146, 2010.

[73] Ashley Napier, Peter Corke, and Paul Newman. Cross-calbration of push-broom 2d lidars and cameras in natural scenes. In *Proceedings of 2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013.

[74] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[75] Paul Michael Newman. MOOS-mission orientated operating suite. *Massachusetts Institute of Technology, Tech. Rep*, 2299(08), 2008.

[76] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652–I–659 Vol.1, 2004.

[77] David Nistér. *Automatic dense reconstruction from uncalibrated video sequences*. PhD thesis, KTH, 2001.

[78] David Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.

[79] Takayuki Ohno, Akihisa Ohya, and S Yuta. Autonomous navigation for mobile robots referring pre-recorded image sequence. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 672–679. IEEE, 1996.

[80] Taragay Oskiper, Zhiwei Zhu, Supun Samarasekera, and Rakesh Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[81] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Extrinsic calibration of a 3d laser scanner and an omnidirectional camera. *Intelligent Autonomous Vehicles*, 2010.

[82] Gaurav Pandey, James R. McBride, Silvio Savarese, and Ryan M. Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. *In Proceedings of the AAAI National Conference on Artificial Intelligence*, 2012.

[83] Sooho Park and Sejoon Lim. Fast shadow detection for urban autonomous driving applications. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1717–1722. IEEE, 2009.

[84] Valentine Peretroukhim. Optimizing camera perspective in stereo visual odometry. Master's thesis, University Of Toronto - Faculty Of Engineering, 2013.

[85] Mordohai. Philippos, Frahm. Jan-Michael, Akbarzadeh. A, et al. Real-time video-based reconstruction of urban environments. *ISPRS Working Group*, 4, 2007.

[86] O Pink and F Moosmann. Visual features for vehicle localization and ego-motion estimation. *Intelligent Vehicles Symposium, IEEE*, pages 254–260, 2009.

[87] O Pink and C Stiller. Automated Map Generation from Aerial Images for Precise Vehicle Localization . *Inteligent Transport Systems Converence (ITSC) 2010*, pages 1517–1522, August 2010.

[88] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8):986–1004, 2003.

[89] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.

[90] Kümmerle. R, Steder. B, Dornhege. C, and Kleiner. A. Large scale graph-based SLAM using aerial images as prior information. *Proc. of Robotics: Science and Systems*, 2009.

[91] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[92] Stergios I Roumeliotis, Andrew E Johnson, and James F Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4326–4333. IEEE, 2002.

[93] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. *Intelligent Robots and Systems, IROS. IEEE/RSJ International Conference on*, 2007.

[94] Ahmed Shaker, Wai Yeung Yan, and Nagwa El-Ashmawy. The effects of laser reflection angle on radiometric correction of the airborne lidar inensity data. *International Society for Photogrammetry and Remote Sensing*, 2011.

[95] CE Shannon. The mathematical theory of communication. *Bell system technical journal*, 27:379–423, 1948.

[96] Mark Sheehan, Alastair Harrison, and Paul Newman. Self-calibration for a 3d laser. *The International Journal of Robotics Research*, 2012.

[97] Mark Sheehan, Alastair Harrison, and Paul Newman. Continuous vehicle localisation using sparse 3d sensing, kernelised renyi distance and fast gauss transforms. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2013)*, Tokyo, Japan, November 2013.

[98] G Sibley. *Long range stereo data-fusion from moving platforms*. PhD thesis, University of Southern California, 2007.

[99] G Sibley, C Mei, I Reid, and P Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. *The International Journal of Robotics Research*, 2010.

[100] G Sibley, G Sukhatme, and L Matthies. Constant time sliding window filter SLAM as a basis for metric visual perception. *International Conference on Rotobics and Automation (ICRA) Workshop*, Jan 2007.

[101] Piotr Skrzypczynski. How to recognize and remove qualitative errors in time-of-flight laser range measurements. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2958–2963. IEEE, 2008.

[102] F Steinbrucker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 719–722. IEEE, 2011.

[103] Alex Stewart and Paul Newman. Laps - localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds. *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2012.

[104] Dennis Strelow and Sanjiv Singh. Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195, 2004.

[105] N Sunderhauf, Marcus Obst, Gerd Wanielik, and Peter Protzel. Multipath mitigation in GNSS-based localization using robust optimization. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 784–789. IEEE, 2012.

[106] Lixin Tang and Shin'ichi Yuta. Vision based navigation for mobile robots in indoor environment by teaching and playing-back scheme. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 3072–3077. IEEE, 2001.

[107] J-P Tardif, Yanis Pavlidis, and Kostas Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538. IEEE, 2008.

[108] S Thrun. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 2006.

[109] Sebastian Thrun, Wolfram Burgard, Dieter Fox, et al. *Probabilistic robotics*, volume 1. MIT press Cambridge, 2005.

[110] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[111] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment a modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 2000.

[112] Chris Urmson, J Andrew Bagnell, Christopher R Baker, Martial Hebert, Alonzo Kelly, Raj Rajkumar, Paul E Rybski, Sebastian Scherer, Reid Simmons, Sanjiv Singh, et al. Tartan racing: A multi-modal approach to the darpa urban challenge. Technical report, 2007.

[113] Felix von Hundelshausen and Rahul Sukthankar. D-nets: Beyond patch-based image descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2941–2948. IEEE, 2012.

[114] Jung-Ming Wang, Yun-Chung Chung, CL Chang, and Sei-Wang Chen. Shadow detection and removal for traffic images. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 1, pages 649–654. IEEE, 2004.

[115] Michael Warren, David McKinnon, and Ben Upcroft. Online calibration of stereo rigs for long-term autonomy. In *Proceedings of the International Conference on Robotics and Automation*, 2013.

[116] William M Wells III, Paul Viola, Hideki Atsumi, Shin Nakajima, and Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, March 1996.

[117] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(10):965–980, 1992.

[118] Paul R Wolf and Charles D Ghilani. *Elementary surveying: An introduction to geomatics*, volume 10. Prentice Hall Englewood Cliffs, New Jersey, 2002.

[119] Enzhan Zhang, Weili Jiang, Yujun Kuang, and MA Umer. Active RFID positioning of vehicles in road traffic. In *Communications and Information Technologies (ISCIT), 2011 11th International Symposium on*, pages 222–227. IEEE, 2011.

[120] Qilong Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder. *Intelligent Robots and Systems (IROS). Proceedings. IEEE/RSJ International Conference on*, 2004.

[121] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.