Automatic Markerless Calibration of Multi-Modal Sensor Arrays

Zachary Jeremy Taylor

A thesis submitted in fulfilment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

Submitted August 2015; revised October 2015

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Zachary Jeremy Taylor

25 October 2015

Abstract

Zachary Jeremy Taylor
The University of Sydney

Doctor of Philosophy
October 2015

Automatic Markerless Calibration of Multi-Modal Sensor Arrays

This thesis presents a novel system for calibrating the extrinsic parameters of an array of cameras, 3D lidars and GPS/INS sensors without the requirement for any markers or other calibration aids.

To achieve this, a new multi-modal metric, the gradient orientation measure is first presented. This metric operates by minimising the misalignment of gradients between the outputs of two candidate sensors and is able to handle the inherent differences in how sensors of different modalities perceive the world.

This metric is successfully demonstrated on a range of calibration problems, however to calibrate the systems in a reliable manner the metric requires an initial estimate to the solution and a constrained search space. These constraints are required as repeated and similar structure in the environment in combination with the limited field of view of the sensors result in the metric's cost function being non-convex. This non-convexity is an issue that affects all appearance-based markerless methods.

To overcome these limitations a second cue to the sensors' alignment is taken, the motion of the system. By estimating the motion that each individual sensor observes, an estimate of the extrinsic calibration of the sensors can be obtained. In this thesis standard techniques for this motion-based calibration (often referred to as hand-eye calibration) are extended by incorporating estimates of the accuracy of each sensor's readings. This allows the development of a probabilistic approach that calibrates all sensors simultaneously. The approach also facilities the estimation of the uncertainty in the final calibration. Finally, this motion-based approach is combined with appearance-based information to build a novel calibration framework. This framework does not require initialisation and can take advantage of all available alignment information to provide an accurate and robust calibration for the system.

Acknowledgements

Firstly I would like to thank Juan Nieto, whose supervision and knowledge has been invaluable in shaping this thesis. I would also like to thank David Johnson and the other Academics and Staff within the ACFR.

Thank you to all my fellow PhD students, with special mention of Andrew, Rishi, Victor and Lloyd for the lunchtime discussions. I am also grateful to my friends and family for their support.

Finally I wish to thank the Rio Tinto Centre for Mine Automation, the Australian Centre for Field Robotics and the University of Sydney for supporting my research.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

List of Symbols

$\mathcal{L}$**:** Likelihood function

s: Scale factor

$\sigma^{\mathbf{2}}$**:** Variance

cov(**X**)**:** Covariance matrix of X

R: Rotation matrix

A: Rotational axis

θ**:** Angle of rotation

ω**:** Rotational velocity

τ**:** Timing offset

t: Translation vector

T: Transformation matrix

$\mathbf{T}^{\mathbf{x}}_{\mathbf{y}}$**:** The transformation from sensor x to sensor y.

$\mathbf{T}^{\mathbf{x,a}}_{\mathbf{y,b}}$**:** The transformation from sensor x at timestep a to sensor y at timestep b

$\mathbf{T}^{\mathbf{x,a}}_{\mathbf{y,b}}{}_{\mathbf{m,n}}$**:** The element in the m'th row and n'th column of the transformation from sensor x at timestep a to sensor y at timestep b

Note, the various indexing used on the transformation matrix can also be applied to all other symbols.

Throughout this thesis a right-handed Cartesian co-ordinate system is assumed.

All rotations are given in degrees, and all distances are in metres. Roll is taken to be rotation about the X axis, pitch about the Y axis and yaw about the Z axis. The rotation sequence is Yaw, Pitch, Roll (ZYX). When not specified otherwise, for a ground vehicle we take Z to be in the direction normal to the ground plane, X to be aligned with the vehicle's prominent direction of motion and Y perpendicular to this. This can usually be thought of simply as X is forward, Y is left and Z is up.

Any transformation or rotation conventions not specified in this thesis conform to the conventions given by the Matlab 2015 Aerospace Toolbox [45].

List of Acronyms

Hardware

CPU	Central Processing Unit
GPU	Graphics Processing Unit

Imaging

CCD	Charge-Coupled Device
IR	InfraRed
RGB	Red Green Blue
SWIR	Short Wave InfraRed
VNIR	Visible and Near InfraRed

Institutions

ACFR	Australian Centre for Field Robotics
KITTI	Karlsruhe Institute of Technology and Toyota technology Institute
RTCMA	Rio Tinto Centre for Mine Automation

Medical

CT	Computed Tomography
MRI	Medical Resonance Imaging

Metrics

GOM	Gradient Orientation Measure
ICP	Iterative Closest Point
IM	Intensity-Motion
MI	Mutual Information
NMI	Normalised Mutual Information
SIFT	Scale Invariant Feature Transform
SSD	Sum of Squared Differences

Optimisation

CMA-ES	Covariance Matrix Adaptation Evolution Strategy
MAPSAC	Maximum A Posteriori SAmple Consensus
NEWUOA	NEW Unconstrained Optimisation Algorithm
RANSAC	RANdom SAmple Consensus

Sensors

GPS	Global Positioning System
GPS/INS	Global Positioning System and Inertial Navigation System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
RTK	Real-Time Kinematic

Statistics

PDF	Probability Density Function

Glossary

Throughout this thesis we will look at both registration and calibration problems. We view these as two related yet distinct problems and define them as follows:

Registration The mapping of data from one sensor to the data from another sensor. The goal of registration is to map the data and may make use of techniques such as affine transforms or local warping whose parameters would only be valid for the scan they were used on.

Calibration The calculation of the parameters that define the sensor, for example in a camera, the 3D location of the Charge-Coupled Device (CCD) and the parameters of the lens. The goal of calibration is to model the sensors such that any data they receive can be mapped onto the data of any other sensor.

Alignment A more general term that encompasses both registration and calibration.

Chapter 1

Introduction

Sensor calibration is a vital step in the operation of a wide range of sensor systems. An accurate calibration is especially important when multiple sensors of different modalities are utilised. This importance is because in the multi-modal case, each sensor only provides part of the required information about an area of interest. Without calibration, the system would be unable to combine the information given by each sensor and would have a far less informative representation of the surroundings.

This thesis is concerned with the calibration of multi-modal sensors in a robust and automatic manner without the requirement for any special markers or other calibration aids being present in the scene. To this end the thesis examines different cues that provide information about the calibration parameters of the sensors. The result of this research has led to the development of methods via which calibration can be performed in the field by non-expert users, while placing minimal constraints on the configuration of the sensors being calibrated.

1.1 Motivation

Due to advances in technology the price, size and power requirements of a large range of sensors have begun to rapidly decrease. When this is combined with the steady

improvements in the capability of the sensors, it is now becoming common for multiple sensors of different modalities to be used for intelligent perception tasks. These tasks can range from mobile robotics, where navigation sensors are combined with cameras and lidars to allow the exploration of an environment, to fields such as geology, where high-resolution range sensors are combined with hyper-spectral cameras to analyse the mineral content in rock faces. As an example of the benefits that fusing multiple sensor modalities can give, consider the raw sensor data shown in Figure 1.1 compared to the fused representation given in Figure 1.2.

In all of these fields, before these different sensor modalities can be combined in a meaningful way, the data must be aligned so that a point in one sensor's frame of reference can be projected into another sensor's. These sensors are typically manually calibrated by either placing markers in the scene or by hand labelling control points in the sensor outputs [19, 70, 96, 103]. These types of methods have significant limitations. They suffer from being time consuming and labour intensive; the method used is generally only applicable to a single pair of sensor types, it requires the sensors utilised to have an overlapping field of view, and requires a user with a significant level of technical knowledge of the sensor operations. Furthermore, in some circumstances these manual methods may give results that contain significant error [35].

Traditionally, while the sensors were only utilised on expensive experimental robots that operated in constrained environments and were maintained by a team of experts, manual calibration methods were a viable strategy for aligning a system's sensors. While tedious, the calibration was not onerous for the staff and could conceivably last the lifetime of the robot. However, as these systems begin to be utilised by end users in unconstrained environments, such as farms and mine sites, due to factors such as rough motion, collisions, repairs or modifications, recalibration of the sensors may be required on a semi-regular basis. This calibration may also need to be performed by people with only a basic understanding of the sensor's operating principals. Another consideration is that, due to the low price point of some sensors (particularly IMU, GPS and Cameras) the entire system may have been assembled by a non-expert, who has little knowledge of the methodology required to calibrate the sensors.

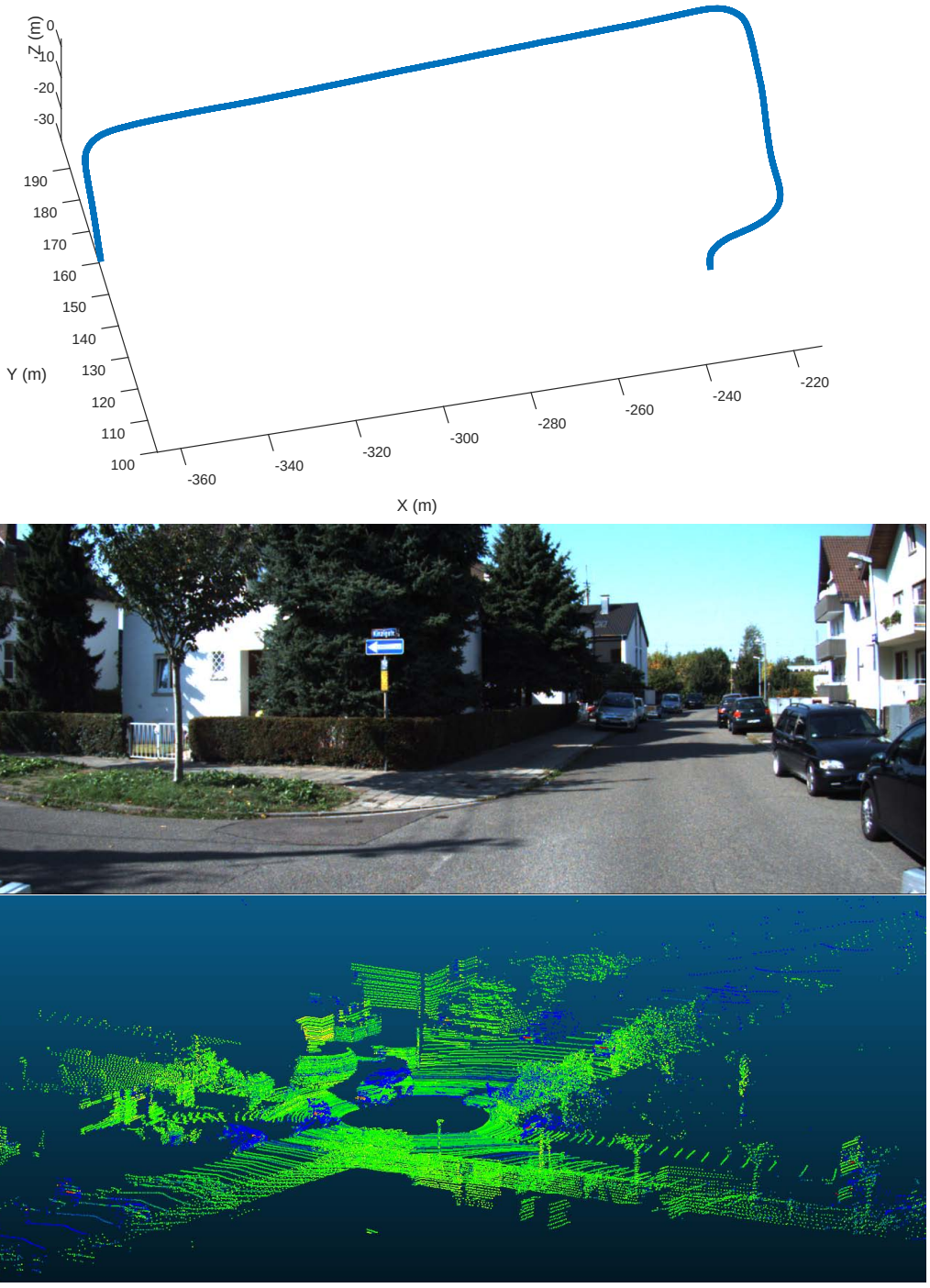


Figure 1.1 – Data taken from a section of the KITTI dataset. Top: an INS/GPS plot of the vehicle's movements. Middle: The view from one of the vehicle's mounted cameras. Bottom: One of the point clouds generated by the vehicle's 3D lidar.

Figure 1.2 – The fusion of all of the information shown in Figure 1.1 into a single coloured 3D map of the area. The fused map provides a far richer view of the environment than any one sensor could generate alone.

Because of this, there is a steadily increasing need for methods that can automatically calibrate the sensors, without the requirement of any markers or human interaction. The aim of this thesis is to analyse the observations obtained by sensors of different modalities during their typical operation and use this information to develop methods via which the sensor outputs may be aligned.

This registration, utilising the surrounding environment, also introduces new issues as, for any sensor and metric combination it is possible for there to be an environment in which one or more of the sensor's parameters is either unobservable or barely observable. In the unobservable case the method will fail to give a calibration and the user or robotic system can take action accordingly. However in the barely observable case, the calibration will be heavily influenced by the noise in the sensors, resulting in an exceptionally poor calibration. To account for this issue, we also develop methods via which an indication of the calibration uncertainty can be obtained.

1.2 Contributions

Specifically this thesis makes the following contributions:

- The development of a new multi-modal appearance-based metric, the Gradient Orientation Measure (GOM). This metric operates by aligning the gradients present in different sensor modalities.
- An analysis of the issues surrounding appearance-based metrics including the difficulty of their optimisation and accuracy estimation.
- A robust method for temporally aligning sensor outputs using the magnitude of the observed angular ego-motion. The method can operate on any number of sensors, is robust to outliers and considers the accuracy of the input readings.
- An extension of traditional hand-eye calibration techniques into a robust probabilistic framework that considers the uncertainty of each sensor's readings. This

formulation allows the technique to be applied to multiple sensors simultaneously and for the uncertainty in the final calibration to be evaluated.

- The combination of appearance-based calibration and motion-based calibration. This is done by using the motion-based calibration to constrain and guide the optimisation of the appearance-based metrics, improving the robustness of their calibration.

- The development of a second new multi-modal metric, the Intensity-Motion (IM) metric for aligning lidar with cameras. This metric is designed for use with mobile vehicle-based systems and utilises both appearance and motion characteristics in its refinement.

- Extensive evaluations of all methods presented with different datasets and comparisons with state-of-the-art algorithms.

Many of these contributions have resulted in publications. Our initial examination of the lidar-camera calibration problem is presented in [85]. The development of GOM was first presented in [89]. Additional experimentation and examination of the nature of GOM's search space led to the publication of [90] and a brief overview of the entire system was also presented in the workshop paper [86]. We first examined motion-based calibration with an initial workshop paper [87]. The process presented in this paper underwent significant revision and improvement before being presented in [88] where we also considered metrics that utilised both motion and appearance information.

During our research, several of the methods developed were utilised to align data for a range of applications. In [50] and [51], 3D lidar scans were combined with hyperspectral images to estimate the minerals present in a cliff face. Finally, in [29] Red Green Blue (RGB) and InfraRed (IR) images of almond trees were aligned to allow for the detection of almonds.

All of the source code used in generating the results found in this thesis has been made publicly available at [84].

1.3 Thesis Structure

Chapter 2 presents an overview of the existing literature in the field of multi-modal sensor calibration. The work is divided into two main areas, appearance-based and motion-based calibration.

Chapter 3 examines appearance-based calibration in further detail, mainly focusing on the calibration of lidar-camera systems. In this chapter we analyse several state-of-the-art methods and present our metric, GOM. We also present a framework for its implementation in calibrating these systems.

Chapter 4 looks into estimating the accuracy of the resulting calibrations. Several approaches are examined with their limitations, computational times and other relevant properties considered. In this chapter we also explore the inherent limitations of the accuracy analysis that can be performed when using markerless calibration techniques.

Chapter 5 departs from appearance-based techniques and analyses methods that can capture calibration information from the motion of the system. The advantages and disadvantages of this approach are presented. Traditional hand-eye calibration techniques are then examined and extended to allow the incorporation of the uncertainty of readings, multiple sensors and outliers in a probabilistic framework. This framework also provides an estimate of the overall confidence in the calibration. Finally, this chapter considers the combination of motion-based techniques with appearance-based techniques to give a method that exhibits the strengths of both approaches.

Chapter 6 presents the experimental results obtained by the methods outlined in Chapters 3, 4 and 5, and compares these results to several of the state-of-the-art techniques examined in Chapter 2.

Chapter 7 summarises the contributions made and looks towards future work before concluding this thesis.

Appendices

Appendix A gives a series of applications that the work from this thesis has been applied to.

Appendix B compares the results obtained in this thesis with those in the literature, and also examines the reasons why different authors report significantly different results when using the same methods.

Appendix C gives a brief overview of some of the practical considerations encountered when fusing the aligned data.

Appendix D looks at the different conventions for representing rotations.

Appendix E briefly outlines the finite difference methods used in this thesis.

Appendix F examines the trade-offs and limitations of the Monte Carlo and delta method.

Appendix G looks at several options for mitigating the influence of outliers.

Chapter 2

Literature Review

The literature on multi-modal sensor calibration can be divided into two distinct groups. The alignment of sensors based on the appearance of the surrounding environment, and alignment based on the motion experienced by the sensors.

2.1 Appearance-Based Metrics

A large body of work exists for the calibration of sensors based on appearance information. To provide an in-depth review of the methods most relevant to our proposed techniques, we have limited the scope of this section to methods that are both multi-modal and markerless. To highlight the similarities and differences among these methods we further divide the appearance-based metrics into four sections. First, we present a description of mutual information, followed by three application-based sections. We review methods that operate on two dense images, then methods that match a single high-resolution point cloud to an image, and finally, methods that work on data gathered from a moving platform and optimise over a large number of data frames.

2.1.1 Mutual Information

Many of the common multi-modal image matching techniques are based on a metric providing statistical dependency between two signals known as Mutual Information (MI). This technique is widely used in medical image registration and a survey of MI-based techniques has been presented by Pluim et al. [65]. MI is at the core of many multi-modal registration techniques, including methods presented in all three of the following sections of this literature review and two of the methods evaluated in this thesis. Due to this widespread application, we begin our examination of the literature with a brief description of the technique.

MI was first developed by Shannon [74] in information theory using the idea of Shannon entropy, which is a measure of how much information is contained in a signal. Its discrete version is defined as:

$$H(X) = H(p_X) = \sum_{i=1}^{n} p_i \log(\frac{1}{p_i}) \tag{2.1}$$

where X is a discrete random variable with n elements and the probability distribution $p_X = (p_1, ..., p_n)$.

When two variables are statistically independent, their joint entropy is equal to the sum of their individual entropies. As shown in Equation 2.2, MI uses this fact to give a measure of the signal's dependence by taking the difference between the individual and joint entropy $H(M, N)$.

$$MI(M, N) = H(M) + H(N) - H(M, N) \tag{2.2}$$

To align sensor outputs using MI, the output of one sensor is transformed and mapped onto the other sensor's output. After this mapping has been performed, points that contain information from both sensors are used to generate the MI value. An optimisation strategy is then applied to find the parameters of the transformation that

maximise the given MI score, assuming that this will correspond to the optimal mapping of the data.

There are two main issues with the MI metric. Firstly, it makes no use of any local structural information in the data [5], and secondly, it makes no use of the locations where only one sensor reading exists. This second issue means that, when used for registration, MI can be influenced by the total amount of information contained in images. This causes it to favour mappings that give less overlap, as was shown in [80]. This drawback is somewhat mitigated by using Normalised Mutual Information (NMI), which is defined as:

$$NMI(M,N) = \frac{H(M) + H(N)}{H(M,N)} \tag{2.3}$$

In practice, for images, the required probabilities $p(M)$, $p(N)$ and $p(M,N)$ are typically estimated using a histogram of the distribution of intensity values.

2.1.2 Multi-Modal Image-Image Systems

A vast number of methods have been proposed for solving the problem of multi-modal image matching and the related problem of multi-modal stereo correspondence. Many of these methods were first developed for the alignment of Medical Resonance Imaging (MRI) and Computed Tomography (CT) scans for use in medical imaging [56]. An example of the typical data these methods operate on is shown in Figure 2.1.

The most common methods used in medical image registration are the MI and NMI methods that have already been discussed. However, many other methods exist. The correlation ratio used by Roche et al. [68] provides a measure of the functional dependence of the intensities. A second closely related metric is the correlation coefficient; this is a less general metric providing a measure of the linear dependence of the two images' intensities. A method known as gradient correlation takes the normalised cross-correlation between two gradient images created using a Sobel filter [63]. Zana and Klein [102] used a Hough transform to register retinal images; the vessels in the

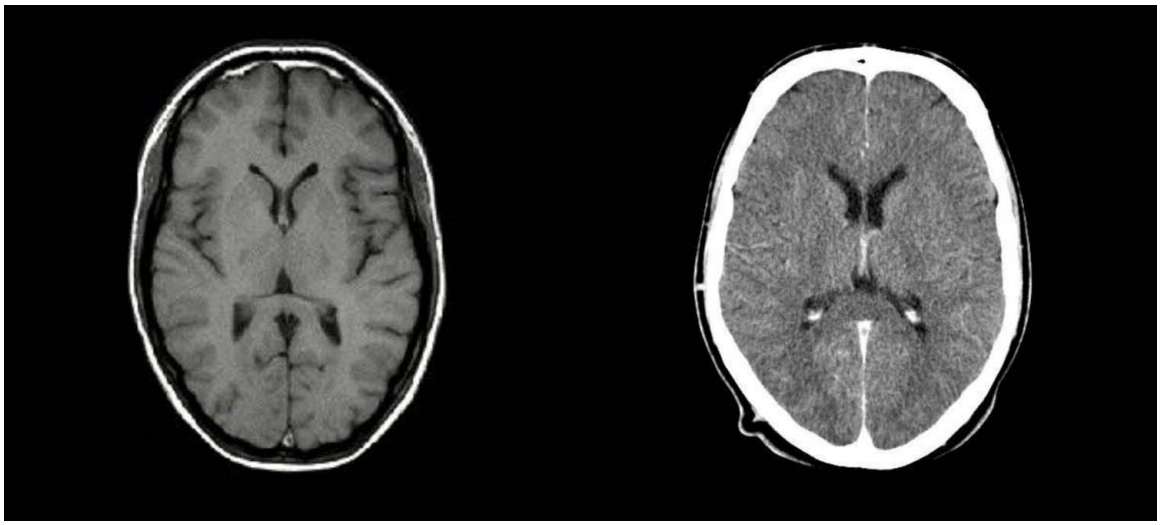


Figure 2.1 – An example of the type of medical data the image-image registration problem typically operates on. Left: A MRI scan of a human brain [41]. Right: a CT scan of a human brain [21].

eye provided strong lines for this method to detect and align. Wachinger and Navab [98] developed a method called entropy Sum of Squared Differences (eSSD) that uses the entropy of patches of the images for registration of T1 and T2 MRI scans. The method works by first creating images where each pixel's intensity is equal to the entropy of the pixels in an n-by-n patch around it. Matching is then performed by taking the Sum of Squared Differences (SSD) of the two generated entropy images. In the majority of these presented medical alignment methods, the metrics used were developed by observing the relationship between intensities in corresponding scans and developing a method to exploit any correlation observed. This means that while the methods operate well in the scenario they were designed for, they may not generalise well to other multi-modal registration problems. The medical images are also obtained in a controlled environment with a much smaller range of targets and scenes than the sensors found on robotic systems. This reduces the need for the metrics to be robust to issues such as outliers, sensor noise, dynamic objects and large offsets that are encountered in robotic applications.

A method known as self-similarity was initially developed by Shechtman and Irani [75] to identify an object in a scene from a rough sketch. It works by assuming that

differently coloured areas in one image will be more likely to be coloured differently in the other modality. Several attempts have been made to use self-similarity for multi-modal image matching, usually with slight changes to the implementation to increase performance. Torabi and Bilodeau [91] used self-similarity to perform multi-modal stereo correspondence between visual and IR images. Heinrich et al. [25] made use of an altered version they called the Modality Independent Neighbourhood Descriptor (MIND) to register MRI with CT scans of the human brain. The main differences between MIND and self-similarity are that the patches were a single pixel in size and no conversion to log-polar bins was made.

Bodensteiner et al. [10] successfully matched images of different modalities using the mono-modal feature descriptor called the Scale Invariant Feature Transform (SIFT) [39]. In most multi-modal applications, however, it is found that the assumptions made by SIFT about the directions and relative magnitudes of the gradients do not hold [25]. To attempt to overcome this issue, Chen and Tian [12] developed a version of SIFT that was based on the absolute value of the gradient so that no distinction was made as to whether the gradient was increasing or decreasing. This method however, is still limited due to SIFT's assumption of only linear changes between images.

2.1.3 Single Laser-Image Scan Systems

These systems operate by matching a single high-resolution scan of an environment with its corresponding image. High-resolution scans are usually produced by mounting a single lidar on a rotating platform and even with current state-of-the-art scanners can take up to an hour to integrate all of the data into a single scan. An example of the type of data aligned by these systems is shown in Figure 2.2.

A method proposed by Li et al. [36] makes use of edges and corners. Their method works by constructing closed polygons from edges detected in both the lidar scan and images. Once the polygons have been extracted, they are used as features and matched to align the sensors. The method was only intended for aerial photos of

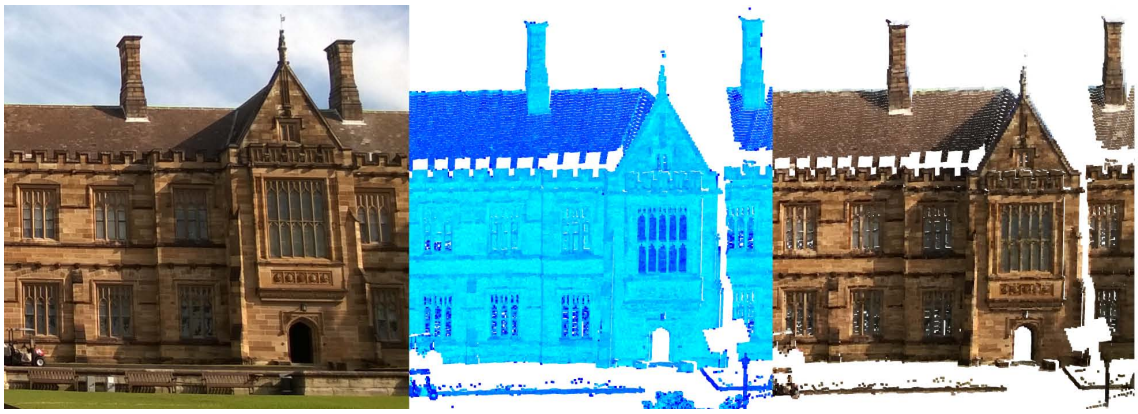


Figure 2.2 – An example of the type of data the single-scan registration problem typically operates on. Left: an image of Sydney University's Great Hall. Middle: a section of a high-resolution lidar scan of the same building. Right: the image projected onto the scan after alignment.

urban environments, where polygons can be easily observed in the data.

Mastin et al. [44] achieved registration of an aerial lidar scan by creating an image from it using a camera model. The intensity of the pixels in the image generated from the lidar scan was either the intensity of the laser return or the height from the ground. The images were compared using the joint entropy of the images and optimisation was done via downhill simplex. The method was only tested in an urban environment where buildings provided a strong relationship between height and image colour.

A method for aligning ground-based lidar scans of cliffs with hyperspectral images of the same area was developed by Nieto et al. [54]. The method makes use of a pre-calibrated second camera that is rigidly attached to the lidar to give RGB colour to the lidar scan points. A camera model is then used to generate a colour image from this laser scan. The hyperspectral camera's image is matched to this generated image by using SIFT features to perform an affine transform on the image. The matching is then further refined using a local warping that utilises the normalised cross-correlation between patches. While this method worked well for the application presented, it had the drawback of requiring a second pre-calibrated camera. In our own earlier work, we developed a method to perform the same task without this limitation [85]. The method operates by creating an accurate camera model that emulates the hyperspec-

tral camera and using it to project the lidar points onto the camera's images. Some of the lidar point clouds did not have usable intensity information, therefore a new intensity was assigned to the points, based on an estimation of the direction of their surface normals. These lidar points were compared to the points in the image that they were projected onto, using NMI. It was assumed that when this measure was maximised, the camera model would have the same parameters as the actual camera used, allowing it to relate each point in the image to its corresponding point in the lidar's output point cloud.

For the alignment of fixed ground-based scans in urban environments, several methods exist that exploit the detection of straight edges in a scene [34, 38]. These straight lines are used to calculate the location of vanishing points in the image. While these methods work well in cities and with images of buildings, they are unable to correctly register natural environments due to the lack of strong straight edges.

A more theoretical view on calibration is presented by Corsini et al. [14] where the authors looked into different techniques for generating a synthetic image from a 3D model so that MI would successfully register the image with a physical photo of the object. They used NEW Unconstrained Optimisation Algorithm (NEWUOA) optimisation in their registration and looked at using combinations of the silhouette, normals, specular map and ambient occlusion to create an image that would robustly be registered with the real image. They found surface normals and a combination of normal and ambient occlusion to be the most effective.

2.1.4 Mobile Systems

While in theory a similar problem to the single lidar scan case explored above, mobile sensing systems typically have some key differences. First, the scanning systems are generally of much lower resolution due to the time constraint on the scans. This lower resolution means that most methods developed for the above systems give poor results on these datasets. An example of the type of data these systems typically operate on is shown in Figure 2.3. A second key difference is that the sensors are

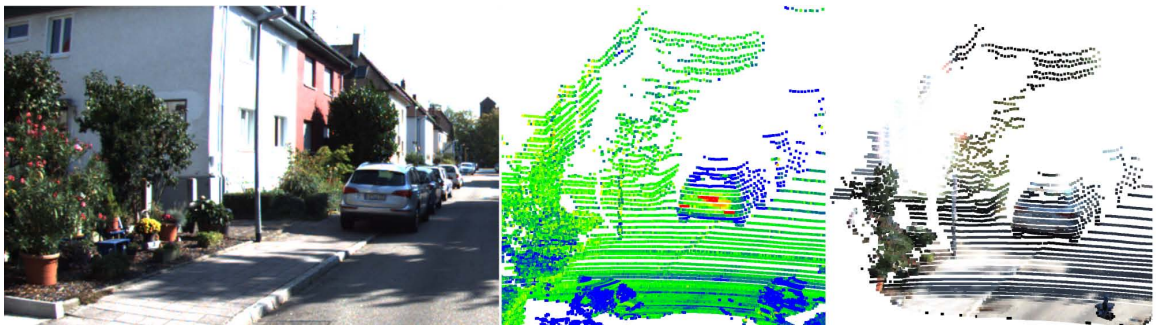


Figure 2.3 – An example of the type of data the multi-scan registration problem typically operates on. Left: an image taken by the KITTI sensor vehicle. Middle: a Velodyne scan of the same area. Right: the image projected onto the scan after alignment.

typically mounted in a rigid, fixed configuration on the sensor vehicle. This permits the use of multiple observations to estimate the calibration parameters.

Widespread availability of 3D lidars capable of operating from a moving platform did not happen until the Velodyne HDL-64E was released in 2007; because of this the previous work in this field is rather limited. One of the first approaches that did not rely on markers was presented by Levinson and Thrun [35]. Their method operates on the principle that depth discontinuities detected by the lidar will tend to lie on edges in the image. Depth discontinuities are isolated by measuring the difference between successive lidar points and removing points with a depth change of less than 30 cm. An edge image is produced from the camera that is then blurred to increase the capture region of the optimiser. The average of all of the edge images is then subtracted from each individual edge image to remove any bias to a region. The two outputs are combined by projecting the isolated lidar points onto the edge image and multiplying the magnitude of each depth discontinuity by the intensity of the edge image at that point. The sum of the result is taken and a grid search used to find the parameters that maximise the resulting metric. A drawback of this method is that the averaging step means the process cannot be directly applied to the single-scan case.

Two very similar methods have been independently developed by Pandey et al. [61] and Wang et al. [100]. These methods use the known intrinsic values of the camera and

estimated extrinsic parameters to project the lidar's scan onto the camera's image. The MI value is then taken between the lidar's intensity of return and the intensity of the corresponding points in the camera's image. When the MI value is maximised, the system is assumed to be perfectly calibrated. The only major difference between these two approaches is in the method of optimisation used; Pandey et al. make use of the Barzilai-Borwein (BB) steepest-gradient ascent algorithm, while Wang et al. make use of the Nelder-Mead downhill simplex method. In both implementations, aggregation of a large set of scans is required for the optimisers used to converge to the global maximum.

More recently an approach was developed by Napier et al. [52] for registering a push broom 2D lidar with a camera. To form an image from the 2D scanner, its scans are first combined with an accurate navigation solution for the mobile system to generate a 3D scan. A 2D image is then produced from this 3D scan using a camera model. The two images have the magnitude of gradients present in them calculated and normalised over a small patch around them. The camera and lidar are assumed to be aligned when the sum of the differences in these gradient magnitude images are minimised. The metric also has an additional weighting that favours areas with higher resolution scans.

The techniques developed for the appearance-based calibration of lidar-camera systems mounted on mobile platforms are the most directly applicable to our area of research. Because of this, several of the techniques outlined in this section are compared to our methods in the experiments carried out in Chapter 6. A comparison of the accuracy of these methods reported by different authors in the literature is also presented in Appendix B.

2.2 Motion-Based Metrics

These methods exploit the motion of rigidly mounted sensors to calibrate the system's parameters. The most common application of these techniques is in the calibration of systems that incorporate robotic arms through the use of 'hand-eye' calibration.

This is not the only application though, as in recent years several methods have also applied this approach to ground vehicles and other sensor systems.

2.2.1 Hand-Eye Calibration

The hand-eye calibration problem is a well known problem in robotics [76, 93, 99]. It is usually expressed as follows: if two points are rigidly connected and both points undergo a series of known transformations, how can the transformation between the two points be recovered? The name is derived from one of the first robotic applications of the problem, where a camera was mounted onto the 'hand' of a robotic arm and the transformation between the 'hand' and the camera or 'eye' needed to be calculated. The problem is also sometimes referred to as $AX = XB$, as if A and B are the transformations the two sensors undergo, X is the transformation between them. A depiction of the calibration process for a robotic arm is shown in Figure 2.4.

In its most basic form the problem is well understood, with techniques developed by Tsai, Lenz and others in the late 80s giving efficient methods that are optimal in the least squared sense [93]. These methods operate by first finding the rotation axis for each transform. All points on a rigid body share the same rotation axis, and so this can be found before the translation is known. Once these axes are found, they can be optimally aligned using an approach known as the Kabsch algorithm [32]. After the rotation has been found, the translation can be obtained using simple linear algebra.

These techniques have the disadvantage of requiring calibrated markers to be placed in the scene to allow the camera's transformations to be calculated. More recently, several methods have made use of visual odometry methods to remove this limitation [2, 26]. As visual odometry utilising a monocular camera has no scale associated with it, this additional parameter must be estimated for every transformation pair.

Another recent modification of the problem was given by Ackerman et al. [1] where a solution is calculated for two sensors that operate asynchronously and provide data at undefined intervals. In this approach, the fact that the magnitude of a rotation is constant for all points on a rigid body is exploited. This magnitude allows the

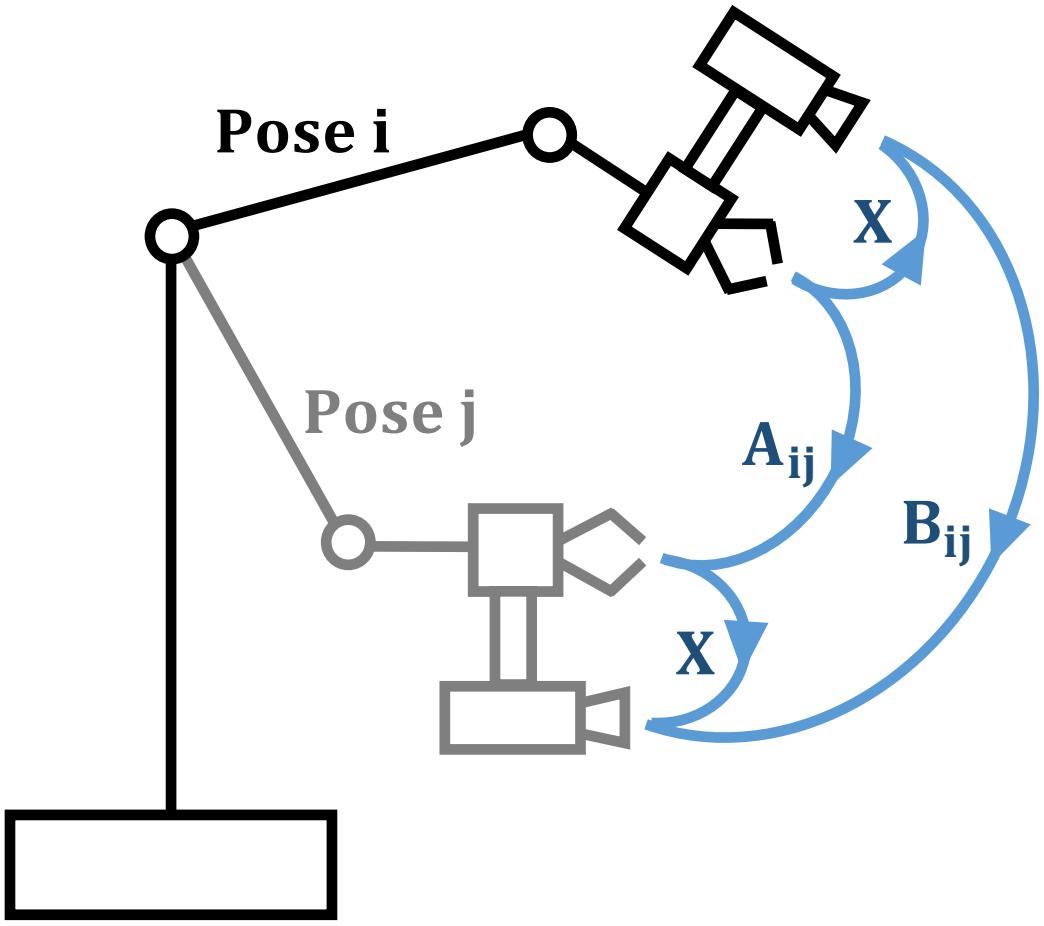


Figure 2.4 – The standard hand-eye calibration problem. A robotic arm is moved through a series of poses. By examining the transformation experienced by the robotic hand A and the Camera (aka the 'Eye') B, the transformation between the camera and hand can be recovered using the relationship $AX = XB$.

authors to find the most likely correspondences of the data. Ovren and Forssen [57] also made use of this in setting the timing between a camera and IMU.

This technique has also been adapted for real-time applications, for example Schneider et al. [72] presented a method that makes use of a Kalman filter to calibrate an INS system with a stereo camera rig. A method for online calibration of a phone's IMU and camera was presented by Li et al. [37]. In this method a large number of parameters are estimated in addition to the extrinsics to account for the imperfect nature of the phone's low-cost sensors.

As well as extrinsic parameters, motion-based methods can be used to calibrate the intrinsic properties of a camera. Keivan and Sibley [33] used an IMU to calibrate the focal length and centre point of a camera. The process also detected if the calibration parameters undergo significant change and in this event recalibrates.

2.2.2 Lidar-GPS Calibration

Underwood [95] utilised the motion of a vehicle to calibrate a Global Positioning System and Inertial Navigation System (GPS/INS) to operate with 2D lidar. To overcome the inherent observability issues of using a 2D lidar, Underwood constrained the environment to a field with a single vertical pole that was used as a target. The authors also considered the variance of the resulting calibration.

2.2.3 Camera-Camera Calibration

In work presented by Heng et al. [27], four cameras with non-overlapping fields of view are calibrated on a vehicle. The method operates by first using visual odometry in combination with the car's motion provided by odometry, to give a coarse estimate of the cameras' position. This is refined by matching points observed by multiple cameras as the vehicle makes a series of tight turns. Bundle adjustment is then performed to refine the camera position estimates. The main limitation of this method

is that it was specifically designed for vision sensors and makes use of feature matching between multiple sensors to refine the calibration.

2.3 Summary

Multi-modal sensor calibration is a problem that is present in a wide range of fields and applications. In many of these fields, the research into the calibration techniques has focused solely on the specific sensor setup they are utilising and generally gives little consideration to the techniques used to calibrate similar setups in unrelated fields. This has resulted in fundamentally different calibration approaches being established for use with medical systems, high resolution lidars, mobile robotic systems and robotic arms. This is reflected in the presentation of the literature where each group of calibration approaches is largely dominated by a single application.

In the coming chapters we will examine the advantages current techniques developed in each of these fields offer, and when possible, combine them. This will lead to a calibration framework that can provide accurate results while placing minimal constrains on the layout and sensors that can be calibrated.

In Chapter 3 an appearance-based metric, GOM, that takes inspiration from both medical and robotic approaches will be presented. It will then be demonstrated that this metric can be used in a large range of lidar-camera and camera-camera calibration problems. After this, Chapter 4 examines methods for estimating the accuracy of the calibration, and in doing so, highlights several shortcomings of the appearance-based approach. To overcome this Chapter 5 looks at the motion-based methods utilised in robotic arm calibration and extends these approaches to consider issues encountered in the mobile robotics domain. Finally, the strengths of both the appearance- and motion-based methods are combined into a single robust framework.

Chapter 3

Appearance-Based Metrics

3.1 Introduction

In order to reliably utilise appearance information to align two sensor outputs of different modalities, an observed location must be unambiguously identifiable in each sensor's output. This requirement significantly limits the sensors that can be calibrated using these techniques. The most common sensors where appearance metrics are used for calibration are cameras of various modalities (IR, RGB, multi-spectral, hyper-spectral, etc.), structured light sensors and lidar sensors. In this chapter we examine the challenges of this calibration problem and present a metric we have developed, the Gradient Orientation Metric (GOM), which can effectively align the output of sensors of this type. Unlike most current calibration approaches, the new metric is also able to calibrate from a single-scan pair. This property makes our approach suitable for a broad range of applications since it is not restricted to calibration based on multiple observations from sensors attached to a rigid mount. For example, our approach is appropriate for registration of multi-temporal data and data collected from non-rigidly mounted sensors. One of the applications that motivated the development of this metric was to align images and lidar scans taken at a mine site. The camera and lidar scanner were operated by two different teams at different times, and the only location provided was through use of a hand-held GPS system. Further

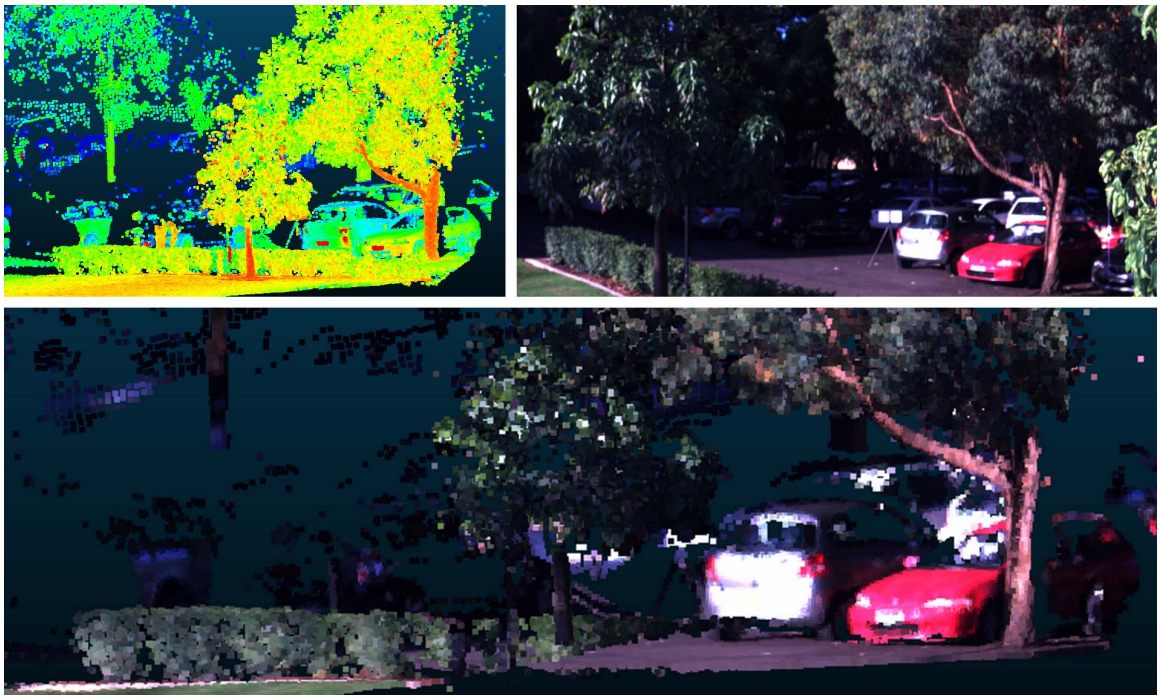


Figure 3.1 – Camera and lidar scan being combined. Top left: Raw lidar data. Top right: the corresponding camera image. Bottom: The textured map obtained after alignment with GOM.

details of this case can be found in Appendix A.1. Figure 3.1 shows an example of the result given after aligning a camera and lidar sensor with the GOM metric.

Specifically this chapter makes the following contributions:

- The formulation of a pipeline for comparing 2D/3D sensor data with information captured by a camera.
- The development of a new multi-modal error metric, the Gradient Orientation Measure (GOM).
- The examination of possible features for use in giving intensity information to 3D data.
- The exploration of the issues surrounding the optimisation of appearance-based multi-modal metrics and their challenging search space.

3.2 Multi-Modal Sensor Approach

Figure 3.2 illustrates the overall idea of our approach to appearance-based multi-modal registration and calibration. The method can be divided into two main stages: feature computation and optimisation.

The feature computation stage converts the sensor data into a form that facilitates comparisons of different alignments during the optimisation stage. The initial step is to assign an intensity value to each data point. For 2D data the average of the colour channels is used. For 3D data, the user selects one of several possible features, usually dependant on the exact sensor and application (the features considered will be presented in Section 3.3). After this is done, histogram equalisation is performed to ensure high contrast in the data. Next, a gradient detector is applied to the data to estimate the intensity and orientation of gradients at each point; the gradient detector used also depends on the dimensionality of the data. The strength of the gradients is histogram equalised to ensure that a significant number of strong gradients are present. This gradient information is finally passed into the optimisation, completing the feature computation step.

The sensors' outputs are aligned during the optimisation. This is done by defining one sensor's output as fixed (called the base sensor output) and transforming the other sensor's output (referred to as the relative sensor output). In our framework, to allow simple mapping of the relative sensor's output onto the base output, we set the base sensor output to always be a dense image. In the case of two 2D images, an affine transform is used, and for 2D-3D alignment, a camera transform is used to project the 3D points of the relative output onto the 2D base output. Once this has been done, the base output is interpolated at the locations that the relative output was projected onto to give the gradient magnitudes and directions at these points.

Finally, GOM is used to compare the gradient features between the two outputs and to provide a measure of the quality of the alignment. This process is repeated for different transformations until the optimal set of parameters is found. For the optimisation, our approach uses particle swarm [47] for one-off data registration, and

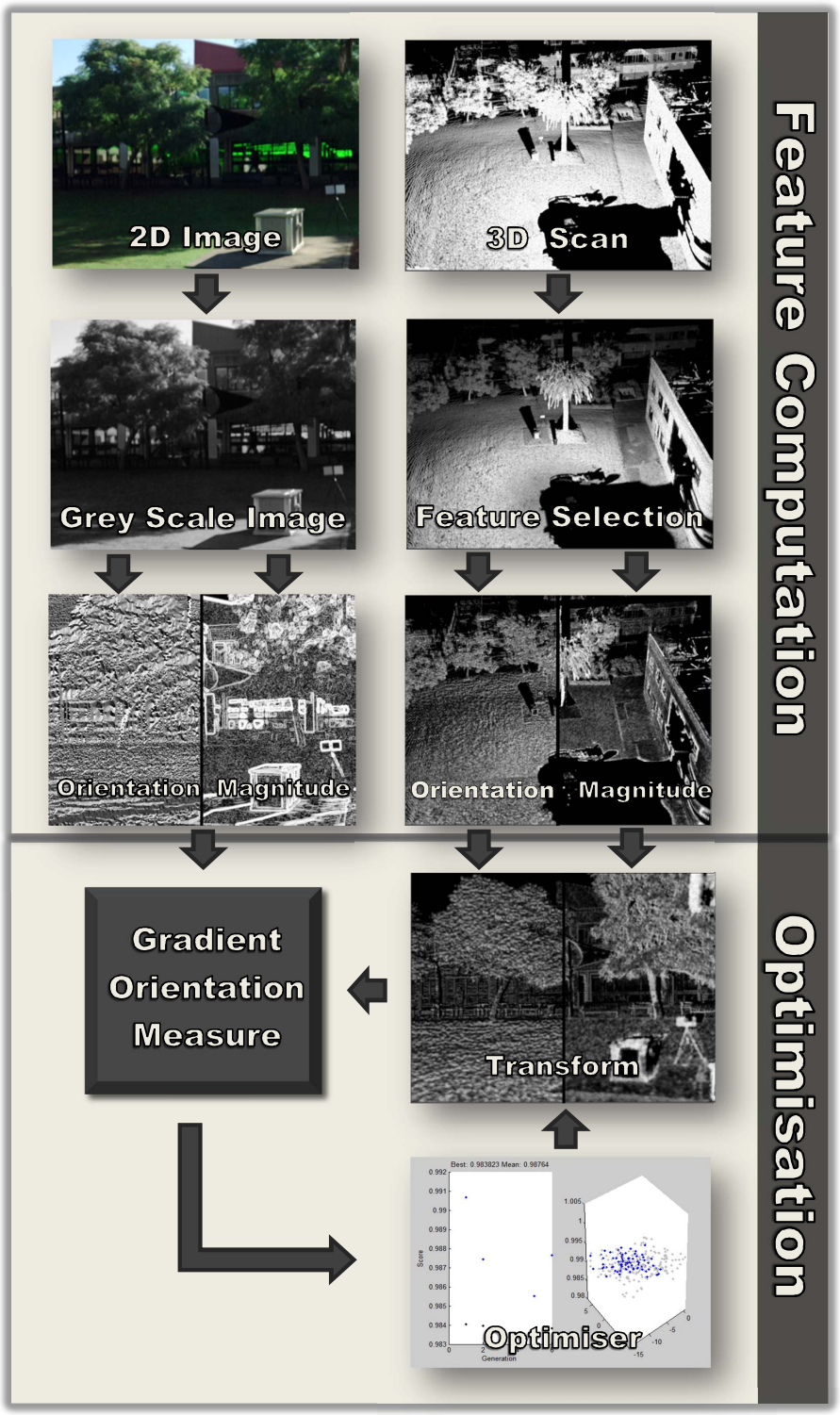


Figure 3.2 – An example of the steps of our approach. This diagram shows the alignment of a camera image with a high-resolution lidar scan coloured by its return intensity.

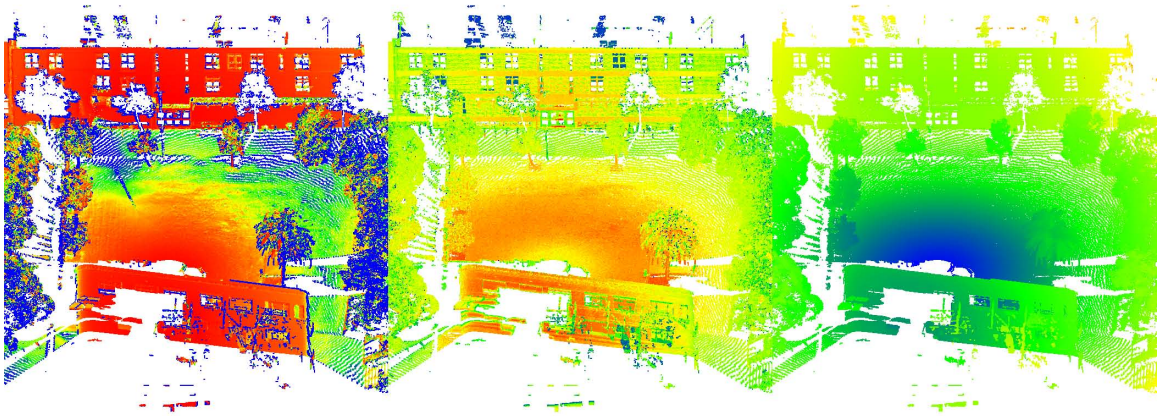


Figure 3.3 – An example of the features used to give intensity information to the lidar scans. Left: normals of points. Middle: return intensity. Right: range.

Nelder Mead simplex [53] for multi-sensor platform calibration. Different optimisers are used as, in the platform calibration case, scans can be aggregated. This aggregation smooths the search space and allows a local optimiser to be used.

3.3 Features

To allow the chosen metric to correctly calibrate the system, there has to be a strong relationship between the intensity of corresponding points. For 3D range sensors, several possible features exist that can be used to set the points' intensities. The features analysed in this work are: (i) the normals of the points, (ii) the return intensity and (iii) the distance of points from the sensors. Histogram equalisation is performed on all features to improve contrast. An example of these features being used to colour a lidar scan is shown in Figure 3.3

Normals of points: One of the most significant factors influencing the appearance of an object in a camera image is the angle between the camera, the object's surface, and any light sources. Because of this, there is a relationship between a point's normals and the camera's pixel intensities. To obtain an estimate of the normals, a plane is approximated at the location of each point. This is done by first placing the points into a k-d tree [8], from which the eight nearest neighbours to each point are

found. The normal vector is calculated from the eigenvectors and eigenvalues of the covariance matrix C, given by Equation 3.1 [69]:

$$C = \frac{1}{8}\sum_{i=1}^{8}(p_i - c)(p_i - c)^T \tag{3.1}$$

where p_i is the i-th nearest neighbour location and c is the location of the point. The eigenvector corresponding to the smallest eigenvalue of C is the best estimate of its normal vector to the plane. Once the normals have been calculated, the three values that make up the normal vector are converted into a single intensity value by calculating the difference in angle between the normal vector and a line between the point and the origin of the scan. While other methods based on the angle of the points can be used, this method was empirically found to give good results.

Return intensity: Lidars provide a measure of the return strength of the laser from each point. This usually gives a strong relationship between the intensity of matching points. This occurs as both laser reflectance and the camera pixel intensity rely on the reflectance of the target material. While most lidar systems will output the intensity reading, it cannot be used in all situations as the intensity of return is dependent on the distance of the object from the sensor. This means that when multiple scans from different locations are combined, the intensity readings associated with a point will depend on which lidar recorded it. A second issue with this method occurs when using systems that make use of multiple lasers. In these systems, each laser scans a different section of the environment. For example, in the Velodyne HDL-64E, the scan is built up by rotating its head containing 64 separate lasers [30]. Each of these lasers has slightly different characteristics and can give substantially different intensity of returns for the same object [79].

Range: The distance from the scanner to a point is a simple, fast, and often effective way of generating intensity values for 3D points. The relationship between point distance and intensity, however, is not as strong as for the other features. It operates on the assumption that, in environments with a large number of distinct items, a relationship between the distance to each item and the colours of that item will

typically exist. This feature works best when used in fairly cluttered scenes with a large number of objects at different distances from the camera, such as on a busy street or in a garage. In open environments such as fields or highways, however, the method generally fails due to a lack of sharp changes in depth.

3.3.1 Transformation

The transformation applied to align the sensors' outputs depends on the dimensionality of the two sensors. If one sensor outputs 3D data, for example a lidar, and the other sensor is a camera, then a camera model is used to transform the 3D output. If both sensors provide a dense 2D image, then an affine transform is used to align them.

Camera Models

To convert the data from a list of 3D points to a 2D image that can be compared to another sensor's image, the points are first passed into a transformation matrix that aligns the sensor's axis. After this has been performed, one of two basic camera models is used. For most sensors, we utilise a pin-hole camera model that is defined as:

$$x_{cam} = x_0 - \frac{cx}{z} \tag{3.2}$$

$$y_{cam} = y_0 - \frac{cy}{z} \tag{3.3}$$

Where:

x_{cam} , y_{cam} are the X and Y position of the point in the image.

x, y, z are the coordinates of points in the environment.

c is the principle distance of the model.

x_0 , y_0 are the location of the principle point in the image.

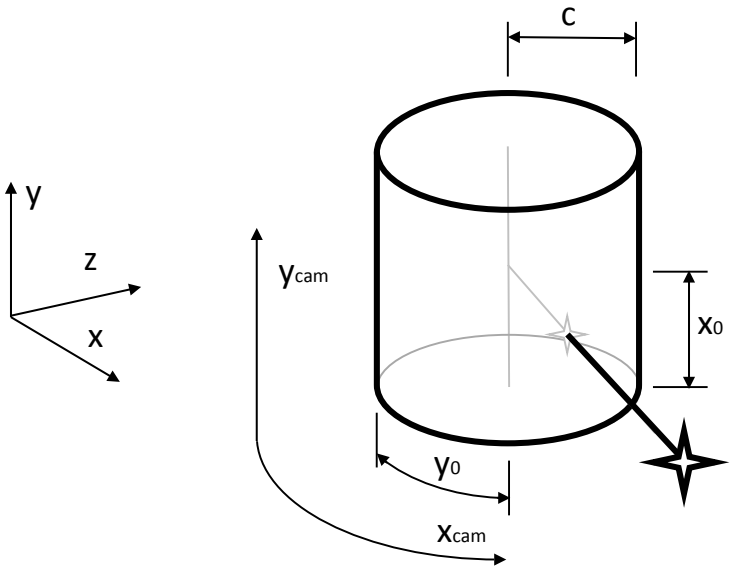


Figure 3.4 – Cylinder model used to represent a panoramic camera.

For some of our datasets, the images were obtained from a panoramic camera. In regular cameras, an image is created when light strikes a 2D CCD. However, in this panoramic camera, the CCD is a single vertical line array mounted on top of a motor and slowly rotated to build up a panoramic image of the environment. To account for this, a camera model that projects the points onto a cylinder must be used. A rough depiction of this is shown in Figure 3.4. This model projects the points using [71]:

$$x_{cam} = x_0 - c \arctan(\frac{-y}{x}) \tag{3.4}$$

$$y_{cam} = y_0 - \frac{cz}{\sqrt{x^2 + y^2}} \tag{3.5}$$

A depiction of how the camera model operates on a point cloud can be seen in Figure 3.5.

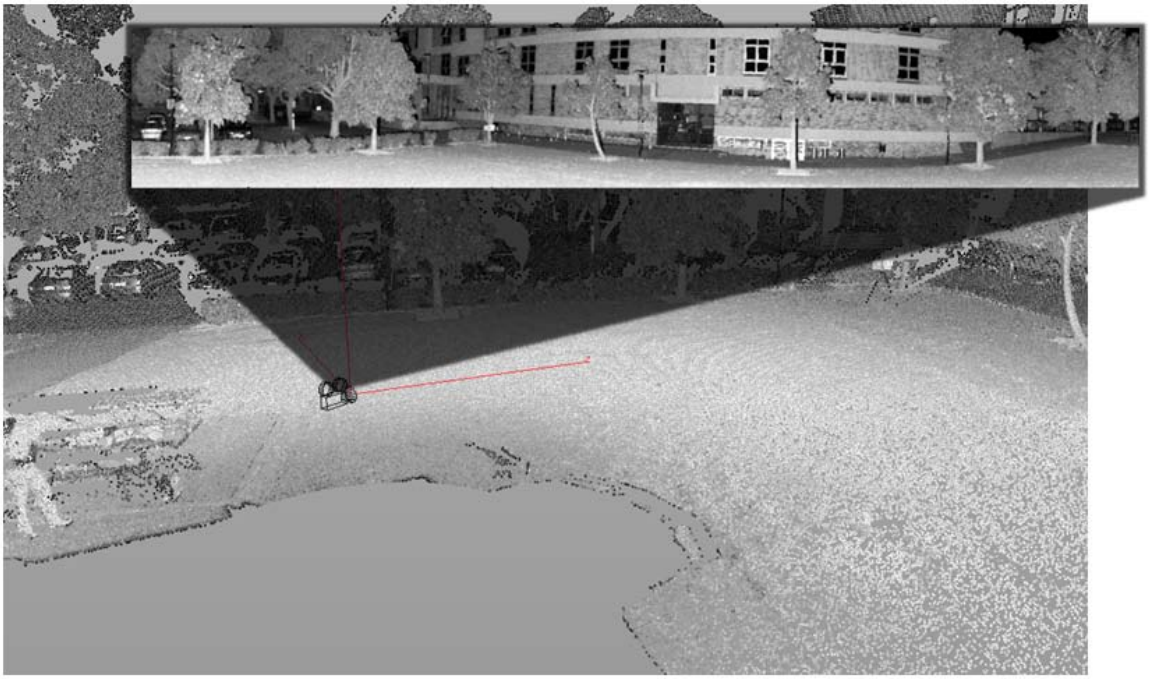


Figure 3.5 – An image of a lidar scan. A virtual camera has been placed in the scan and is generating an image from the scan that will later be aligned with a real image of the same scene.

Affine Transform

To perfectly align all points in two images taken by two cameras at different locations, the distance to each point in the images must be known. This is due to parallax changing where objects appear in each image. This alignment can be estimated using stereo-vision methods. However, the correspondence between pixels must be recalculated for each frame. Due to the difficulty of performing such alignment on two cameras of different modalities, we instead calibrate the camera images using a simple affine transform. While not perfect, for two cameras with only small differences in location and orientation, an affine transform can give high-quality image registration [104].

3.3.2 Gradient Calculation

Once the features have been calculated and used to assign intensity values to the data, the gradient at each point must be calculated. This is required as our method utilises the angle between corresponding gradients to determine the strength of alignment in the sensor outputs.

The magnitude and orientation of the gradient of a camera's image intensity is calculated using the Sobel operator [78]. Calculation of the gradient from 3D data sources is slightly more challenging. The gradient of the points in the 3D data should be from the perspective of the camera so that, once transformed to the camera's frame of reference, the orientations for both sensors will be aligned.

To do this we provide two different approaches. In both methods however, the first step is the same. Initially, the points are projected onto a sphere that is centred at the estimated location of the camera using Equations 3.6 and 3.7.

$$x_{sphere} = \arccos(\frac{z}{\sqrt{x^2 + y^2 + z^2}}) \tag{3.6}$$

$$y_{sphere} = \arctan(\frac{y}{x}) \tag{3.7}$$

A sphere is used rather than a plane as in image generation, as with a plane, points in front of and behind the camera can adversely be projected onto the same location. It was also found to be more resilient to error in the estimated location of the camera.

From this point the first method uses linear interpolation to generate an image from these points. We set the resolution of the image so that the number of pixels in the image is approximately equal to the number of points being interpolated. Once the interpolation is done, a Sobel operator is used to find the gradients. After this has been done, we again use linear interpolation to find the gradient intensities at the corresponding lidar points. This method gives good results on high-resolution scans such as those produced by the Riegl lidar VZ-1000 and LMS-Z420i. However, it can

perform poorly with lower resolution scans such as those produced by the Velodyne HDL-64E.

Because of this, unless otherwise stated, we make use of a second method. In this method each point on the sphere has its 8 nearest neighbours evaluated before the gradient is calculated by adding the gradient vectors to each neighbouring point. The process for this is shown in Algorithm 3.1.

Algorithm 3.1: Gradient calculation for 3D sensors

Let
p_x, p_y, p_v be the current point's x-position, y-position and intensity-value, respectively
n_x, n_y, n_v be the neighbouring points' x-position, y-position and intensity-value, respectively
$g, temp, g_{mag}, g_{or}$ be the gradient vector, a temporary vector, the gradient vector's magnitude and the gradient vector's orientation, respectively

$g = 0$
for *neighbouring point* n **do**
  $x = p_x - n_x$
  $y = p_y - n_y$
  $v = p_v - n_v$
  $temp_{mag} = \frac{v}{8\sqrt{x^2+y^2}}$
  $temp_{or} = \arctan2(y, x)$
  $g = g + temp$

As the gradient is dependent on the location of the camera, it requires re-estimation every time the camera's extrinsic parameters are changed. However, as this process is computationally expensive, for the purpose of gradient calculation in our process it is assumed that the initial parameters are approximately equal to the final parameters. This assumption allows the gradients to be pre-calculated and gives a large reduction in the computational cost. We concluded that this assumption would be valid for most practical cases as the search space used for optimising the lidar's extrinsic calibration is usually at most 1m and 10 degrees, whereas the distance to most objects in the environment in all our experiments was well over 10m. This meant that there would have been only minor changes in the calculated gradients' magnitude and orientation. To test the validity of this, a simple experiment was run on one of our datasets. A lidar scan of the Australian Centre for Field Robotics (ACFR) building (ACFR scan 1 presented in Section 6.3.3) was first aligned using GOM without the simplifying

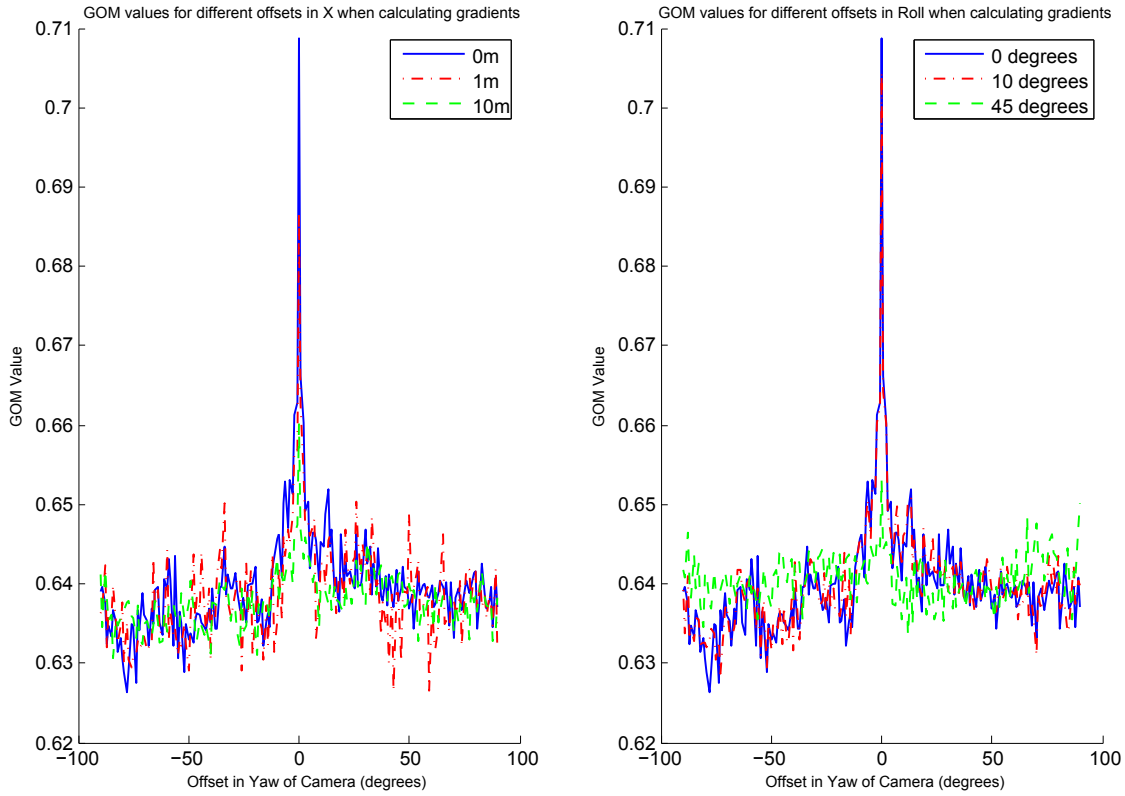


Figure 3.6 – The plots illustrate how offsets in the position and orientation of the camera affect GOM, when assuming constant gradient values. GOM is plotted for a range of yaw values to show the global maximum for each run.

assumption of constant gradient values. The gradients were then recalculated for different levels of offset from the position and perspective of the camera. These scans were used to calculate the GOM values over a range of camera yaw values. The results are shown in Figure 3.6.

Three different levels of offset were introduced into the X location at which gradients were calculated at. These offsets were 0m, 1m and 10m. Despite the different locations where the gradients were calculated, the global maximum for GOM still occurred in the same location for all three runs. While the value of the global maximum for the 1m and 10m error runs was slightly lower than that of the 0m run, it was still clearly distinct from other local maxima. Similar results were also obtained from different datasets and introducing errors into the Y and Z dimension.

The same experiment was performed for the orientation of the lidar. For its roll, three different errors of 0, 10 and 45 degrees were used. A roll offset of 10 degrees has little impact on the results. However, a roll offset of 45 degrees significantly reduced the value of the global maximum. This is expected, as a change in roll has the most direct impact on the orientation of the gradients, and therefore an initial offset as large as 45 degrees breaks the assumption that the initial parameters are approximately equal to the final parameters. Pitch was found to have less impact, and yaw is independent of the gradients. While this assumption would slightly degrade the value of the global maximum, this experiment showed that for our application it would be unlikely to shift it significantly, making the large reduction in computational time offered by the assumption worthwhile.

3.4 The Gradient Orientation Measure (GOM)

The formation of a measure of alignment between two multi-modal sources is a challenging problem. Strong features in one source can be weak or missing in the other. A reasonable assumption when comparing two multi-modal images is that, if there is a significant change in intensity between two points in one image, then there is a high probability there will be a large change in intensity in the other modality. This correlation exists as these large changes in intensity usually occur due to a difference in the material or objects being detected. This correspondence even exists between seemingly unrelated features such as range and reflectance. For example, a sharp change in distance usually indicates a change in the object being detected. There is a high probability that these objects will be made of materials with different reflectance properties, meaning it is likely that there will be a significant change in reflectance at the same location.

GOM exploits these differences to give a measure of the alignment. GOM was inspired by a measure proposed by Pluim et al. [64] for use in medical imaging registration. The presented measure, however, has several differences as Pluim et al.'s method is un-normalised, uses a different calculation of the gradient's strength and direction,

Figure 3.7 – A pair of example images showing how gradients may reverse direction between modalities. In the left image, taken at 950 nm, the trees are white with a black sky behind them. However in the right image, taken at 418 nm, the trees appear black and the sky white. This means that if the gradient between the trees and sky is calculated, the gradients will be 180 degrees out of phase for the two images.

and is combined with mutual information. GOM operates by calculating how well the orientation of the gradients are aligned between two inputs. For each pixel, it gives a measure of how aligned the points are by taking the absolute value of the dot product of the gradient vectors:

$$alignment_j = |g_{(1,j)} \cdot g_{(2,j)}| \tag{3.8}$$

where $g_{(i,j)}$ is the gradient in image i at point j. The absolute value is taken, as a change going from low to high intensity in one modality may be detected as going from high to low intensity in the other modality. This means that for two aligned images, the two corresponding gradients may be out of phase by 180 degrees. An example of this occurring is shown in Figure 3.7.

Summing over the values of these points results in a measure that is dependent on the alignment of the gradients. An issue, however, is that this measure will favour maximising the strength of the gradients present in the overlapping regions of the sensor fields. While this issue could be corrected by normalising the vectors before

taking the dot product, sharper gradients represent features that are more likely to be preserved between images. The stronger gradients also mean that the direction of the gradient calculated will be less susceptible to noise, and thus, more accurate. This means that these points should be given an increased weight, which normalising at this stage would remove. To correct for this bias, the measure is normalised after the sum of the alignments has been made, by dividing by the sum of all of the gradient magnitudes. This gives the final measure as shown in Equation 3.9:

$$GOM = \frac{\sum_{j=1}^{n} |g_{(1,j)} \cdot g_{(2,j)}|}{\sum_{j=1}^{n} \|g_{(1,j)}\| \|g_{(2,j)}\|} \tag{3.9}$$

The measure has a range from 0 to 1, where, if 0, every gradient in one image is perpendicular to that in the other, and 1 if every gradient is perfectly aligned. Something of note is that if the two images were completely uncorrelated, we would expect the measure to give a value of 0.5. This means that if two images have a GOM value of less than 0.5, the score is worse than random and it is a fairly safe assumption that the system is in need of calibration. Some typical GOM values for a range of images is shown in Figure 3.8. The NMI values are also shown for comparison.

3.5 Normalisation and Bias

The normalisation performed prevents the metric from exhibiting a bias towards maximizing the overlap between sensors. Unfortunately, in doing so it creates a second bias which results in the metric favouring alignments with exceptionally small numbers of overlapping points. The reason for this bias is that, as the number of points reduces, the chance of two unrelated areas matching increases.

In our examination of the literature, all markerless appearance metrics we have encountered are susceptible to one of these two biases. MI's bias to minimising overlap is well known [80, 65] and, in the extreme case, NMI will report perfect alignments for a single overlapping point regardless of its value. In Levinson's method [35], adding

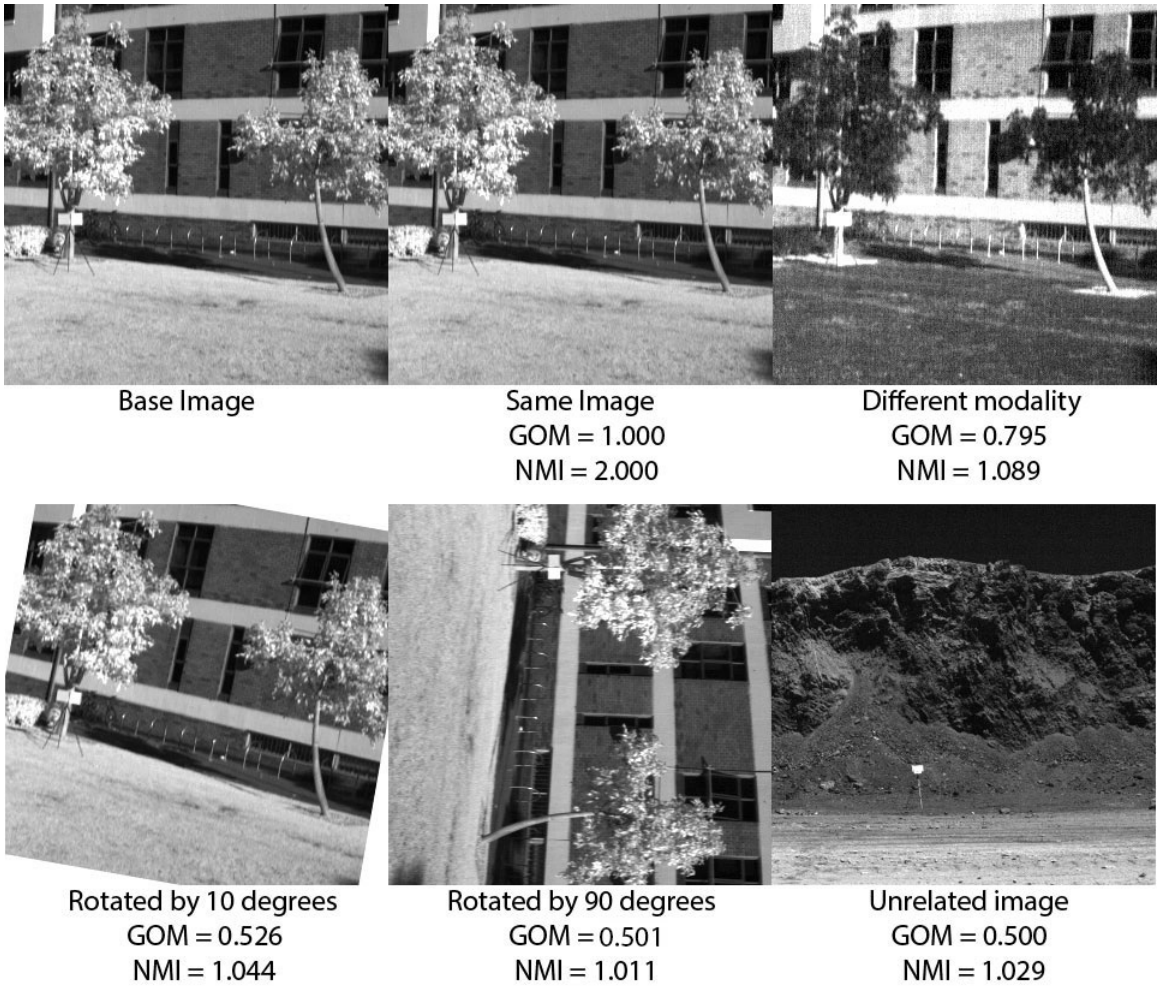


Figure 3.8 – GOM and NMI values when the base image shown on the left is compared with a range of other images.

more points increases the perceived alignment strength, giving a strong bias to maximising overlap. Several methods have been looked at for reducing the effect of these biases, such as a simple threshold, weighting by overlapping points or examining the entropy of the overlapping region. However, no solution could be found that both removed the bias and maintained the possibility of aligning sensors that did only contain small areas of overlap.

In problems with a tightly constrained search-space (as formed by an accurate initial guess to the parameters of the system), the bias towards minimising overlap generally does not play a significant role. This is due to the search-space only containing calibrations that have sufficient overlapping points to mitigate its effect.

3.6 Projecting 3D Points to 2D Images

Several issues arise when attempting to create an image from the point cloud produced by a 3D sensor. The sparse nature of the scans (especially those obtained from mobile platforms) causes the majority of the pixels to have no associated intensity value. Aliasing issues also occur from forcing the points onto the discrete grid that makes up an image. These issues significantly degrade the quality of the alignments, especially for methods based on edge directions, such as GOM. A typical image produced from Velodyne data is shown in Figure 3.9. To overcome these issues, a range of different post-processing options and interpolation or blurring techniques were attempted. However, it was found that most of these techniques would destroy sharp edges and do little to improve results.

To prevent these issues, the generation of a traditional image from the 3D data is only done for visualisation purposes. Instead, the points are kept in a list, and when they are projected using the camera model their position is not discretised. To get the matching points from the base sensor's image, linear interpolation is performed at the coordinates given by the point list.

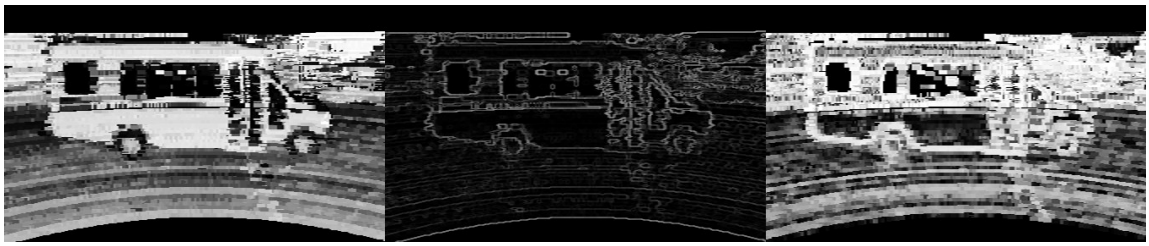


Figure 3.9 – An image generated from Velodyne lidar data is shown on the left. The centre image shows the image's gradients calculated using a standard Sobel filter; the right image shows the gradients calculated from the point cloud by our own technique. Due to the sparse nature of the lidar points when a Sobel filter is used, its gradients tend to depend more on the distribution of points than their intensity. In this instance this results in almost all the gradients being detected as horizontal or vertical lines.

3.7 Optimisation

The registration of one-off scans, and the calibration of a multi-sensor system tend to have significantly different constraints on their optimisation. Because of this, our approach for optimising each problem differs, as outlined below.

3.7.1 Multi-Sensor Calibration

When multiple scans can be aggregated, as in the case of multi-sensor platform calibration, the optimisation is performed using the Nelder-Mead simplex method. This method is a well established local optimisation technique developed by Nelder and Mead [53]. It creates a simplex on the function and uses the points of this simplex to estimate the position of the maximum. It then uses this information to replace one of the points in the simplex. This process repeats until the function converges to a solution. For this optimiser to reliably converge, the search space must be convex. If this condition is not met, the optimisation can become trapped at a local maximum. Generally, this requirement would not be satisfied by any of the metrics analysed (GOM, MI ,NMI or the metric presented by Levinson and Thrun [35]). However, for the case of multi-sensor calibration, in some cases, it can be a reasonable assumption for three key reasons:

1) In calibrating a multi-sensor system, an accurate initial guess as to the alignment of the sensors is often known. This is because the sensors are mounted in a rigid setup with most uncertainty caused by the location of the sensors within their housings.

2) As the sensors are rigidly mounted, a large number of sensor readings can be aggregated. Aggregating readings is beneficial as it both provides additional information and helps to minimise the impact of alignments between unrelated areas. The effect of this is to reduce the strength and number of local maxima produced by the metric.

3) The basin of attraction for GOM's global maximum can be substantially increased by applying a Gaussian blur to the output of one of the sensors it is aligning. This works as the GOM exploits the strong gradients present at overlapping edges as an indicator of an accurate alignment, and, unless the edges are partially overlapping, there is nothing to show how close to the correct alignment the measure is. When blurring is applied, the size of the edges is increased, providing more overlap between them, and an improved indication of the alignment. An issue with this blurring is that it removes many of the small edges that can be used to indicate a precise alignment, reducing the accuracy of the metric. To prevent this issue, an optimisation pyramid was implemented where the result of a level of the pyramid is used as the initial guess for the optimisation of the same image with less blurring. In our experiments, four layers were used, with Gaussians with σ of 4, 2, 1 and 0 applied. The effect of the aggregation and blurring can be seen in Figure 3.10.

3.7.2 Registration

Optimisation of the registration problem with single scans is generally significantly more challenging than the calibration problem. The main reason for this is that the error in the initial guess for the sensors' alignments can be very large. In one of the situations we explored, the initial conditions were only given using consumer GPS systems or a simple note claiming that the sensors were positioned close to each other. This can lead to initial position errors in excess of 1m and 10 degrees. There is also no rigid connection between the sensors, preventing scan aggregation from being

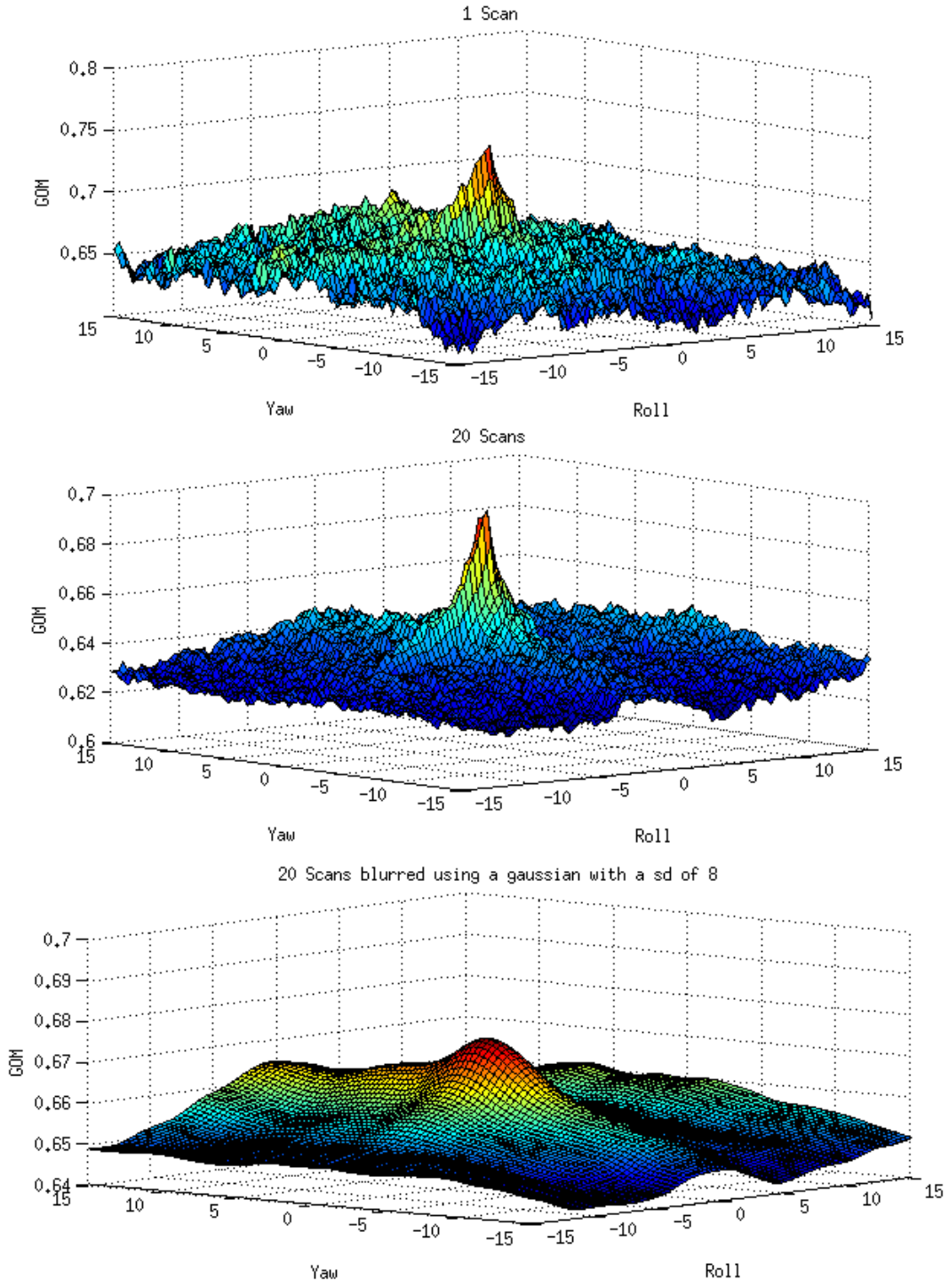


Figure 3.10 – GOM values plotted for the roll and yaw of a typical lidar-photo alignment using one scan-image pair (top) and 20 scan-image pairs (middle). Also shown is the result of applying a Gaussian blur to the 20 images (bottom). Aggregation and Gaussian blurring both significantly smooth the function, allowing for easier optimisation of the metric.

used. These limitations result in a highly non-convex search space that requires a global optimisation technique to find the maximum.

We evaluated several different global optimisation methods such as pattern search [3], global search [94], genetic algorithms [20] and particle swarm optimisation [47]. Particle swarm was found to perform the fastest of the options while still being robust. It was also fairly intuitive, allowing the progress of the optimisation to be evaluated during operation.

Particle swarm optimisation works by randomly placing an initial population of particles in the search space. On each iteration, a particle moves to a new location chosen by summing three factors: i) it moves towards the best location found by any particle, ii) it moves towards the best location it has ever found itself and iii) it moves in a random direction. The optimiser stops once all particles have converged. The process of registration is shown in Algorithm 3.2. The implementation of particle swarm used was developed by Chen [13]. In our experiments we used a particle swarm optimiser with 500 particles.

3.8 Summary

In this chapter we have outlined a framework for calibrating 3D lidar and camera sensors based on the appearance of an arbitrary surrounding environment. This process operates by creating a virtual camera and using it to project the lidar data onto the real camera's image. Once projected, a measure of the similarity between the data is generated. To this end, we have developed a metric GOM to act as an indication of the alignment of the two sensors. The optimisation of this metric is dependent on whether the setup allows the aggregation of multiple scans, and the accuracy of the initial guess, as the search space is inherently non-convex.

One of the key features required for deploying robotic systems into the service of non-expert users is self-diagnostics. To perform this the system will require a measure of the quality of any alignment performed, as well as ways to estimate and propagate

Algorithm 3.2: Particle swarm algorithm

Let
$\boldsymbol{r^i(t)}$ be the position of particle $\boldsymbol{i}$ at time $\boldsymbol{t}$
$\boldsymbol{v^i(t)}$ be the velocity of particle $\boldsymbol{i}$ at time $\boldsymbol{t}$
$\boldsymbol{p_n^{i,L}}$ be the local best of the $\boldsymbol{i}$th particle for the $\boldsymbol{n}$th dimension
$\boldsymbol{p_n^g}$ be the global best for the $\boldsymbol{n}$th dimension
$\boldsymbol{n \in 1, 2, ...N}$
$\boldsymbol{t}$ be the time
$\boldsymbol{\Delta t}$ be the time step
$\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are the cognitive and social factor constants
$\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$ are two statistically independent random variables uniformly distributed between 0 and 1
$\boldsymbol{w}$ be the inertial factor

for *each iteration* l **do**
  if $f(r^i(l+1)) > f(p^{i,L}(l))$ **then**
    $p^{i,L}(l+1) = r^i$
  end
  if $f(r^i(l+1)) > f(p^g(l))$ **then**
    $p^g(l+1) = r^i$
  end
  $v_n^i(t+\Delta t) = wv_n^i(t) + c_1\phi_1[p_n^{i,L} - x_n^i(t)]\Delta t + c_2\phi_2[p_n^g - x_n^i(t)]\Delta t$
  $r_n^i(t+\Delta t) = r_n^i(t) + \Delta t v_n^i(t)$
end

any uncertainty in its calibration parameters. In the coming chapter we examine error estimation techniques that can be applied to these metrics and any challenges and limitations faced in doing so.

Chapter 4

Variance Estimation

Estimation of the variance in the parameters is an important component of sensor registration and calibration, as without some indication of accuracy the results are of limited value. However this is an area that is often overlooked; in our examination of the appearance-based markerless calibration literature, only one paper provided any estimate of the uncertainty [62] and this formed a lower bound on the error.

The only method utilised for estimating the accuracy of a calibration is often by a user observing the projection of one sensor's output onto another. In many situations this method is impractical and error prone; it also cannot be applied to autonomous systems. Part of the reason for the lack of methods to estimate calibration uncertainty is the difficulty of its calculation in multi-modal markerless registration problems.

This chapter presents the following contributions:

- An examination of the challenges regarding the estimation of the variance for appearance-based markerless metrics and the constraints required for accurate and timely estimation.
- An exploration of methods that can be utilised in estimating the variance of an algorithm's output.
- The consideration of the assumptions these variance approximation techniques make and the examination of issues, such as any bias or outliers present.

4.1 Variance in Appearance-Based Metrics

Most approaches to estimating the uncertainty of parameters obtained by maximising a metric make the implicit assumption that the search-space is convex, or at the very least has no local optima. This assumption stems from the methods only examining the structure either at, or in a small region around the global maxima. Both the delta method [55] and Cramér–Rao lower bound [67], which are examined later in this chapter, implicitly assume this. This limitation links the need to calculate the covariance in a robust and efficient manner to the need to develop a convex search space, an area that poses serious difficulty to metrics that rely on the environment's appearance.

This difficultly arises, as regardless of the approach taken, local maxima will occur due to repetitions in the environment. To illustrate this, consider an example where two sensors are placed in two different, but identical rooms. Given only observations of the environment, it is impossible for the sensors to tell that they are not viewing the same location and the metrics value would reflect this confusion. While this is an extreme example, repetitions are abundant in man-made environments.

This means that when a single sensor reading is used to align two sensors, the local variance, and hence confidence in the method may be very different to the actual. An example of this problem is shown in Figure 4.1, where GOM optimisation has been run aligning a single lidar scan with a camera's image. The distribution contains a large number of very 'spiky' local maxima. As a result of this, if we were to evaluate our confidence in one of these local maxima being the correct solution, by looking at the local curvature or nearby points, this would cause us to give a vastly overconfident assessment of the situation.

While no closed-form covariance estimate exists in these cases, a measure of the uncertainty can still be obtained using sampling based methods. To do this, in situations where the variance was required (See Section A.1) we applied a global optimisation technique in combination with bootstrapping of the data. While this method was found to be effective it has two significant drawbacks.

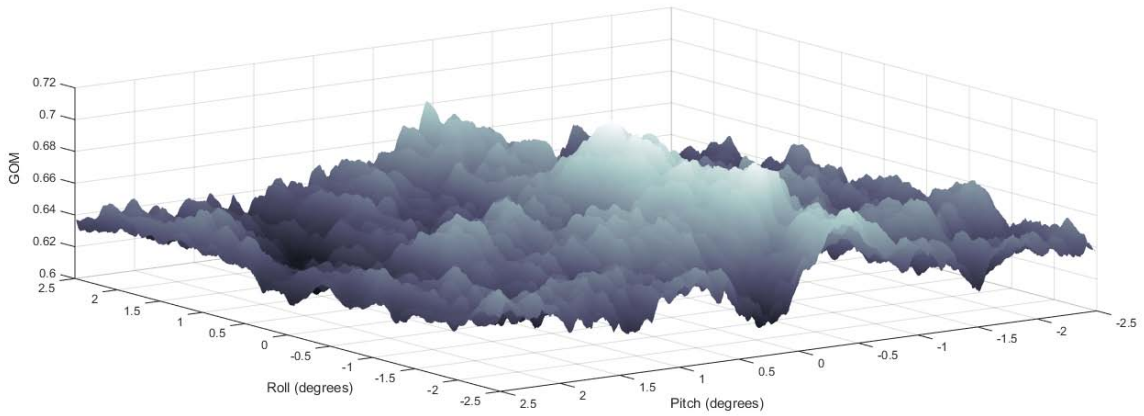


Figure 4.1 – The GOM values when roll and pitch are adjusted for a single Velodyne scan aligned with an image. The resulting distribution is highly multi-modal with a large number of local optima.

The first issue is that neither bootstrapping nor global optimisation methods provide any form of formal guarantee, or even bound, on the accuracy of their results. The second issue is that both methods require large numbers of function evaluations. The method described in Section A.1 took several hours to complete, despite the use of efficient Graphics Processing Unit (GPU) routines for the evaluation of the metrics, a heavily constrained search space and only 20 samples used in the bootstrapping process.

To avoid these issues, methods that utilise the environment typically recommend that they are calibrated using a large number of readings from a variety of locations. The hope is that any chance alignments, similar areas, repeated structure and other idiosyncrasies will be unique to that one pair of sensor readings. If this holds then for sufficiently large numbers of scans, all of these chance alignments and their associated local optima will be smoothed away. Ideally this will result in a perfectly convex cost function. As previously discussed this convexity facilitates the analysis of the variance and observability of the parameters using simple analytical methods.

To further assist in developing a convex cost function, the search space of possible solutions may also be clamped to a small range around the correct solution. This is done as, even for large numbers of scans, most appearance metrics will never become globally convex. The main reason for this is that most appearance-based multi-

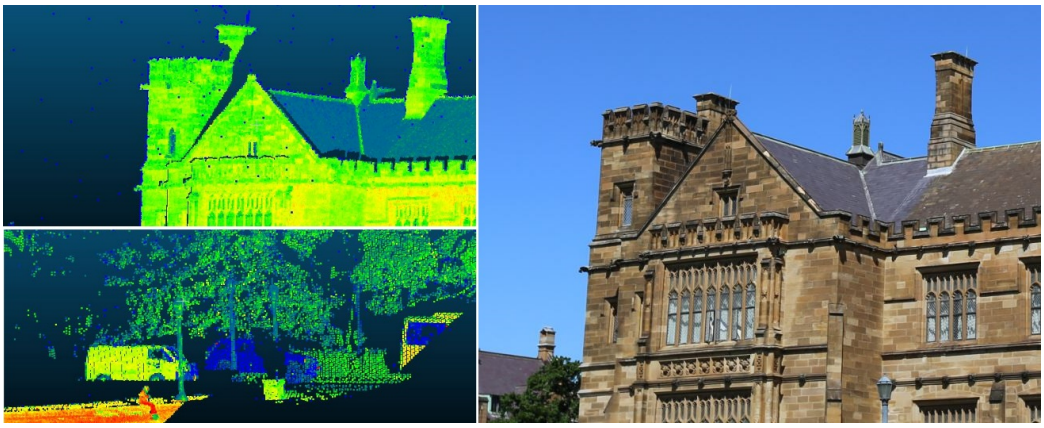


Figure 4.2 – A camera image and two sections of the corresponding lidar scan. The top left scan has a close initial alignment, whereas the bottom left scan is far from the correct solution.

modal metrics only utilise overlapping regions of the sensors. This limitation will, in almost all cases, result in a non-convex search space. For example, consider aligning a camera with a high-resolution lidar by projecting the lidar's generated point cloud onto the camera's image, as shown in Figure 4.2. In the case of the top left scan, while the parameters used to create this scan differ slightly from those of the actual camera, they are close to correct. This results in a similar image for which, if the correct multi-modal metric is applied, could conceivably have a smooth and easily optimised convex search space between its current position and the true parameters. On the other hand, if the initial guess to the alignment is very poor (bottom left scan in Figure 4.2) none of the lidar points projected onto the image will correspond to anything in the camera's field of view. This means that there is nothing to indicate how the alignment parameters must change to bring the two sensors into a closer alignment. With the exception of highly contrived environments, adding more scans will not do anything to alleviate this issue.

The need to clamp the search space to as small a search region as possible is further motivated by the so-called 'curse of dimensionality' that increases the difficulties that the non-convex nature of the search space presents. To illustrate this, consider an example where we optimise parameters of an n dimensional problem. For each parameter of this problem, let us assume we have a metric that provides a smooth

and convex surface for 50% of the search space. For a one-dimensional problem this results in a randomly initialised gradient ascent optimiser having a 50% chance of correctly finding the global maxima. This would mean that running an optimiser such as this for several different initial conditions would be an effective and efficient method for locating the maximum.

However, in all of our problems we have a minimum of 6 unknown variables corresponding to the extrinsic calibration parameters to find. Thus the chance of a randomly initialised local optimiser converging to the correct solution would be 0.5^6 or roughly 1.6%. This small convex region now means that the previously suggested optimisation method of starting a series of local optimisers has become far more computationally expensive. These problems only get worse as more parameters are optimised over (focal length, camera centre, lens distortion parameters, timing offset, etc.).

The clamping of the search space also acts to alleviate two other issues. The first is the bias to overlap that was discussed in Section 3.5. The second is that in data gathered from ground vehicles in urban environments, there typically also exists a strong local optima when a sensor is rotated 180 degrees from the correct position. This is due to the symmetry often present in the shape of roads and the position of buildings to either side of them.

In a system being calibrated by an expert user, many of the issues presented here would not pose a significant issue as they could quickly be recognised and accounted for. However, in the case of a fully automated system, one set up by a hobbyist or being maintained by non-experts these issues must be addressed.

To facilitate a more robust solution to the automatic calibration problem we examine other cues, such as sensor motion, that can be used for calibration; this is presented in Chapter 5. Before this area is explored, we require the ability to estimate the variance of the output of an arbitrary function given the variance of its inputs. Thus, we first present the methods we have made use of to achieve this.

4.2 Estimating the Variance of Algorithms

In calculating the uncertainty present in a system, there are many situations where the variance of the output of an algorithm is desired given the variance of its inputs. It is also desirable that the output is a Gaussian distribution as this simplifies the equations and the propagation of probability estimation of future steps [58].

Estimating the variance of the output provided by an algorithm can be approached in different forms, and it is in general problem specific. In our work we make use of a range of approaches listed below, in roughly the order of preference with the most desirable first.

4.2.1 Exact Covariance Calculation

The exact covariance can be calculated when an analytical method for formulating the output uncertainty from the input uncertainty is given, and the resulting distribution is Gaussian. For example, the rotation of a vector x by a rotation matrix R gives the equation $z = Rx$ and has its covariance given by:

$$\text{cov}(z) = R\text{cov}(x)R^T \tag{4.1}$$

4.2.2 Approximate Analytical Variance

An analytical method that gives an approximate covariance is used in situations where the output variance is either non-Gaussian or intractable to calculate, but a close approximate formulation exists. An example of this situation is the rotation of a vector by a rotation matrix R, where both the vector and the rotation matrix have an associated covariance. In this situation we make use of the angle-axis vector form of the rotation r and note that for 'small' deviations in r designated Δr, the change in the corresponding rotation matrix R, designated ΔR is:

$$\Delta R = \begin{bmatrix} 0 & -\Delta r_3 & \Delta r_2 \\ \Delta r_3 & 0 & -\Delta r_1 \\ -\Delta r_2 & \Delta r_1 & 0 \end{bmatrix} \tag{4.2}$$

Barfoot and Furgale [6] make use of this relationship to form several approximate analytical covariance methods of varying complexity and accuracy for this situation.

4.2.3 The Delta Method

The approximate approach given above can allow accurate estimation of the variance for a large range of problems. However, the approximate formula used is unique to each problem, and it may be difficult to find an approximation that remains accurate for all possible input values and variances.

In these situations we make use of the delta method [55]. This method is a simple and powerful approach to approximating the variance of a function. It operates by forming a first-order approximation to the function at the point of interest and finding the variance of this approximation. Let $z = f(x)$, where f is an arbitrary function relating two variables x and z; the delta method gives an approximation to the covariance of this equation by:

$$\text{cov}(z) \approx \frac{df}{\partial x}\text{cov}(x)\frac{df}{\partial x}^T \tag{4.3}$$

As this method is only used when no analytical or simple approximation to the variance exists, the analytical calculation of $\frac{df}{\partial x}$ is typically also intractable. Because of this we make use of simple finite difference methods (Appendix E) in its calculation. In our work this approximation is utilised when calculating the variance of our translation estimates during optimisation.

4.2.4 Approximate Covariance of a Minimisation

Consider the case where we are attempting to find the values of z, $\hat{z}$, that minimise the function $y = f(z, \hat{x})$ given data $\hat{x}$. Let this situation be expressed as:

$$\hat{z} = g(\hat{x}) = \text{argmin}_{\text{x}} f(z, \hat{x}) \tag{4.4}$$

then applying the delta method from before yields

$$\text{cov}(\hat{z}) \approx \frac{dg}{\partial x} \text{cov}(x) \frac{dg}{\partial x}^T \tag{4.5}$$

However, the issue with this formulation is the term $\frac{dg}{\partial x}$. As $g(\hat{x})$ is a minimisation, any analytical calculation of its derivative is usually beyond our abilities to compute. We also face the issue that if the finite difference method is used to solve this term, it will require the minimisation process to be repeated at least once per element of $\hat{x}$. This minimisation process is typically computationally expensive and would likely form a significant bottleneck in any approach that made use of it.

Fortunately this term can be approximated by a relationship presented in [11]. As we have performed a minimisation, the gradient of $g(x)$ at $\hat{x}$ must be zero; from this the implicit function theorem means that at $x = \hat{x}$ and $z = g(\hat{x})$

$$\frac{dg}{\partial x} = -(\frac{d^2 f}{\partial z^2})^{-1} \frac{d^2 f}{\partial x \partial z} \tag{4.6}$$

This equation removes the need to find the partial derivative of the minimisation function and simplifies the calculations[1]. In many cases the underlying function f is also a relatively simple function such as a least squares calculation, allowing for simple calculation of the needed derivatives.

[1] See [11] for the full derivation.

4.2.5 Cramér–Rao Lower Bound

In situations where the algorithm utilises probabilities, an absolute lower bound can be calculated on the variance through the use of the Cramér–Rao lower bound [67]. This process calculates the minimum possible covariance an unbiased estimator can possess. It is the inverse of the Fisher information matrix, which is calculated by finding the negative expected log likelihood of the second derivative of the output of the algorithm of interest. In our work it has been found that this method is typically too optimistic in its estimates to be used alone. Instead, we typically make use of it as a simple sanity check as any variance given by other methods should be greater than this lower bound.

4.2.6 Monte Carlo Simulation

The delta method makes the assumption that the local shape of a cost function can be extended to the whole space. As we discussed in Section 4.1 this will not be the case for functions with multiple local optima, as are frequently observed in multi-modal appearance metrics. In these situations Monte Carlo simulations can be used to estimate the output [42]. The process is very simple; a point is drawn from the input distributions and evaluated. This point is recorded and the process repeated a large number of times and the final covariance is calculated from these points. This random sampling method is the most general approach, however it is usually the most computational expensive.

4.2.7 Bootstrapping

In some situations the covariance of one or more input variables is not known. If, in these cases, the system is also over-defined by these variables, an indication of the output covariance can still be calculated. To do this a technique known as ‘bootstrapping’ is used [17]. In bootstrapping a new subset of the variables is created by sampling the original variables with replacement, the algorithm is run and the result

recorded. This is repeated a large number of times and the covariance calculated using the outputs. This method can be combined with Monte Carlo simulation when some variables have known covariance.

4.3 Issues with Approximations

In all of our methods we represent the results using the distribution's first and second moments, which are optimally described by the Gaussian distribution [31]. While this gives an accurate approximation for many situations, in some cases the utilised Gaussian is a poor fit to the actual distribution. One of the most common cases where we encountered this in our work is dividing by a value that is approximately 0. For example, this happens in practice when we calculate the translation between two sensors for a timestep where little rotation has been observed. Figure 4.3 shows the case of calculating $\frac{1}{x}$ where x has a μ of 0.01 and σ of 0.05. This gives a distinctly multi-modal output. However in cases where $\sigma \ll \mu$ these multi-modal peaks disappear and the Gaussian approximation becomes valid.

This is a trend present in most cases we have encountered. As sigma reduces, the validity of the Gaussian approximation increases, thus in practice, as long as a significant number of low covariance readings occur, accurate results can still be obtained. Our outlier rejection process also acts to remove the cases where these issues result in the variance being underestimated by a significant margin. These issue as well as the trade-offs in performance and speed given by the Monte Carlo and delta method are further explored in Appendix F.

A second problem is that bootstrapping and random sampling both rely on a sufficient number of samples to give accurate results. This is an issue, as in general the more samples obtained, the more accurate the estimation will be, but at the expense of computation time. In all cases where we make use of Monte-Carlo or bootstrapping methods, unless explicitly stated otherwise, 1000 samples are used. It was empirically found that for most situations this number gave what we considered, to be a reasonable trade-off between accuracy and run time [16].

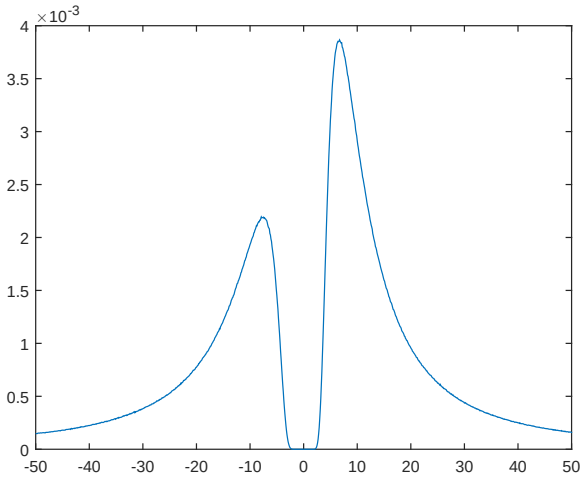


Figure 4.3 – The PDF of $\frac{1}{x}$ when x has a μ of 0.01 and σ of 0.05. This PDF gives a multi-modal distribution which would be poorly approximated by a Gaussian distribution preventing many of the covariance estimation strategies from operating correctly.

4.4 Bias

In our work we generally make the assumption that our methods produce estimates that are unbiased, however this is not always the case. The main source of bias in our work stems from the assumption that the intrinsic calibration of our sensors is perfect. The result of errors in this calibration will have different effects depending on the sensors. As one example of the possible issues these biases can cause, consider the Velodyne sensor used at the ACFR. Originally this Velodyne sensor had some slight error in its intrinsic calibration that resulted in all scans appearing slightly 'squished' vertically. This intrinsic calibration would lead to all movement in this direction being underestimated. As most ground-based robots typically undergo little vertical movement, errors in this direction tend to be amplified and even a small error can have a significant effect on the accuracy of the resulting calibration.

While it is possible to estimate the bias in the sensors, even a simple linear model would double the number of parameters that must be estimated during calibration. This would increase computation time and more importantly would increase the sensitivity of the method to noise. As significant errors in the sensor intrinsics would

generally mean that any extrinsic calibration would be of limited use, we have chosen to assume no bias in our setup. While for the systems examined this assumption appears valid, the estimation and compensation for any bias is an important topic, though one whose examination lies beyond the scope of this thesis.

4.5 Outliers

In some circumstances the estimated covariance can form a poor representation of the true error present in an estimate. Cases where the variance has been overestimated have little impact on the system other than to undervalue some information, and while not desirable, will not adversely affect the results by any significant amount. However, points where the variance has been greatly underestimated will have a large impact on a method's accuracy and reliability. These points are deemed to be outliers and their impact must be mitigated to allow the system to converge to a correct solution.

There are a large range of possible approaches to the handling of outliers present in the literature [43]. In our approach we deal with outliers by using trimmed means. Trimmed means (also known as truncated means) is a process via which the data is sorted and a set ratio of the worst performing data is labelled as outliers and discarded before the mean of the remaining data is taken. In our work we set 25% of the data as outliers, however the process is fairly insensitive to the exact value used as long as it is greater than the true ratio of outliers. We chose trimmed means as it gave an intuitive process and worked with our local optimisation framework; for the full justification as well as comparisons to other options refer to Appendix G.

4.6 Summary

In this chapter we first examined the issues associated with the non-convex search space given by the appearance metrics and their implications for the robustness of any optimisation. We then examined the situations and constraints that allow an

estimate of the calibration's uncertainty to be obtained. After this we explored the more general problem of estimating the uncertainty of the output of an algorithm given its input uncertainty. Finally we explored the issues with the approximations made to obtain these uncertainty estimates and looked at the issue of outliers in the data. The techniques outlined in this chapter lay the ground work for motion-based calibration approaches that will be presented in Chapter 5.

Chapter 5

Motion-Based Metrics

5.1 Introduction

While appearance-based metrics have the potential to quickly and automatically calibrate the sensors of a system, they suffer from two main drawbacks: i) they require the sensors to have an overlapping field of view, and ii) they generally require a highly constrained search space in order to locate the metric's global maximum in a robust and timely manner. These limitations are inherent to how these metrics operate; a sensor cannot align to what it does not observe and an unconstrained environment can contain repetitive structure and chance similarities that will yield local maxima.

Therefore, to improve the robustness and reliability of automated registration and calibration, and to increase the range of problems to which it can be applied, we must look to other sources of alignment information. In a large number of applications the sensors are positioned in a rigid mounting and provide continuously updated information about their surroundings. The most common example of this is the rapidly growing field of mobile robotics.

In this chapter we examine the problems involved in motion-based calibration and present a pipeline for exploiting the motion of a vehicle travelling through an arbitrary environment to provide the extrinsic calibration of its sensors. The intuition behind

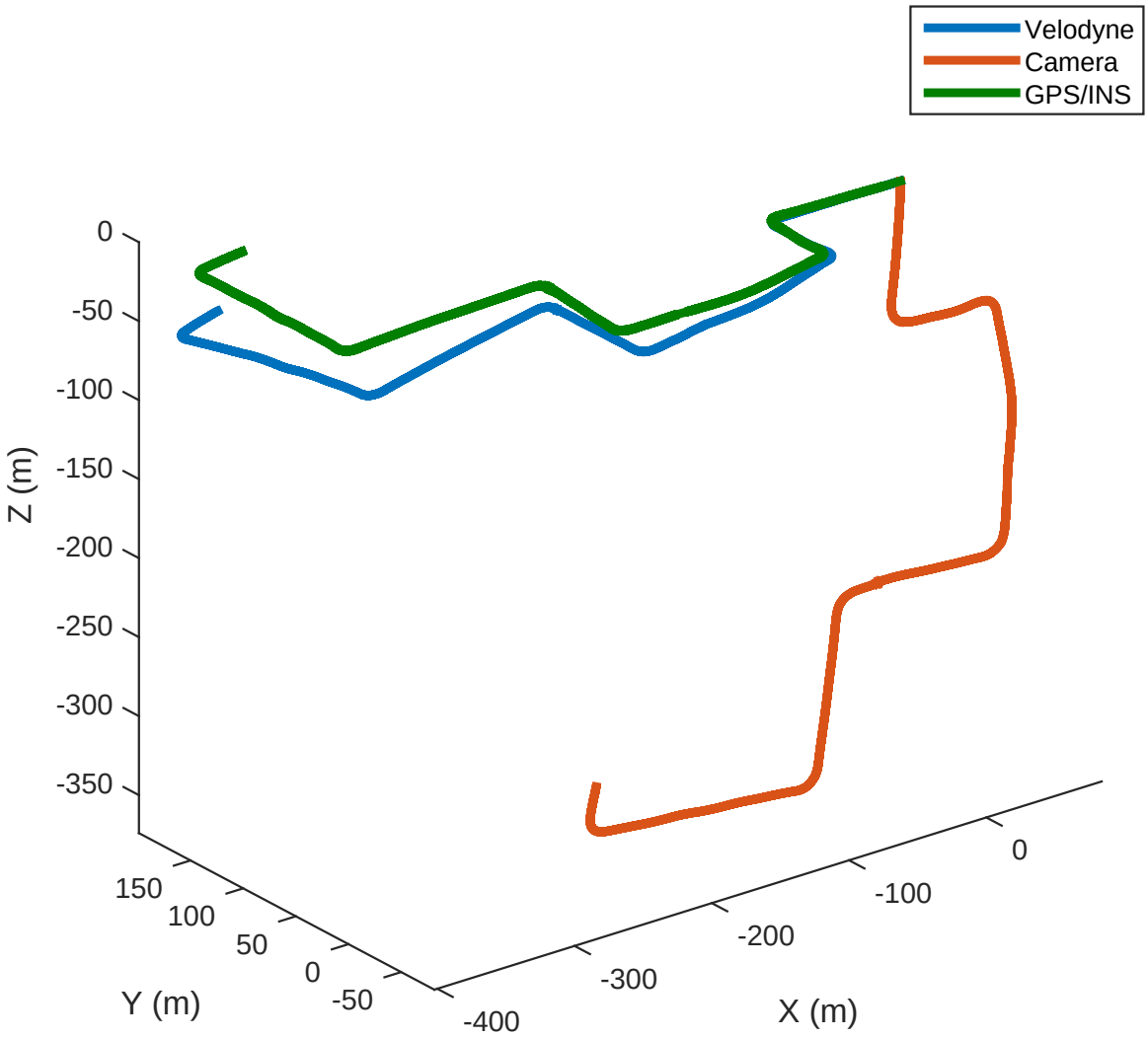


Figure 5.1 – A plot of the detected motion of a camera, Velodyne and GPS/INS system that are part of the KITTI vehicle during a drive. Simple observation of the paths allows a user to see how the axes must be rotated to align the co-ordinate systems of the sensors. More detailed analysis can also give the translation between the sensors.

this approach is very simple, as illustrated in Figure 5.1.

In this Figure, a sensor vehicle equipped with a camera, Velodyne and GPS/INS system has been driven through a short series of turns. The motion that each sensor has undergone has then been independently estimated using sensor-specific techniques. From observing the plots of this motion, a user can quickly identify the rough rotations each sensor must undergo to bring its co-ordinate systems into alignment. More

detailed analysis also allows the translation to be estimated. This is the intuition behind a category of calibration techniques known as 'hand-eye' calibration.

In the following sections we will begin by deriving the constraints and equations that relate the motion sensors on a vehicle undergo, to the offset between the sensors. With these relationships in place, we will formulate motion-based error metrics for the rotation, translation and timing offset between the sensors. These metrics are constructed in a probabilistic manner that makes use of the uncertainty present in each sensor reading. This allows for the comparison of the offset between any number and combination of cameras, 3D lidars and navigation sensors.

A process for automatically initialising the sensor offset parameters, and optimising them using the developed error metrics is then presented. This process not only provides the most likely estimate for the sensor layout, but also gives the confidence in the parameters obtained. Furthermore, this system, unlike the previously examined appearance-based techniques, does not require the user to provide any form of initial guess as to the configuration of the sensors.

As this motion-based approach does not require a constrained search space it is complementary to previous appearance-based approaches, by providing the accurate initialisation needed by appearance-driven methods (such as those presented in [35, 52, 61, 85]). We will finally present new techniques that make further use of the sensor motion in combination with the environment's appearance.

Specifically this chapter makes the following contributions:

- The formulation of a pipeline for calibrating an array of 3D lidars, cameras and GPS/INS sensors on a mobile vehicle moving through an arbitrary scene.
- The extension of 'hand-eye' calibration techniques to a probabilistic form that incorporates the uncertainty of each sensor's readings.
- The use of motion-based techniques to constrain and guide the optimisation of appearance-based approaches.
- The development of a new alignment metric, the Intensity-Motion (IM) metric, which is designed for aligning lidar-camera systems on mobile vehicles through

a combination of motion and appearance cues.

- The estimation of the uncertainty present at all stages of the calibration.

Algorithm 5.1 presents an overview of our process for motion-based calibration. In this chapter we will present the theory, design decisions and practical considerations that led to the development of this approach.

Algorithm 5.1: High level overview of motion-based calibration approach.

1. Given $\boldsymbol{n}$ sensor readings from $\boldsymbol{m}$ sensors.
2. For each sensor $\boldsymbol{i}$ at time $\boldsymbol{k}$ convert sensor readings into sensor motion $\boldsymbol{T_{i,k}^{i,k-1}}$. Each transformation has the associated covariance matrix $\mathbf{cov}(\boldsymbol{T_{i,k}^{i,k-1}})$.
3. Arbitrarily set a sensor to be the base sensor $\boldsymbol{B}$.
4. Use the sensor transformations to find their rotational velocity $\boldsymbol{\omega_{i,k}^{i,k-1}}$ and its associated variance.
5. Use the rotational velocity to find the timing offsets $\boldsymbol{\tau_i^B}$ that minimises the trimmed sum of the variance of $\boldsymbol{\omega_{i,(k+\tau_i^B)}^{i,(k-1+\tau_i^B)}}$ weighted by the inverse of its variance.
6. Approximate the variance of the resulting timing offset $\boldsymbol{\tau_i^B}$ and transfer it to the uncertainty of the motion $\mathbf{cov}(\boldsymbol{T_{i,k}^{i,k-1}})$.
7. Use the timing offset to interpolate all sensor readings to when the slowest updating sensor obtains readings and reject uninformative data.
8. Convert $\boldsymbol{R_{i,k}^{i,k-1}}$ to angle axis form $\boldsymbol{A_{i,k}^{i,k-1}}$.
9. For every sensor $\boldsymbol{i}$, find a coarse estimate for $\boldsymbol{R_i^B}$ by using the Kabsch algorithm to solve $\boldsymbol{A_{i,k}^{i,k-1} = R_i^B A_{B,k}^{B,k-1}}$ weighting the elements by their variance.
10. Refine $\boldsymbol{R_i^B}$ by finding the values that maximise the trimmed sum log-likelihood of $\boldsymbol{A_{j,k}^{j,k-1} = R_j^i A_{i,k}^{i,k-1}}$ for all combinations of two sensors $\boldsymbol{i}$ and $\boldsymbol{j}$.
11. Approximate the variance of the resulting rotations $\boldsymbol{R_i^B}$.
12. For every sensor $\boldsymbol{i}$, find a coarse estimate for $\boldsymbol{t_i^B}$ by finding the least squares solution to the system of equations given by considering $\boldsymbol{t_i^B = (R_{i,k}^{i,k-1} - I)^{-1}(R_i^B t_{B,k}^{B,k-1} - t_{i,k}^{i,k-1})}$ at each time-step, weighting the equations by their parameters variance.
13. Estimate the scale of the transformations provided by the monocular cameras and their associated uncertainty.
14. Refine $\boldsymbol{t_i^B}$ by finding the values that maximise the trimmed sum log-likelihood of $\boldsymbol{t_j^i = (R_{j,k}^{j,k-1} - I)^{-1}(R_j^i t_{i,k}^{i,k-1} - t_{j,k}^{j,k-1})}$ for all combinations of two sensors $\boldsymbol{i}$ and $\boldsymbol{j}$.
15. Approximate the variance of the resulting translations $\boldsymbol{t_i^B}$.
16. Find all sensors that have overlapping field of view.
17. Use sensor-specific metrics to estimate the transformation between sensors with overlapping field of view.
18. Combine results to give final $\boldsymbol{R_i^B}$, $\mathbf{cov}(\boldsymbol{R_i^B})$, $\boldsymbol{t_i^B}$ and $\mathbf{cov}(\boldsymbol{t_i^B})$.

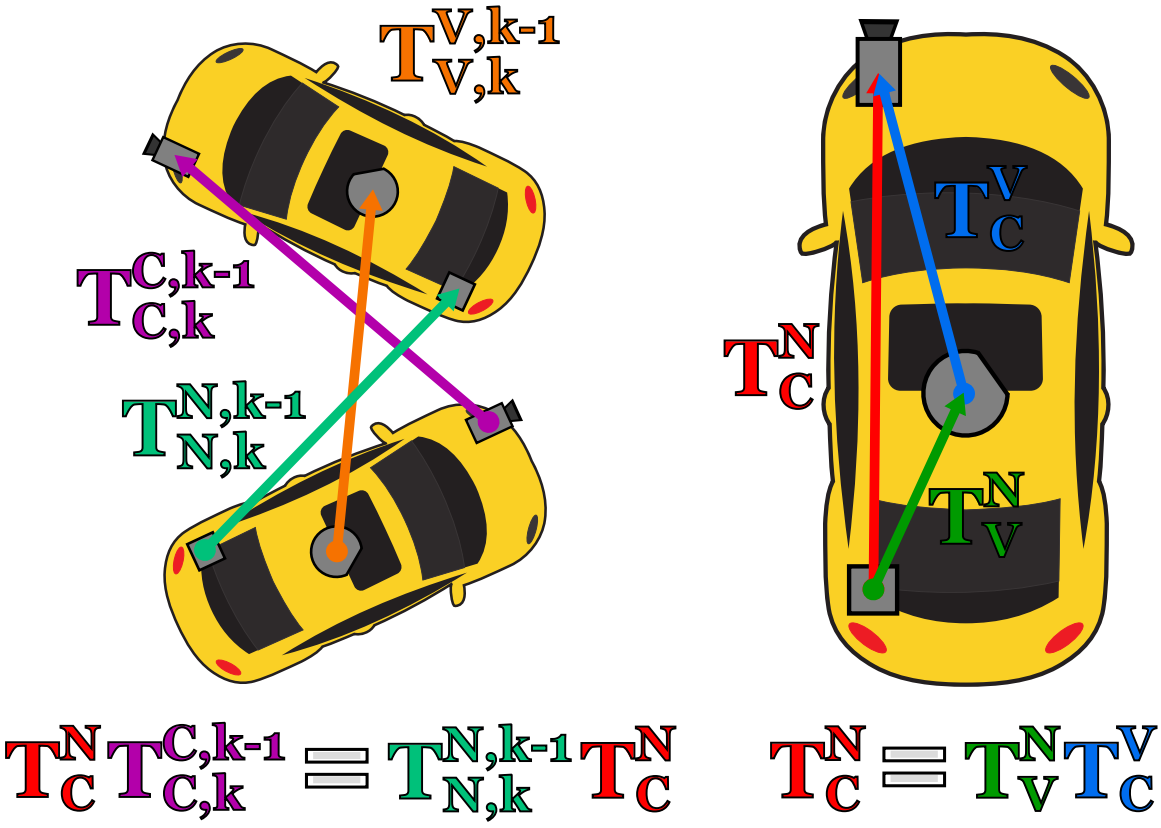


Figure 5.2 – A diagram of a car with a camera (C), Velodyne lidar (V) and navigation sensor (N). The image on the right shows the position of the three sensors on the vehicle. The image on the left shows the transformation these sensors undergo at timestep k.

5.2 Estimating Sensor Extrinsics from Motion

Before we can utilise sensor motion to calibrate a system, we must first explore how this motion is related to the sensors' parameters. Figure 5.2 shows a depiction of how each sensor's motion and relative position are related. As the vehicle moves, the transformation between any two rigidly mounted sensors x and y at timestep k can be given by:

$$T_y^x T_{y,k}^{y,k-1} = T_{x,k}^{x,k-1} T_y^x \tag{5.1}$$

This is the basic equation from which all of the motion-based alignment information is derived.

Our objective in performing the motion-based calibration of a system is to find the values of the inter-sensor rotation R_y^x, translation t_y^x and timing offset τ_y^x for which Equation 5.1 holds. In order to achieve this, we begin by first examining Equation 5.1 in terms of its rotational and translational components. This yields two equations:

$$R_y^x R_{y,k}^{y,k-1} = R_{x,k}^{x,k-1} R_y^x \tag{5.2}$$

and

$$R_y^x t_{y,k}^{y,k-1} + t_y^x = R_{x,k}^{x,k-1} t_y^x + t_{x,k}^{x,k-1} \tag{5.3}$$

Importantly, if the terms of the rotational component (Equation 5.2) are examined, we see that this equation is not dependant on the translational offset. This allows the rotational component R_y^x to be found before the translational component t_y^x is known, decoupling these two terms.

Until now, we have assumed that no time offset is present in the sensor readings. This assumption is not required however. If we take two sensors that were started independently and assign variables a and b to represent their individual timesteps then Equation 5.2 becomes:

$$R_y^x R_{y,b}^{y,b-1} = R_{x,a}^{x,a-1} R_y^x \tag{5.4}$$

which for non-constant rotational velocities will only hold when:

$$\tau_y^x = b - a \tag{5.5}$$

Furthermore, the magnitude of the angle through which a rotation matrix rotates a point is independent of the frame of reference. Thus, if the angular magnitude of the rotation is given by θ, the relationship can be expressed as:

$$\tau_y^x = b - a \implies \theta_{y,b}^{y,b-1} = \theta_{x,a}^{x,a-1} \tag{5.6}$$

This equation is now completely independent of the extrinsic transformations between the sensors, only depending on the timing offset τ_y^x. This allows the timing offset between the sensors to be found before any extrinsic transformation information is known.

These three Equations (5.2, 5.3 and 5.6) are the standard equations that many authors ([1, 2, 26, 93] among others) have made use of in aligning sensor data from motion. If the assumption of a rigid mounting holds and the sensors undergo sufficient non-degenerate motion, these equations will give the unknown offset between the sensors.

Due to several factors, by far the most significant being noise in the sensor motion estimates, the observed values will not perfectly conform to these equations. Typically, to account for these differences in readings, an approach that finds the least-squares error for the system of equations is used. While this can be an effective approximation method, it implies that every sensor reading was made with equal accuracy. Furthermore, if the approach was extended to consider more than two sensors, it would imply that each sensor's readings were of equal worth.

In a constrained environment that has been designed to calibrate a specific sensor set, this assumption of equal accuracy in each sensor reading may be approximately true. However, if we wish to calibrate a system composed of a wide range of sensors, as it moves through an unconstrained environment, these assumptions will detrimentally affect our system. Issues such as a Real-Time Kinematic (RTK) GPS losing connection with its base-station, or a visual-odometry system encountering an area with insufficient near-field objects to accurately estimate its translation, must be accounted for. In order for a system to handle a wide range of sensors, as well as the range of possible situations an unconstrained environment will provide, it must be able to reason about the value of each reading it receives.

If the readings were given only as the simple point estimates that they are often taken as, such a system would not be possible. Fortunately, the sensors utilised

generally provide a far richer range of information than this single parameter. From this information it is usually feasible to not only estimate a sensor's motion, but also the variance of this estimate.

To account for this, we reformulate the calibration problem to one that not only considers the value of the sensor readings, but also the confidence that each sensor has in its values. This allows the formation of an approach that uses the evidence given by each sensor reading and reasons as to the most probable sensor configuration. Importantly, as this system takes into consideration the accuracy of each sensor reading and the effect it has on the resulting system, it is also able to give an accurate assessment of its confidence in the final configuration. We see this as one of the key advantages of our approach as it allows a user or robotic system to know the confidence it can place in the obtained configuration.

To facilitate this approach, we extend the above equations relating sensor offsets to motion, to include the uncertainty in the readings. In the following sections, this extension will enable us to develop error functions that can give the relative likelihood of any set of offsets between the sensors being correct.

5.2.1 Timing Offset

In our work we utilise a three-element angle-axis form to represent our angles, with a second three-element vector giving the variance of each element. This representation will be covered in further details in Section 5.5.1. In this form, if $[r_x, r_y, r_z]$ are the rotational elements, the magnitude of the rotation angle is given by:

$$\theta = |||[r_x, r_y, r_z]||_2 \tag{5.7}$$

and if $[\sigma_{rx}^2, \sigma_{ry}^2, \sigma_{rz}^2]$ are the associated variance, then the variance in the rotation magnitude is given by:

$$\text{cov}(\theta) = |||[\sigma_{rx}^2, \sigma_{ry}^2, \sigma_{rz}^2]||_1 \tag{5.8}$$

The rotation of two sensors x and y at timestep k is linked via Equation 5.6. Using this and modelling the sensor observations by their first and second moments, the relative likelihood of these sensor observations for this system is given via:

$$\mathcal{L}_\theta(x, y, k) = \frac{1}{\sqrt{2\pi(\text{cov}(\theta_{x,k}^{x,k-1}) + \text{cov}(\theta_{y,k}^{y,k-1}))}} \exp\left(-\frac{(\theta_{x,k}^{x,k-1} - \theta_{y,k}^{y,k-1})^2}{2(\text{cov}(\theta_{x,k}^{x,k-1}) + \text{cov}(\theta_{y,k}^{y,k-1}))}\right) \tag{5.9}$$

This formulation is equivalent to the evaluation of the PDF of a Gaussian distribution. The distribution is formed using the difference between the two sensor readings for the angle and is evaluated at 0.

5.2.2 Rotational Offset

As previously stated the rotations of the sensors are related by:

$$R_y^x R_{y,k}^{y,k-1} = R_{x,k}^{x,k-1} R_y^x \tag{5.10}$$

Due to our use of an angle-axis rotation representation, this equation can be simplified. If A is our rotation vector, then the sensor rotations are related by:

$$A_{y,k}^{y,k-1} = R_y^x A_{x,k}^{x,k-1} \tag{5.11}$$

From this starting point we perform a process similar to that done for the timing offset. We first utilise the above equation to find the error present for the readings of two sensors x and y at timestep k. This is expressed as:

$$R_{err}(x, y, k) = A_{y,k}^{y,k-1} - R_y^x A_{x,k}^{x,k-1} \tag{5.12}$$

with associated variance:

$$\text{cov}(R_{err}(x,y,k)) = \text{cov}(R_{y,k}^{y,k-1}) + R_y^x \ \text{cov}(R_{x,k}^{x,k-1}) \ (R_y^x)^T \tag{5.13}$$

These equations are then combined to give the likelihood of the readings for the given system:

$$\mathcal{L}_R(x,y,k) = \frac{1}{\sqrt{2\pi \ \text{cov}(R_{err}(x,y,k))}} \exp\left(-\frac{R_{err}(x,y,k)^2}{2 \ \text{cov}(R_{err}(x,y,k))}\right) \tag{5.14}$$

This is equivalent to evaluating the likelihood of a Gaussian distribution with mean R_{err} and covariance $\text{cov}(R_{err}(x,y,k))$ at 0.

5.2.3 Translational Offset

The translational offset is related to the sensor motion and rotational offset via:

$$R_y^x t_{y,k}^{y,k-1} + t_y^x = R_{x,k}^{x,k-1} t_y^x + t_{x,k}^{x,k-1} \tag{5.15}$$

Again, taking the error present for two sensors at timestep k we can calculate the error in the system as:

$$t_{err}(x,y,k) = (R_{y,k}^{y,k-1} - I)t_y^x + t_{y,k}^{y,k-1} - R_y^x t_{x,k}^{x,k-1} \tag{5.16}$$

However, unlike in previous calculations, the variance of the above system cannot be given using a simple analytical combination of the components. This is due to the interaction of multiple variables containing uncertainty, and the need to convert several rotation estimates from angle-axis to rotation-matrix form. We overcome this issue by making use of the aforementioned delta approximation (Section 4.2.3) to estimate $\text{cov}(t_{err}(x,y,k))$.

Once this approximation has been performed, this information is again used to give the likelihood of the readings via the equation:

$$\mathcal{L}_t(x, y, k) = \frac{1}{\sqrt{2\pi \text{ cov}(t_{err}(x, y, k))}} \exp\left(-\frac{t_{err}(x, y, k)^2}{2 \text{ cov}(t_{err}(x, y, k))}\right) \tag{5.17}$$

Scale Ambiguity in Translational Offset

For monocular cameras, unless further assumptions about the system are made, the translational motion estimates will be normalised with no sense of scale. In the case of a system solely comprised of cameras, this prevents the calculation of the translational offset. In any other setup however, we will be able to utilise the camera's motion to give the offset. This is done through the introduction of a scale term s_k for the translation that is estimated at the same time as the translational offset t_y^x. If in this instance sensor y is the camera, Equation 5.16 would become:

$$t_{err}(x, y, k) = (R_{y,k}^{y,k-1} - I)t_y^x + s_k t_{y,k}^{y,k-1} - R_y^x t_{x,k}^{x,k-1} \tag{5.18}$$

With this new term for estimating the error in the modelled translation in place, all other steps in the translation offset likelihood estimation would proceed in the same manner.

5.2.4 Combining Multiple Readings

The above processes gives expressions for the likelihood of a pair of sensor readings fitting our rigid sensor model, for a given set of offsets. Assuming that each sensor reading is independent of the others, the joint likelihood of scans is then found by multiplying together the likelihood functions. In practice, for numerical convenience this is done through summing log-likelihoods.

Finding the parameters that maximise each of these pairwise log-likelihood sums would result in a series of pairwise offsets between the sensors. Unfortunately, due to sensor noise, it is unlikely that these estimates would form a consistent system. That is, if there are three sensors x, y and z, then for this process $T_y^x T_z^y \neq T_z^x$. To

prevent this issue from occurring we consider the likelihood of all possible pairwise offsets simultaneously, summing their pairwise log-likelihood to give a single measure of the likelihood for the entire system. This likelihood should be maximised when the correct timing, rotation and translation parameters are utilised.

More formally, if the system is comprised of n sensors each making m readings, our objective is to find the parameters that maximise the following expression:

$$\sum_{x=1}^{n} \sum_{y=1}^{n} \sum_{k=1}^{m} \log \mathcal{L}(x, y, k) \tag{5.19}$$

for each of the likelihood expressions given by Equations 5.9, 5.14 and 5.17.

5.3 Optimisation

Given the motion of the individual sensors, the process for calibrating the system can now be divided into three stages:

1. Timing offset estimation
2. Rotational offset estimation
3. Translational offset estimation

At each stage the optimal values for the unknown parameters are found by maximising the log-likelihood equations outlined in Section 5.2. This maximisation must be performed using an optimisation process. In our implementation, this is accomplished using a local optimisation strategy, the Nelder-Mead Simplex Optimiser [53]. This approach was used as it is a derivative-free method that can efficiently find the required maximum in a timely manner. The initial point used in starting this optimisation is specific to the parameters being optimised and will be discussed in Section 5.4.

After the optimisation has located the most likely parameters, the uncertainty in this estimate is found. This is calculated by taking the maximum values obtained from

two strategies. We first make use of the Cramér–Rao lower bound. This estimate is a lower bound however, and in many situations was found to be optimistic about the actual accuracy of our readings. To overcome this issue we combine it with the modified delta method presented by Censi and La [11]. This method, was discussed in Section 4.2.4 and extends the delta method to allow for efficient variance estimates on optimisation problems.

While the delta method should have an estimate that is always greater or equal to the Cramér–Rao lower bound, in practice, rounding and numerical issues can sometimes prevent this relationship from holding. Taking the maximum of the two estimates helps to generate a more robust estimate.

5.3.1 Propagating Uncertainty

In this optimisation process, each stage utilises the previous stage and thus will be influenced by the accuracy of its solutions. The dependence of the translation on the rotational offset estimation's accuracy is captured in the calculation of $\mathcal{L}_t(x, y, k)$. However, the same is not true of the timing offset. While both the rotational and translational offset equations depend on it, their equations implicitly assume exact correspondence between the scans. To overcome this limitation we transfer the uncertainty from the time parameter to the sensor motion estimates.

This is accomplished by assuming that the timing offset estimate is sufficiently accurate that the rate of change of the sensor motion between the actual time and the estimated time is roughly constant. That is, if τ represents the timing offset, $\hat{\tau}$ the true value of the timing offset, T the transformation the sensor undergoes and $\Delta\tau$ the error in our timing offset estimation, then:

$$\left.\frac{dT}{d\tau}\right|_{\tau=\hat{\tau}} \approx \left.\frac{dT}{d\tau}\right|_{\tau=\hat{\tau}+\Delta\tau} \tag{5.20}$$

With this assumption, the angular and linear rotational velocity can be taken to be approximately constant for this time range. This means that the relationship between

timing error and position error can be linearly approximated as:

$$\Delta T \approx \Delta\tau \left.\frac{dT}{d\tau}\right|_{\tau=\hat{\tau}} \tag{5.21}$$

The variance given by this equation is added to the variance already present in the transformation estimates.

5.3.2 Scale Term Optimisation

When monocular cameras are used a scale term must be estimated for every timestep. To accomplish this at each function evaluation, Equation 5.18 is rearranged to solve for the scale parameter. This yields 3×(the number of non-camera sensors) values for each scale parameter. The delta method is used to estimate the variance of each of these estimates before they are combined into a single estimate. This estimate and its variance, in combination with the other parameters, are then used to assess the likelihood of the system's configuration.

5.4 Initialisation

Our decision to consider all pairwise transformation estimates and sensor readings in the calculation of the likelihood estimation results in a measure that can be relatively computationally expensive to calculate. This also poses the problem that while the likelihood functions will tend to be convex in a large region around the optimal parameters, the system is not guaranteed to be globally convex. Because of these issues, we wish to initialise the optimiser as close to the correct solution as possible. This will ensure we are searching in the convex basin of attraction and minimise the number of function evaluations required.

For the timing offset we simply initialise the optimiser assuming no offset between the sensors. This assumption will be close to correct for most systems. In the case of

the rotation and translation estimates the initialisation is more complex. However, unlike the appearance-based methods, an initial guess does not need to be provided to the system. Instead an accurate initialisation can be obtained from simplifying our treatment of the sensor motion.

5.4.1 Rotational Initialisation

In the case of the rotational offset, the parameters used to initialise the optimisation framework of Section 5.3 are found by simplifying our treatment of the variance and considering a single pair of sensors at a time. This allows an efficient analytical approximate solution to the system to be obtained.

In this approximation, one of the sensors is first arbitrarily designated to be the base sensor B. We then proceed to calculate the approximate rotation between this sensor B and each other sensor i, R_i^B. This is accomplished through the use of a slightly modified version of the Kabsch algorithm.

The Kabsch Algorithm

The Kabsch algorithm, presented in [32], is an approach that calculates the rotation matrix between two sets of vectors providing the least squared error. The method operates by first finding the covariance matrix between the sets of vectors. If V_1 and V_2 are the vector sets, the covariance matrix Σ between them is given by:

$$\Sigma = (V_1)^T V_2 \tag{5.22}$$

This covariance matrix can then be used to find the rotation, R via [32]:

$$R = (\Sigma^T \Sigma)^{\frac{1}{2}} \Sigma^{-1} \tag{5.23}$$

If the two sets of vectors provided are the angle-axis form of the rotations provided by the sensors B and i, this process will give an estimate for R_i^B. In our approach

the Kabsch algorithm had been modified to give a non-equal weighting to each sensor reading. The weight assigned to the readings at each timestep is given by:

$$W_k = \frac{1}{\sqrt{\sum_{i=1}^{3} \text{cov}({A_{x,k}^{x,k-1}}_{i,i}) + \text{cov}({A_{y,k}^{y,k-1}}_{i,i})}} \tag{5.24}$$

where $\text{cov}(A)$ is the covariance matrix of the rotation axis. The summing of the diagonal elements converts the variance into a form that is independent of the rotation and reduces it to a single number.

5.4.2 Translational Initialisation

As in the case of the rotation, the translational initialisation operates by simplifying the variance estimation and only considering a single pair of sensors to allow the formation of an efficient analytical approximation.

Section 5.2 found that the relationship between motion and the translational offset between two sensors x and y at timestep k is given by:

$$R_y^x t_{y,k}^{y,k-1} + t_y^x = R_{x,k}^{x,k-1} t_y^x + t_{x,k}^{x,k-1} \tag{5.25}$$

The terms of Equation 5.25 can be rearranged and combined with information from n other timesteps to give the following system of Equations:

$$t_y^x = \begin{bmatrix} R_{y,2}^{y,1} - I \\ R_{y,3}^{y,2} - I \\ \vdots \\ R_{y,n}^{y,n-1} - I \end{bmatrix}^{-1} \begin{bmatrix} R_y^x t_{x,2}^{x,1} - t_{y,2}^{y,1} \\ R_y^x t_{x,3}^{x,2} - t_{y,3}^{y,2} \\ \vdots \\ R_y^x t_{x,n}^{x,n-1} - t_{y,n}^{y,n-1} \end{bmatrix} \tag{5.26}$$

The only unknown here is t_y^x. Solving this system of equations will yield a least squares estimate for the translation and is the approach taken by Tsai and Lenz

[93]. Just as in the case of rotation, we can improve the estimate of t_y^x given by this equation by weighting each of the terms with a value derived from taking the inverse of a simplified sum of its variance.

In cases where one of the sensors is a monocular camera, the motion is only given up to scale ambiguity. Because of this, the scale of the motion at each timestep must also be estimated. One possible approach to solving this problem is simultaneously solving for the scale terms s_k. If we assume that the scale ambiguity is in sensor y, this gives:

$$\begin{bmatrix} t_y^x \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} R_{y,2}^{y,1} - I & t_{y,2}^{y,1} & 0 & \dots & 0 \\ R_{y,3}^{y,2} - I & 0 & t_{y,3}^{y,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ R_{y,n-1}^{y,n} - I & 0 & 0 & \dots & t_{y,n}^{y,n-1} \end{bmatrix}^{-1} \begin{bmatrix} R_y^x t_{x,2}^{x,1} \\ R_y^x t_{x,3}^{x,2} \\ \vdots \\ R_y^x t_{x,n}^{x,n-1} \end{bmatrix} \tag{5.27}$$

This system gives the least squared solution to the offset t_y^x as well as each of the unknown scales s_k, however if n is the number of sensor readings, its calculation requires the inversion of a $3n$ by $n+3$ matrix. This inversion presents a significant computational load in situations where several thousand estimates are made. As this calculation is only used to initialise a second refinement stage, exact accuracy of the solution is not required and so, to ease the difficulty of computation a rough approximation is made. We assume for the sake of this initial calculation that the offset between the sensors is approximately 0. With this approximation in place, the scale can be estimated as:

$$s_k \approx R_y^x t_{x,k}^{x,k-1} (t_{y,k}^{y,k-1})^{-1} \tag{5.28}$$

This will yield three estimates for the scale. Again, making a rough approximation we assume that the scale term corresponding to the largest element of $t_{y,k}^{y,k-1}$ will be the most accurately calculated. This term is used to give absolute scale to the camera's motion, for use with Equation 5.26.

We wish to clearly note here that this scale approximation is only used for this one equation, with all other stages considering the offset when estimating the scale. The offset calculated through this approximate method will have a bias towards estimating zero offset. We do not see this bias as detrimental however, as when the exact formulation given by Equation 5.27 is used, outliers can in some cases cause the method to give unreasonable estimates (for example t_y^x where the elements can exceed $100m$). This zero bias helps to mitigate the effect of these outliers.

5.5 Practical Considerations

The process outlined so far gives a method for accurately determining the calibration of a system of sensors using their motion. However to allow for an efficient and robust implementation of the system, a range of factors must still be considered. The following subsections present several implementation details that were important to the development of our framework.

5.5.1 Transformation Representation

In the majority of our work, transformations are stored as six element vectors $[x, y, z, rx, ry, rz]$. The first three elements $[x, y, z]$ give the translational component of the transform. The next three elements are an angle-axis representation of the rotation where $[rx, ry, rz]$ give the rotation axes and $||[rx, ry, rz]||_2$ gives the rotation angle. This representation was chosen as it is a minimal three-element vector and the rotation is in a form directly usable by hand-eye calibration techniques [93]. The angle-axis system as well as other rotation systems are outlined in Appendix D, where their strengths and weaknesses are also examined.

To represent uncertainty we assume that the six elements under consideration are independent, giving a second six-element vector containing the associated uncertainty.

When the offset between two cameras is incorporated with the offset between other sensors, we make use of a seven-element vector $[nx, ny, nz, s, rx, ry, rz]$. In this vec-

tor, instead of a three-element translation vector, we have a four-element vector $[nx, ny, nz, s]$. This is made up of a normalised translation vector $[nx, ny, nz]$ and the scale $[s]$. We use this notation as several situations arise where the camera-camera transformation offset is known very accurately up to scale ambiguity, but the scale is either noisy or completely unknown. This representation allows us to correctly capture and represent this uncertainty.

5.5.2 Consistent Offset Representation

To ensure that all inter-sensor offsets form a consistent system, the optimisation only generates offset estimates with respect to the base sensor. These offsets are then combined to generate the parameter values with respect to each other sensor.

A second consideration is that for two sensors x and y, under our motion models, $\mathcal{L}(x, y, k) = \mathcal{L}(y, x, k)$. This reduces the number of calculations required at each stage.

5.5.3 Outlier Rejection

There are a large range of approaches that can add robustness against outliers to our framework. In our implementation, we made used of the trimmed-means outlier rejection strategy described in Section 4.5. This trimmed value is used when log-likelihood readings are summed to generate the overall likelihood in the optimisation process.

5.5.4 Interpolation

Where interpolation of transformations is required, we make use of simple linear interpolation. In the case of rotations, linear interpolation can yield highly non-linear angular velocities between the two points [15]. However, due to the high update rate our sensors provide, the maximum rotation a vehicle underwent between two sensor

readings was 0.1 radians, with the average reading being closer to 0.0001 radians. This meant that the small angle approximation of $sin(\theta) \approx \theta$ is appropriate and linear approximation can be used without these issues impacting accuracy.

5.5.5 Temporal Alignment

Sensors on a vehicle generally operate in an asynchronous manner. On a typical robotic system, each sensor reading undergoes processing within the sensor before being transferred to a central system via a route that may involve other sources of delay. This creates an unknown timing offset between when the data was taken and when the robot retrieves it. In scenarios where the sensors are not used as an input to the control systems, these unknown timing offsets can be even more substantial. In these systems, sensors may be individually started and record their data locally, for combination after all the data has been transferred. In these situations, while the sensor update rate is known, the unknown offsets between the sensors must be corrected to allow high-quality calibration and fusion from the sensor information.

This situation is slightly different from the problem we have been looking at up until this point. The presented process for finding the timing offset between the sensors assumed that the sensors obtained readings simultaneously and the unknown offset is given as a discrete number of timesteps.

To compensate for the asynchronous nature of the sensors, before the likelihood of the data is assessed, we incorporate several additional processing steps. First, we apply the current estimate for the timing offsets to the data before interpolating it at n equally spaced intervals to create synchronous data. In our implementation, $n = 10,000$ steps was chosen as this was found to give a suitable balance between accuracy and runtime.

A second processing step is the differentiation of the data to give velocities rather than positions. This step is undertaken because the absolute angular magnitude will often suffer from drift as small errors in the estimated angle accumulate. Finally, sections

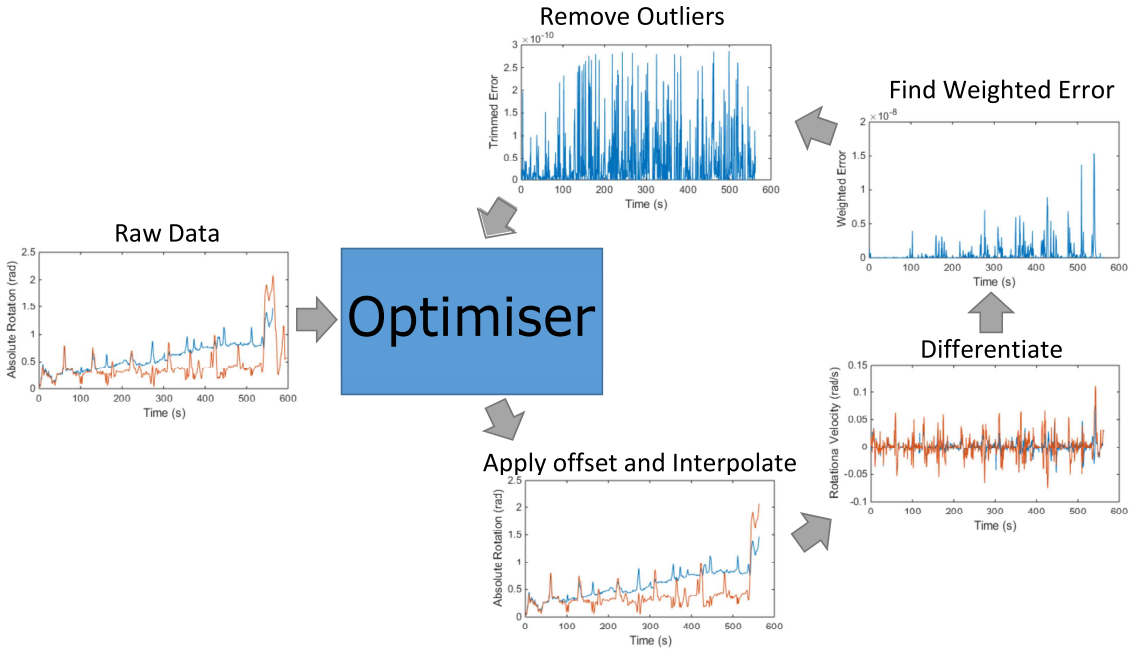


Figure 5.3 – An overview of the steps taken to find the timing offset between the sensors.

of the sensor's data that have no corresponding reading from the other sensors, are discarded.

In practice, due to the simple relation between sensor angular magnitudes, the comparison of pair-wise offsets can also be simplified. In this case it can be found by solving the equivalent problem of minimising the weighted variance of the sensors readings at each timestep. An overview of the process for two sensors is shown in Figure 5.3.

Once the timing offsets have been found, synchronised sensor-motion readings are obtained. To accomplish this the motion of each sensor is interpolated at the times when the slowest-updating sensor obtained readings. In the experimental systems used this was typically at a rate of 5 to 10 Hz.

5.5.6 Data Pre-Processing

After the temporal correspondence between the sensors has been established, we found that performing a simple pre-processing step can eliminate many of the outliers. The weighted mean rotational magnitude is found at each timestep. Points where one of the sensor readings lies over 10 standard deviations from this mean are labelled as outliers and removed. This pre-processing is fairly insensitive to the number of standard deviations used as the threshold, as most outliers tend to be hundreds of standard deviations from the mean.

At this stage we also remove uninformative data points. Motion with a rotational velocity below a threshold value, or extremely high rotational variance, are all removed. While not essential to the operation of our method, these points will have weightings of approximately 0 when calculating offsets. Eliminating these points at this stage reduces the computational demands of the process.

5.5.7 Camera-Camera Translation Estimation

While the approach given allows for the estimation of the translational offset between two cameras through two scaling parameters (assuming that at least one sensor with absolute scale is also present in the system), in practice these estimates were not utilised. This was done because the two unknown parameters per timestep resulted in the translation of two cameras relative to each other conveying minimal additional information about the system. As in most systems, cameras are the most numerous sensor; considering these correspondences also greatly increased the computation time of the system.

5.6 Sensor Transformation Estimation

To facilitate the motion-based calibration process described above, individual sensor motions, as well as an estimate of their uncertainty, is required. As different sen-

sor modalities provide different types of information, the approach used for motion estimation is sensor dependent. The following subsections explain how to estimate sensor self-motion for the main sensor modalities found in mobile robots.

5.6.1 3D Lidar

To calculate the transform from one scan to the next, the Iterative Closest Point (ICP) [9] algorithm is used. A point-to-plane variant of ICP is employed and the previous transformation is used for the initial guess as to the transformation. We also ignore points with an error of over a threshold distance (in our implementation set to 0.2m), treating them as outliers.

Most 3D lidar systems, such as the Velodyne HDL-64E, map their surroundings through a continuously rotating lidar head. This causes issues when the lidar is mounted on a moving vehicle as even at sedate urban speed, the position of the lidar at the beginning and end of its scan will have varied significantly. For example, if the lidar is set to provide one scan every 0.1 seconds and is mounted to a vehicle travelling at 20 kmph, the sensor will have moved by over half a metre during the recording. To allow for accurate scan matching this offset must be compensated for.

To do this, the time at which each point was recorded must first be found. Some platforms such as the Shrimp system used in our experiments, individually timestamp each point recorded. However, the majority of systems used only record when the scan starts and ends. In systems where the original point order is preserved and lidar points with no return are recorded, we model the system as scanning points at a constant rate over the duration of the scan. In systems where the point ordering has not been preserved (for example the KITTI dataset) we assume a lidar model where the scanner head rotates in a clockwise direction at a constant speed, scanning points in a vertical line. This simplification has the drawback that it does not capture the actual pattern of lasers used in the lidar's head and can cause significant errors in the area around where the lidar starts and finishes its scan. Am example of this issue is in the Velodyne HDL-64E, where the actual laser configuration results in the sawtooth

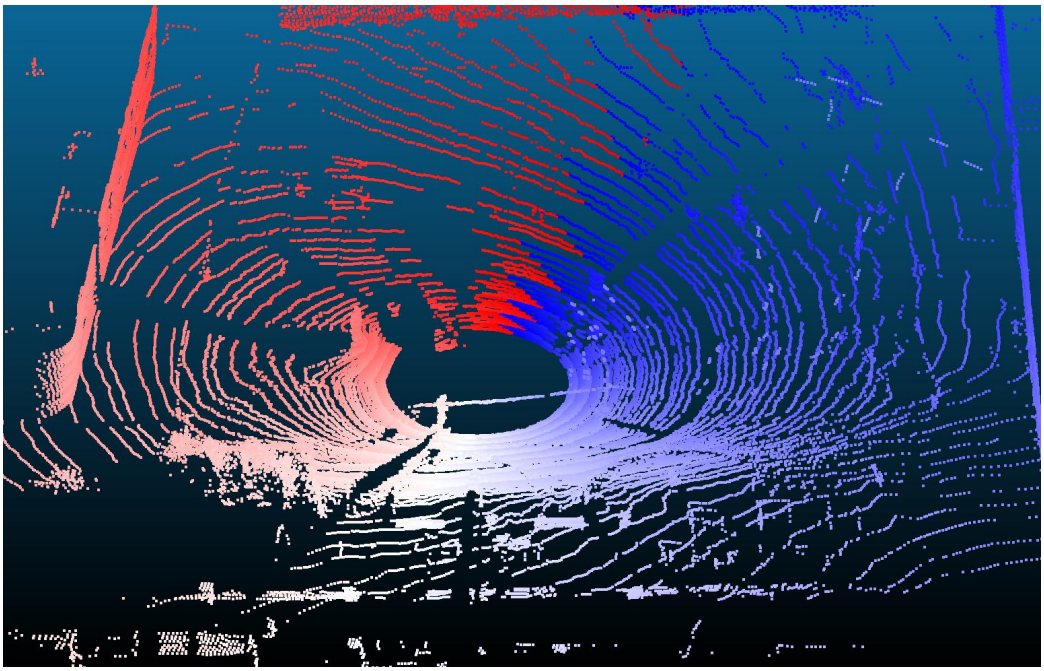


Figure 5.4 – Velodyne scan coloured by when the points were recorded (red for scan start, blue for the end). Note the sawtooth pattern where the blue and red meet showing where the scan starts/ends.

pattern shown in Figure 5.4. This means that some points recorded when the lidar scan started will be incorrectly timestamped as occurring near the end of the scan. To compensate for this the first and last 10% of the scan are discarded.

Once the time at which each point was scanned has been found and if we assume the surrounding environment is static, then the transformation between scans can be used to correct the position of the points. We assume that the velocity between scans is constant and utilise linear interpolation to correct the scan. This means that every point appears in the position it would have been in relative to the scanner when 50% of the scan had been completed. As, at this stage the transformation to the current scan is unknown, we use the transformation given by the previously matched scan. An example of this motion compensation can be seen in Figure 5.5

To estimate the variance of the calculated transformation vector we utilise the variance approximation method from Section 4.2.4 that makes use of the delta method approximation. For this method, while Velodyne specifies that the lasers have a stan-

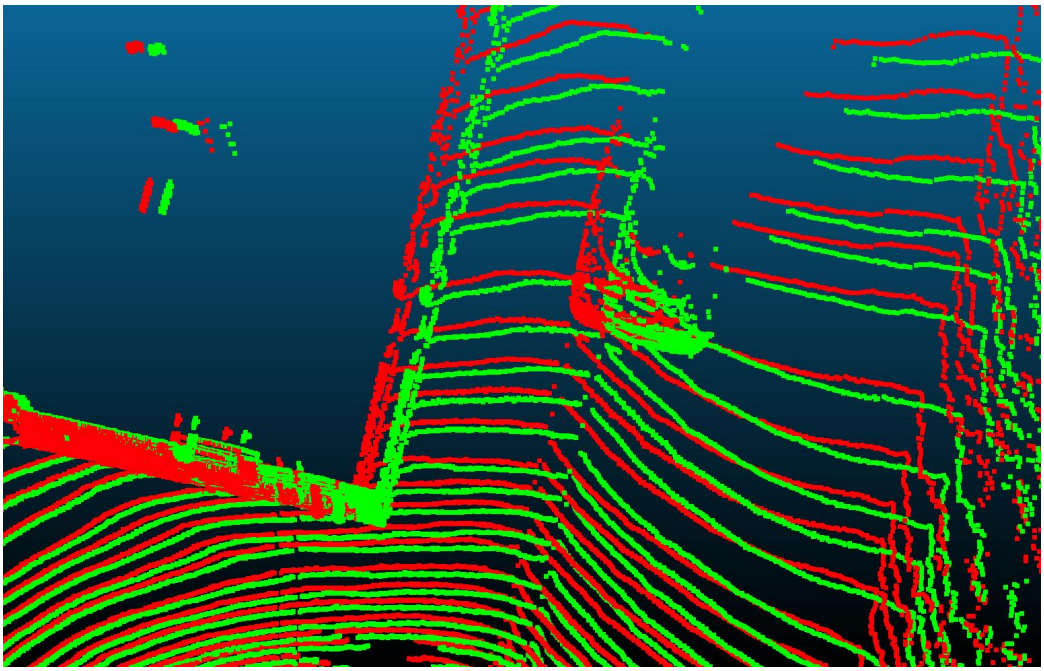


Figure 5.5 – Section of a Velodyne scan from the KITTI dataset made during a tight turn with (green) and without (red) motion compensation.

dard deviation in their range of approximately $2cm$, we use a σ of $10cm$ to account for any imperfections in the intrinsic parameters of the setup.

5.6.2 Cameras

The camera transforms are found using a fairly standard visual odometry approach. First, Harris corners are detected in an image before Lucas-Kanade optical flow [40] is used to find the location of these points in the next image. An example of this process is shown in Figure 5.6. Optical flow was used as opposed to feature-based approaches as it was found to generally find more matches and have fewer outliers. The use of optical flow meant that matches could only be done between images from similar viewpoints, which prevents loop closure from being used to refine the transformations. Loop closure, however, is generally only needed when attempting to build large consistent maps. In our approach we are only interested in the motion of the sensors and not their exact absolute position.

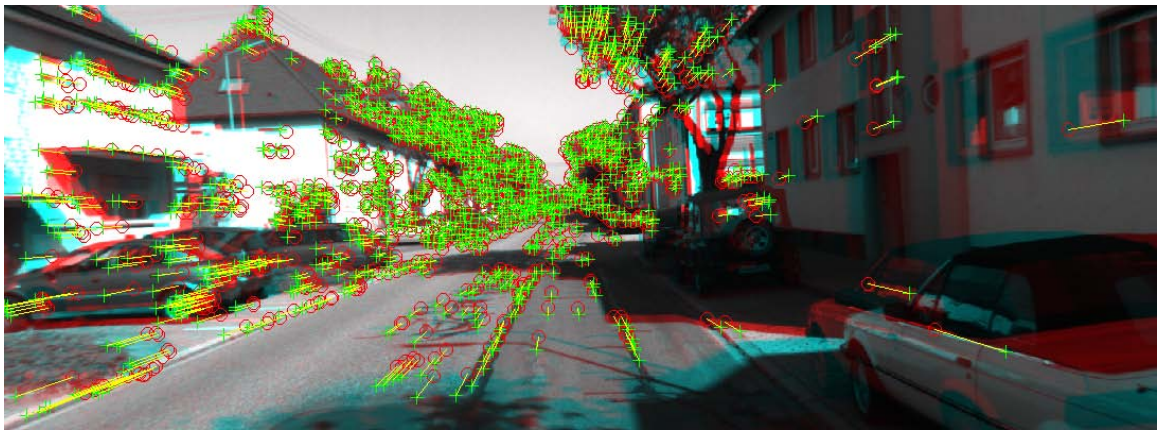


Figure 5.6 – Two frames from the KITTI dataset with the inlier matching points plotted.

A Maximum A Posteriori SAmple Consensus (MAPSAC) [92] implementation is used to reject outlier points. The inliers are used to find the fundamental matrix using the 8-point normalised algorithm [24]. This is combined with the known camera intrinsics to give the essential matrix. The essential matrix is finally used to find the transformation matrix between the positions.

As the only sensor employed is a monocular camera and its position on the vehicle is unknown, no sense of absolute scale of the movements can be obtained. Several methods were examined for providing an estimate of the relative scale between each of the cameras frames, however issues were encountered with these approaches. When a vehicle is travelling in a relatively straight path the scale of its movement relative to previous frames is highly sensitive to noise and generally has significant variance. This problem is made worse by the reliance of each frame on the scale estimated at previous frames, meaning that a single poor scale estimate can detrimentally affect all future estimates. Because of this, in order to gain reasonable scale estimates using only the camera data we would need to make further assumptions about our system, such as a constant height from a ground plane. This would reduce the flexibility of the approach. Therefore, only normalised camera transformations are used and the scale is estimated during the inter-sensor transformation estimation stage, where the information from additional sensors can be incorporated.

Once the transformation vector has been found, an estimate of its variance is computed. The calculation of this variance has no closed form solution. In our implementation we overcome this using a sampling strategy. This is done by either bootstrapping, or in most cases, assuming the tracked points have a variance in their position of 1 pixel and utilising the delta method.

5.6.3 GPS/INS Systems

GPS/INS sensors give the transformations directly and the variance of the sensor is given either by the manufacturer as a fixed value or in the case of the translation, is given as one of the outputs of the sensor. In order to convert the variance from the given roll, pitch and yaw values into the form used by our approach, a simple Monte-Carlo sampling approach was used.

5.7 Utilising Appearance to Refine the Calibration

While the accuracy of the motion-based methods depends on the dataset, motion and sensors used, through experimentation several common trends appear. Under all but the most extreme cases (for example datasets with a vehicle driving in a straight line at constant speed) accurate rotation with low uncertainty is obtained. The translation estimates however show more uncertainty. Generally for a non-holonomic ground vehicle the translation will be most accurately estimated in the direction tangential to the ground plane and perpendicular to the vehicle's main direction of motion. This accuracy stems from the large differences in translation the sensors will undergo during a typical turning manoeuvre, and for our test systems was in the range of 10 to 50mm. The least accurately estimated parameter was the translation in the direction of the normal of the ground plane. This is to be expected as with only planar movement this parameter would be completely unobservable. This means that this parameter has only been observed through the subtle movements caused by defects

in the road surface and the rocking of the system on its suspension. This parameter's accuracy was typically in the range of 100 to 1000mm.

With this level of error possible in the translation estimation, the output of the motion calibration provided by near-planar motion is of limited direct practical value. The calibration has not been without purpose though, as we can now use it to guide the optimisation of appearance-based metrics to further refine the transformation.

Initially, the true calibration between the sensors could take any parameter in a large six-dimensional search space. As was outlined in Section 4.1, for appearance-based metrics, this search space is typically highly non-convex and contains a large number of local optima. Therefore searching the entire search space is error prone and often intractable. However, our motion-based calibration estimates and associated variance vastly reduce the feasible regions the solution could lie within. Generally only one or two parameters still contain large uncertainty, with the remainder constrained to small regions. Because of this, our motion-based calibration is complementary to these appearance-based methods through the initialisation and constraining of their search space. The improvements in accuracy possible using this method are qualitatively shown in Figure 5.7.

For cameras and 3D lidar scanners, methods such as those presented by [35] or GOM can be utilised for the calibration refinement. These metrics operate by examining points at a single timestep, however we can exploit the already obtained motion information in combination with the appearance information to form a new lidar-camera alignment metric

5.7.1 Lidar-Camera Intensity-Motion Metric

Markerless metrics used to match between lidar scans and camera images either rely on correlating the intensity of the two modalities (MI,NMI) or aligning edges (Levinson's method [35], GOM). The problem with these metrics is that as the two sensors perceive the world in fundamentally different ways, there will be many cases where

Figure 5.7 – These images show a camera image from the KITTI dataset projected onto its corresponding Velodyne scan. The top image shows the result obtained by the motion only approach, while accurate rotation values have been estimated there is significant uncertainty and error in the translation. The bottom image shows the combined approach where the motion method has constrained an appearance-based optimisation, allowing a superior calibration to be obtained.

the features of interest are not present in both modalities. Overcoming these inconsistencies is one of the main issues in developing a robust multi-modal metric.

The addition of motion information, however, greatly simplifies our problem as it now becomes possible to align lidar and cameras using only mono-modal matching. As the metric used to achieve this alignment makes use of both intensity cues and motion cues in its development, we will henceforth refer to it as the Intensity-Motion (IM) Metric. An overview of the process used to achieve this is shown in Figure 5.8.

In this metric a camera image and its corresponding lidar scan are found. A static environment is assumed and the motion information estimated by the lidar used to compensate for any timing difference between when the camera image was taken and the lidar scan was recorded. Once this has been done the camera image is projected onto the lidar scan giving each of the lidar points an associated colour. The same lidar scan is then matched to the next camera image with the same process of using the motion information to compensate for the timing difference. This image is again

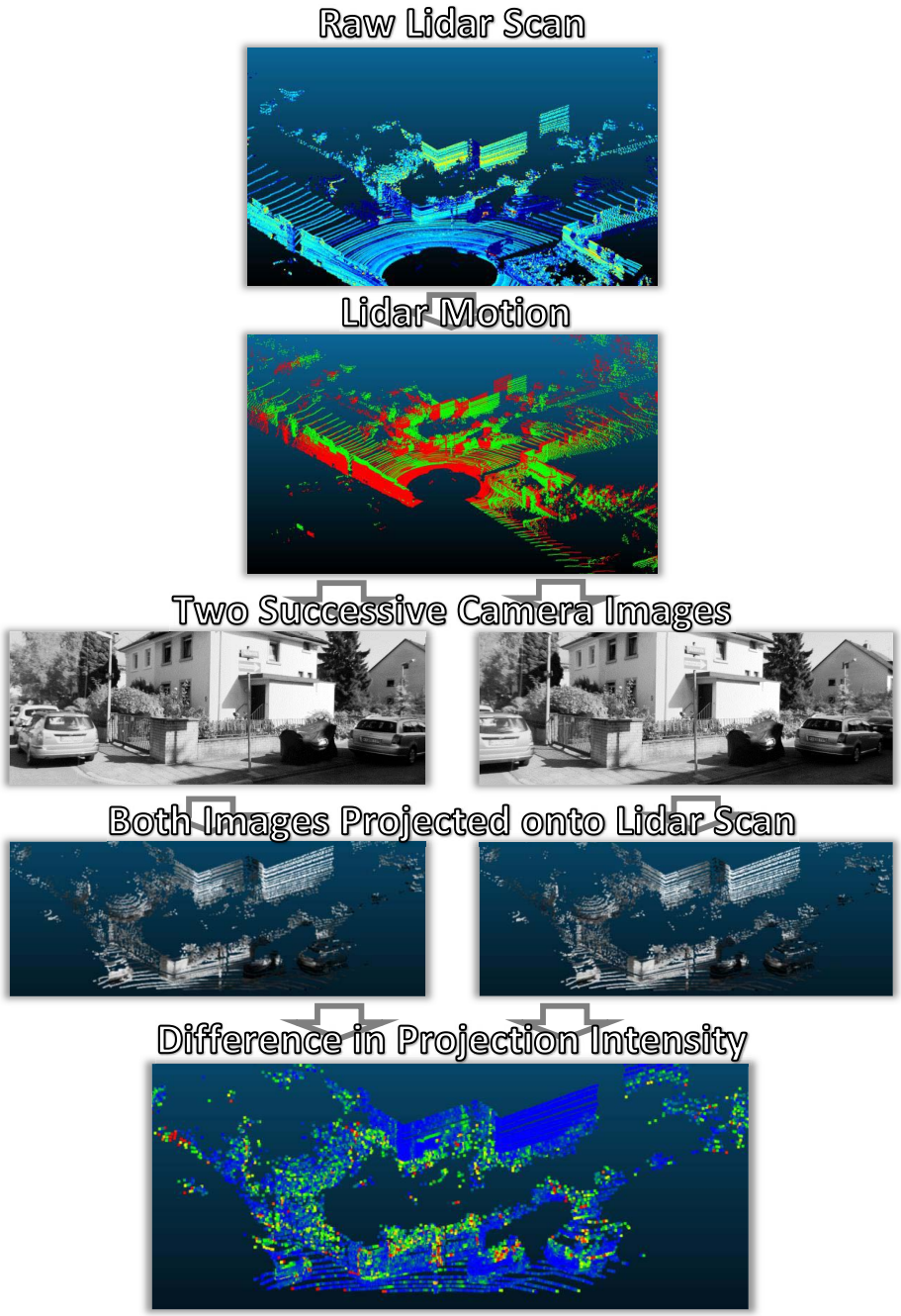


Figure 5.8 – Overview of the intensity-motion (IM) metric. The motion undergone by a lidar scan is used to project it into two successive camera images. The difference in the image intensity at the same position on the scan forms the metrics alignment error.

projected onto the lidar scan. This process has resulted in two different images being used to give colour information to the same lidar scan. If we assume constant lighting conditions, camera settings and a static environment, it would then be expected that, if the camera offset was correct both possible colourings would be identical. Because of this we can take the mean squared difference in their intensity as an indication of the sensor's alignment. Minimising this error should result in the correct lidar camera offset.

While in this simple form the metric will operate well in many scenarios to increase its reliability, several changes are made. First a Gaussian blur is applied to the image. In its original form, when there is a sharp edge of an object, a misalignment of a single pixel can result in no correlation between the intensities at this point. However when a Gaussian blur is applied, the error for small misalignments will be roughly proportional to the misalignment. This results in a smoother search space increasing an optimiser's likelihood of correctly converging to the true offset.

The second change is made to help minimise the impact that occlusions will have on the image. As the two images used are taken from two different perspectives, occlusions will impact the alignment. These occlusions will also occur due to the difference in the location of the lidar and camera. While a ray-tracing process could be implemented to calculate which points are likely to be occluded at each estimation step, this is a computationally expensive process. Instead we opt for a simple strategy that pre-emptively removes points that will have a high chance of becoming occluded during the sensor calibration. In this process we first project the points onto a sphere and use a kd-tree to find the 50 closest neighbours for each point. The neighbouring point closest to the camera is found and the distance between these points is found. This distance is than divided by the distance from the sensor to the points. If the final value is greater then a threshold (in our case set to 0.1) the point is rejected as having a high probability of being occluded. An example of the results of this process are shown in Figure 5.9.

The final and most significant change is to operate on a gradient image. By using a gradient image the importance of a metric correctly aligning the boundaries between

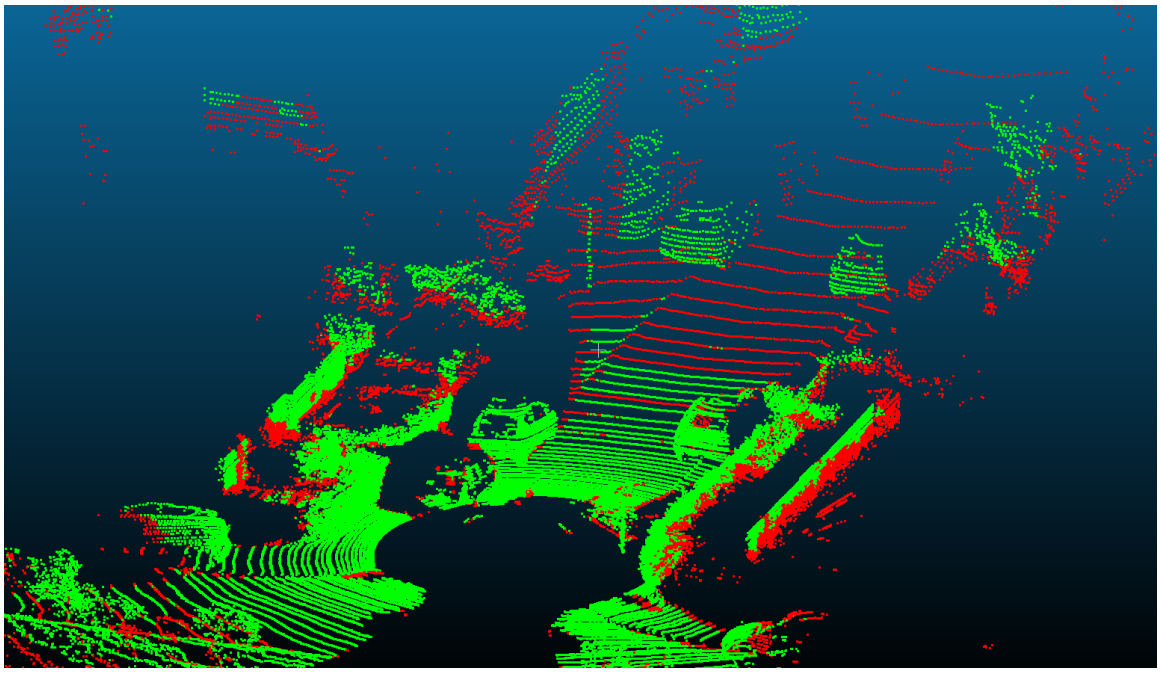


Figure 5.9 – A Velodyne scan filtered to remove points that have a significant chance of being occluded by others. The occluded points, shown in red, will be removed before the appearance-based alignment proceeds.

objects is significantly increased. It also allows us to remove regions of low texture by removing points with a gradient of 0. Removing these points prevents the metric from attempting to align the entire scan with a low texture region of the image, such as the sky.

The metric also allows for an estimation of its accuracy. This is done by once again making use of the delta method on the optimisation of the metric, with the variance of the motion used to find the variance in the output parameters. This estimation ignores several possible sources of error such as the noise in the camera intensities, noise in the lidar points, errors in the intrinsics and motion/changes in the observed scene. The justification for ignoring these effects is that in most cases the impact of these parameters will be orders of magnitude less than the motion error being considered.

The method also assumes that the correct global maxima has been found. This is a serious concern for appearance-based metrics, as was discussed in Section 4.1. In

our processing we assume that the constraining of the search space in combination with the aggregation of scans has been sufficient for this to be a valid assumption. To improve the convexity of the search space and the reliability of the delta method in these situations, a slight modification is made. Only points that are projected onto the image for all steps of the finite difference approach are used. This smooths the metric and increases the robustness of the estimate.

5.7.2 Lidar-Camera Optimisation

When optimising these methods we wish to make use of both the estimated solution and its associated variance provided by the motion estimation process. As we are operating with an appearance metric we also wish to make use of a global optimiser, as while significantly constrained, the metric may still have multiple optima within the feasible search space. To meet these requirements we make use of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimisation technique [23]. This technique randomly samples the search space using a multivariate normal distribution that is constantly updated. This optimisation strategy works well with our approach as the initial normal distribution required can be set using the variance from the estimation. This means that the optimiser only has to search a small portion of the search space and can rapidly converge to the correct solution.

5.7.3 Camera-Camera Optimisation

If both of the sensors are cameras and they have overlapping fields of view, then the calibration provided by the motion-based calibration may be refined using simple mono-modal matching. This is done by detecting and matching SURF [7] features present in the two cameras' field of view. MAPSAC is then used to reject outliers and the normalised transformation between the two cameras is found. The process of forming a transformation estimate from the inliers is bootstrapped 100 times to give an estimate of the transformational variance.

5.7.4 Combining the Refined Results

The pairwise transformations calculated in the appearance-based refinement step will not represent a consistent set of solutions to the transformation of the sensors. This is due to each pair of sensors obtaining their appearance-based calibration independently. This means, we again face the issue that if there are three sensors x, y and z, then for the current estimates $T_y^x T_z^y \neq T_z^x$. The camera-to-camera transformations also contain scale ambiguity.

To correct for this and find a consistent solution, the transformations are combined. This is done by using the calculated parameters to find a consistent transformation that has the highest probability of occurring. We do this by first using the transformations to the base sensor to generate an initial guess. The probability of this solution occurring, given the transformation and variance of all the pairwise transforms, is calculated and used as a cost function that is optimised using Nelder-Mead Simplex optimisation. In this optimisation we make use of the seven-element transformation vector $[nx, ny, nz, s, rx, ry, rz]$ as the additional scale term allows the use of the calculated camera-to-camera translation through setting the variance of the scale parameter to ∞.

5.8 Reducing the Uncertainty of the Final Calibration

The accuracy of motion-based methods is dependent on the range of motion observed and the accuracy of the motion generated by the sensors. In the case of a system undergoing purely rotation motion, information about the translational offset between the sensors can still be obtained. This is because the sensors will observe translation that is proportional to their distance from the axis of rotation.

In the case of pure planar motion it is fairly simple to show that the offset perpendicular to this plane is unobservable. At first glance this would appear to prevent

the system from being used on most ground vehicles whose motion is typically approximately planar. However on most systems, factors such as defects in the surface, the rocking of suspension and give in the tires result in sufficient motion to provide an estimate of the perpendicular offset. However the accuracy of this measurement is typically over an order of magnitude less accurate than the others. While we have presented an appearance-based refinement step to help alleviate this issue, for systems calibrating a GPS/INS system or without overlapping fields of view in its sensors, this issue will remain. However, if a ground vehicle is being driven purely for the purpose of calibration this issue can still be overcome.

An ideal situation would be to perform a series of tight turns over an uneven surface. For example driving up a multi-storied parking lot or performing a three-point turn over a speed bump would meet these requirements. The tight turns in combination with the uneven ground would maximise the observed motion in each direction and thus the accuracy. Accurate lidar- and camera-motion estimation also operate best in static environments with a range of distinct objects in their view. Alternatively, if the vehicle is small or the sensor array removable, simply holding and randomly waving the system around by hand would yield an accurate calibration. In the case of aerial systems, planar motion is less of an issue and simply performing banked turns should be sufficient to allow accurate calibration.

The framework outlined can perform the calibration of a uni-modal system, albeit with two limitations. The first, is that in systems consisting of only cameras, there is no measure of absolute scale and so no translation parameters can be estimated. The second, is that the framework makes limited use of uni-modal inter-sensor point matching. This allows it to calibrate multi-modal systems, but in the uni-modal case, means it does not fully utilise a possible source of alignment information. Due to this limitation, the results of the calibration may be less accurate then if a process designed specificity for the modality of interest is used.

5.9 Summary

In this chapter we have explored the idea of calibrating a mobile system's sensors from the individually observed sensor motion. We have extended current hand-eye calibration techniques by incorporating a measure of each readings uncertainty into the process. This transforms the process of the sensor calibration from a pair-wise least squares metric, to a probabilistic method that can reason about the value of information and present an estimate of its confidence in the results produced. The process also allows the possibility of simultaneously calibrating more than two sensors.

This motion-based approach has several advantages over appearance-based metrics. It does not require any initial guess as to the sensors' setup and can operate when the sensors lack an overlapping field of view. It also gives a single unified approach for aligning any type of sensor.

It has a distinct disadvantage however; it requires a large range of motions to give accurate calibration results for all of the parameters considered. For vehicles that operate on a roughly planar surface this can result in significant uncertainty about the offset in the direction of the plane's normal.

As this large uncertainty is typically only in one or two dimensions the motion result can be used to constrain and guide an appearance-based alignment. This combines the strengths of both approaches and eliminates the need for an accurate initial guess that is typically one of the key limitations when using appearance-based methods.

In the next chapter, we evaluate the accuracy and applicability of all of the methods described in this chapter and Chapter 3. This is done through a series of experiments where the methods are utilised to calibrate sensors over a range of challenging datasets.

Chapter 6

Experimental Results

6.1 Introduction

This chapter performed an experimental evaluation of the approaches proposed in this thesis as well as comparisons to several state-of-the-art methods from the literature. Briefly the experiments presented in this section examine the following areas;

- The alignment of lidar-scans with images using appearance-metrics.
- The alignment of multi-modal image data.
- The impact of using point distance or normals as features and the basin of attraction of metrics during optimisation.
- The accuracy of motion-based calibration in finding the timing, rotational and translational offset of various sensor systems.
- The impact that simultaneously calibrating more then two sensors has on the resulting calibration
- The impact that using the motion-based solution to constrain appearance-based optimisation methods has on the accuracy and robustness of the solutions.
- The overall accuracy and robustness of the system when all stages of the process are combined and an evaluation of the estimated uncertainty.

6.2 Experimental Platforms

Over the course of the thesis a large range of experiments were performed utilising a variety of sensor platforms. A brief description of each of these platforms is outlined below.

6.2.1 KITTI Dataset Car

The KITTI dataset is a well known publicly available dataset obtained from a sensor vehicle driving in the city of Karlsruhe in Germany [18]. The sensor vehicle is equipped with two sets of stereo cameras, a Velodyne HDL-64E and a GPS/INS system. The setup is shown in Figure 6.1. This system was chosen for calibration due to the ease of availability and the excellent ground truth available due to the recalibration of its sensors before every drive.

All of the results presented here were tested using drive 27 of the dataset. In this dataset the car drives through a residential neighbourhood. Drive 27 was selected as it is one of the longest of all the drives provided in the KITTI dataset, giving 4000 consecutive frames of information on which to test the calibration method.

6.2.2 Ford Campus Vision and Lidar Dataset Car

The Ford campus vision and lidar dataset is a second publicly available dataset. It was produced by driving a Ford F-250 pick-up truck around Dearborn, Michigan in the United States [60]. The car is equipped with a Ladybug spherical camera, a Velodyne HDL-64E, two push broom forward-looking Rigel lidars and a GPS/INS system. This setup is shown in Figure 6.2.

6.2.3 ACFR's Shrimp

Shrimp is a general purpose sensor vehicle used by the ACFR to gather data for a wide range of applications. Due to this it is equipped with an exceptionally large

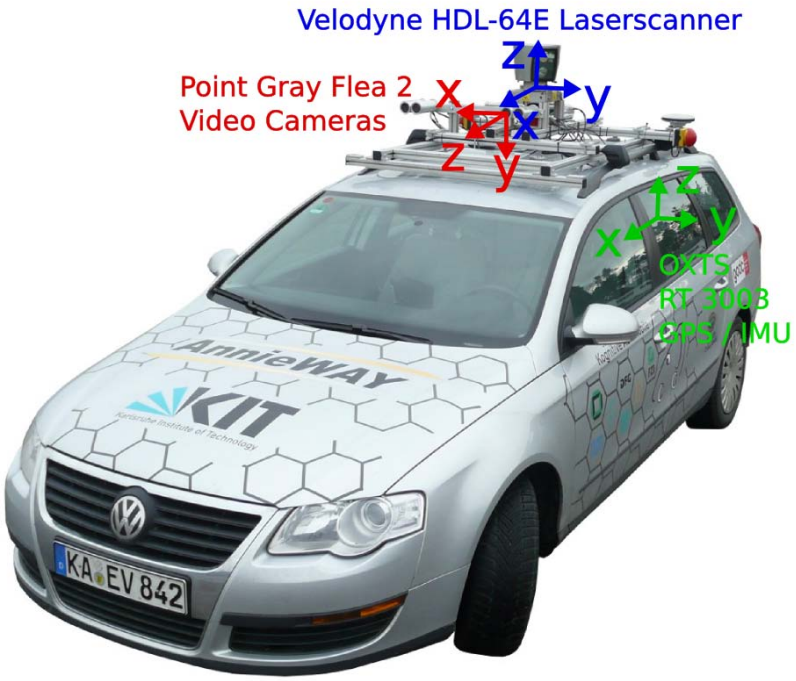


Figure 6.1 – The car used to gather the KITTI dataset equipped with all of its sensors.

Figure 6.2 – The system used to gather the Ford campus vision and lidar dataset.

Figure 6.3 – ACFR's shrimp robot.

array of sensors. These sensors included a Ladybug spherical camera, Velodyne HDL-64E lidar, Bumblebee XB3 stereo camera, thermal IR camera, two Sick lidar's and a Novetal GPS/INS system. The setup is shown in Figure 6.3. For our experiments the shrimp vehicle was driven around a quadrangle outside the ACFR at the University of Sydney. An extrinsic calibration between the system's lidar and GPS/INS had previously been generated using a target-based method. While a calibration also existed between the lidar and cameras this had been obtained using an appearance-based approach evaluated in this thesis. Because of the possible bias this lidar-camera calibration may show, it was not used in evaluating any of our results.

Figure 6.4 – ACFR's autonomous light vehicle.

6.2.4 ACFR's Autonomous Light Vehicle

The ACFR has a Toyota Hilux that has been converted for autonomous driving as well as holding a suite of sensors. The sensors mounted depend on the application, however it is generally equipped with two Sick lidars, Specim hyperspectral Visible and Near InfraRed (VNIR) and Short Wave InfraRed (SWIR) cameras, a Rigel VZ-1000 lidar and a Novetal GPS/INS system. The setup is shown in Figure 6.4.

6.2.5 RTCMA Tripod-Based Setup

The Rio Tinto Centre for Mine Automation (RTCMA) possesses high-resolution scanning lidar systems in the form of the Rigel VZ-1000 and LMS-Z420i. These systems can quickly be mounted onto a tripod and used to make a single high resolution scan of an area. A camera that is either hand held or mounted to a separate tripod is then used to take images of the same area. Over the course of the thesis a large range of cameras have been used with this setup, ranging from scanning hyperspectral cameras to smart phones. These include, but are not limited to, the Neo HySpex hyperspectral camera, Specim VNIR camera, Specim SWIR camera, Cannon 5D, Cannon Powershot A720, Samsung Galaxy 3, Iphone 5, Sony Xperia Z1 and a Nokia Lumia 520. A

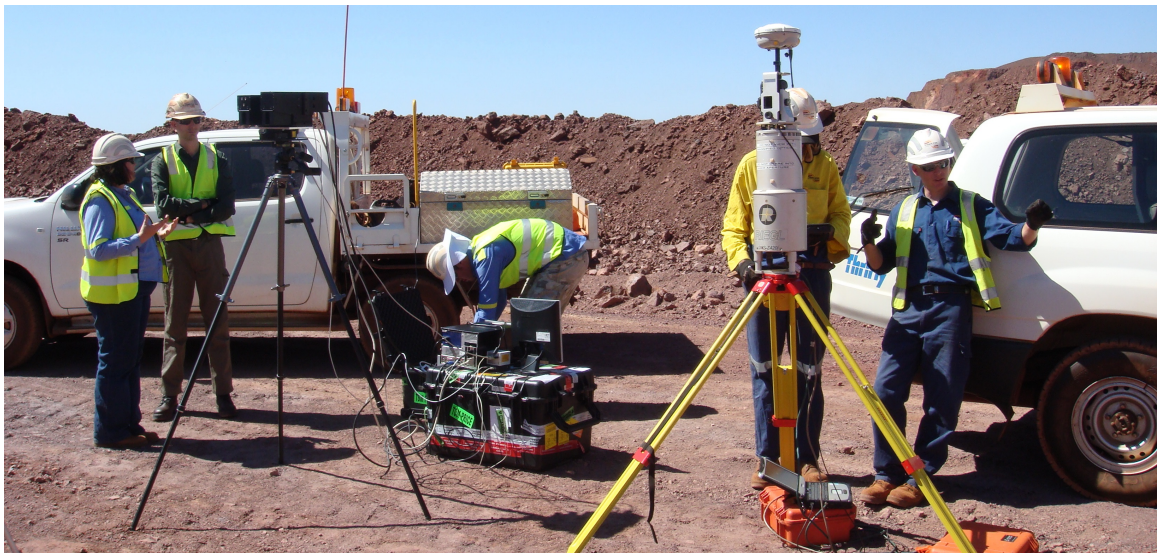


Figure 6.5 – Tripod-based experimental setup used to gather a dataset of several cliff faces in a mine site.

typical setup is shown in Figure 6.5

6.2.6 Ground Truth

In all of the following experiments the methods used are compared to existing calibration values that were provided with the datasets and taken to be the 'ground truth'. The nature of how these ground truth values are generated depends on the sensor and dataset, however most will have been generated using a marker-based method overseen and verified by an expert. This should in most instances yield a highly accurate solution, however any estimate to the actual accuracy of this ground-truth is typically unknown.

In our results the inaccuracies present in this ground-truth will, in most cases detrimentally affect the perceived accuracy of the tested methods. Inspite of this we have chosen to compare our results to these pre-existing calibration values rather than simply reporting the precision of the method. This choice was made as the given precision can, in some cases, greatly overestimate the accuracy of a method and fails to correctly capture any bias a technique may exhibit.

6.2.7 Implementation

The majority of the code for testing the methods we have outlined was written in Matlab, with Mex files written in C and C++ used to optimise bottlenecks in the code. The most computationally expensive sections of the appearance-based techniques (the transformation of the point cloud, interpolation of images and the evaluation of the metrics) were further optimised by making use of Cuda to allow their evaluation on a GPU.

All of the results were generated on a desktop outfitted with an i7-4770 Central Processing Unit (CPU), 8 GB of system memory, a GTX760 GPU and 2 GB of graphics memory. On this system the transformation, interpolation and GOM evaluation of a Velodyne scan containing 80,000 points with a camera image requires roughly 4 ms. The source code used in the generation of the results in this chapter is available at [84].

6.3 Appearance-Based Metrics

6.3.1 Metrics Evaluated

In this section, a series of metrics are evaluated on three different datasets. The metrics evaluated are as follows:

- MI - mutual information, the metric used by [61] in their experiments on the Ford dataset [60].
- NMI - normalised mutual information, a metric we had used in our previous work on multi-modal calibration [85].
- The Levinson method [35].
- GOM - the gradient orientation measure presented in Chapter 3
- SIFT - scale invariant feature transform, a mono-modal registration technique included to highlight some of the challenges of multi-modal registration and calibration.

For MI, NMI, GOM and the Levinson method, we wrote our own code and optimised each one using the same optimisation process and parameters. The SIFT implementation was taken from code written by Vedaldi and Fulkerson [97] and the best match was found using a RANdom SAmple Consensus (RANSAC) implementation. The SIFT method was only applicable, and thus used, on the hyperspectral image dataset.

6.3.2 Parameter Optimisation

For the evaluation of the appearance-based metrics we initialise the optimisation using either the ground truth (when available) or a manually calibrated solution. We then added a random offset to it. The random offset is uniformly distributed, with the maximum value used given in the details of each experiment. This random offset is introduced to ensure that the results obtained from multiple runs of the optimisation are a fair representation of the method's ability to converge to a solution reliably. When particle swarm optimisation is used, the search space of the optimiser is set to be twice the size of the maximum offset.

On datasets where no ground truth was available, the search space was always constructed so that the space was much greater than twice the estimated error of the manual calibration, to ensure that it would always be possible for a run to converge to the correct solution. All experiments were run 10 times with the mean and standard deviation from these runs reported for each dataset.

6.3.3 Registration of a Single Image to a High-Resolution Lidar Scan

ACFR's Autonomous light vehicle was used to take four high-resolution lidar scans and hyperspectral images of our building, from the grass courtyard next to it. The scanner output gave the location of each point as its latitude, longitude and altitude. The focal length of the hyper-spectral camera was adjusted between each scan. This

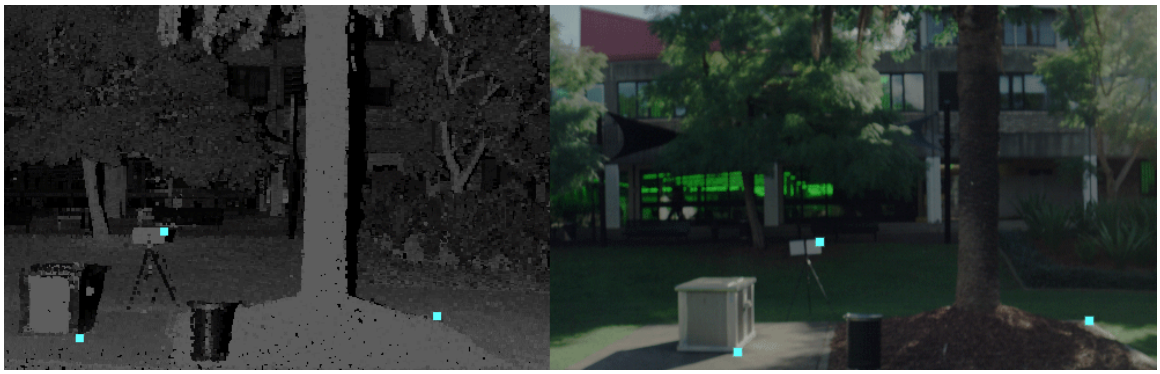


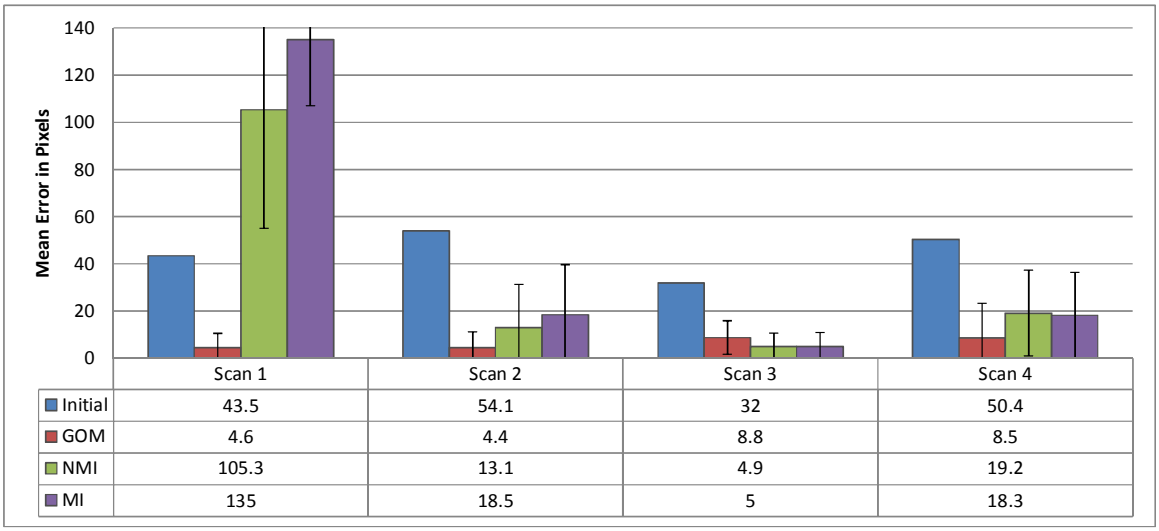
Figure 6.6 – Hand labelled points for a section of ACFR scan 1 aligned by GOM. The left image shows the lidar scan with three of the labelled points highlighted in blue. The right image shows the camera image with the three corresponding points highlighted.

was done due to the different lighting conditions and to simulate the actual data collection process in the field.

This dataset required the estimation of an intrinsic parameter of the camera, its focal length in addition to its extrinsic calibration. To test the robustness and convergence of the methods, each scan was first roughly manually aligned. The search space was then constructed assuming the roll, pitch and yaw of the camera were each within 5 degrees of the lasers. The camera's focal length was within 40 pixels of correct (for this camera focal length ≈ 780) and the X, Y and Z coordinates were within 1 metre of correct. These values represent a realistic range in which the sensors parameters may lie.

No accurate ground truth is available for this dataset. To overcome this issue and allow an evaluation of the accuracy of the method, 20 points in each scan-image pair were matched by hand. An example of this is shown in Figure 6.6. An evaluation of the accuracy of the method was made by measuring the distance in pixels between these points on the generated images. The results are shown in Table 6.1.

For this dataset GOM significantly improved upon the initial guess for all four of the tested scans. Scans 1 and 2 were however more accurately registered than scans 3 and 4. These last two scans were taken near sunset, and the long shadows and poorer light may have played a part in the reduced accuracy of the registration. NMI gave mixed



	Scan 1	Scan 2	Scan 3	Scan 4
Initial	43.5	54.1	32	50.4
GOM	4.6	4.4	8.8	8.5
NMI	105.3	13.1	4.9	19.2
MI	135	18.5	5	18.3

Table 6.1 – Accuracy comparison of different methods on ACFR dataset. All distances in pixels.

results on this dataset, misregistering scan 1 by a large margin and giving results far worse than GOM's for scans 2 and 4. It did however outperform all other methods on Scan 3. MI gave a slightly worse, but similar, performance. Levinson's method could not be evaluated on this dataset as it requires multiple images to operate.

6.3.4 Multi-Modal Image Registration

To test each method's ability to register different modality camera images such as IR-RGB camera alignment, two scenes were scanned with a hyperspectral camera. Hyperspectral cameras capture images from a large number of light wavelengths at the same time. This made them ideal for testing the method's accuracy as the different modality images are initially perfectly aligned, due to all the different wavelength images being captured at the same time onto the same CCD. This removes any difference in the camera intrinsics or extrinsics, and means that a perfect ground truth exists and it is easy to quantify any error a registration method has.

For the hyperspectral camera images, bands near the upper and lower limits of the camera's spectral sensitivity were selected, so that the modality of the images compared would be as different as possible, providing a challenging dataset on which to

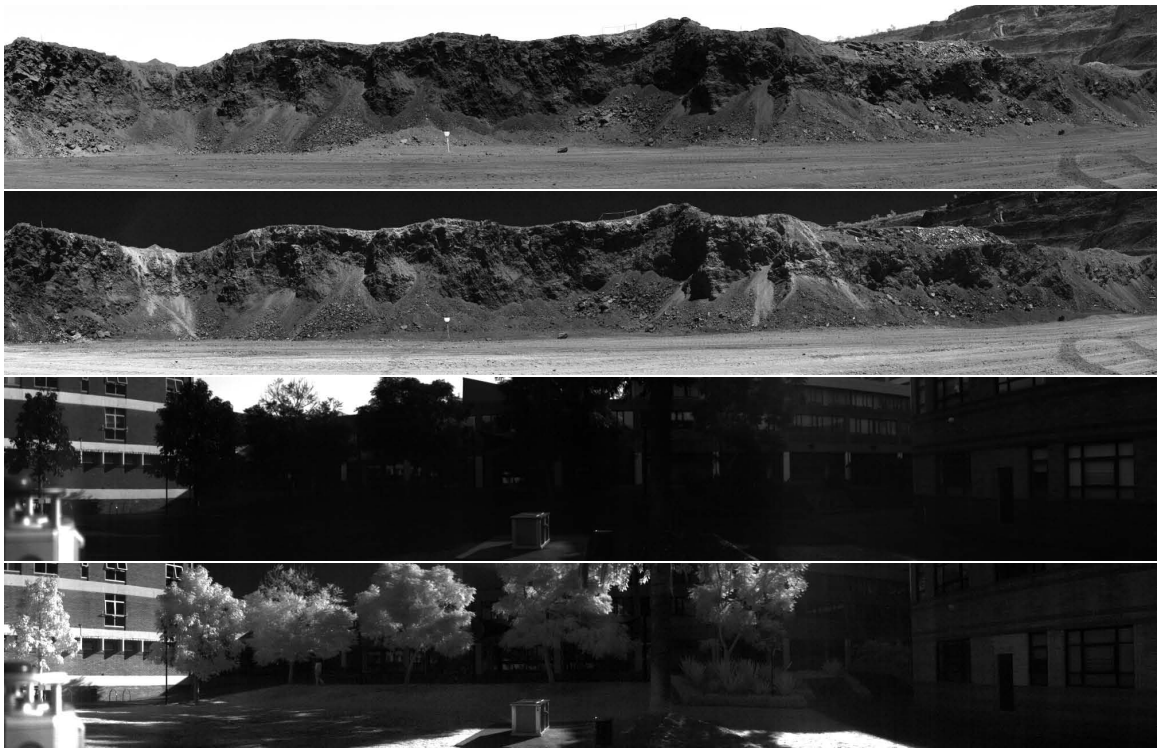
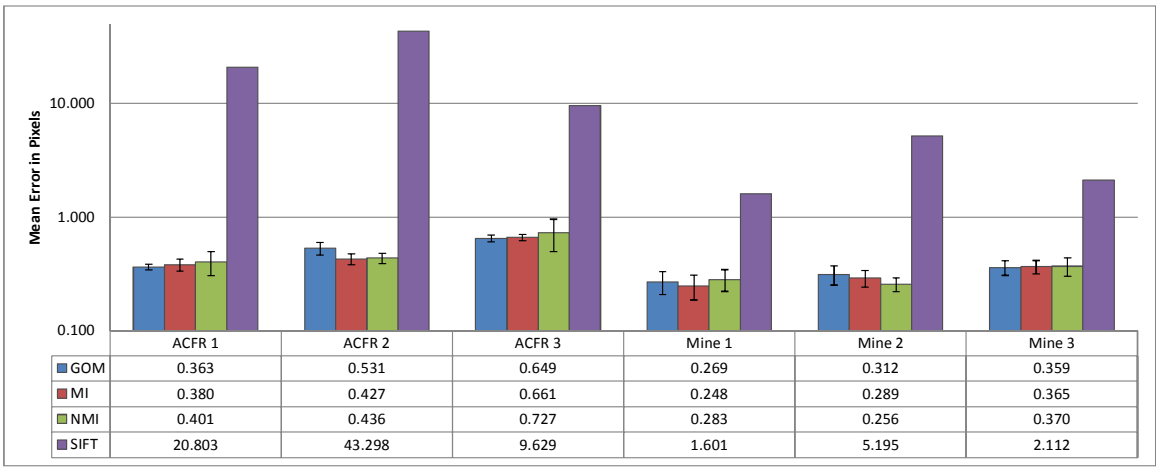


Figure 6.7 – Images captured by hyperspectral camera. From top to bottom: 420nm mine 1, 950nm mine 1, 420nm ACFR 1, 950nm ACFR 1.

perform the alignment. The bands selected were at 420 nm (violet light) and 950 nm (near IR). The camera was used to take a series of three images of the ACFR building and three images of cliffs at a mine site. An example of the images taken is shown in Figure 6.7.

The search space for the particle swarm optimiser was set up assuming the X and Y translation were within 20 pixels of the actual image, the rotation was within 10 degrees, the X and Y scale were within 10% and the X and Y shear were within 10%.

In addition to the GOM, MI and NMI methods that have been applied to all of the datasets, SIFT features were also used. SIFT was used in combination with RANSAC to give the final transform. To measure how accurate the registration was, the average difference in position between each pixel's transformed position and its correct location was obtained. The results of this registration are shown in Table 6.2. The images taken at the ACFR were 320 by 2010 pixels in size. The width of the



	ACFR 1	ACFR 2	ACFR 3	Mine 1	Mine 2	Mine 3
GOM	0.363	0.531	0.649	0.269	0.312	0.359
MI	0.380	0.427	0.661	0.248	0.289	0.365
NMI	0.401	0.436	0.727	0.283	0.256	0.370
SIFT	20.803	43.298	9.629	1.601	5.195	2.112

Table 6.2 – Error and standard deviation of different registration methods performed on hyperspectral images. Error is given as the mean per-pixel error in position. Note that the chart's axis uses a log scale.

images taken at the mine varied slightly, but were generally around 320 by 2500 pixels in size.

SIFT performed rather poorly on the ACFR dataset and reasonably on the mining dataset. The reason for this difference was most likely due to the very different appearance vegetation has at each of the frequencies tested. This difference in appearance breaks the assumption SIFT makes of only linear intensity changes between images, and therefore the grass and trees at the ACFR generate large numbers of incorrect SIFT matches. In the mine sites that are devoid of vegetation, most of the scene appears very similar, allowing the SIFT method to operate and give more accurate results.

Looking at the mean values for each run MI, NMI and GOM gave similar performance on these datasets, all achieving sub-pixel accuracy in all cases. There was little variation in the results obtained using the multi-modal metrics, with all three methods always giving errors between 0.2 and 0.8 pixels.

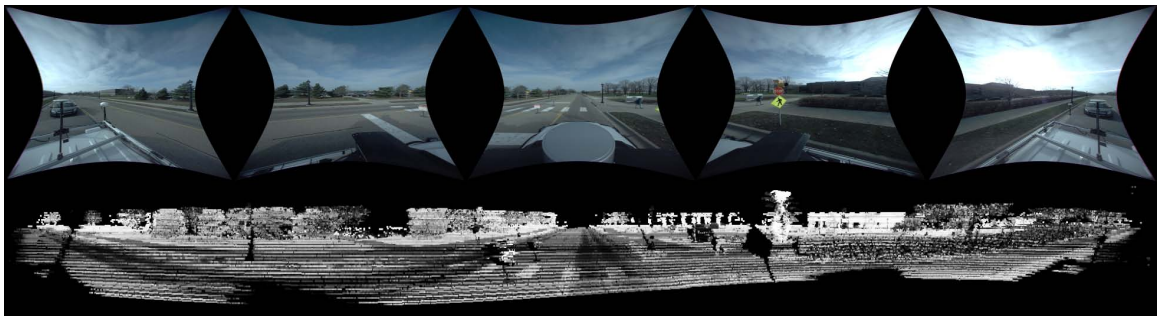


Figure 6.8 – Example scan-image pair from the ford dataset. Top: Camera images. Bottom: lidar scan coloured by the intensity of laser return.

6.3.5 Camera-Velodyne Calibration from Multiple Scans

The camera-Velodyne evaluation was performed on the Ford campus vision dataset as it offers a variety of environments, and the Velodyne scanner used in this test has been calibrated to account for the different return characteristics of each laser. An example of the data is shown in Figure 6.8. The methods were tested on a subset of 20 scans. These scans were chosen as they were the same scans used in the results presented by Pandey et al. [61]. Similarly, the initial parameters used were those provided with the dataset. As all of the scan-image pairs on this dataset shared the same calibration parameters, aggregation of the scans can be used to improve the accuracy of the metrics. Because of this, each experiment was performed three times, aggregating 2, 5 and 10 scans.

The Ford dataset does not have a ground truth. To overcome this problem we used a measure of the calibration accuracy through the use of the Ladybug camera. The Ladybug consists of five different cameras all pointing in different directions (excluding the camera pointed directly upwards). The extrinsic location and orientation of each of these cameras is provided by the manufacturer with respect to one another. This means that if the calibration is performed for each camera independently, the error in their relative location and orientation will give a strong indication as to the method's accuracy.

All of the cameras are calibrated independently. An example of the process of registering one of the camera's outputs is shown in Figure 6.9. This calibration was

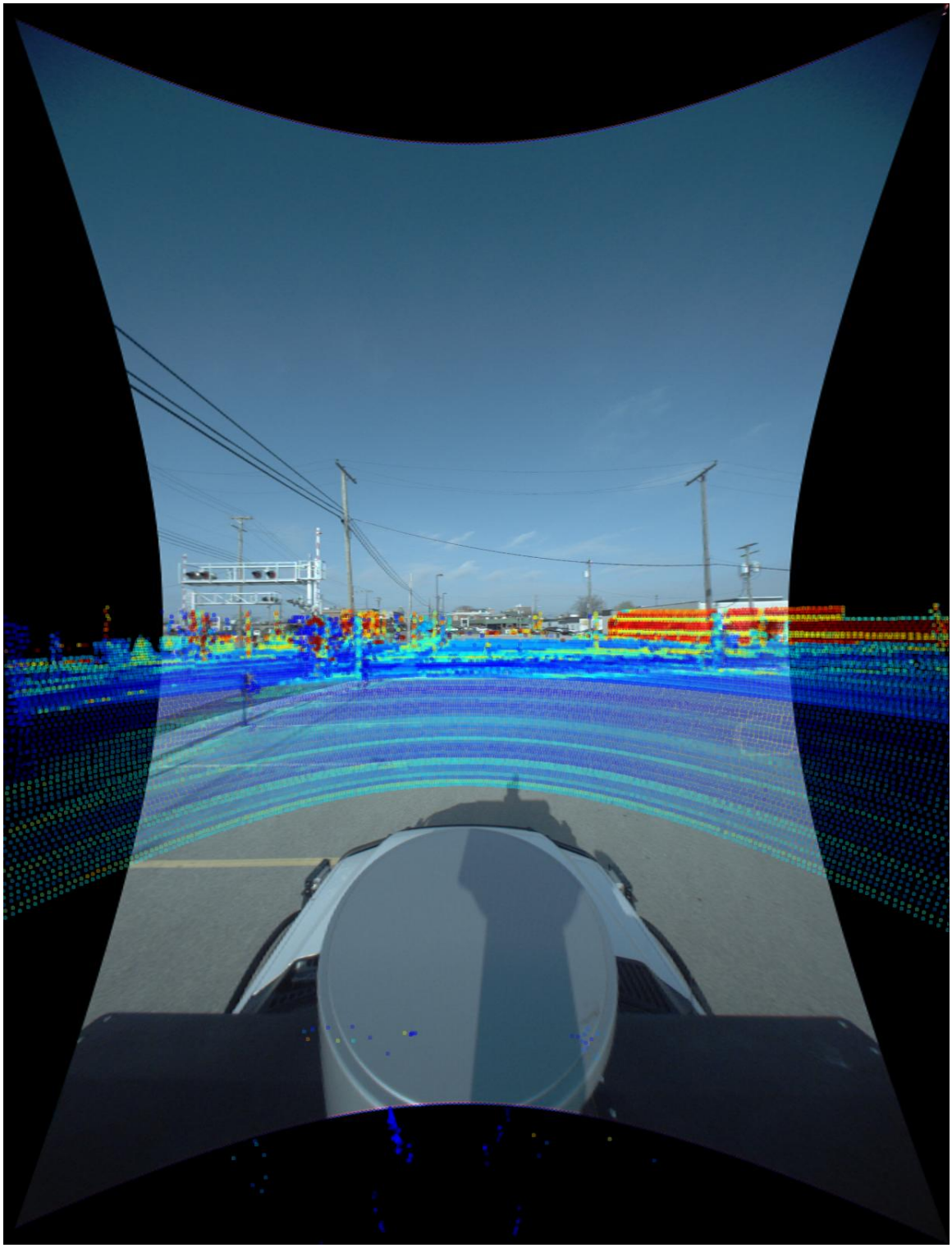
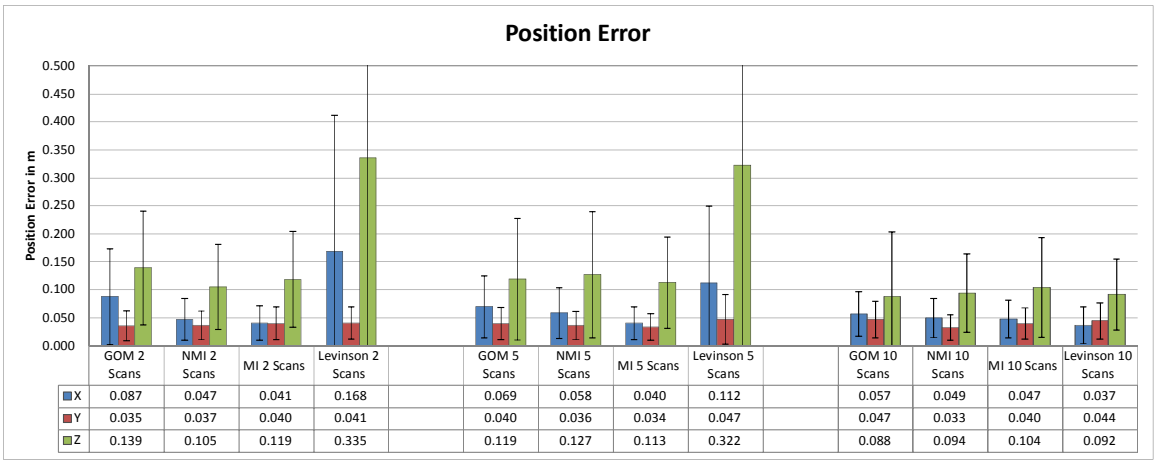
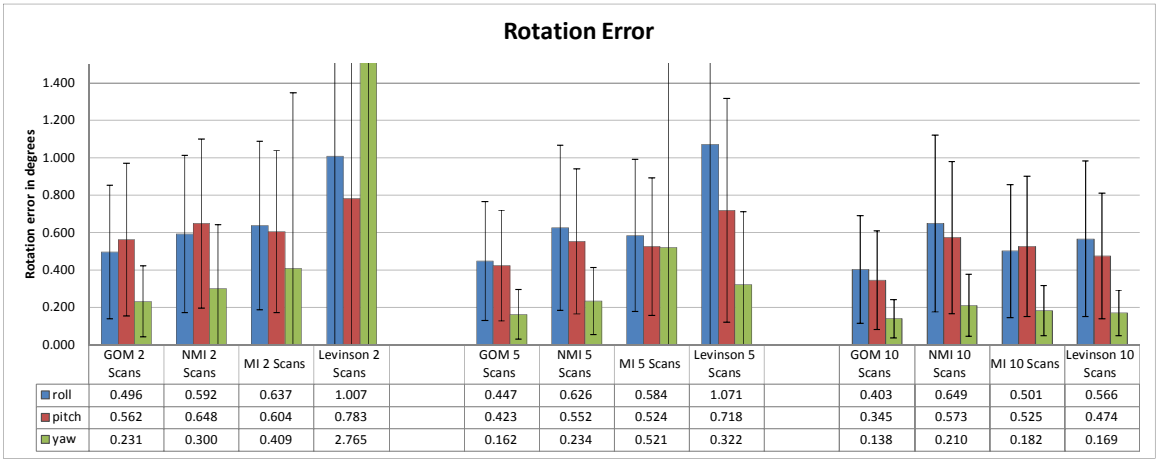


Figure 6.9 – Camera and Velodyne scan being registered. The current estimate for the extrinsic calibration has been used to project the Velodyne data onto the camera image.



	GOM 2 Scans	NMI 2 Scans	MI 2 Scans	Levinson 2 Scans	GOM 5 Scans	NMI 5 Scans	MI 5 Scans	Levinson 5 Scans	GOM 10 Scans	NMI 10 Scans	MI 10 Scans	Levinson 10 Scans
X	0.087	0.047	0.041	0.168	0.069	0.058	0.040	0.112	0.057	0.049	0.047	0.037
Y	0.035	0.037	0.040	0.041	0.040	0.036	0.034	0.047	0.047	0.033	0.040	0.044
Z	0.139	0.105	0.119	0.335	0.119	0.127	0.113	0.322	0.088	0.094	0.104	0.092



	GOM 2 Scans	NMI 2 Scans	MI 2 Scans	Levinson 2 Scans	GOM 5 Scans	NMI 5 Scans	MI 5 Scans	Levinson 5 Scans	GOM 10 Scans	NMI 10 Scans	MI 10 Scans	Levinson 10 Scans
roll	0.496	0.592	0.637	1.007	0.447	0.626	0.584	1.071	0.403	0.649	0.501	0.566
pitch	0.562	0.648	0.604	0.783	0.423	0.552	0.524	0.718	0.345	0.573	0.525	0.474
yaw	0.231	0.300	0.409	2.765	0.162	0.234	0.521	0.322	0.138	0.210	0.182	0.169

Table 6.3 – Average error between two aligned Ladybug cameras. All distances are in metres and angles are in degrees.

performed 10 times for each camera, using randomly selected scans each time. The error in each camera's relative position to each other camera in all trials was found and the average error shown in Table 6.3.

In these tests GOM, NMI and MI gave similar results. GOM tended to give the most accurate rotation estimates while MI gave the most accurate translation estimates. For all three of these metrics, scan aggregation slightly improved the accuracy of angles and position. Levinson's method presented the largest improvement in accuracy when more scans were aggregated, resulting, however in the largest error with 2 and 5 scans and giving similar results to the other methods with 10 scans.

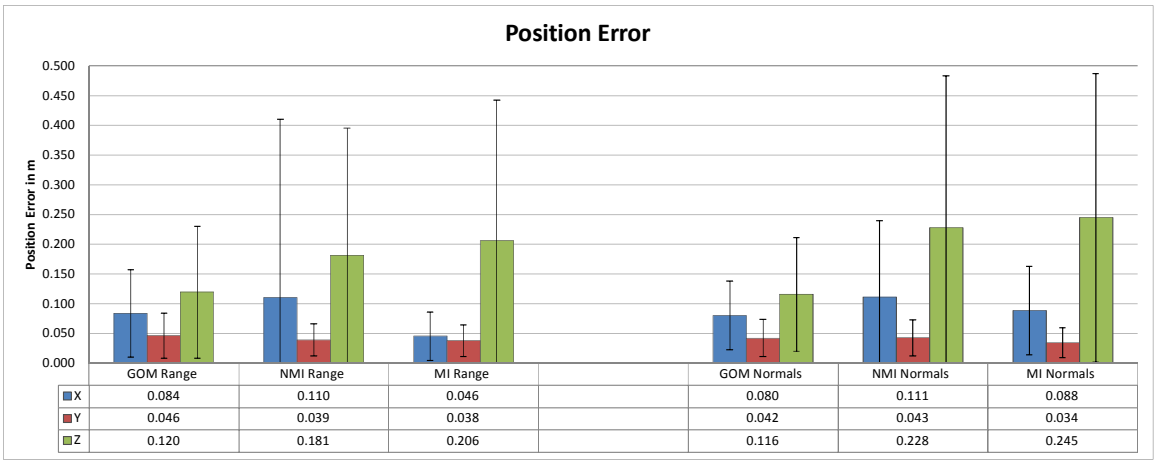
In this experiment, any strong conclusion about which metric performed the best is

difficult to draw as the difference between any two metrics for 10 aggregated scans is significantly less than the variance in their values. In almost all of the tests, the estimate of the cameras Z position was significantly worse than the X and Y estimates. This was expected, as the metric can only be evaluated in the overlapping regions of the sensor's fields of view. The Velodyne used has an extremely limited vertical resolution (64 points, one for each laser). Thus making the parallax error that indicates an error in the Z position difficult to observe. The narrow beam width of the Velodyne is also why the yaw shows the lowest error, as there are more overlapping points that can be used to evaluate this movement.

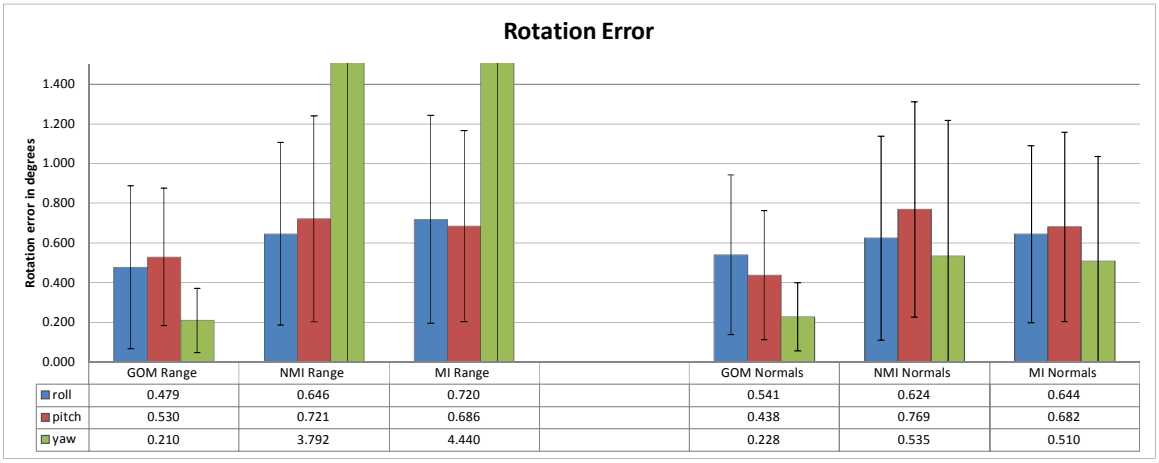
The actual error of a Ladybug-Velodyne system calibrated using all five cameras simultaneously would give a far more accurate solution than the results obtained here. There are several reasons for this. Individually the single camera systems have a narrow field of view. Therefore, a forward or backward translation of the camera is only shown through subtle parallax error in the position of objects in the scene. This issue is significantly reduced in the full system due to the cameras that give a perpendicular view that clearly shows this movement. In the single camera problem, movement parallel to the scene is difficult to distinguish from a rotation. This is also solved by the full system due to the very different effects a rotation and translation have on cameras facing in significantly different directions. Finally the full system also benefits from the increase in the amount of overlap between the sensors' fields of view.

6.3.6 Feature Comparison

In many lidar datasets, return intensity is not available. This may be due to the fusing of uncalibrated scans, the sensor having a low number of intensity bits (many lidar only have 3 bit return intensity) or the sensor returns tuned to detect corner reflectors giving almost zero intensity for all other points. In these situations the alternative features suggested in Section 3.3, that utilise either the normals or distance of the points from the sensor, must be used. To test their accuracy the Ford dataset was



	GOM Range	NMI Range	MI Range		GOM Normals	NMI Normals	MI Normals
X	0.084	0.110	0.046		0.080	0.111	0.088
Y	0.046	0.039	0.038		0.042	0.043	0.034
Z	0.120	0.181	0.206		0.116	0.228	0.245



	GOM Range	NMI Range	MI Range		GOM Normals	NMI Normals	MI Normals
roll	0.479	0.646	0.720		0.541	0.624	0.644
pitch	0.530	0.721	0.686		0.438	0.769	0.682
yaw	0.210	3.792	4.440		0.228	0.535	0.510

Table 6.4 – Average error between two aligned Ladybug cameras for different 3D features. All distances are in metres and angles are in degrees.

evaluated using these different features, with all other parameters kept the same and 10 scans aggregated for MI, NMI and GOM.

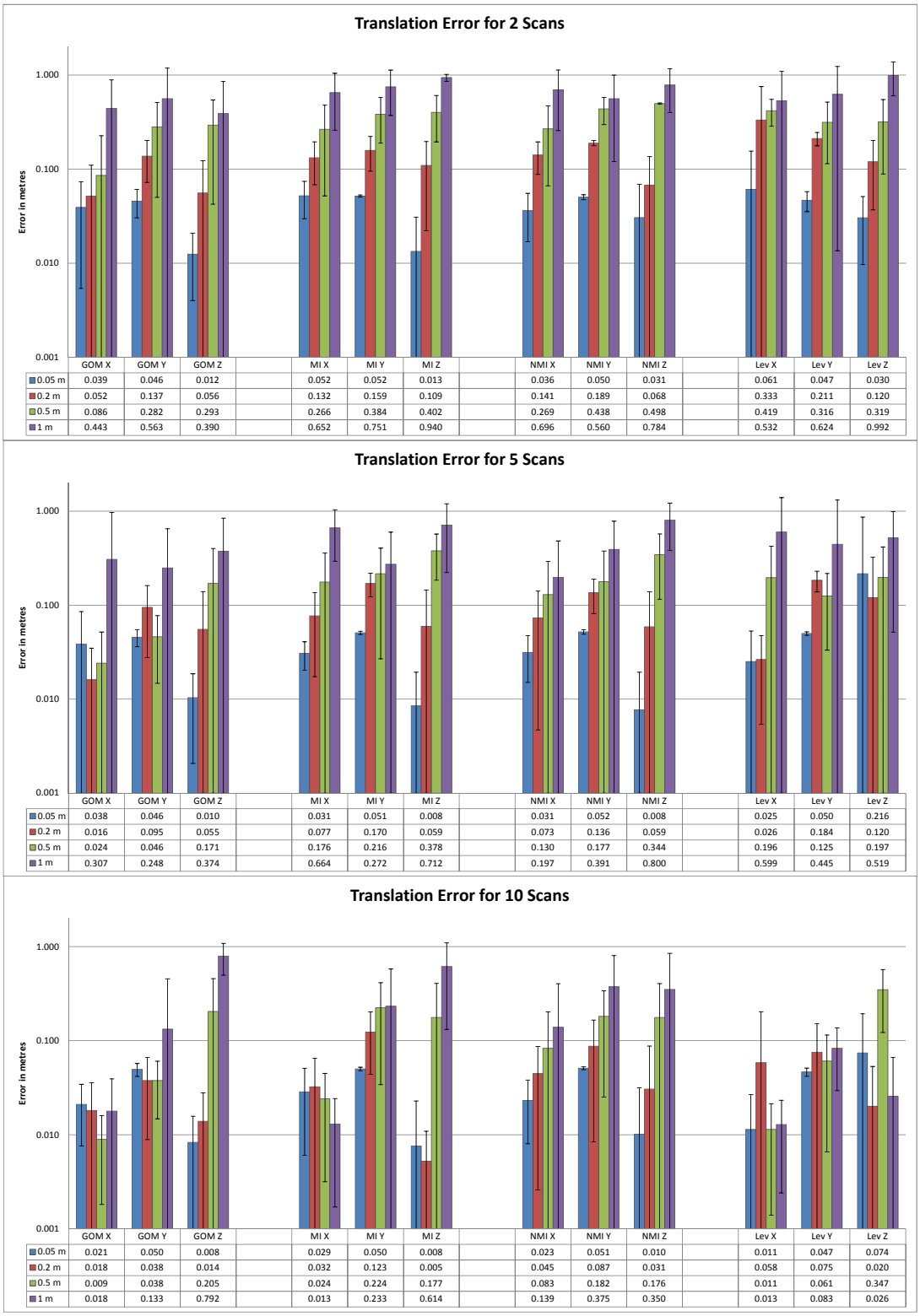
The results of these tests are presented in Table 6.4. For most of the results the metrics gave values that were slightly worse, but otherwise similar to what had been obtained using intensity information. The one exception to this was the yaw angle for MI and NMI when using range as a feature, which showed considerably larger error. While all of these results are worse than those obtained using the return intensity, they are accurate enough to provide a viable option in circumstances where the intensity information is not available.

6.3.7 Basin of Attraction

It is preferable to be able to use a local optimisation technique to find the maximum of a metric due to the substantially faster run time. However, as previously stated, the optimiser will only converge to the correct maximum if the initial guess as to the calibration is within its basin of attraction. The size of this region depends on the metric, the number of scans being aggregated and the scene being observed. In an effort to give an indication of the size of the basin for the different metrics, and how scan aggregation affects it, an experiment was designed.

The maximum of a metric was first found by the same method used to test the metrics' accuracy in the Ford dataset. A predetermined error was then added to one of the solution's parameters, and the experiment was re-run using this point as the initial guess. The difference between the initial solution found and the solution found after the error was added, was recorded. This was performed for X, Y and Z errors of 0.05, 0.2, 0.5 and 1.0 metres. Roll, pitch and yaw errors of 1, 5, 15 and 30 degrees were also tested. These results were repeated for aggregating 2, 5 and 10 scans. Each experiment was repeated 10 times with random scans.

The results of this experiment are presented in Tables 6.5 and 6.6. For the position of the sensor, optimisation significantly reduced the initial error in almost all of the cases. This implies that even when offsets as large as 1m are present, all four metrics provide an indication of the direction of the correct alignment. However, for low numbers of scans or large offsets, while the optimiser reduced the error it still converged to solutions a significant distance from the previously located maximum. When using 2 scans, all X and Y positions showed large error and the Z position required a starting location only 0.05m from the maximum to converge. The accuracy of all of the solutions increased significantly for 5 scans, and again for 10 scans. This meant, in the 10 scans case both the GOM metric and MI metric provided accurate results for all positions with starting offsets of 0.2m or less. They also gave accurate results in X for all offsets tested. The Levinson method appeared to give the least consistent results. For example, when 10 scans were used, it generally gave the most accurate X positions. However, occasionally it converged to an incorrect location far from the

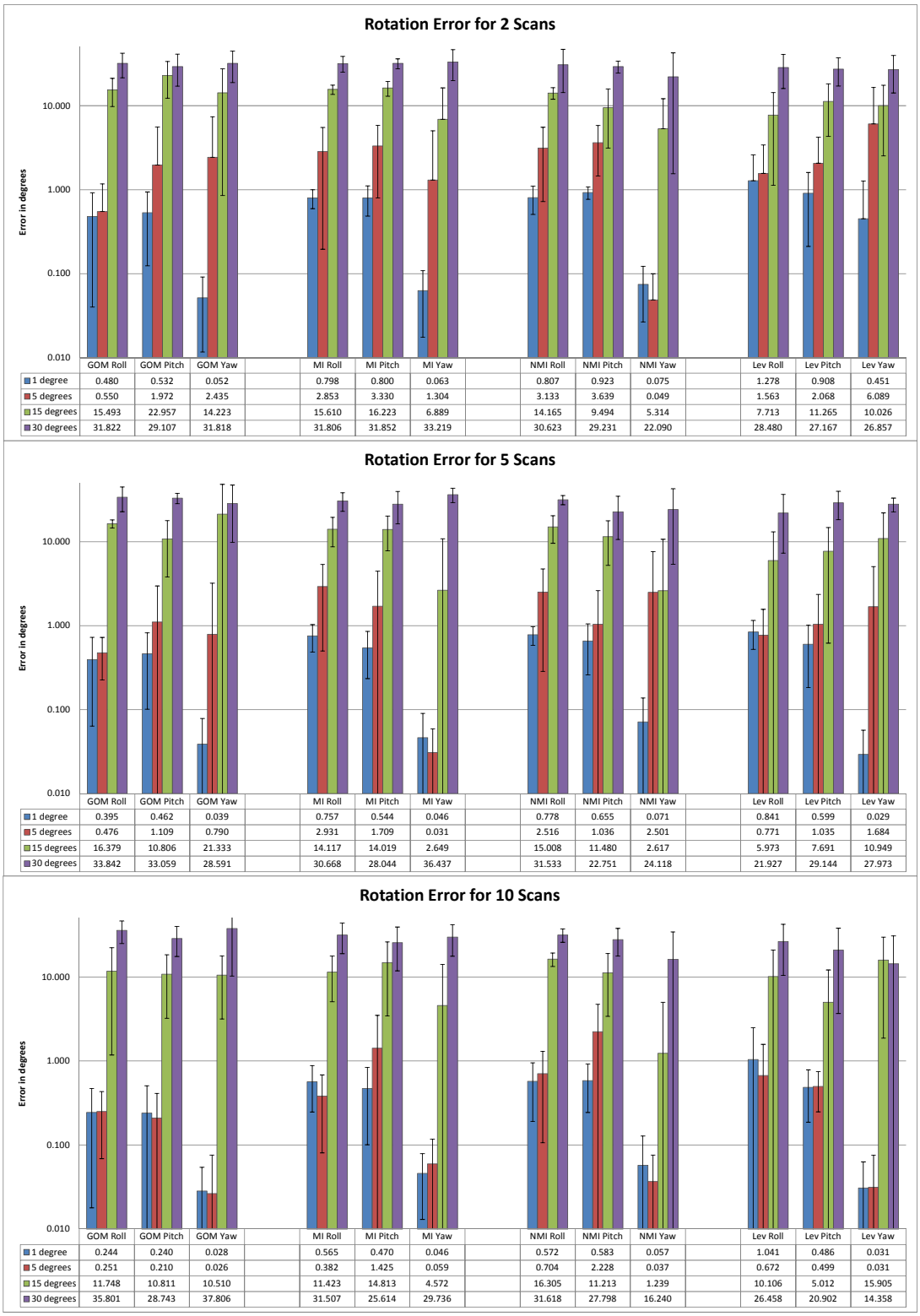


	GOM X	GOM Y	GOM Z	MI X	MI Y	MI Z	NMI X	NMI Y	NMI Z	Lev X	Lev Y	Lev Z
0.05 m	0.039	0.046	0.012	0.052	0.052	0.013	0.036	0.050	0.031	0.061	0.047	0.030
0.2 m	0.052	0.137	0.056	0.132	0.159	0.109	0.141	0.189	0.068	0.333	0.211	0.120
0.5 m	0.086	0.282	0.293	0.266	0.384	0.402	0.269	0.438	0.498	0.419	0.316	0.319
1 m	0.443	0.563	0.390	0.652	0.751	0.940	0.696	0.560	0.784	0.532	0.624	0.992

	GOM X	GOM Y	GOM Z	MI X	MI Y	MI Z	NMI X	NMI Y	NMI Z	Lev X	Lev Y	Lev Z
0.05 m	0.038	0.046	0.010	0.031	0.051	0.008	0.031	0.052	0.008	0.025	0.050	0.216
0.2 m	0.016	0.095	0.055	0.077	0.170	0.059	0.073	0.136	0.059	0.026	0.184	0.120
0.5 m	0.024	0.046	0.171	0.176	0.216	0.378	0.130	0.177	0.344	0.196	0.125	0.197
1 m	0.307	0.248	0.374	0.664	0.272	0.712	0.197	0.391	0.800	0.599	0.445	0.519

	GOM X	GOM Y	GOM Z	MI X	MI Y	MI Z	NMI X	NMI Y	NMI Z	Lev X	Lev Y	Lev Z
0.05 m	0.021	0.050	0.008	0.029	0.050	0.008	0.023	0.051	0.010	0.011	0.047	0.074
0.2 m	0.018	0.038	0.014	0.032	0.123	0.005	0.045	0.087	0.031	0.058	0.075	0.020
0.5 m	0.009	0.038	0.205	0.024	0.224	0.177	0.083	0.182	0.176	0.011	0.061	0.347
1 m	0.018	0.133	0.792	0.013	0.233	0.614	0.139	0.375	0.350	0.013	0.083	0.026

Table 6.5 – Mean translation error in optimisation for different levels of initial offset. All distances in metres. Note the chart axis uses a log scale.



	GOM Roll	GOM Pitch	GOM Yaw	MI Roll	MI Pitch	MI Yaw	NMI Roll	NMI Pitch	NMI Yaw	Lev Roll	Lev Pitch	Lev Yaw
1 degree	0.480	0.532	0.052	0.798	0.800	0.063	0.807	0.923	0.075	1.278	0.908	0.451
5 degrees	0.550	1.972	2.435	2.853	3.330	1.304	3.133	3.639	0.049	1.563	2.068	6.089
15 degrees	15.493	22.957	14.223	15.610	16.223	6.889	14.165	9.494	5.314	7.713	11.265	10.026
30 degrees	31.822	29.107	31.818	31.806	31.852	33.219	30.623	29.231	22.090	28.480	27.167	26.857

	GOM Roll	GOM Pitch	GOM Yaw	MI Roll	MI Pitch	MI Yaw	NMI Roll	NMI Pitch	NMI Yaw	Lev Roll	Lev Pitch	Lev Yaw
1 degree	0.395	0.462	0.039	0.757	0.544	0.046	0.778	0.655	0.071	0.841	0.599	0.029
5 degrees	0.476	1.109	0.790	2.931	1.709	0.031	2.516	1.036	2.501	0.771	1.035	1.684
15 degrees	16.379	10.806	21.333	14.117	14.019	2.649	15.008	11.480	2.617	5.973	7.691	10.949
30 degrees	33.842	33.059	28.591	30.668	28.044	36.437	31.533	22.751	24.118	21.927	29.144	27.973

	GOM Roll	GOM Pitch	GOM Yaw	MI Roll	MI Pitch	MI Yaw	NMI Roll	NMI Pitch	NMI Yaw	Lev Roll	Lev Pitch	Lev Yaw
1 degree	0.244	0.240	0.028	0.565	0.470	0.046	0.572	0.583	0.057	1.041	0.486	0.031
5 degrees	0.251	0.210	0.026	0.382	1.425	0.059	0.704	2.228	0.037	0.672	0.499	0.031
15 degrees	11.748	10.811	10.510	11.423	14.813	4.572	16.305	11.213	1.239	10.106	5.012	15.905
30 degrees	35.801	28.743	37.806	31.507	25.614	29.736	31.618	27.798	16.240	26.458	20.902	14.358

Table 6.6 – Mean rotation error in optimisation for different levels of initial offset. All angles in degrees. Note the chart axis uses a log scale.

initial point giving the larger error for the 0.2 m starting offset. Overall, the basin of attraction for GOM and MI appeared to be similar in size, with the Levinson method and NMI's being slightly larger.

In rotation, starting offsets of 15 and 30 degrees often led to errors larger than before optimisation. This implies that at these angles from the true rotation, the metrics could not provide any strong indication as to the direction of the previously found maximum, and therefore the metrics often converged to an incorrect local maximum in a random direction. While scan aggregation reduced this issue for the 15 degree offset, it was present in all the 30 degree offset results. The 1 degree offset did not significantly impede any of the metrics from obtaining an accurate result. However using 10 scans, only the GOM solution obtained an accurate result for a 5 degree offset for roll, pitch and yaw. For NMI and MI, significant error was always present in the pitch, while the Levinson method generally had significant pitch and yaw error.

From these results, it can be concluded that all four metrics have a similar basin of attraction, although GOM appeared to handle a slightly wider range of angles than the other methods. In all cases, scan aggregation noticeably improved the results. These results would suggest that for 10 scans on the Ford dataset, a metric must have an initial guess within at least 5 degrees and 0.2m of the correct solution to obtain reliable calibration results.

6.4 Motion-Based Metrics

6.4.1 Finding Timing Offset

We began the testing of motion-based metrics by looking at the method described in Section 5.2.1, and its ability to compensate for timing offset in sensor data. In this experiment the approach was used to correct for timing offset in the cameras of the KITTI, Ford and Shrimp dataset. Only the cameras were made use of due to the accurate ground truth available between these sensors. In the Ford and Shrimp dataset, a Ladybug spherical camera, is used. This camera system uses a global

shutter where all the cameras are triggered simultaneously. Similarly the KITTI dataset uses a system that allows for near-simultaneous triggering of the camera shutters. If any other sensors were to be incorporated we would be forced to rely on the time stamping of the system's which is of unknown quality in the KITTI and Ford dataset, and known to result in a lag as large as 60 ms for the Shrimp dataset.

To test the system, random contiguous sections of the drives were taken ranging from 10 to 200 seconds in length in 10 second increments; the cameras then had a random timing offset applied to them. The calibration was run and the mean absolute error in the timing found. Three experiments were run; in the first experiment a random timing offset between -0.1 seconds and 0.1 seconds was first applied to each of the cameras. In the second experiment up to 1 second of random offset was applied; in the final experiment 5 seconds of offset was used. Each experiment was performed 100 times. The results are shown in Figures 6.10, 6.11 and 6.12.

For all three datasets, several common trends appear. For the shortest time period tested (10 seconds) the results are generally quite poor. In the case of 5 seconds of random offset the results are comparable to random guessing. This is to be expected as the method only operates on the overlapping regions of data and for short time intervals there is a substantial chance that the vehicle will not exhibit any significant rotational cues from which to estimate the offset. The accuracy of the method rapidly improves over the next few timesteps. After this initial period of improvement, while the robustness of the method continues to improve (seen through the reduction in outlier points) the mean accuracy of the method does not change significantly.

The maximum error of the outliers tends to increase with the length of information used. The reason for this is that the maximum possible error a frame can be mismatched by is the length of the dataset used.

For 200 seconds of data and 1 second of initial offset the KITTI dataset gives a final median error of roughly 6 ms with a worst case error of 40 ms. The Shrimp dataset gives a median error of 2 ms with a worst case error of 9 ms and the Ford dataset has a median of 70 ms and worst case of 370 ms. Given that the sensors used give readings every 100 ms for the KITTI dataset and 120 ms for the other two datasets,

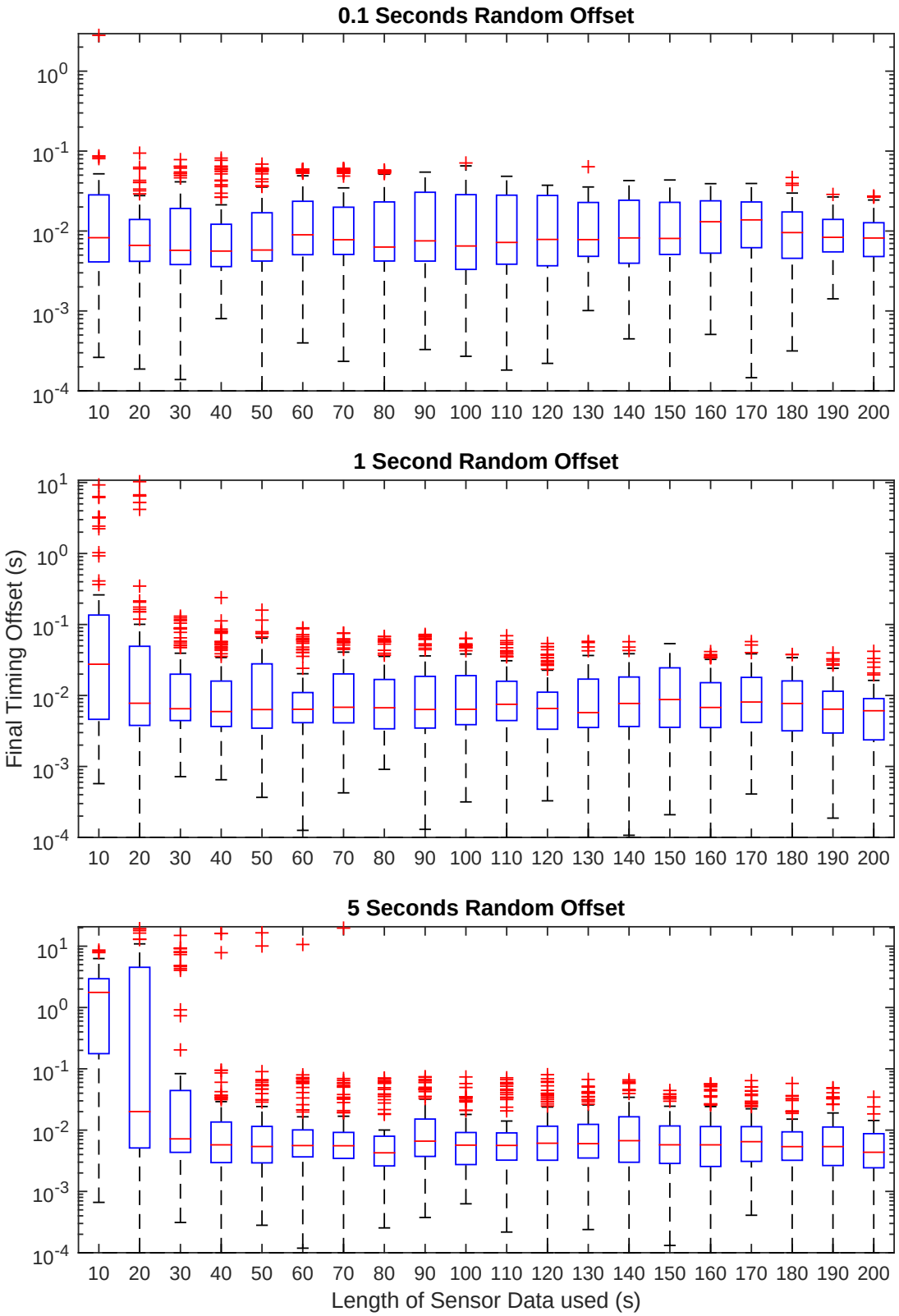


Figure 6.10 – Error in estimated sensor timing offset in the KITTI dataset.

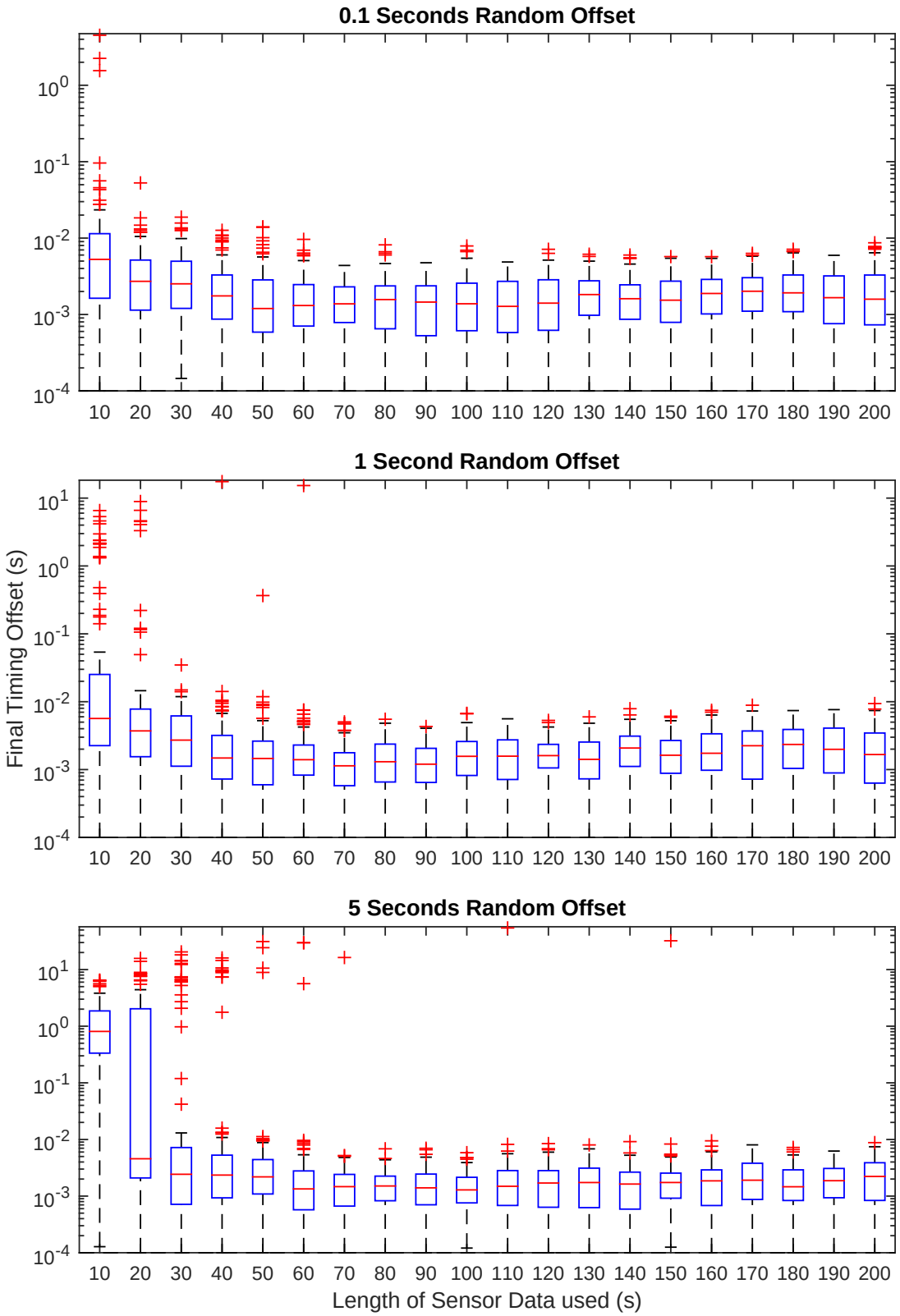


Figure 6.11 – Error in estimated sensor timing offset in the Shrimp dataset.

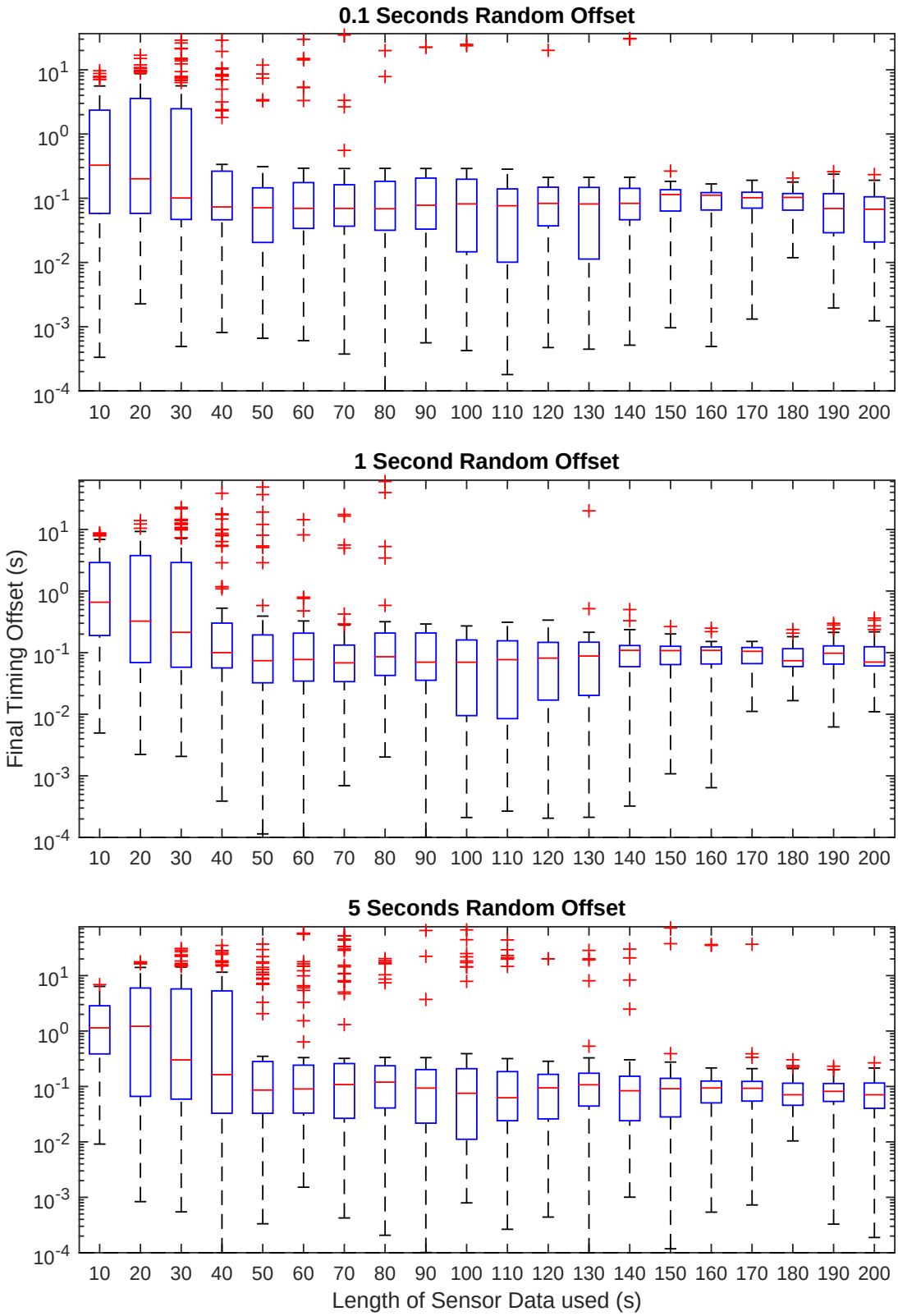


Figure 6.12 – Error in estimated sensor timing offset in the Ford dataset.

Dataset	Median Rotation Speed (rad/s)
Kitti	0.0572
Shrimp	0.0649
Ford	0.0118

Table 6.7 – Median rotational speed for each dataset.

this level of error is acceptable for our application as in the vast majority of cases the accuracy is sufficient to correctly provide the closest matching frames between two sensors.

The reason for the large difference in accuracy of the method on the three datasets however bears further examination. Part of the reason for this difference is the different vehicles, terrain and behaviour during the data collection. In the KITTI dataset a car is driven through a series of winding side streets in a small city; the Shrimp dataset has sensors mounted on a tall mast on a Segway vehicle driving back and forth over a grassy park. Finally, the Ford dataset drives a large Ford pickup truck around a single loop of a city block, travelling on wide main streets and stopping for traffic lights at several points.

From the perspective of timing calibration we are interested in the affect these differences have on the rotational speed of the sensors. Figure 6.13 shows the rotational speed of the GPS/INS system for each of the test systems. While the Shrimp dataset does not undergo any tight turns that the other systems experience, moving over unpaved terrain results in most timesteps experiencing a significant rotation. The Ford dataset, on the other hand, only experiences 8 significant turns in its six minute run, and it also remains stationary for over a minute waiting at a set of lights. The median rotational speed for each dataset is shown in Table 6.7. The low median rotational speed exhibited by the Ford dataset is the main reason for its poor performance in comparison to the other two datasets, as the large sections of little rotation result in insufficient features for an accurate alignment.

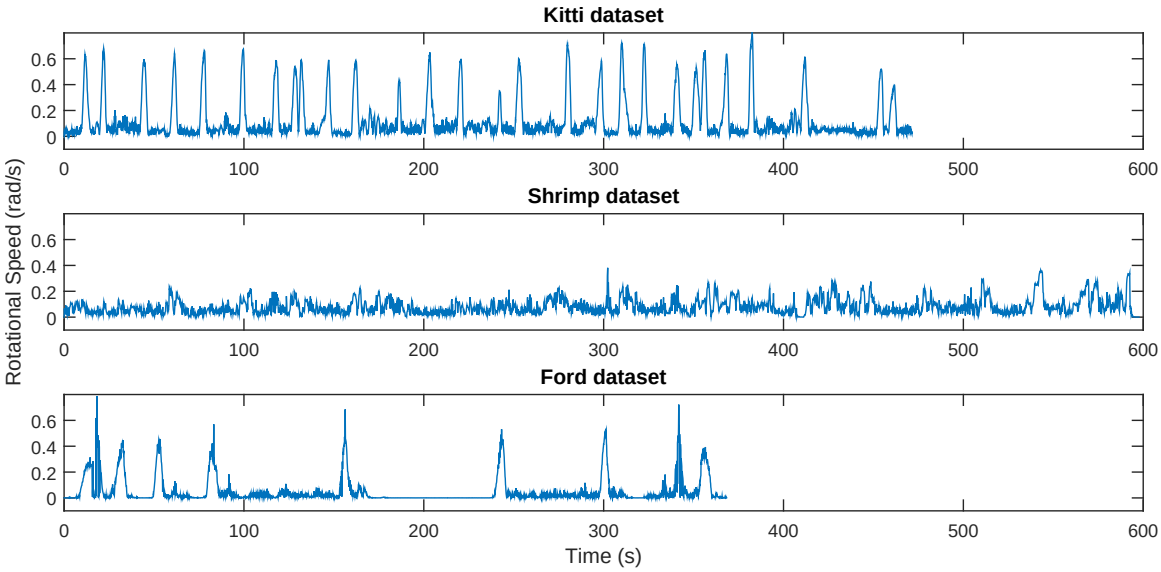


Figure 6.13 – Rotational speed experienced by the GPS/INS sensor during each dataset.

6.4.2 Aligning Two Sensors

To test the ability of the motion-based approach described in Chapter 5 to align two sensors with no overlap in their field of view, two experiments were performed. In these experiments the calibration between a GPS/INS system and a Velodyne lidar was found. In the first experiment the calibration was performed on the KITTI dataset. In the second experiment the Shrimp dataset was used. The results were also compared with a least squares approach that does not make use of the reading's variance estimates. In the experiment a set of continuous sensor readings is selected at random from the dataset and the extrinsic calibration between them found. This is compared to the known ground truth provided with the dataset. The length of information was set between 10 seconds and 300 seconds in 10 second increments. For each time period the experiment was repeated 500 times with the mean reported. While our method calculates the rotation in angle-axis format to allow for intuitive understanding, we convert this to Euler angles when displaying the results. The associated estimated standard deviation is also converted using a simple Monte-Carlo method.

Figure 6.14 shows the absolute error of the calibration. For all readings our calibra-

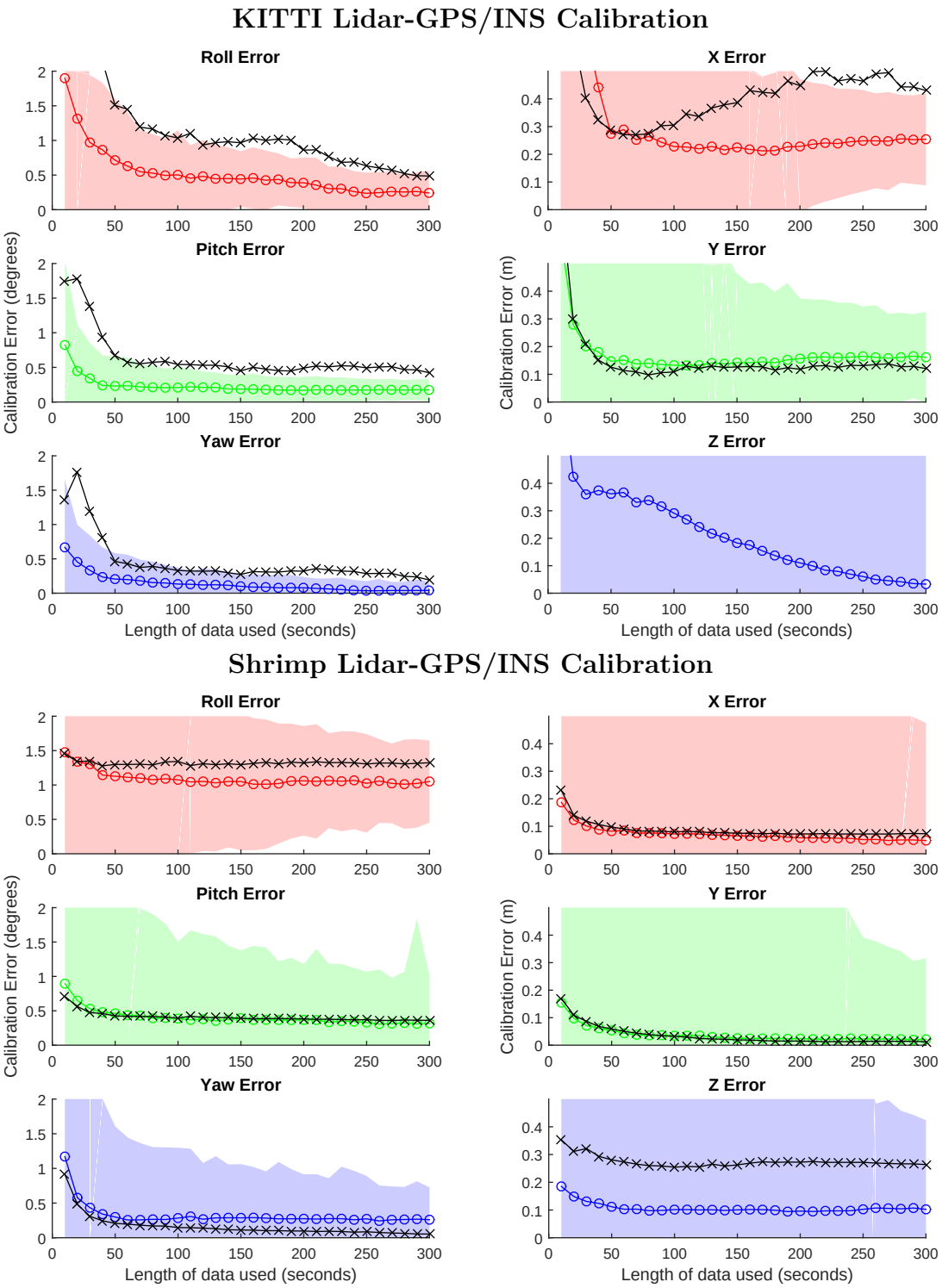


Figure 6.14 – The error in rotation in degrees and translation in metres for varying numbers of sensor readings. The black line shows the least squares result, while the coloured line gives the result of our approach. The shaded region gives one standard deviation of the estimated uncertainty provided by our approach.

tion significantly improves as more readings are used for the first few hundred scans, before slowly tapering off. In rotation, yaw was the most accurately estimated. This was to be expected as the motion of the vehicle is roughly planar giving less motion from which roll and pitch can be estimated. For large numbers of scans the method estimated the rotation in the KITTI dataset to within 0.5 degrees of error and in the Shrimp dataset to within 2 degrees of error. Our method outperformed the least squares method on the KITTI dataset while the least squares method gave similar results on the Shrimp dataset. The reason for the smaller difference in performance on the Shrimp platform was probably due to all the driving in this dataset occurring within a single small courtyard. This meant that there was less variation in the accuracy of the lidar and GPS readings during the calibration then in the KITTI dataset. Because of this, the least squares method not considering the sensors' uncertainty had less of an impact on the results. However, even in cases where our method does not outperform the least squares approach it offers the advantage of providing an estimate of the uncertainty in its values.

The calibration for the translation was poorer than the rotation. This is due to the reliance on the rotation calculation and the sensors tending to give noisier translation estimates. For translation our method significantly outperformed the least squares method. The least squares method, in many instances, also began to perform more poorly as the data used increased. The most likely explanation for this decrease in performance is that the more data is used for the calculation, the higher the probability an outlier will be present in the data. The largest difference in performance can be seen when estimating the Z offset. This difference is due to the roughly planar motion that makes the Z axis the most sensitive parameter to noise and outliers, which the simple least squares method fails to account for.

The accuracy of the predicted variance of the result is more difficult to assess than that of the calibration. However all of the errors were within a range of around 1.5 standard deviations from the actual error. Viewing the data suggests that the estimated translation variance may be slightly conservative in its estimates. Overall the method gives a consistent indication of the estimation's accuracy.

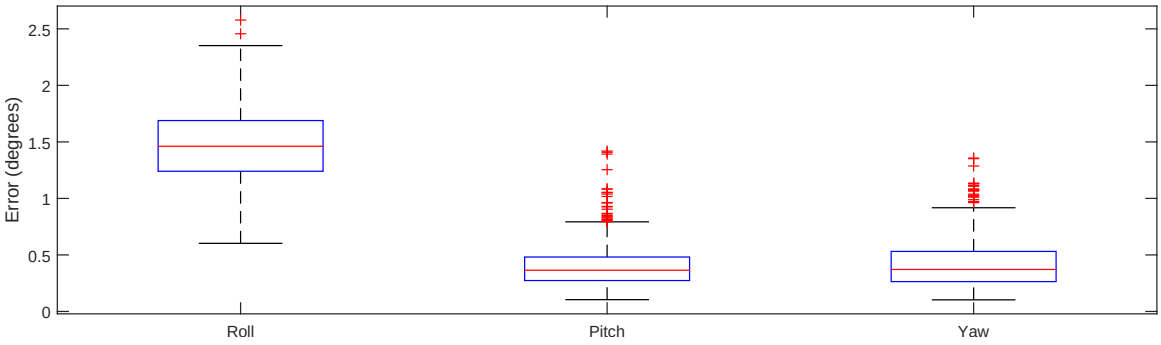


Figure 6.15 – The absolute error in rotation, in degrees, when calibrating 5 of Shrimp's cameras.

6.4.3 Calibrating Panoramic Camera Systems

To give a near-complete image of the surroundings many platforms make use of panoramic camera systems. These systems fuse multiple images from cameras facing in different directions into a single image. The cameras in these systems often have little to no overlap in their views, making their calibration challenging. However, our motion-based approach can still estimate calibration parameters for this type of system.

To demonstrate this an experiment was run calibrating the five horizontally facing cameras of the Shrimp platform's Ladybug camera system. As only monocular camera matching is performed, no sense of scale is present and so only the rotational offsets can be estimated. This limitation does not significantly impede the use of multiple cameras in forming a panoramic image, however as in most practical systems, the cameras are placed in close proximity. This means that assuming no translation between the cameras is usually valid. Even when the translation is known, panoramic camera systems often make no use of it as the translation cannot be used to correct the image panorama without some knowledge of the structure of the scene [81].

For the calibration process 30 seconds of continuous data was used and the process was repeated 500 times. While in this case the cameras had a small amount of overlap that could be used to refine the alignment, this was not utilised in this experiment. The error in this calibration was found using the ground truth provided by the manufacturer to find the mean error between any combination of two cameras. The results

Figure 6.16 – Spherical panoramic image created from the 5 individual cameras. The top image was created using the manufacturer's values and the bottom with the mean results from the experiment.

of this experiment are shown in Figure 6.15.

As in previous experiments, roll was the least accurately estimated due to the vehicle undergoing roughly planar motion, limiting the observability of this parameter. Overall the results gave a median error in roll of roughly 1.5 degrees and an error in yaw and pitch of around 0.5 degrees. Once the calibration has been performed the results can be combined with the sensor intrinsics to create a spherical panoramic image.

While the accuracy of the above calibration could be improved by utilising more data or making use of point matching in the small regions of overlap, this would not significantly improve the appearance of the generated panoramic image. This is because the inaccuracies in the lens distortion model and intrinsics have a significant impact on the quality of the image. This is shown qualitatively in Figure 6.16 where both the manufacturer's ground truth and generated calibration values are used to fuse the Ladybug's images. For a human viewing these images it is challenging to discern with any certainty which calibration is more accurate.

6.4.4 Simultaneous Calibration of Multiple Sensors

An experiment was conducted to evaluate the effect that simultaneously calibrating all sensors has on the results. This was done using the KITTI dataset and aligning its Velodyne, GPS/INS unit and two of its cameras. The experiment was first performed using 200 seconds of data to calibrate the sensor's and combining all of the readings. In a second experiment the sensors were again aligned, however this time only the error with respect to the Velodyne sensor was optimised. The experiment was repeated 100 times and the mean error in rotation and translation was found and shown in Figure 6.17. From the figure it can be seen that the error with respect to the Velodyne sensor was similar for both optimisations. The simultaneous calibration generally resulted in slightly more accurate rotation estimates, however the translation estimates are roughly the same or marginally worse. The difference comes when the error between the other sensors is considered. When the error with respect to one of the cameras is considered, it can be seen that a significant reduction in error is present. This is to be expected as simultaneously considering the error between all sensors will result in a calibration that, while not necessarily more accurate for a single sensor pair, will provide the most accurate solution for the system as a whole.

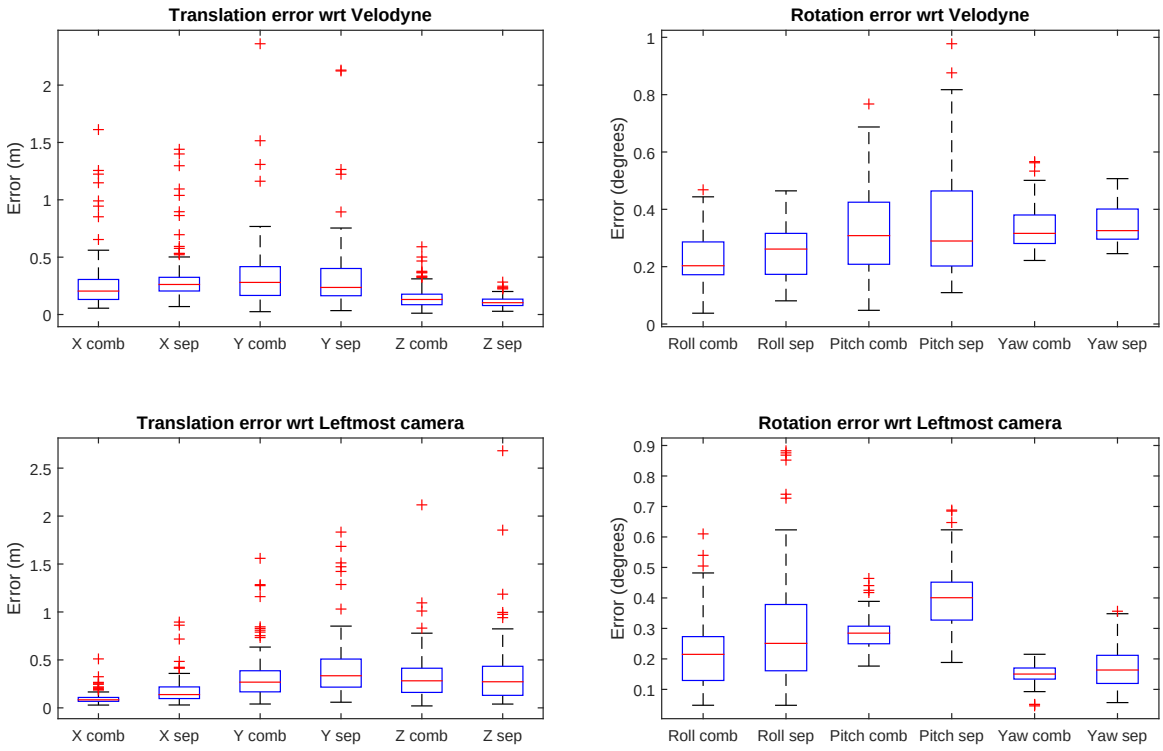


Figure 6.17 – Box plot of the error in rotation, in degrees, and translation, in metres, for combining sensor readings and performing separate optimisations. In the separate optimisations only the error with respect to the Velodyne was considered.

6.4.5 Constraining the Search Space of Appearance-Based Metrics

To evaluate the use of motion-based calibration for constraining the search space of appearance metrics, an experiment was performed. Initially 100 seconds of data from the KITTI dataset was used to align the system's Velodyne with its leftmost camera. From this initial estimate appearance-based alignment of the sensors is then performed. The search space for the problem is defined so that all rotations are considered and the magnitude of the X, Y and Z offset of the two sensors is limited to be under 1 metre. Note that, due to the offset between the sensors this results in a maximum possible error in the calibration of 1.29m in X, 1.01m in Y and 1.06m in Z.

From this starting point a CMA-ES optimiser is first run considering the entire search space. For this initial experiment we follow the advice of Hansen [22] that when no other method of determining the starting variance is present, it should be set so the correct solution lies within 2σ of the initial estimate. Thus the initial multivariate Gaussian is set to have a σ that is 50% of the extent of our search space.

It should also be noted that in the optimisation the first point evaluated is the solution provided by the motion stage. This means that any parameters output by the CMA-ES optimisation will give equal or superior metric values to this initial parameter.

Once this optimisation has been performed the approach is rerun, this time making use of the variance estimate provided by the motion stage and using this information to initialise CMA-ES's Gaussian distribution. In the appearance stage 25 scan-image pairs were used; these were randomly selected from the available data. The experiment was repeated 50 times and the GOM, NMI, Levinson and IM metric were evaluated. The results of this experiment are presented in Figure 6.18 and Table 6.8; a depiction of some of the resulting calibrations is shown in Figure 6.19.

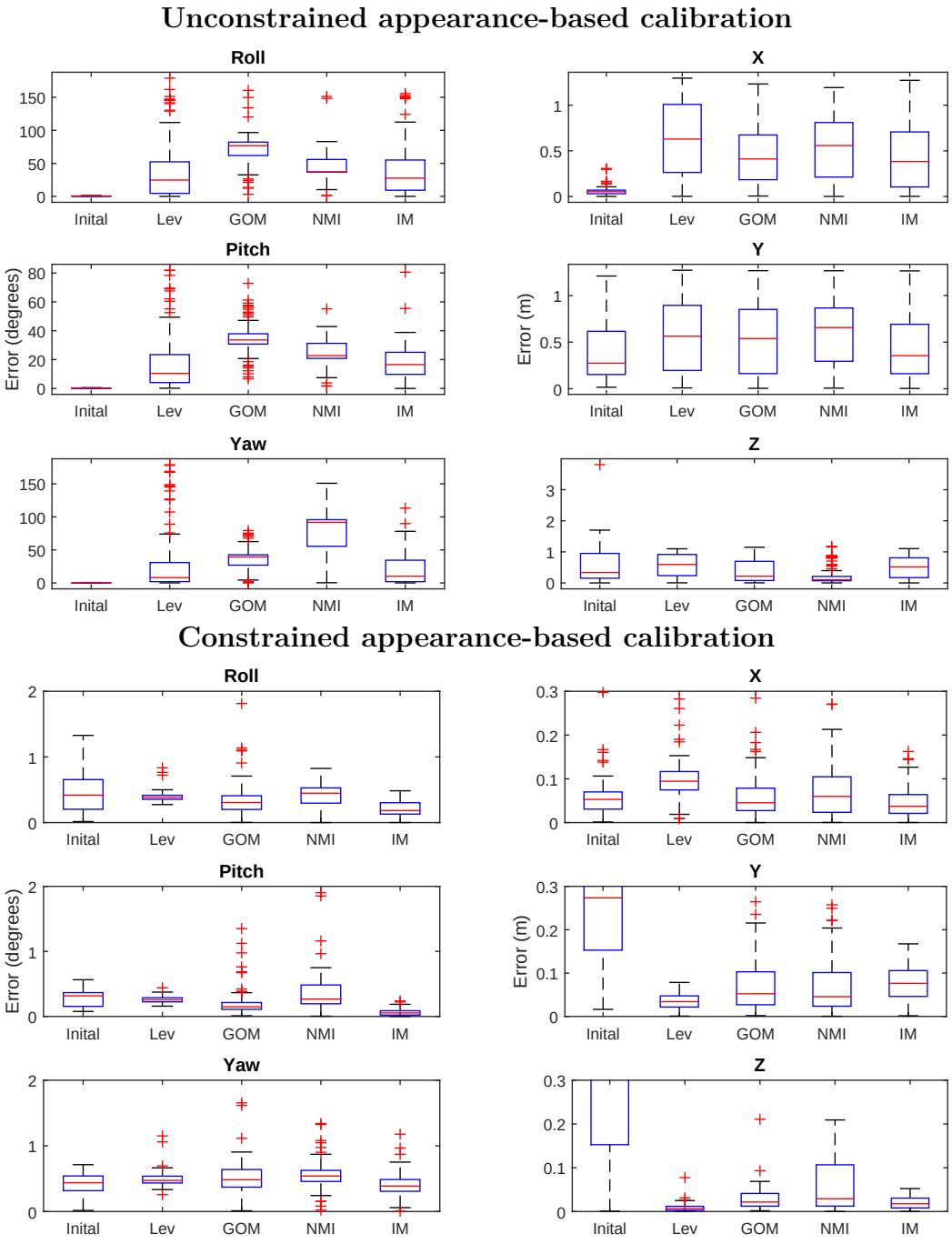


Figure 6.18 – Box plot of error in alignment for appearance-based metrics. The unconstrained results search the entire viable search space, whereas the constrained results made use of the variance given by the motion-based calibration. Note the axis of the constrained optimisation excludes several outliers; the number of these outliers can be found in Table 6.8. All rotations are in degrees and translations in metres.

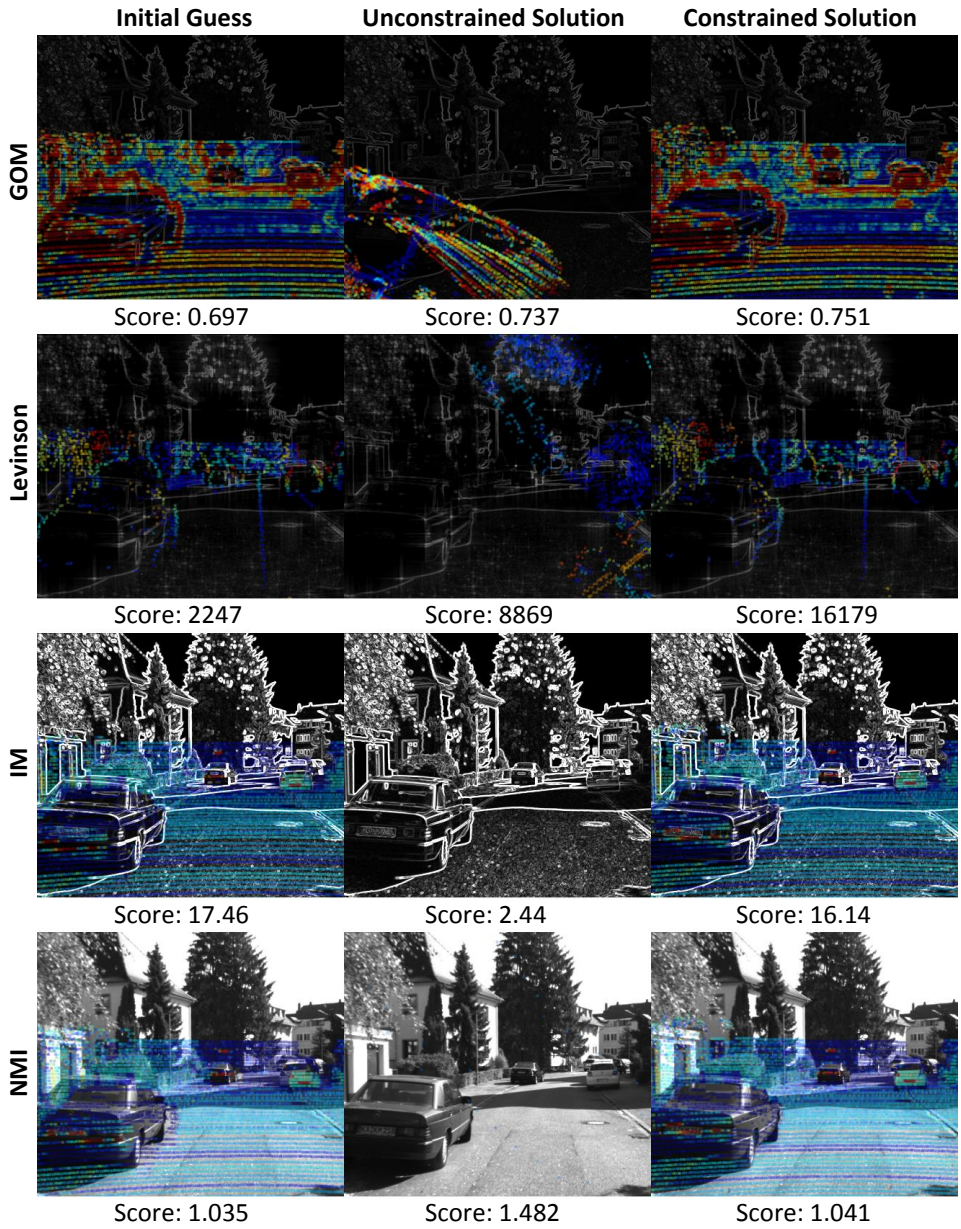


Figure 6.19 – Some typical results when the Velodyne and Camera in the KITTI dataset are aligned. Each image shows the Velodyne projected onto a camera's image after the preprocessing steps of each metric have been carried out. The initial guess, constrained solution and unconstrained solution, along with the metric scores, are shown. While the unconstrained solution improves the metric scores in all cases (GOM and IM scores are minimised, Levinson and NMI scores are maximised) the solution found has little relation to the correct parameters. By constraining the optimisation, only solutions that are likely under the motion-estimation step are evaluated, thereby improving the robustness and accuracy of the calibration.

	Initial	NMI	GOM	Levinson	IM
Full Search Space	42%	0%	0%	2%	10%
Constrained	42%	82%	88%	96%	100%

Table 6.8 – Percentage of solutions whose parameters are within 0.3m and 2 degree of the correct calibration.

For the case of the constrained optimisation, the Levinson method, GOM and the IM metric all typically improved the calibration of the system. The one exception to this was the yaw estimate which was already estimated by the motion-based calibration to a high degree of accuracy. For many of the metrics, in some instances the optimisation failed and the system converged to a result with significant error. The rate at which this occurred for the different metrics is shown in Table 6.8. In this experiment the constrained IM metric was the only method that did not suffer from these occasional outliers.

For the case where the appearance metrics were optimised over the entire search space, the results were exceptionally poor. In many cases metrics gave results similar or in some cases worse then random guessing. As Table 6.8 shows, NMI and GOM both failed to have a single result that was within what we considered an acceptable range of 2 degrees and 0.3m of error in its solution.

These findings show the limitations of appearance metrics in regards to initialisation, and the issues encountered when the metrics are applied without regard for them. There are two issues that combine to give these poor results. The first issue is the bias to sensor overlap that was discussed in Section 4.1. This can be seen in the unconstrained results for the IM and NMI metric in Figure 6.19. In both cases this issue causes a global optima in a location that does not correspond to the correct calibration parameters. The second issue is when the system is simply unable to locate the global optima. This issue occurs because the CMA-ES optimiser used has the number of points it evaluates at each step given by the heuristic $(4 + \text{floor}(3\log(N)))$, where N is the number of dimensions the data has (6 in this case). This means that at each iteration the function is defined by 9 samples. These samples are unable to capture the complex nature of the entire search space given by the appearance

	Roll	Pitch	Yaw	X	Y	Z
Full Search Space	90.00	45.00	90.00	0.50	0.50	0.50
Constrained	0.24	0.27	0.26	0.04	0.14	0.57

Table 6.9 – Mean standard deviation of the Gaussian used to initialise the optimisation. All angles in degrees and distances in metres

metrics, causing these poor results.

Instead of using the motion-based methods solution and estimated variance, another option to deal with the large search space problem would be to use more samples, either with CMA-ES or an alternative optimiser. This however presents the issue of the methods runtime. Depending on the metric used the evaluation of the appearance metric on one lidar-image pair requires, on average, between 2 and 6 ms. Taking the best-case option of 2 ms then to evaluate 25 images requires 50 ms. Using CMA-ES with 9 function evaluations per step, the method typically requires around 200 steps to converge, resulting in a run time of 90 seconds. Now we must examine how many function evaluations must be made to truly represent the search space without constraining it. If we were to sample the search space as densely as the solution that makes use of the variance from the motion-based estimation, we would require:

$$evaluations = 9 \prod_{i=1}^{6} \frac{\sigma_{full_i}}{\sigma_{con_i}} \tag{6.1}$$

Where σ_{full_i} is the standard deviation of dimension i of the full search space and σ_{con_i} is the standard deviation of dimension i of the motion-based solution. For this experiment the mean standard deviation of these search spaces is given by Table 6.9. This means in this instance Equation 6.1 would give $\approx 7,600,000,000$ function evaluations per step of the optimisation. Clearly this approach would be completely intractable to evaluate in any sort of timely manner.

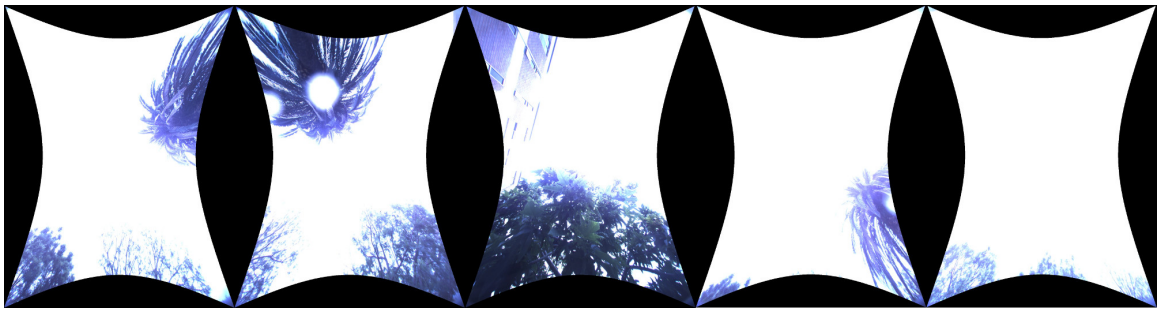


Figure 6.20 – Five images from the upward-facing camera (camera 6) of the Ladybug in the Shrimp dataset. Overexposure and few motion cues make this a challenging sensor to register.

6.4.6 Impact of Noisy Inputs

While the Ladybug system is composed of six cameras, the sixth camera points directly upwards. Due to this positioning, in many applications this camera will only give images of the sky and so is of limited use (The Ford Michigan dataset does not even provide it). However the Shrimp dataset was gathered in a quadrangle with some trees around the edge. This means that when the vehicle was near the edges of the area, the camera was able to see parts of trees and the upper floors of some buildings.

This camera provides a challenging test to evaluate our method's robustness against noisy observations. As was already mentioned, the most significant issue is that most of the captured images only have a small number of objects near the edges of the frame. A second factor that degrades the image quality is that during the capture of this dataset, the sun was almost directly overhead. In the Ladybug camera system the exposure time of all of the cameras is set to be equal. This means that when this exposure is set to an ideal value for the horizontal cameras, the vertical camera pointing at the sun has all of its images overexposed by a significant amount. These images also experience substantial lens flare. A series of example images from the camera is shown in Figure 6.20.

Due to the low quality of the images, the visual odometry gained from the camera is exceptionally poor, with many missing frames and noisy readings. Figure 6.21 shows

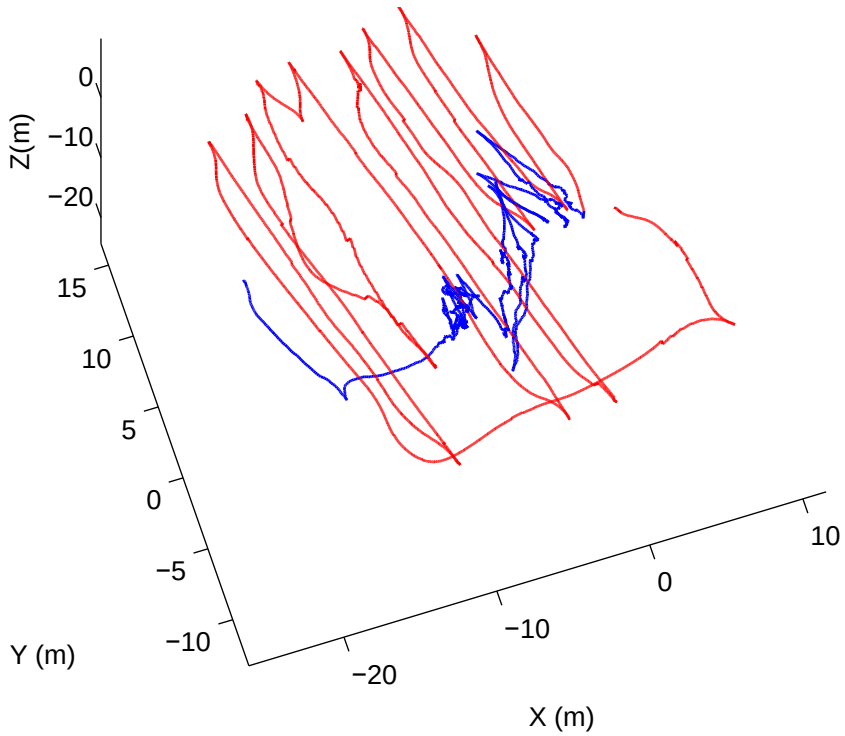


Figure 6.21 – The visual odometry result from the upwards-facing camera (blue) compared to the RTK GPS unit (red) on the Shrimp dataset.

the path given by our camera transformation calculation step when the GPS unit is used to provide absolute scale to its transformations.

To test the method's ability to handle this challenging sensor an experiment was run. A section of continuous data was randomly selected from the Shrimp dataset and used to find the rotational offset between Shrimp's front-facing and upward-facing camera. This experiment was repeated for trimmed means that rejected 0% (standard mean), 25% and 75% of the data, as well as data lengths of 20, 100 and 200 seconds. The process was repeated 100 times. No accurate ground truth for the Ladybug's position with respect to the other Shrimp sensors exist, so it was not included in the calibration. This limitation means that only monocular cameras were compared and so no estimate of the translation between the sensors can be calculated.

The results of the experiment are shown in Figure 6.22. In this experiment the rotational error rapidly decreases with the length of data used. However, using a mean where no data is trimmed results in poor calibration results for all runs. This is

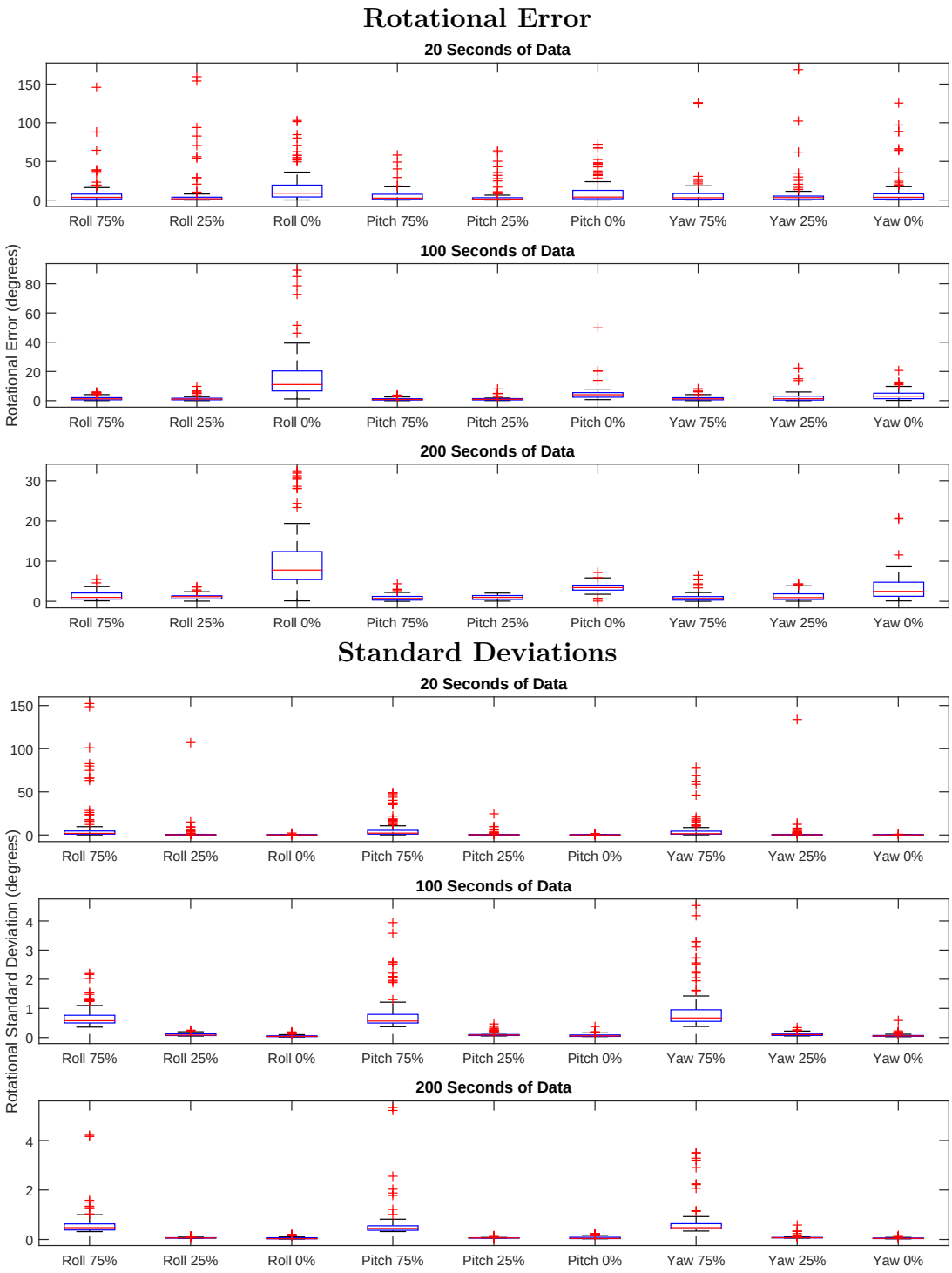


Figure 6.22 – Alignment of Shrimp's upward-facing camera for 20, 100 and 200 seconds of data. Trimmed means rejecting 0%, 25% and 75% have been applied. The estimated standard deviation is also provided. In this noisy dataset only the 75% trimmed means robustly prevents outliers from affecting its estimate and standard deviation.

almost certainly due to the large impact outliers have on this dataset. The trimmed means rejecting 75% and 25% give similar accuracy over the scans for this dataset. The estimated standard deviations, on the other hand, displayed very different results. For these sensors the 75% rejection trimmed means was the only measure that did not consistently underestimate the true error in the resulting calibration by a significant margin.

Two factors contribute to the increased σ of the 75% trimmed means. The first is that it utilises less data then the other two methods; however the impact of this is not on the scale of the differences witnessed in this dataset. The second is that any outliers that are included in the probability calculation will greatly increase the method's certainty in its results. This result would imply that, due to the extreme noise unique to the motion of this upward-facing camera, the simple approach we have utilised, of rejecting 25% of the worst fitting data, may be failing to remove all of the outlier points. Because of this, either increasing the trimmed means rejection ratio or another form of robust estimator is required in these extreme circumstances. The results of the experiment show that if a robust framework is used, the calibration pipeline still works even in very noisy conditions, though some care must be taken to ensure the noisy readings are correctly identified.

6.4.7 Full Alignment of Multiple Sensors on the KITTI Dataset

To test how the entire process performs when applied to the calibration of a system, all stages of the process—the timing estimation, motion estimation and refinement, were used to align four cameras and the Velodyne scanner in the KITTI dataset. 100 seconds of contiguous data was taken for the test and the IM metric was used for refining the alignment. A Gaussian with a σ $\frac{1}{500}$th of the image width was used for blurring the images in IM's image preprocessing stage. The experiment was repeated 100 times. The results of those runs are presented in Figure 6.23 and the mean error given in Table 6.10.

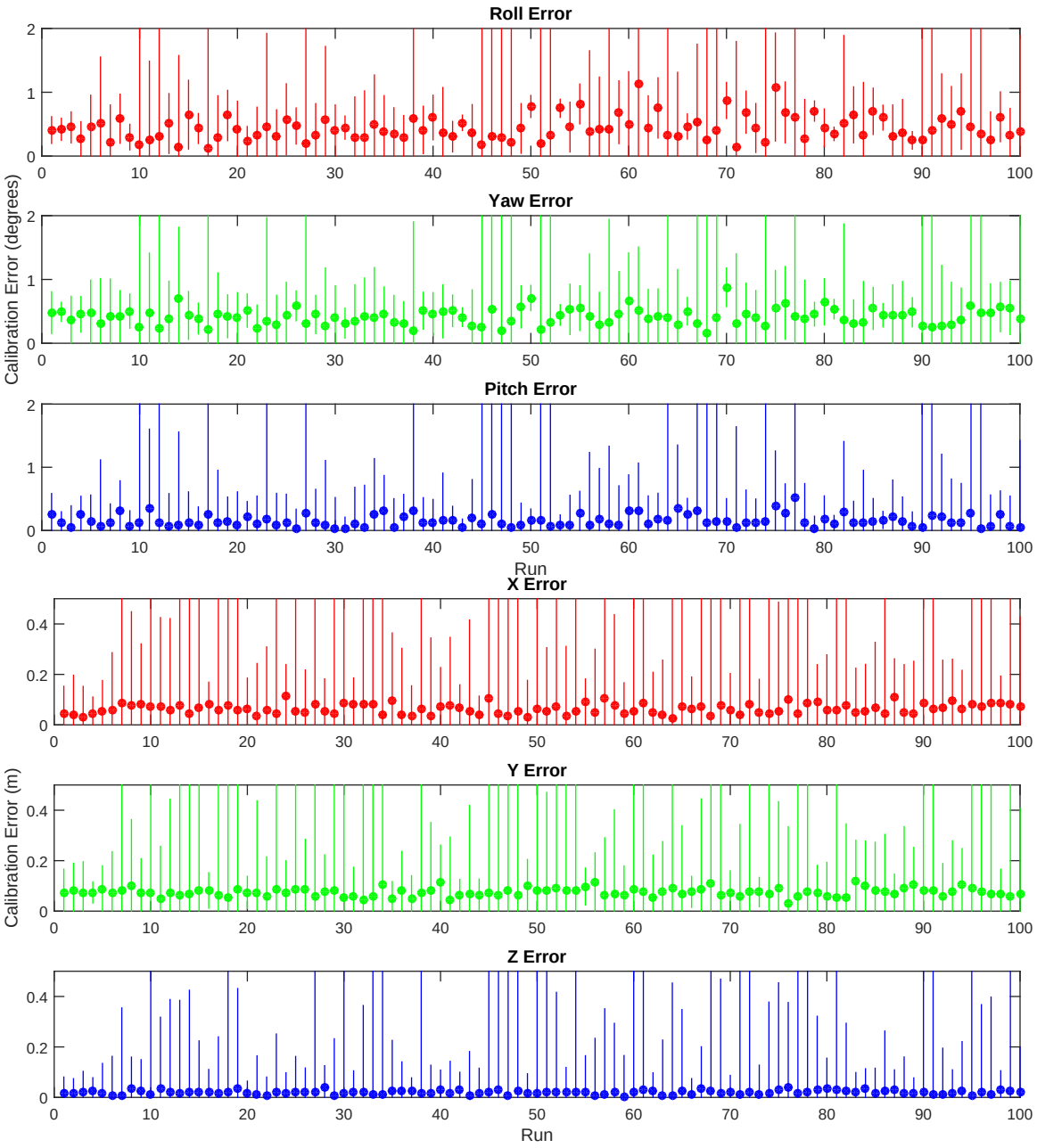


Figure 6.23 – Plot of the resulting error when the full calibration method is used to align the cameras and Velodyne lidar in the KITTI dataset. The mean error between the sensors is shown for all 100 runs with error bars giving the estimated standard deviation.

X	Y	Z	Roll	Pitch	Yaw
0.0633	0.0747	0.0202	0.4384	0.4152	0.1540

Table 6.10 – Mean error when the full method is used to calibrate the Velodyne and cameras on the KITTI dataset. Distances in metres and angles in degrees.

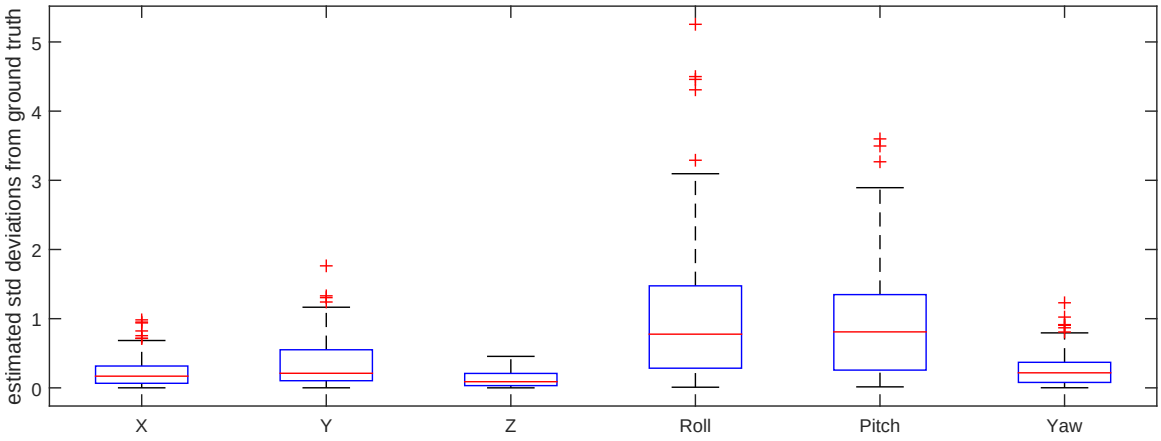


Figure 6.24 – Box plot of the number of estimated standard deviations the estimated solution lies from the ground-truth

This process gave an accurate calibration in almost all cases. To give an indication of the accuracy of the estimated variance our method returns, we analysed the number of standard deviations this estimate placed between our calibration and the ground truth. This information is shown in Figure 6.24.

If the parameters were Gaussian distributed with the estimated standard deviation, we would expect 50% of the data to lie within 0.67σ of the ground truth. From viewing the results it appears that for this experiment the X,Y,Z and yaw were slightly conservative in their estimation predicting greater uncertainty than is present. Roll and Pitch demonstrated the opposite behaviour slightly overestimating the confidence in the results. In general, all confidence estimates gave an estimated uncertainty that was sufficiently close to the true uncertainty to be of practical value in assessing the performance of the calibration.

6.5 Summary

In this chapter we have presented a thorough evaluation of the algorithms presented in previous sections of this thesis. Through experimentation using a rich number of datasets, we have demonstrated the accuracy achievable via the techniques presented in this thesis, as well as several state-of-the-art methods.

We first examined appearance-based metrics and compared the performance of the developed GOM metric to current approaches. Its performance in high resolution lidar-camera registration, image-image registration and mobile lidar-camera calibration was assessed, and shown to be comparable or superior to other state-of-the-art methods. We then performed experiments with methods for generating intensity information from 3D data and observed how this impacted the quality of the calibration obtained. Finally, we examined the accuracy of the initial guess required to calibrate a mobile system's sensors, via an experiment that tested the size of the basin of attraction of several methods.

After this we turned our focus to the motion-based metrics, first demonstrating that for systems undergoing sufficient motion, our methods could accurately recover the timing offsets between the sensors. The rotation and translation offset estimation methods accuracy was then assessed, before an experiment was performed demonstrating the improvement considering the offset between all sensors had on the resulting estimates. We then examined the performance of a system that combined both motion and appearance-based information in its calibration. This experiment showed, that through the combination of our motion-based approach and our IM metric, a robust and accurate calibration could be given. It was also shown that this solution was only possible when the full motion-based approach was utilised to guide the appearance-based refinement stage. We then examined the methods performance under significant noise. Finally, we implemented all of the stages of our approach (timing, rotation and translational offset estimation from motion, with appearance based refinement) in a single experiment. This experiment accurately calibrated the sensors of a vehicle, while also providing a believable indication of the uncertainty in its values.

Chapter 7

Conclusion

The calibration of multi-modal sensors is typically a time consuming and challenging process that must be performed by someone highly knowledgeable about the sensors' operating principles. As the application of systems with multiple cameras, lidars, GPS, INS and other sensors becomes more common, this calibration will become a stumbling block for non-expert users that need to operate these systems. This issue will also impact autonomous systems that need to operate without assistance for long periods of time. Because of this, this thesis has focused on the development of automatic extrinsic calibration techniques that make a minimal number of assumptions about the system they are calibrating. Initially the focus of our research was on high-resolution lidar and their fusion with camera systems. To this end the GOM metric was developed. It was then found the metric was also suited to 3D lidar-camera calibration for mobile vehicles. The calibration approach however is limited due to the challenging optimisation that is experienced by all metrics that only make use of the appearance of the surrounding environment.

To overcome these issues we looked to other cues of alignment and found that on ground vehicles, motion provides a rich source of calibration information. From this, a motion-based approach that could operate on any number of cameras, 3D-lidar and GPS/INS systems, was developed. The system was also able to correct for timing offsets between the sensors. It was found that this system can be combined with

the appearance-based approaches and assisted in reducing the difficulties of their optimisation by constraining the feasible search space within which a solution could lie. It was also found that by utilising both appearance and motion cues, a new metric (the Intensity-Motion metric) could be developed for use in calibrating 3D lidar-camera systems for mobile vehicles. Finally, throughout all of the motion-based calibration, careful consideration of the variance in each estimate was made to allow the final approach to give an estimate of the confidence in any calibration it obtained. A qualitative demonstration of the results obtainable via the methods developed are shown in Figure 7.1. Further images and videos demonstrating the procedure can also be found in [83].

7.1 Contributions

The specific contributions of this thesis are as follows:

- The development of a new multi-modal appearance-based metric, the Gradient Orientation Measure (GOM). This metric operates by aligning the gradients present in sensor modalities. The metric was designed for use with aligning single, high-resolution lidar scans with images of the same scene without any markers, a task most existing metrics are unable to perform. The metric has also been shown to perform well on other tasks such as IR-RGB alignment and the calibration of low-resolution lidar with a camera via the aggregation of multiple frames. Further details of the metric are presented in Section 3.4.

- The examination of the issues surrounding appearance-based metrics, including the difficulty of their optimisation and accuracy estimation. The details of this are outlined in Section 4.1.

- The extension of hand-eye calibration techniques into a framework that allows for the estimation of the timing, rotational and translational offset between any number of sensors. The method operates on observations provided by a mobile

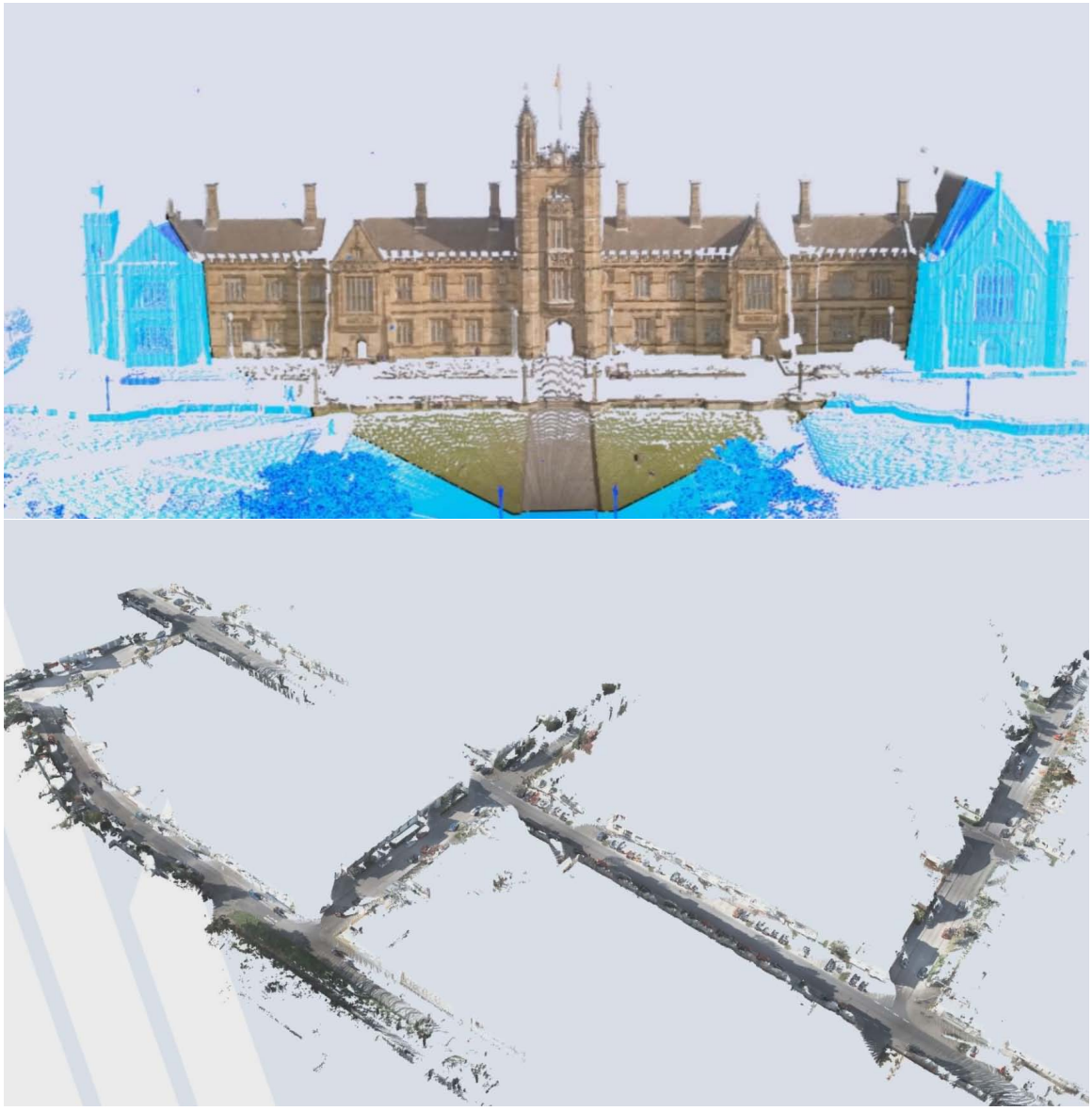


Figure 7.1 – Top: The University of Sydney's Great Hall. A camera image was projected onto a high-resolution lidar scan after alignment with GOM. Bottom: A section of the KITTI dataset where the Velodyne has first been coloured using the leftmost camera. The calibration was performed using our motion-based approach and IM metric. The Velodyne's motion was then used to project all the 3D points onto a single image.

platform moving through an arbitrary environment and does not require any calibration aids or even overlap in the sensors field-of-view. The technique also considers the variance of the sensor readings at each stage and reasons about the given information in a probabilistic manner. The details of this approach are given in Section 5.2.

- The estimation of the uncertainty present in the calibration results found. These estimates are generated utilising statistical techniques that are presented in Section 4.2.

- The combination of the strengths of both appearance-based and motion-based calibration. This is done by using the motion-based calibration to remove the requirement for an accurate initial guess to the calibration, that is typically a feature of appearance-based metrics. We are unaware of any other approach that combines both types of sensor information. The details of this combination are given in Section 5.7.

- The development of a second new multi-modal metric, the Intensity Motion (IM) metric for aligning lidar with cameras. This metric is designed for use with mobile vehicle-based systems and utilises both appearance- and motion characteristics in its refinement. The use of both of these calibration cues allows for an indication of the sensors' alignment while only making mono-modal comparisons. This is an advantage over previous methods as mono-modal matching will typically be more robust then multi-modal matching. This robustness is due to the simpler relationship between mono-modal sensor readings. The details of this metric are outlined in Section 5.7.1.

- Evaluation of all of the proposed metrics and methods described above on real-world datasets. The methods are also compared to techniques present in the literature and are all compared with known ground truth where available. The evaluations carried out are given in Chapter 6

7.2 Future Work

There are five promising areas via which this research could be continued. The first is the use of sensors that are only able to observe 2D and 1D transformations. Currently, the method has only been applied to sensors that are able to observe the full 3D transformation of the surrounding environment. However, this limitation is not a requirement for any stage of the approach as, if the sensor transforms are given infinite variance in directions where the motion is unobservable by the sensor, the methods may operate in their current form. The issue here, and the reason why this area requires further research, is that this limited observability will tend to result in exceptionally large variance in the calibration process and a fairly uninformative resulting calibration. The addition of extra environmental constraints or assumptions would likely be required before these sensors could accurately be calibrated to work with the system.

The second area would be to look at placing this framework into a real-time structure. This would allow a user to see the estimated calibration as the vehicle was collecting the data and assess how accurate the current calibration was, and what motions or observations would be required to improve it further.

The third area would be the inclusion of all of the sensor-intrinsic parameters in the optimisation. Currently we assume that all of the sensors have a known internal calibration, however for the system to be used in plug-play sensing where a non-expert users can change sensor configurations, the system will need to be robust to uncertain initialisation.

The fourth area is the examination of other outlier rejection techniques. The analysis of different robust estimation options was beyond the scope of this thesis and the trimmed means was chosen based on empirical testing that found it to be a simple, robust and effective means of outlier elimination. However, it discards the value of the readings for a significant portion of the provided data. This means that it is possible that a detailed analysis of the problem would yield a more efficient estimation technique.

Finally, the code aspect of this work is currently not in a form that would be user friendly to someone with limited programming knowledge. Work could be done to improve the approachability of the code through the development of a graphical user interface to allow non-expert users to make use of the system.

List of References

[1] Martin Kendal Ackerman, Alexis Cheng, Bernard Shiffman, Emad Boctor, and Gregory Chirikjian. Sensor calibration with unknown correspondence: Solving AX=XB using Euclidean-group invariants. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1308–1313, November 2013.

[2] Nicolas Andreff, Radu Horaud, and Bernard Espiau. Robot Hand-Eye Calibration Using Structure-from-Motion. *The International Journal of Robotics Research*, 20(3):228–248, March 2001.

[3] Charles Audet and J.E Jr Dennis. Analysis of Generalized Pattern Searches. *SIAM Journal on Optimization*, 13(3):889–903, 2002.

[4] Asma Azim and Olivier Aycard. Detection, classification and tracking of moving objects in a 3D environment. In *IEEE Intelligent Vehicles Symposium*, pages 802–807, 2012.

[5] Anton Bardera, Miquel Feixas, Imma Boada, and Mateu Sbert. High-dimensional normalized mutual information for image registration using random lines. In *Biomedical Image Registration*, volume 2057, pages 264–271, 2006.

[6] Timothy D. Barfoot and Paul T. Furgale. Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems. *IEEE Transactions on Robotics*, 30(3):679–693, June 2014.

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF : Speeded Up

Robust Features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[8] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[9] Paul Besl and Neil McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[10] Christoph Bodensteiner, Wolfgang Hubner, Kai Jungling, Peter Solbrig, and Michael Arens. Monocular Camera Trajectory Optimization using LiDAR Data. In *Computer Vision Workshops (ICCV Workshops)*, pages 2018–2025, November 2011.

[11] Andrea Censi and Roma La. An accurate closed-form estimate of ICP's covariance. In *IEEE International Conference on Robotics and Automation*, number April, pages 10–14, 2007.

[12] Jian Chen and Jie Tian. Real-time multi-modal rigid registration based on a novel symmetric-SIFT descriptor. *Progress in Natural Science*, 19(5):643–651, May 2009.

[13] Sam Chen. Another Particle Swarm Toolbox. 2009. URL `http://au.mathworks.com/matlabcentral/fileexchange/25986-another-particle-swarm-toolbox`.

[14] Massimiliano Corsini, Matteo Dellepiane, Federico Ponchio, and Roberto Scopigno. Image to Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties. *Computer Graphics Forum*, 28(7):1755–1764, 2009.

[15] Erik B Dam, Martin Koch, and Martin Lillholm. Quaternions , Interpolation and Animation. Technical report, 1998.

[16] Fred Daum and Jim Huang. Curse of Dimensionality and Particle Filters. In *IEEE Aerospace Conference*, volume 4, pages 1979–1993, 2003.

[17] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[18] Andreas Geiger and Philip Lenz. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, June 2012.

[19] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE International Conference on Robotics and Automation*, pages 3936–3943. Ieee, May 2012.

[20] David Goldberg and John Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[21] Mikael Haggstrom. Image: CT scan of the human brain. URL `https://commons.wikimedia.org/wiki/File:Computed_tomography_of_human_brain_-_large.png`.

[22] Nikolaus Hansen. The CMA evolution strategy: A comparing review. *Studies in Fuzziness and Soft Computing*, 192(2006):75–102, 2006.

[23] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. *Evolutionary Computation*, pages 312–317, 1996.

[24] Richard I. Hartley. In defence of the 8-point algorithm. In *IEEE International Conference on Computer Vision*, 1995.

[25] Mattias P. Heinrich, Mark Jenkinson, Manav Bhushan, Tahreema Matin, Fergus V. Gleeson, Sir Michael Brady, and Julia a. Schnabel. MIND: Modality Independent Neighbourhood Descriptor for Multi-Modal Deformable

Registration. In *Medical Image Analysis*, pages 1423–1435. Elsevier B.V., May 2012.

[26] Jan Heller, Michal Havlena, Akihiro Sugimoto, and Tomas Pajdla. Structure-from-motion based hand-eye calibration using L infinity minimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3497–3503, June 2011.

[27] Lionel Heng, Bo Li, and Marc Pollefeys. CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *IEEE International Conference On Intelligent Robots and Systems*, pages 1793–1800, November 2013.

[28] Lionel Heng, B Mathias, Gim Hee Lee, Paul Furgale, Roland Siegwart, and Marc Pollefeys. Infrastructure-Based Calibration of a Multi-Camera Rig. In *IEEE International Conference on Robotics and Automation*, pages 4912–4919, 2014.

[29] Calvin Hung, Juan Nieto, and Zachary Taylor. Orchard fruit segmentation using multi-spectral feature learning. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5314–5320, 2013.

[30] Velodyne Acoustics Inc. User's Manual and Programming Guide HDL-64E S2 and S2.1. Technical report, 2012.

[31] E. T. Jaynes. *Probability theory: the logic of science*, volume 27. 2005.

[32] Wolfgang Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 34(5):827–828, 1978.

[33] Nima Keivan and Gabe Sibley. Online SLAM with Any-time Self-calibration and Automatic Change Detection. In *IEEE International Conference on Robotics and Automation*, pages 5775–5782, 2015.

[34] Sung Lee, Soon Jung, and Ram Nevatia. Automatic integration of facade textures into 3D building models with a projective geometry based line clustering. In *Computer Graphics Forum*, volume 21, pages 511–519, 2002.

[35] Jesse Levinson and Sebastian Thrun. Automatic Calibration of Cameras and Lasers in Arbitrary Scenes. In *International Symposium on Experimental Robotics*, 2012.

[36] Hui Li, Cheng Zhong, and Xianfeng Huang. Reliable registration of LiDAR data and aerial images without orientation parameters. *Sensor Review*, 32(4), 2012.

[37] Mingyang Li, Hongsheng Yu, Xing Zheng, and AI Mourikis. High-fidelity Sensor Modeling and Self-Calibration in Vision-aided Inertial Navigation. In *IEEE International Conference on Robotics and Automation*, pages 409–416, 2014.

[38] Lingyun Liu and Ioannis Stamos. A systematic approach for 2D-image to 3D-range registration in urban environments. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[39] David Lowe. Object Recognition fromLocal Scale-Invariant Features. In *IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

[40] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.

[41] Kieran Maher. Image: MRI of the human brain. URL `https://commons.wikimedia.org/wiki/File:T1t2PD.jpg`.

[42] Bryan F. Manly. *Randomization, Bootstrap and Monte Carlo Methods in Biology*. CRC Press, 3rd edition, 2006.

[43] Ricardo A. Maronna, R. Douglas Martin, and Victor J. Yohai. *Robust Statistics: Theory and Methods*. John Wiley & Sons, Ltd, 2006.

[44] Andrew Mastin, Jeremy Kepner, and John Fisher III. Automatic registration of LIDAR and optical images of urban scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2639–2646, 2009.

[45] Mathworks. Aerospace Toolbox User's Guide R 2015a. Technical report, 2015. URL `http://au.mathworks.com/help/releases/R2015a/pdf_doc/aerotbx/aerotbx_ug.pdf`.

[46] Christoph Mertz, Luis E. Navarro-Serment, Robert MacLachlan, Paul Rybski, Aaron Steinfield, Arne Suppe, Christopher Urmson, Nicolas Vandapel, Martial Hebert, and Chuck Thorpe. Moing Object Detection with Laser Scanners. *Journal of Field Robotics*, 1:17–43, 2013.

[47] Said M. Mikki and Ahmed a. Kishk. Particle Swarm Optimization: A Physics-Based Approach. *Synthesis Lectures on Computational Electromagnetics*, 3(1):1–103, January 2008.

[48] Louis M. Milne-Thomson. *The Calculus of Finite Differences*. Chelsea Pub Co, 2000.

[49] P Morton, B Douillard, and J Underwood. An evaluation of dynamic object tracking with 3D LIDAR. In *Australasian Conference on Robotics and Automation*, pages 7–9, 2011.

[50] Richard J Murphy, Sven Schneider, Zachary Taylor, and Juan Nieto. Mapping clay minerals in an open-pit mine using hyperspectral imagery and automated feature extraction. In *Vertical Geology Conference*, pages 145–156, 2014.

[51] Richard J Murphy, Zachary Taylor, Sven Schneider, and Juan Nieto. Mapping clay minerals in an open-pit mine using hyperspectral and LiDAR data. *European Journal of Remote Sensing*, 2015.

[52] Ashley Napier, Peter Corke, and Paul Newman. Cross-Calibration of Push-Broom 2D LIDARs and Cameras In Natural Scenes. In *IEEE International Conference on Robotics and Automation*, pages 3679–3684, 2013.

[53] John A Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[54] Juan Nieto, Sildomar Monteiro, and Diego Viejo. 3D geological modelling using laser and hyperspectral data. *Geoscience and Remote Sensing Symposium*, pages 4568–4571, 2010.

[55] Gary W. Oehlert. A note on the delta method. *The American Statistician*, 46 (1):27–29, 1992.

[56] Francisco P.M. Oliveira and João Manuel R.S. Tavares. Medical image registration: a review. *Computer Methods in Biomechanics and Biomedical Engineering*, (July 2015):1–21, 2012.

[57] Hannes Ovren and Per-Erik Forssen. Gyroscope-based Video Stabilisation With Auto-Calibration. In *IEEE International Conference on Robotics and Automation*, pages 2090–2097, 2015.

[58] Andrew W. Palmer, Andrew J. Hill, and Steven J. Scheding. Stochastic collection and replenishment (SCAR): Objective functions. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3324–3331, 2013.

[59] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Extrinsic calibration of a 3D laser scanner and an omnidirectional camera. In *IFAC Symposium on Intelligent Autonomous Vehicles*, volume 7, pages 336–341, 2010.

[60] Gaurav Pandey, James R McBride, and Ryan Eustice. Ford campus vision and lidar data set. In *The International Journal of Robotics Research*, pages 1543–1552, 2011.

[61] 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. *AAAI Conference on Artificial Intelligence*, 26:2053–2059, 2012.

[62] Gaurav Pandey, James R Mcbride, Silvio Savarese, and Ryan M Eustice. Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information. *Journal of Field Robotics*, 2014.

[63] Graeme Penney, Jurgen Weese, John Little, Paul Desmedt, Derek Hill, and David Hawkes. A comparison of similarity measures for use in 2-D-3-D medical image registration. *Transactions on Medical Imaging*, 17(4):586–95, August 1998.

[64] Josien P. W. Pluim, J B Maintz, and Max A. Viergever. Image registration by maximization of combined mutual information and gradient information. *Transactions on Medical Imaging*, 19(8):809–14, August 2000.

[65] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE*, 22(8):986–1004, 2003.

[66] Rishi Ramakrishnan, Juan Nieto, and Steve Scheding. Shadow Compensation for Outdoor Perception. In *IEEE International Conference on Robotics and Automation*, pages 4835–4842, 2015.

[67] Calyampudi Radakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, (37):81–89, 1945.

[68] Alexis Roche, Grégoire Malandain, Xavier Pennec, and Nicholas Ayache. The Correlation Ratio as a New Similarity Measure for Multimodal Image Registration. In *International Conference on Medical Image Computing and Computer-Assisted Interventation (MICCAI)*, volume 1496, page 1115, 1998.

[69] Radu Bogdan Rusu. Semantic 3D Object Maps for Everyday Manipulation in

Human Living Environments. *Künstliche Intelligenz*, 24(4):345–348, August 2010.

[70] Ingo Schiller, Christian Beder, and Reinhard Koch. Calibration of a PMD camera using a planar calibration object together with a multi-camera setup. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.*, 37:297–302, 2008.

[71] Danilo Schneider and Hans-Gerd Maas. Geometric modelling and calibration of a high resolution panoramic camera. *Optical 3-D Measurement Techniques VI*, 2003.

[72] Sebastian Schneider, Thorsten Luettel, and Hans-Joachim Wuensche. Odometry-based online extrinsic sensor calibration. In *IEEE International Conference on Intelligent Robots and Systems*, number 2, pages 1287–1292, November 2013.

[73] Robert A. Schowengerdt. *Remote Sensing, Models and Methods for Image Processing*. Elsevier Inc., 3rd edition, 2007.

[74] Claude Elwood Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[75] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[76] Yiu Cheung Shiu and Shaheen Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB. *IEEE Transactions on Robotics and Automation*, 5(1):16–29, 1989.

[77] Malcolm D. Shuster. A Survey of Attitude Representations. *The Journal of Astronautical Sciences*, 41(4):439–517, 1993.

[78] Irwin Sobel. An isotropic 3× 3 image gradient operator. *Machine Vision for three-demensional Sciences*, 1(1):23–34, 1990.

[79] Bastian Steder and K Rainer. Maximum Likelihood Remission Calibration for Groups of Heterogeneous Laser Scanners. In *IEEE International Conference on Robotics and Automation*, 2015.

[80] Colin Studholme, Derek L.G. Hill, and David J Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32(1): 71–86, January 1999.

[81] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics and Iteractive Techniques*, pages 251–258, 1997.

[82] Levente Tamas and Zoltan Kato. Targetless calibration of a lidar - Perspective camera pair. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 668–675, 2013.

[83] Zachary Taylor. Calibration Demonstrations. URL `http://www.zjtaylor.com/videos`.

[84] Zachary Taylor. Multi-modal Array Calibration Source Code. 2015. URL `http://www.zjtaylor.com/code`.

[85] Zachary Taylor and Juan Nieto. A Mutual Information Approach to Automatic Calibration of Camera and Lidar in Natural Environments. In *Australian Conference on Robotics and Automation*, pages 3–5, 2012.

[86] Zachary Taylor and Juan Nieto. Gradient Based Multi-modal Sensor Calibration. *IEEE International Conference on Robotics and Automation: Modelling, Estimation, Perception and Control of All Terrain Mobile Robots Workshop*, 2014.

[87] Zachary Taylor and Juan Nieto. Parameterless Automatic Extrinsic Calibration of Vehicle Mounted Lidar-Camera Systems. In *IEEE International Conference on Robotics and Automation: Long Term Autonomy Workshop*, 2014.

[88] Zachary Taylor and Juan Nieto. Motion-Based Calibration of Multimodal Sensor Arrays. In *IEEE International Conference on Robotics and Automation*, 2015.

[89] Zachary Taylor, Juan Nieto, and David Johnson. Automatic calibration of multi-modal sensor systems using a gradient orientation measure. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1293–1300, November 2013.

[90] Zachary Taylor, Juan Nieto, and David Johnson. Multi-Modal Sensor Calibration Using a Gradient Orientation Measure. *Journal of Field Robotics*, 2014.

[91] Atousa Torabi and GA Bilodeau. Local self-similarity as a dense stereo correspondence measure for themal-visible video registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 61–67, 2011.

[92] Phil Torr and Andrew Zisserman. Robust computation and parametrization of multiple view relations. In *International Conference on Computer Vision*, 1998.

[93] Roger Tsai and Reimar Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *Robotics and Automation*, 5(3), 1989.

[94] Zsolt Ugray, Leon Lasdon, John Plummer, and Fred Glover. Scatter Search and Local NLP Solvers : A Multistart Framework for Global Optimization. *Information Systems*, (May), 2006.

[95] James P Underwood. *Reliable and safe autonomy for ground vehicles in unstructured environments*. PhD thesis, 2008.

[96] Ranjith Unnikrishnan and Martial Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. 2005.

[97] A Vedaldi and B Fulkerson. VLFeat: An open and portable library of computer vision algorithms. *ACM International Conference on Multimedia*, 2010.

[98] Christian Wachinger and Nassir Navab. Structural image representation for image registration. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 23–30, June 2010.

[99] Ching Cheng Wang. Extrinsic calibration of a vision sensor mounted on a robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175, 1992.

[100] Ruisheng Wang, Frank P. Ferrie, and Jane Macfarlane. Automatic registration of mobile LiDAR and spherical panoramas. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 33–40, June 2012.

[101] Anthony Winning. Photo: Sydney Opera House at night. 2005. URL `https://commons.wikimedia.org/wiki/File:Sydney_Opera_House_Night.jpg`.

[102] F Zana and J C Klein. A multimodal registration algorithm of eye fundus images using vessels detection and Hough transform. *Transactions on Medical Imaging*, 18(5):419–28, May 1999.

[103] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). *IEEE International Conference on Intelligent Robots and Systems*, pages 2301–2306, 2004.

[104] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

Appendix A

Applications

During the development of this thesis the approaches formulated have been used to provide accurate data alignment for a large range of systems and situations. Many of these applications have no form of ground truth against which to evaluate the results, and so have not been included in the main section of this thesis. However as they still demonstrate real-world applications of the method, they have been included in this appendix.

The problems examined in this appendix are as follows:

A.1 Mine Face Classification
A.2 Sydney Opera House Registration
A.3 Illumination Invariant Dataset
A.4 IR-RGB Image Alignment for Almond Detection
A.5 Mine Site Visualisation
A.6 Line Scanner Mobile Rig Calibration

A.1 Mine Face Classification

Hyperspectral cameras detect a large range of spectral information which can be used to classify the materials in a given scene [73]. In this application the camera was used

to map clay minerals through the intensity of their response at different wavelengths on the face of a mine pit. Our methods were used to find the transformation between the hyper-spectral camera and a Riegl lidar scanner mounted on a separate tripod nearby. The combination of the sensors facilitated the mapping of the mineral classification onto the geography of the face, allowing us to obtain surface area estimates for each of the classified minerals. This work has been published in [50] and [51].

A.1.1 Registration

For this registration GOM was used as the alignment metric. The search space for the optimiser was formed around a rough guess as to the alignment, from observing the sensor outputs, and set to have a range of 10 degrees in roll, pitch and yaw, 10 metres in X and Z and 2 metres in Y. The focal length had a range of 100 pixels. An estimate of the variance in this calibration was also generated. This was done by bootstrapping the lidar data before registering it with the camera data. Using the bootstrapping process the registration was performed 20 times with the results used to estimate the variance of the estimated camera parameters.

A.1.2 Area Calculation

The area of the mine pit covered by each of the minerals was calculated in two separate ways. First, a simple estimate of the area was made using the image alone. In the second method the image was projected onto the registered lidar scan. To calculate the area from the point cloud, the space each point covers was calculated by assuming it was a square whose sides are the length of the median distance to the four closest neighbours. Small changes in the camera's position could have a large impact on the location of classified minerals. To account for this, the estimated variance in the camera parameters, previously calculated, was used to calculate a variance in the classified area using a Monte-Carlo sampling approach.

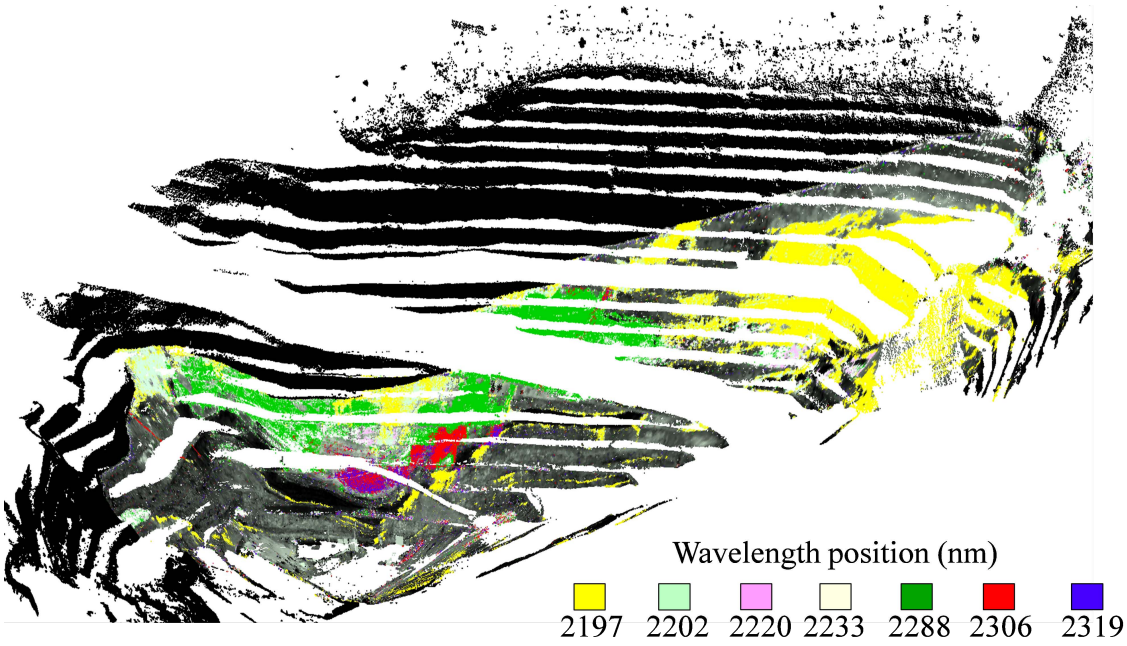


Figure A.1 – Minerals detected at each wavelength position on the cliff face.

Wavelength Position	Image Area (Pixels)	Face Surface Area (m^2)	Image Area (%)	Face Surface Area (%)	Face Surface σ (%)
Unclassified	466110	73461	72.3	67.0	1.0
2197 nm	54913	16351	8.5	14.9	0.3
2202 nm	29753	5867	4.6	5.4	0.3
2220 nm	6192	1251	1.0	1.1	0.3
2233 nm	18134	3007	2.8	2.7	0.2
2288 nm	39405	6318	6.1	5.8	0.4
2306 nm	17331	2329	2.7	2.1	0.2
2319 nm	12642	1053	2.0	1.0	0.1
Total	644480	109637	100	100	

Table A.1 – Area of face classified as each material

A.1.3 Results

The mapping of the classified image onto the face is shown in Figure A.1. The area covered by each of the minerals is shown in Table A.1. This table shows the two key advantages of projecting the image information onto the lidar scan. The first, is that it allows the quantification of the material in absolute units (in this case m^2), the second, is that it prevents the distance of the minerals from the camera effecting their perceived abundance. For example the minerals at wavelength 2197 makes up 8.5% of the minerals in the image but 14.9% of the minerals on the face. This is due to a large amount of this mineral detected near the top of the cliff, far from the camera.

	Manual	Automated
X	263.2	256.2
Y	7.1	10.3
Z	200.9	248.0
Roll	-173.1	-174.6
Pitch	-1.9	-1.9
Yaw	-91.7	-92.0
Focal X	5522	5797
Focal Y	5465	5701
Centre X	370	532
Centre Y	254	259

Table A.2 – Values found during registration, distances in metres and angles in degrees. The registration was performed assuming no lens distortion.

A.2 Sydney Opera House Registration

To allow an unrelated experiment to take place, a Riegl lidar scanner was taken to Sydney's Circular Quay. During the downtime in this experiment a scan of the Sydney Opera House was made. We decided to make use of this scan in demonstrating our alignment method, as the Opera House is a far more interesting target than our usual scans. However, no image of the Opera House had been taken and no one had made note of where the scanner was when it recorded the scan. We decided to register the Opera House with an image found on the internet [101]. This proved a challenging problem as we had very little idea of the initial parameters and had to account for a wide range of possible camera intrinsics as well as the possibility that the image had been cropped.

An initial guess was made by hand matching 12 points in the image to the lidar scan. Interior-point optimisation was then used to find the least squares error in the points' positions when the lidar points were projected onto the image. Once this solution had been obtained, the optimisation was performed using GOM. The results obtained for the manual- and automated method can be seen in Table A.2 and Figure A.2

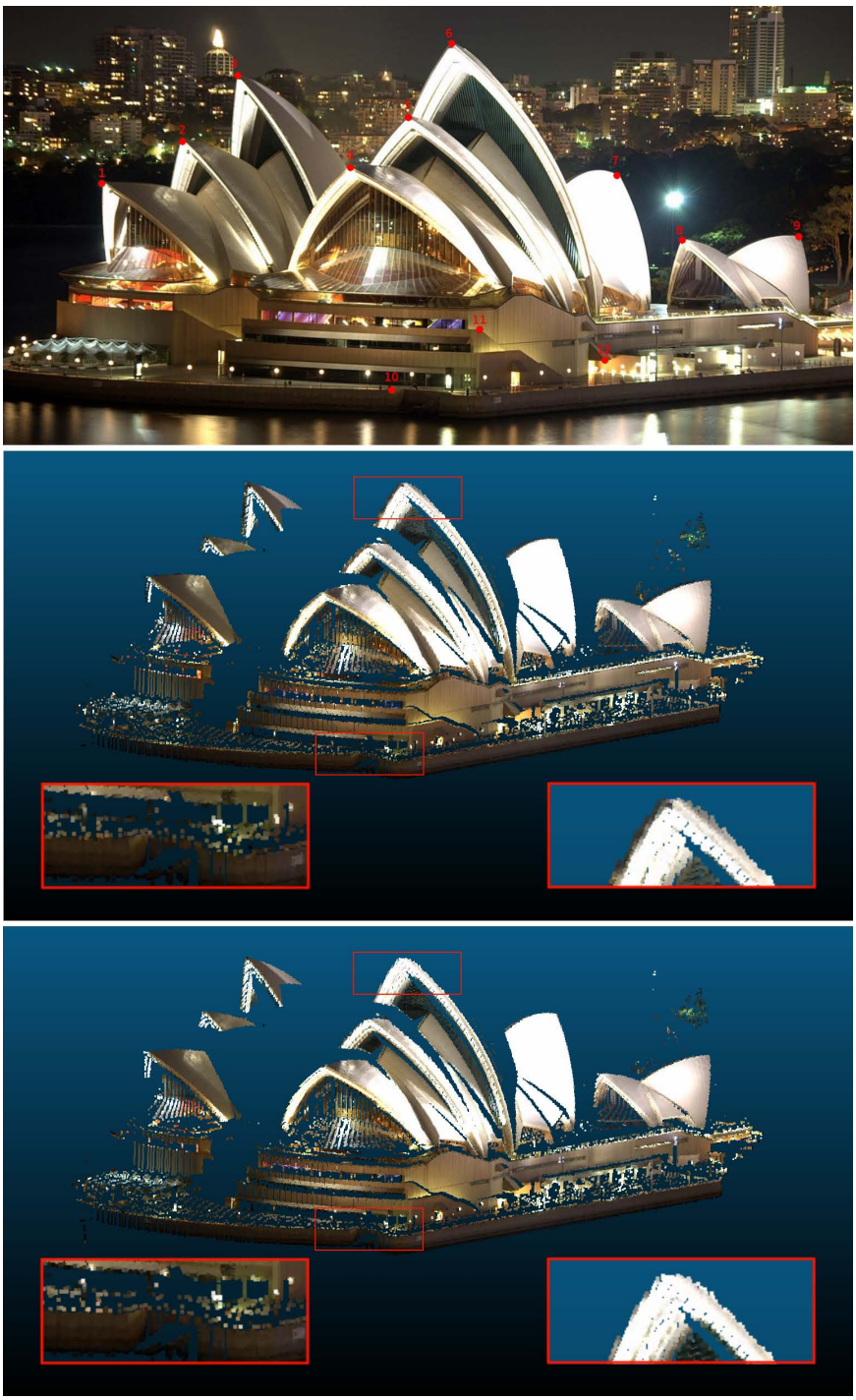


Figure A.2 – Registration of the Opera House image. Top: the image with hand matched points. Middle: the registration resulting from the matched points. Bottom: registration after automated alignment. The differences between the manual- and automated alignment can best be seen by looking at the far edge of the roof or at the light poles by the waterfront.

Figure A.3 – Image with cropped region shown in red assuming the focal point was originally in the centre of the image.

The output of the manual and automatic alignment give very similar looking projections. This is despite some of the parameters used to generate these coloured point clouds differing substantially (for example, the Z position). This is due to many parameters giving similar effects. For example, in this case where we have a single target in the distance, a change in the Z value or a change in the focal length will produce near identical results.

This registration also allows us to infer some additional information about the photo. If we assume that the camera's focal point was roughly in the centre of the image (a reasonable assumption for the majority of modern consumer cameras) we can see that a large section must have been cropped from the top left of the original image, as is shown in Figure A.3.

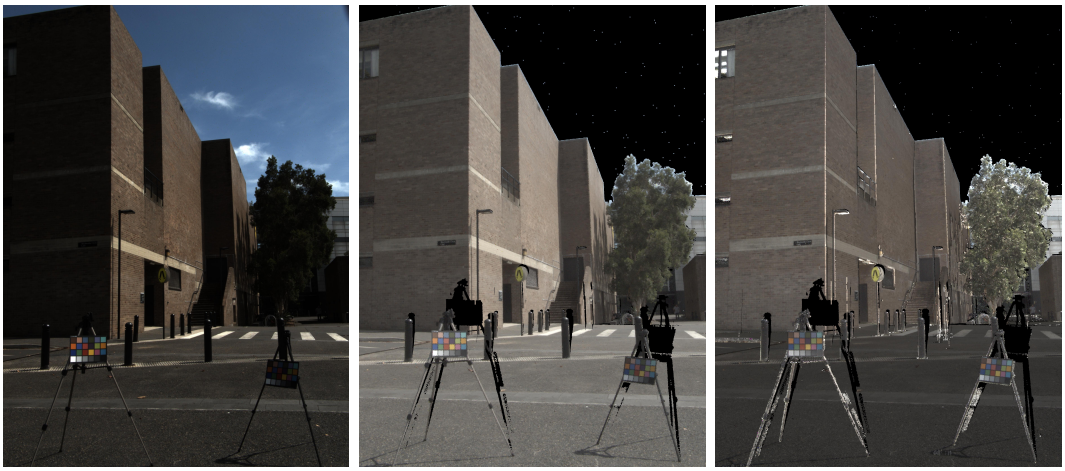


Figure A.4 – Left: the original image. Middle: after combination with the lidar. Right: after illumination invariance.

A.3 Illumination Invariant Dataset

A series of experiments were conducted for use in work by Ramakrishnan et al. [66]. In this work Ramakrishnan developed an approach to remove intensity changes due to lighting variation via the utilisation of an areas 3D structure. The 3D structure was given by making a high-resolution lidar scan of an urban area. Images of the area were then taken at different times of day over several days. While some effort was made to take each image from the same position, small variations in position needed to be compensated for. This registration was done using GOM. An example of the resulting registration after relighting can be seen in Figure A.4.

A.4 IR-RGB Image Alignment for Almond Detection

A multi-class image segmentation approach for automated fruit segmentation was developed by Hung et al. [29]. The method utilised both RGB and IR bands in its classification. The dataset of almond trees was made by taking a photo of an almond tree, swapping the lens to one with an IR filter, and taking a second photo.

This method of collecting the images meant there were some slight differences in the position and focal length of the IR and RGB images. This was compensated for by using an affine transformation and optimising its parameters using NMI. An example of the resulting alignment is shown in Figure A.5.

A.5 Mine Site Visualisation

A mine site in Western Australia had an aerial photo taken of it. Several months later a survey of the mine resulted in a high-resolution lidar scan of the mine being made. For visualisation purposes the image was used to colour the mine lidar scan. This presented several challenges; the aerial image was a combination of a large number of images and the original images were not available. Similarly, the lidar scans were a combination of a large number of scans with no reflectance information recorded. The mine had also been active during the months between, with some areas undergoing significant changes. To align the scans, an image was formed from the lidar scans by using an orthographic projection along its Z axis; the intensity of the resulting image was coloured by its altitude. The rotation, translation and scale between the two images was found using NMI and particle swarm optimisation. The resulting coloured point cloud is shown in Figure A.6.

A.6 Line Scanner Mobile Rig Calibration

A vehicle for quickly scanning roadsides was set up with a side-facing camera and lidar sensor to record a line scan that would be integrated with the navigation to generate a 3D colour map. The lidar sensor was a standard high-resolution Rigel scanner that could rotate. This, combined with the provided dimensions of the bracket both sensors were mounted to, made the problem very simple to solve using the GOM method. The results can be seen in Figure A.7

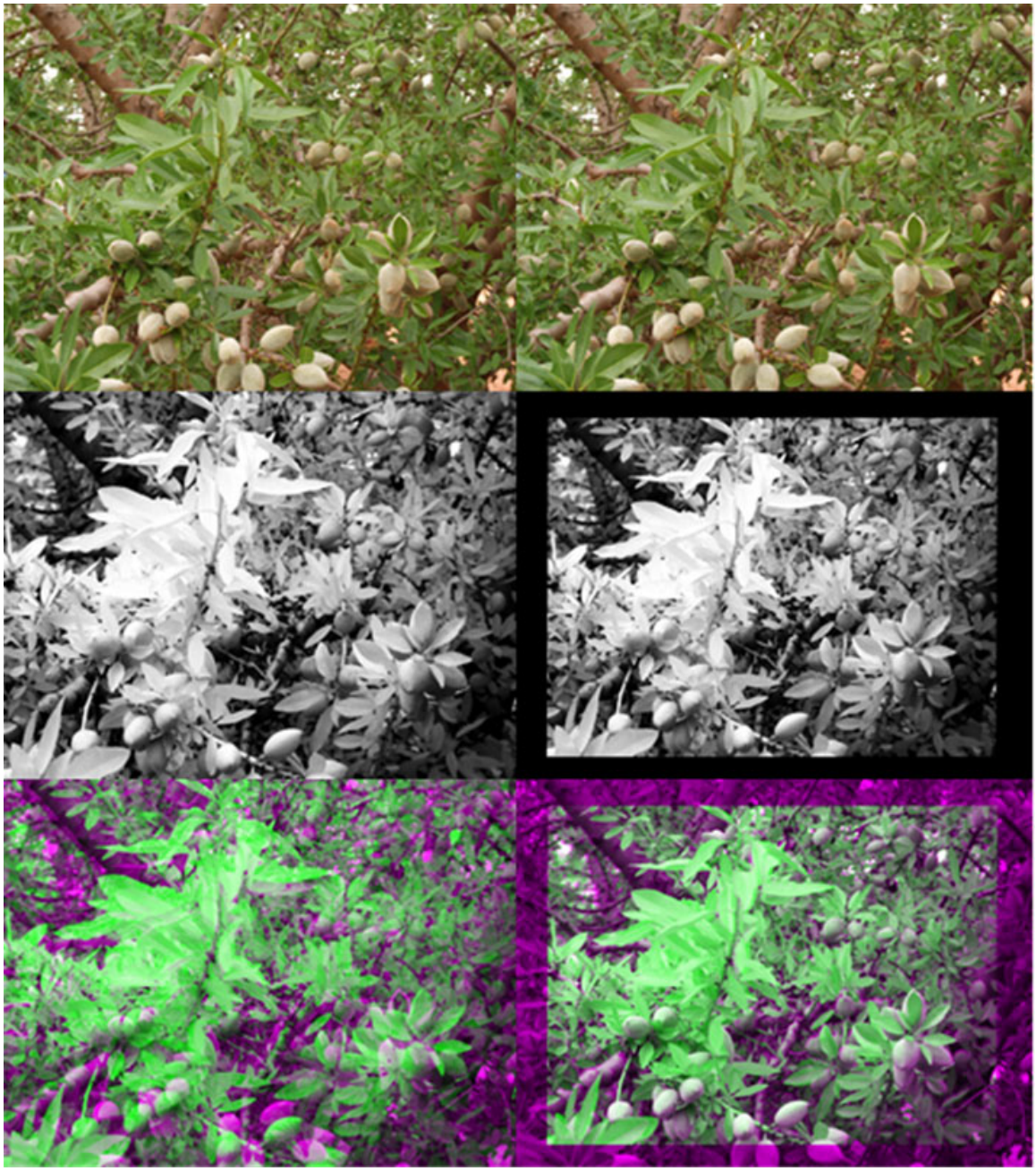


Figure A.5 – One of the images before (left) and after alignment (right). RGB is shown at the top, IR shown in the middle and alignment (two images superimposed) shown at the bottom.

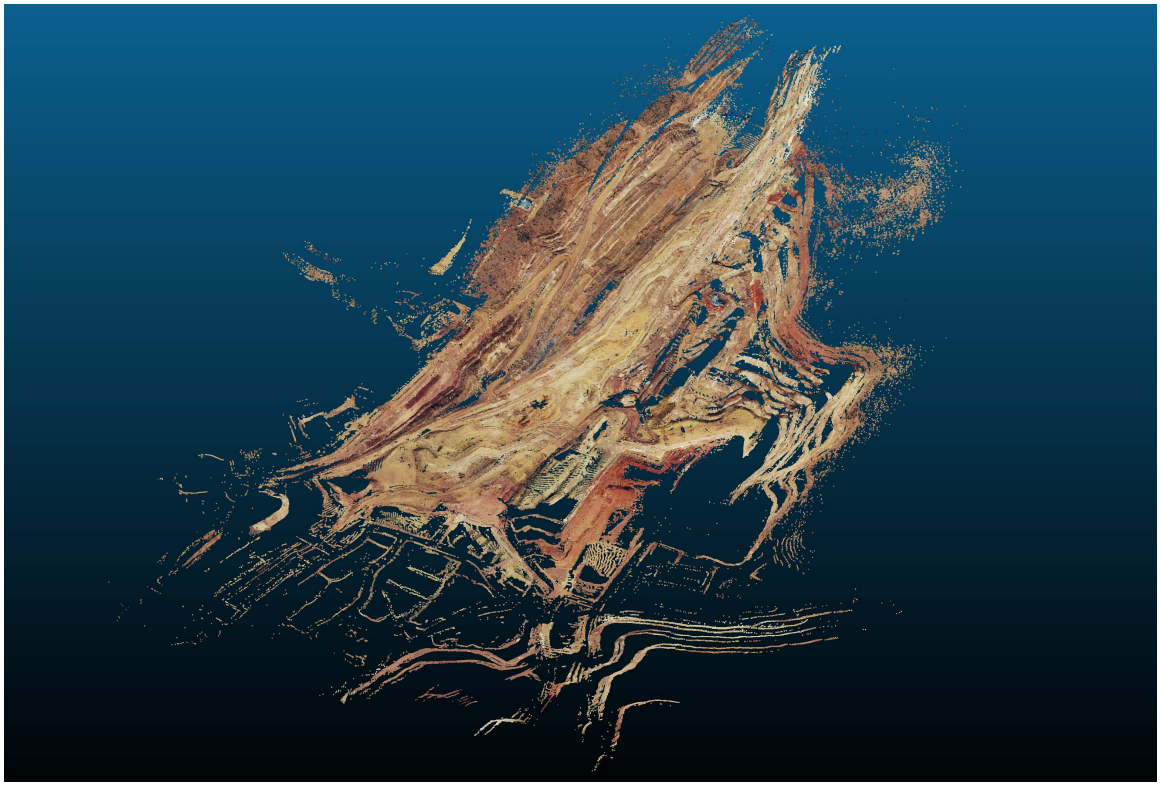


Figure A.6 – Scan of mine site coloured using an aerial image.

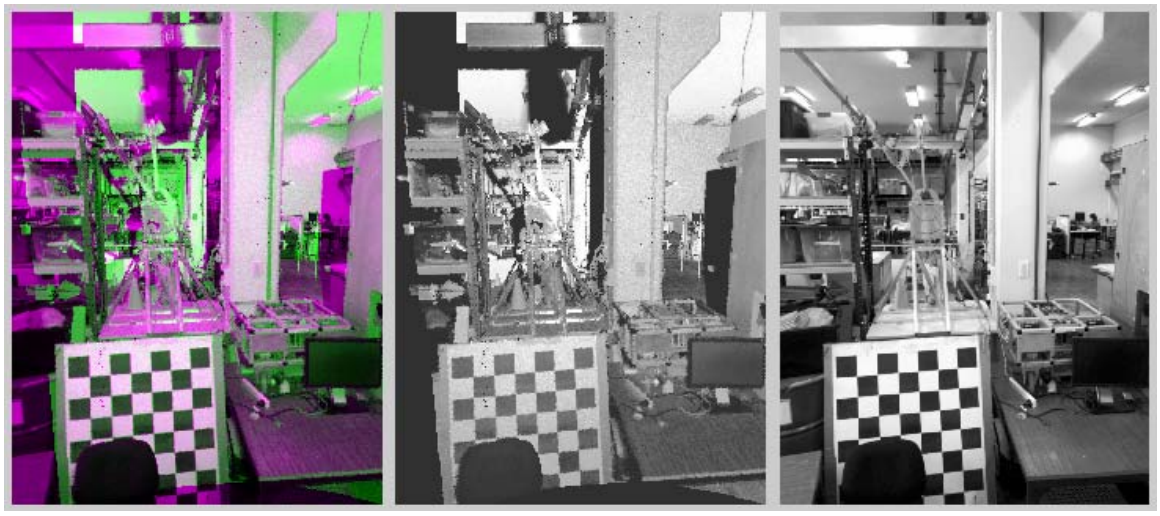


Figure A.7 – Left: calibrated lidar and camera superimposed. Middle: an image formed from the lidar scan Right: the camera image.

Appendix B

Accuracy Comparisons with Existing Literature

In the field of markerless extrinsic calibration of 3D lidar and camera systems, several methods such as MI, NMI and Levinson's method have been tested by several independent groups in a range of conference and journal papers. The claimed accuracy of the methods between papers however varies substantially, often giving over an order of magnitude difference for similar setups. Among these results our own findings are some of the most pessimistic about the accuracy with which markerless methods are able to calibrate a system.

In spite of the large range of results, we believe that all of the experiments we have encountered by different authors testing these systems have been conducted in a fair manner with the results giving an accurate assessment of the method's abilities. We believe the difference in results stems from subtle differences in the sensors, the initial information provided to the systems and the assumptions the authors have made about the system. The aim of this section is to briefly discuss the findings of several prominent authors in the field and comment on some of the finer details that contributed to the final estimated solutions they discovered. This section will mainly focus on the work presented in [28, 35, 52, 61, 100].

B.1 Influences on Calibration Accuracy

One of the largest influences on accuracy, especially relating to translational accuracy, is the use of multiple cameras facing in different directions. In many instances the authors have made use of spherical camera rigs ([35, 61, 100]) such as the Ladybug camera. These setups house several cameras that have factory-calibrated extrinsic and intrinsic parameters. When the camera outputs are combined these systems offer a near complete panoramic view of the world. If the user is prepared to take the calibration between the cameras of this system as correct (usually a good assumption) then these systems greatly improve the observability of most calibration parameters, as was touched on in Section 6.3.5. This is due to three effects; firstly translations perpendicular to the camera direction have a far more pronounced effect on the camera's image than translations towards or away from the camera. In a panoramic system all translations are perpendicular to part of the camera image and so are more clearly observed. Secondly, for one camera some translations and rotations will affect the output in a similar manner; these movements however would have a vastly different effect on cameras facing in opposite directions. Finally, the use of a spherical system alleviates the need to account for the bias a metric will exhibit due to differing overlap in the field of view of the sensors. This is as a result of the spherical camera system possessing a near-360-degree field, which results in all of the lidar points projecting onto the camera image.

A second factor that will have a significant influence on the accuracy of calibrations obtained by a system is the combination of the lidar with a pre-existing GPS/INS system. The use of the pre-calibrated GPS/INS system allows for the movement of the vehicle during the lidar scans to be accounted for and allows for, any timing offset between the scan and the camera images to be overcome. In the case of [52] it also allows a 2D lidar scan to be calibrated via the transformation into a 3D scan. This system is used by [35, 61, 100].

The quality of the initial guess and restriction of the search space also affects the quality of the obtained solution and the likelihood of converging to an erroneous

solution. While the method of determining the initial conditions differed substantially, with the exception of [35] and in some select experiments [61], all authors generally provided the most accurate hand-calibrated solution as an initial starting point for the system.

Finally, to compare the method's accuracy you must first possess a correct ground truth to the extrinsic calibration, something notoriously hard to come by. In our own work we have typically used the ground truth provided by more manual methods, as was covered in Section 6.2.6. This however relies on the reader understanding that a portion of the error in the method they are seeing is due to the error in what has been taken to be the "ground truth". For some authors the error this introduces is not seen as acceptable and so, instead of reporting their method's accuracy, they resort to reporting its precision. That is, they run the method on a range of sections of data and report its standard deviation. This approach, while understandable, will underestimate any true error and not account for any bias the method may exhibit.

B.2 Presented Accuracy of Methods

Due to the possible combinations of sensors, initialisations, methods of measuring error, assumptions and other details discussed above, the range of accuracies found can show substantial variation. A summary of some of the most prevalent methods is shown in Table B.1. In our own work we have placed a strong emphasis on the calibration of sensors while making minimal assumptions. Because of this we have never made use of any pre-calibrated extrinsic relationship between sensors. We have also used search spaces that exhibit far more uncertainty in the initial parameters than most authors consider. This, combined with our comparison of methods in terms of their accuracy with respect to ground truth rather than their repeatable precision, has led to results that tend to appear underwhelming when compared to some of the results in Table B.1. We feel however that our results manage to capture many of the issues that a general system may encounter if these methods were used by non-experts.

Paper	Method	Error Type	Dataset	# of Scans	Roll	Pitch	Yaw	X	Y	Z
[35]	Hand Calibration	σ of precision	Own	100	0.460	0.510	0.620	0.121	0.077	0.126
	Levinson	σ of precision	Own	100	0.014	0.028	0.010	0.006	0.006	0.006
[62]	MI	σ of precision	Ford	40	0.050	0.050	0.100	0.030	0.010	0.010
	Levinson	σ of precision	Ford	40	0.400	0.700	0.200	0.050	0.040	0.090
[59]	Marker-based	σ of accuracy	Simulated	20	0.001	< 0.001	< 0.001	0.001	0.001	0.001
[52]	Napier	σ of precision	Own	20	0.380	0.390	0.440	0.045	0.052	0.046
[28]	Heng	Error wrt CamOdoCal	Own	500	0.004	0.003	0.009	0.002	0.002	0.002
[82]	MI	Error wrt KITTI	KITTI	1	0.800	1.200	0.900	0.007	0.026	0.180
[100]	MI	Error wrt hand labelled	Own	10	Error given in pixels: 0.88 pixels					
This Thesis	NMI	Median error wrt KITTI	KITTI	25	0.516	0.376	0.593	0.059	0.059	0.054
	GOM	Median error wrt KITTI	KITTI	25	0.298	0.138	0.479	0.044	0.055	0.022
	Levinson	Median error wrt KITTI	KITTI	25	0.384	0.255	0.470	0.094	0.035	0.007
	Motion	Median error wrt KITTI	KITTI	25	0.411	0.315	0.449	0.052	0.272	0.331
	IM	Median error wrt KITTI	KITTI	25	0.186	0.055	0.379	0.036	0.074	0.018

Table B.1 – A summary of some of the results found in the literature. For brevity only a single representative result with each metric was taken from each paper. In several instances the results were only displayed graphically and in these instances the exact values were estimated.

Appendix C

Combining Sensor Information

Once the calibration of the system has been performed, the sensor information can be combined to give a more complete representation of the world than any one sensor could provide alone. This section briefly examines several of the practical issues encountered when combining cameras, 3D lidar and GPS/INS sensors to form a coloured map of the world. An example of a section of one of these maps is shown in Figure C.1.

C.1 Image Noise

In creating visually pleasing coloured point clouds, an issue that is often encountered is that the process of projecting an image onto a point cloud can enhance the visibility of any image noise or artefacts present. This is best shown through an example. Consider the white car shown in Figure C.2. On close examination the colouring of the car in some regions of the Velodyne scan may appear unusual. For example, the licence plate contains no white or black; instead this area is filled with patches of bright orange and blue.

The reason for this odd colouring is apparent if we enlarge a small section of the original colour image. This is done in Figure C.3. This figure shows that, although the image taken as a whole displays a white license plate with black lettering, it is

Figure C.1 – A section of a coloured 3D point cloud generated using the lidar, cameras and GPS/INS systems of the KITTI system.

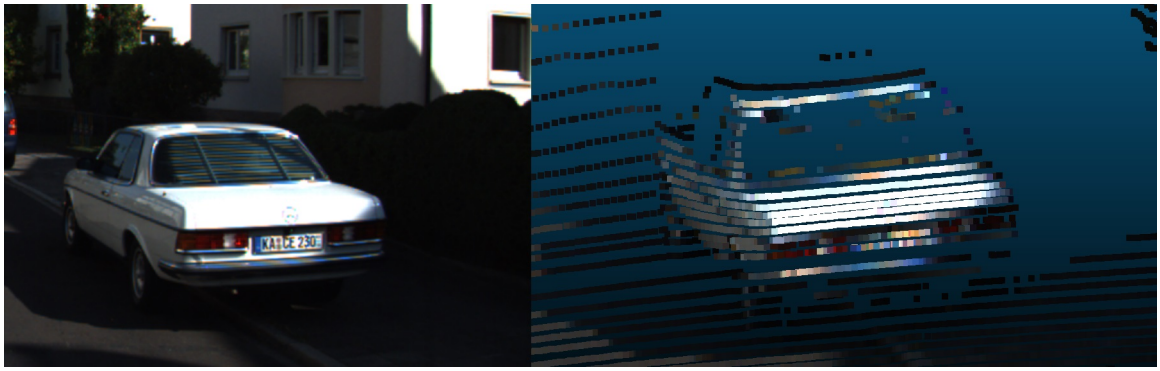


Figure C.2 – A Velodyne scan of a white car that has been coloured by a camera image. The car contains significant regions of bright orange and blue colouring.

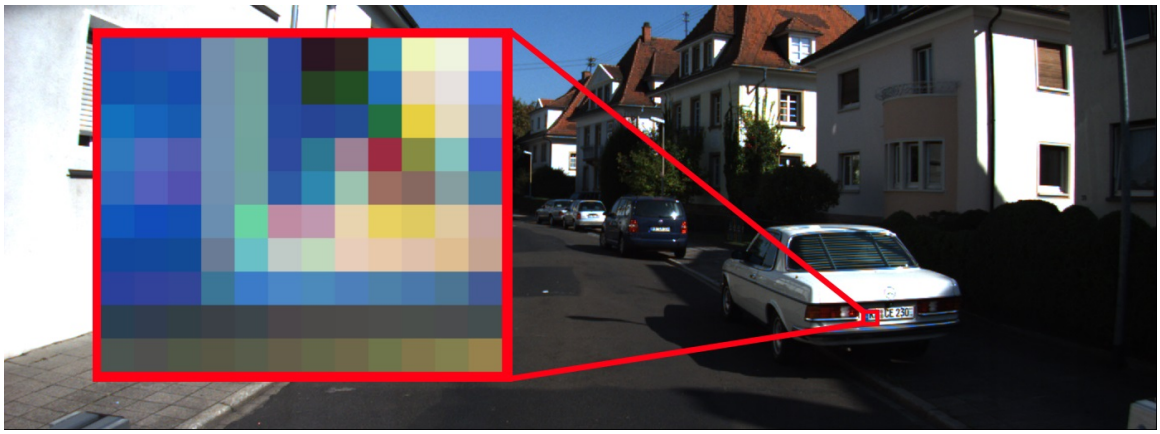


Figure C.3 – The image of the white car where a small section of the image has been enlarged. In this region the colour of the individual pixels differ significantly from the perceived colour of the patch.

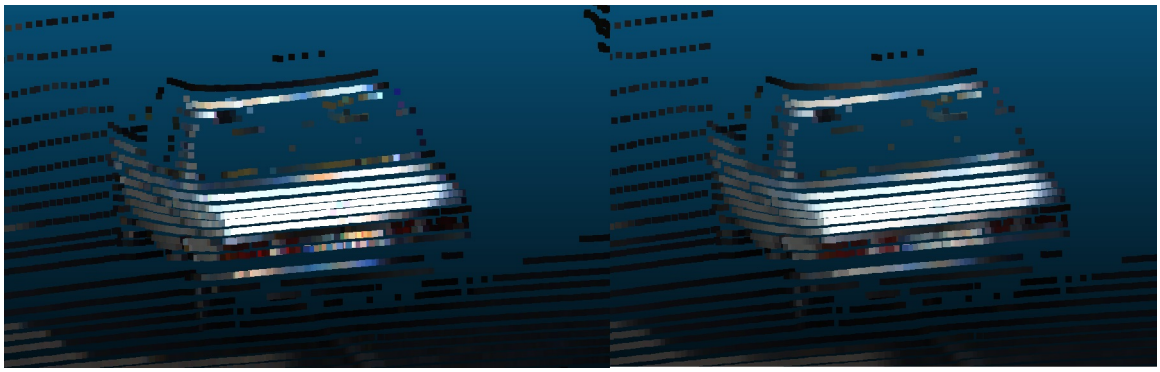


Figure C.4 – Left, a Velodyne scan of a white car coloured by a camera image. Right, the same scan coloured by the image after a Gaussian blur with a standard deviation of 2 was applied. The right-hand image maintains a more natural colouring with fewer bright orange regions then the left-hand image.

actually composed of brightly coloured pixels of a range of hues. This means that when the colours of individual pixels are taken and applied to the much sparser Velodyne scan, the colour of individual points becomes far more noticeable leading to the image appearing unnatural in some regions.

To correct for this we wish each Velodyne point to take on the perceived value of all the pixels in a small region rather than a single pixel intensity value. In practice this can be implemented by simply applying a slight Gaussian blur to the image before projecting the Velodyne onto it. The result of this is shown in Figure C.4.

C.2 Occlusions

As the cameras and 3D lidar are located at different points on the vehicle, the parts of the scene that are occluded will be different for each sensor. Therefore, when it comes to colouring a lidar scan using the cameras' colour information, a naive approach of simply projecting all of the lidar points onto the image and assigning them the colour at that location, will give results such as those seen in Figure C.5. In this figure it can be observed that several regions in the scan have been mis-coloured. The most noticeable areas are the pole and car, which have been projected onto areas behind the actual objects. It should be clearly noted here that this mis-colouration is not due to the mis-calibration of the intrinsics or extrinsics of the system. It has occurred as, from the camera's perspective, the car and pole are occluding sections of the background that the lidar is observing. This is then not accounted for in the camera model, resulting in the camera assigning the same colour to any point that lies behind any foreground objects. The closer an object is to the camera and the further the camera is from the lidar, the more pronounced this issue will become.

To account for this issue we adopt a scheme similar to that used in Section 5.7.1 where points likely to be occluded were removed. To remove occluded points we first project all points onto the image and record their image coordinates. We then find the 10 closest neighbours to each point, if any of these neighbours is over 0.1 metres in front of the point it is deemed to have a high likelihood of being occluded and is not assigned a colour from the image. While this method has the potential to exclude valid points when considering situations such as chain-link fences, it typically gives good results in an efficient manner.

It should also be noted that the points which are excluded will depend on the position of the camera. Therefore if we make use of multiple cameras in different positions, the number of points that cannot be assigned colour information due to occlusion is vastly reduced. Figure C.6 shows the results of our occlusion removal.

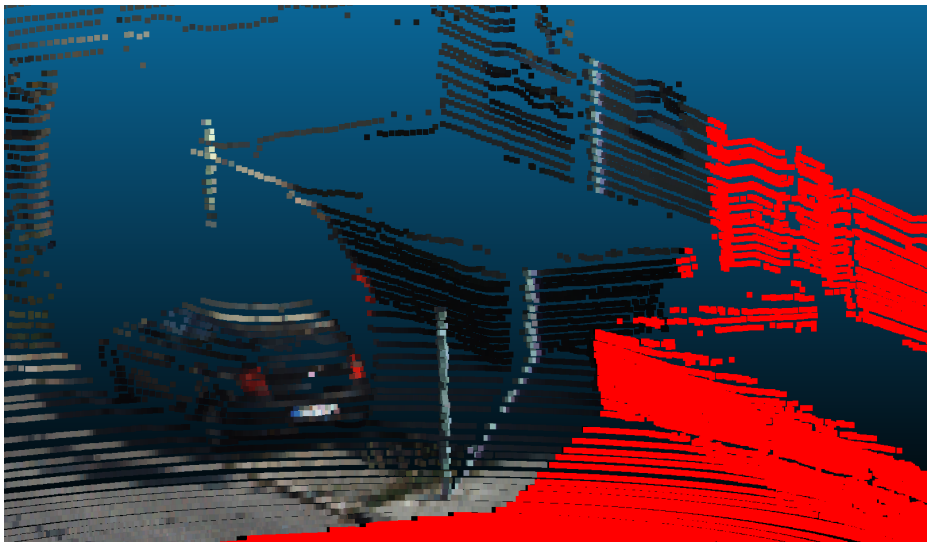


Figure C.5 – A frame from the KITTI dataset where the left colour camera image has been projected onto the Velodyne's lidar scan. The difference in the perspective of the sensors has resulted in some points in the background receiving the colour of foreground objects; for example the sign post and edge of the car have been projected onto the wall behind. Note, red is used to colour the scan where no camera information exists.

C.3 Dynamic Objects

To combine the individual scans into a larger map, dynamic objects will need to be removed. If this is not done the object may be placed into the scene many times, giving a poor output. The detection of these dynamic objects from lidar scans is an area that has seen significant attention [4, 46, 49]. However, the implementation and use of these methods is beyond the scope of this thesis. In this area we simply wish to note that in many cases we have found it sufficient to use simpler approaches, such as defining a region in front of the vehicle, in which points do not contribute to the map. Also if the purpose is simply to generate a 2D image of the map, simply overlaying each new scan over the previous one tends to remove trails left by dynamic objects.

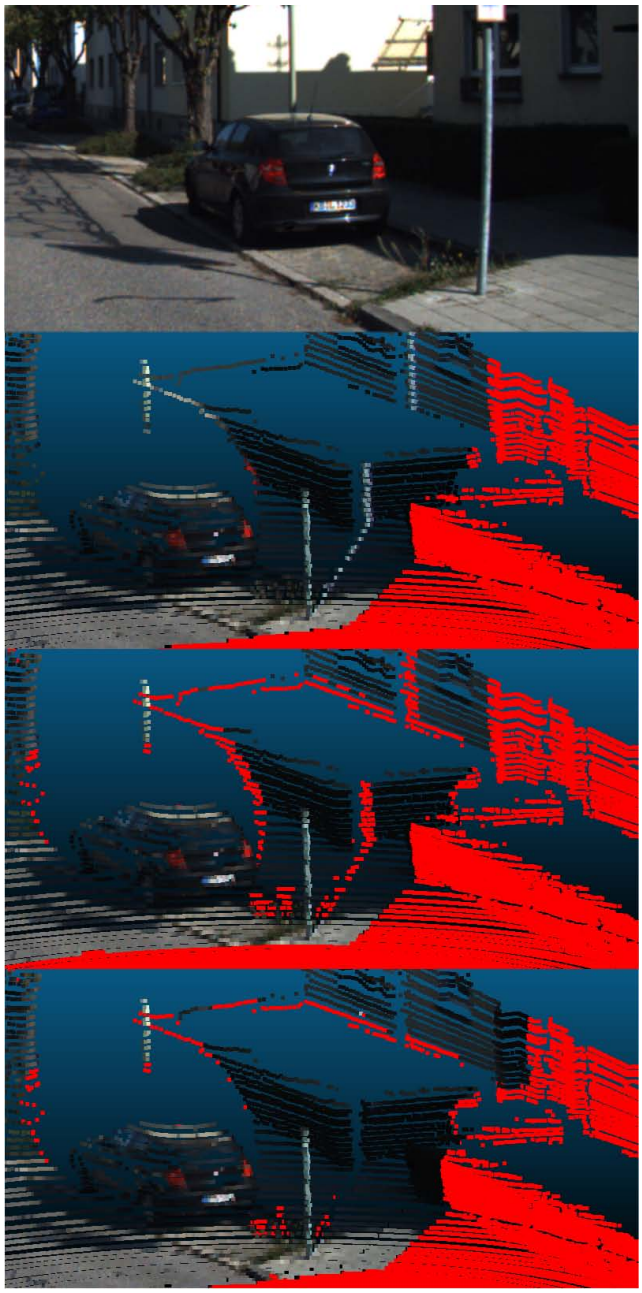


Figure C.6 – From the top down, the first image is the view of the leftmost colour camera of the KITTI vehicle. The second image shows the naive Velodyne colouring approach where foreground points are projected onto the background in occluded regions. In the third image, our occlusion-aware method has not assigned colour values to the occluded regions of the image, thereby removing the errors seen in image 2. Finally, in image 4 both the left and right colour cameras were utilised, resulting in fewer regions where no colour information can be given due to occlusions. In all images red is used to represent regions with no colour information.

Appendix D

3D Rotation Representation

A large number of methods for representing a 3D rotation exist [77], with each method having a range of pros and cons.

The four most common representations are as follows:

D.1 Euler Angles

Euler angles are a compact representation that requires only three values. However, they have the disadvantage of giving singularities when pitch is at 90 or -90 degrees. They also provide a highly non-linear representation of the search space and different combinations of Euler angles may result in the same rotation. An example of the possible issues this representation can cause is best illustrated through a real-world example. Consider the following two rotation matrices A and B:

$$A = \begin{bmatrix} 0.0210 & -0.0035 & 0.9998 \\ -0.9998 & 0.0008 & 0.0210 \\ -0.0008 & -1.0000 & -0.0035 \end{bmatrix}, B = \begin{bmatrix} 0.0174 & -0.0074 & 0.9998 \\ -0.9998 & 0.0035 & 0.0174 \\ -0.0036 & -1.0000 & -0.0073 \end{bmatrix} \quad \text{(D.1)}$$

A is the ground truth for the rotation between the Velodyne and leftmost camera used in the KITTI dataset and B is the result obtained from our calibration. We wish to quantify the difference between these angles in a way that is easy for the reader to interpret. One possible method is to report the difference in the roll, pitch and yaw of the two angles. Performing this action yields:

$$A_{rpy} = \begin{bmatrix} -9.6^\circ & -88.8^\circ & 99.5^\circ \end{bmatrix}, B_{rpy} = \begin{bmatrix} 23.0^\circ & -88.9^\circ & 67.2^\circ \end{bmatrix} \tag{D.2}$$

$$|A_{rpy} - B_{rpy}| = \begin{bmatrix} 32.6^\circ & 0.1^\circ & 32.3^\circ \end{bmatrix} \tag{D.3}$$

This gives extremely large differences in the roll and yaw of the two rotations. Given only this difference, a reader would be forced to conclude that the method had performed poorly, when in fact the large difference has been caused due to pitch values near the singularity at -90 degrees. To prevent this issue we must first find the difference between the two rotation matrices AB^{-1} then convert this to Euler angles

$$|(AB^{-1})_{rpy}| = \begin{bmatrix} 0.3^\circ & 0.2^\circ & 0.6^\circ \end{bmatrix} \tag{D.4}$$

This gives a more accurate representation of the actual error in the method.

D.2 Rotation Matrix

The rotation matrix is an intuitive representation. It is also the only rotation system that does not have multiple standard representations. However, as a 3D rotation can be represented by just three numbers, the nine-element rotation matrix contains a large amount of redundancy with significant constraints placed upon its elements. These constraints make it an exceptionally poor choice for utilising in optimisation, due to the need to ensure a matrix is consistent before it can be evaluated.

D.3 Quaternions

Quaternions use a four-number representation of the angles. However, for a quaternion Q and constant c, Q = cQ = -Q. This allows the quaternion to be represented by three elements by first normalising the Quaternion and ensuring its first element is positive. This means that the first element is no longer required as it can be obtained if the other three elements are known. This does have the disadvantage however of constraining the remaining elements to $||[Q2,Q3,Q4]|| < 1$. One advantage of the quaternions is that for small differences in angle, linear interpolation provides an intuitive and smooth angle.

D.4 Angle-Axis

Like the quaternion this gives a four-number representation of the angle, and like the quaternion it can also be converted to a three-number representation. First, the angle is constrained to be positive and the axis vector is normalised. Then the axis is multiplied by the angle. Unlike the quaternions, no restrictions are placed on the values that the elements of the vector can obtain and while there is the issue of wrap-around for rotations greater than 2π, it is often the best candidate for optimisation. The axis form is useful in hand-eye calibration as it is in a form directly usable by the Kabsch algorithm. It also has the advantage that all points on a rigid body will rotate by the same angle, which makes the angle element useful in aligning timing offsets. For these reasons it is our rotational representation of choice, when appropriate, during this thesis.

Appendix E

Finite Differences

During the approximation of some variance calculations the 1st or 2nd derivative of a function are often required. These functions, however, are commonly complex algorithms with no simple analytical form and thus no tractable way of finding an exact differential. Because of this we make use of the central difference formation [48] which states that for a function f and variable x:

$$f'(x) \approx \frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} \tag{E.1}$$

with error $O(h^2)$.

This is trivial to extend to the case of the second differential via:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{E.2}$$

and second order partial differentials:

$$f_{xy}(x,y) \approx \frac{f(x+\frac{h}{2}, y+\frac{k}{2}) - f(x+\frac{h}{2}, y-\frac{k}{2}) - f(x-\frac{h}{2}, y+\frac{k}{2}) + f(x-\frac{h}{2}, y-\frac{k}{2})}{h^2} \tag{E.3}$$

The choice of h is typically problem specific, however, as a rule of thumb we give it the smallest value for which Equation E.1 is unlikely to be detrimentally affected by the limited precision of the computer's numeric representations.

Appendix F

Performance of Variance Approximations

During this thesis a range of methods for approximating the variance of functions have been utilised. In this appendix we will demonstrate the performance of the Monte Carlo and delta method. These two techniques were chosen, as unlike the other approaches they can be applied to 'black box' situations where the inputs are known, but the process via which the outputs are generated is either unknown, or cannot be represented by a simple analytical relationship. This situation is both challenging and frequently encountered in the field of mobile robotics. In our implementation the delta method uses finite differences in calculating the derivatives to allow it to operate on any function.

The first experiment examines the performance of the Monte Carlo approach on the simple function $y = 10x + 5$. While trivial in nature, it serves to highlight the trade-offs this method exhibits in terms of accuracy and runtime. The experiment was run using a range of samples (2^1 to 2^{20}) with the accuracy and runtime recorded. The input value and standard deviation were both selected from a uniform random distribution between 0 and 1, and the experiment was repeated 1000 times. The results are show in Figure F.1.

As this simple experiment shows, the Monte Carlo method's runtime is proportional

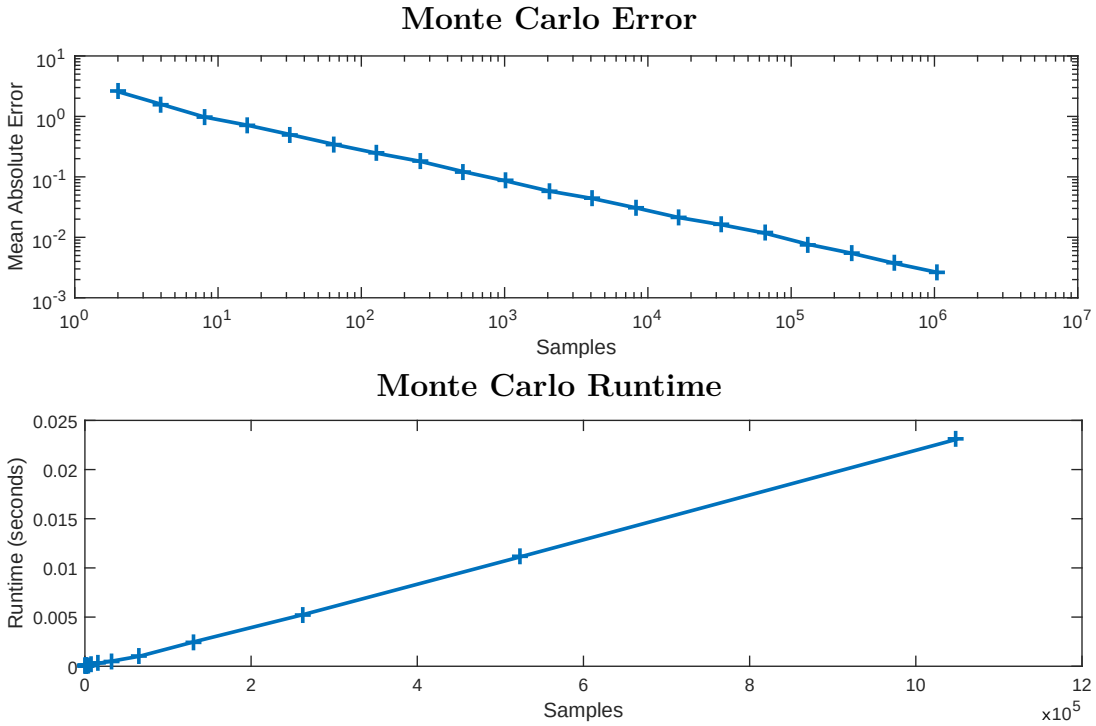


Figure F.1 – The accuracy and runtime of the Monte Carlo method using a Matlab implementation on the function $y = 10x + 5$

	Delta Method	Monte Carlo (2 Samples)	Monte Carlo (2^{20} Samples)
Runtime (seconds)	1.08×10^{-6}	3.38×10^{-5}	2.31×10^{-2}
Mean Absolute Error	4.20×10^{-12}	2.57	2.60×10^{-3}

Table F.1 – Performance of the delta and Monte Carlo Methods on the function $y = 10x + 5$

to the number of samples, while its accuracy is inversely proportional to the square-root of the number of samples. This means care must be taken when selecting how many samples to use with the method so that the desired accuracy is achieved in a practical time frame. The same experiment was conducted using the delta method and compared to the Monte Carlo method in Table F.1.

In this situation the delta method outperforms the Monte Carlo method by a wide margin. However, this is a special case. As the function has a single scalar variable only 2 evaluations are necessary, allowing the fast runtime. The exceptionally low error is also due to the nature of the function. As the equation only makes linear

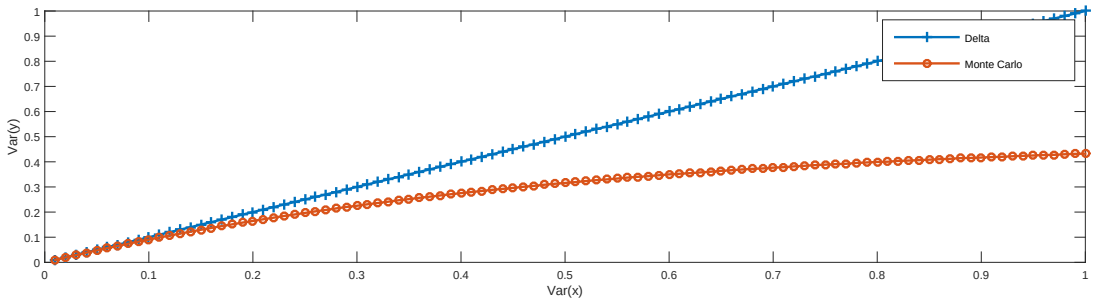


Figure F.2 – Estimated variance of y for given variance of x in the function $y = sin(x)$ at $x = 0$

transformations to x, the linear assumptions made by the delta method are exactly correct. This means the only error present is due to the limited precision of the computer's numeric representation.

The above experiment highlighted the strengths of the delta method, however the method also has significant limitations as the following experiment demonstrates. The function $y = sin(x)$ was taken and the mean of x set to 0. The Monte Carlo and delta method were then used to estimate the variance of y for given variances of x. In this experiment 100000 samples were used for the Monte Carlo Method and the results are shown in Figure F.2.

At $x = 0$ the derivative of $sin(x)$ is 1. This means that in this case the delta method will give the approximation that the variance in y is equal to the variance in x. For small variance values the small angle approximation ensures that this is a reasonable approximation and both methods give similar variance estimates. However, as the variance increases the accuracy of this approximation quickly degrades due to the non-linear relationship between x and y. This means that in this case, when x's variance is 1 the delta method's estimate for the variance of y is more then double the true value. For most non-linear functions this trend of the estimates quality degrading as variance increases holds and must be kept in mind when applying the delta method.

The experiments above demonstrate the performance and some of the potential pitfalls of the Monte Carlo and delta method. The merits of the other methods mentioned in Section 4.2 in terms of performance and application however also bear brief discussion.

Exact Covariance Calculation This method, by definition will always be the most accurate and should typically also be the fastest. However, it can only be applied when the operation modifying the inputs can be stated in an analytically tractable manner, its effect on the variance is well understood and the resulting distribution is of an appropriate form. These requirements prevent its use on the vast majority of practical problems.

Approximate Analytical Variance The speed and accuracy of this method depends on the approximations made. If a linear model is fitted it will behave in a near identical manner to the delta method. The downside of this method is it must be tailored to every situation.

Bootstrapping This method is only applicable when a large number of samples have been gathered from a dataset. It shares many similarities with the Monte Carlo method and has similar trade-offs between performance and runtime.

Appendix G

Outliers

In the field of robust statistics there are several popular methods for dealing with outliers. While we saw this topic as beyond the scope of this thesis, we nevertheless required a method to prevent outliers formed from inaccurate variance estimates from adversely impacting our method. Due to this need, we implemented and examined the range of simple estimation strategies presented below.

G.1 Median

By far the most common method, the value is found by taking the central element after sorting. It has the advantage of being simple and robust to up to 50% of the data being outliers. The disadvantage however, is that it ignores a large amount of information provided by the data in the form of the exact value of the points. This makes it very inefficient, requiring a large amount of additional data points to achieve the same accuracy as taking the mean.

G.2 Threshold

A simple strategy in which all data that falls outside the defined limits is discarded and the mean of the result taken. This method works well when it is known with a large degree of certainty what error any readings will take. This has the downside that it generally needs to be tuned to fit any change in situation. The method also has the disadvantage in that it is the only method presented here that does not make use of all of the data when calculating its output.

G.3 Trimmed means

In trimmed means (also known as truncated means) the data is first sorted and a predefined percentage of the data is removed or trimmed. Discarding no data will result in it giving the mean, while discarding all but 1 point will give the median function. Because of this it can be thought of as a balance between the two. As the amount of discarded data increases so does its robustness, at the expense of its efficiency. This makes the metric useful in cases where far less than 50% of the data are outliers and the median is too extreme.

G.4 Heavy-tailed distributions

This encompasses any method where, while the weighting given to distant points is a strictly increasing function, its gradient is less than that of the standard mean. For example, norms in the range 0 to 2 fit in here. These methods tend to make good use of all the data, however, unlike the other approaches they will suffer if the outliers have a consistent bias to their values.

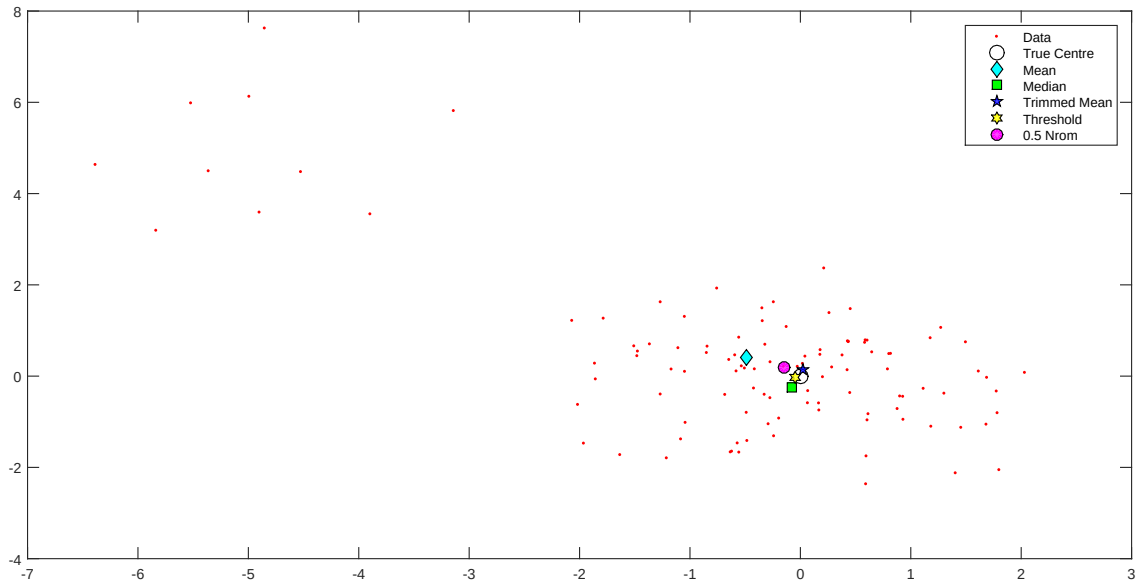


Figure G.1 – The centre found by various robust mean methods on 100 points with Gaussian noise plus 10 outliers points.

G.5 Optimisation

One of the most significant issues when deciding on a robust metric is how it impacts the smoothness and convexity of the search space. Consider for example a sample of 2D positions made from sampling a Gaussian distribution with a mean of [0,0] and standard deviation of [1,1] 100 times. Outliers are then added by sampling a second Gaussian distribution with a mean of [5,5] and standard deviation of [1,1] ten times. Each metric is used to find the centre of the data by minimising the distance to points and the result is plotted in Figure G.1.

In this case all the methods examined perform significantly better than the standard mean. However, to solve this optimisation a global optimiser had to be used. The reason for this can be seen by examining the search space shown in Figure G.2.

While the mean, trimmed means, and 0.5 norm all remain smooth objective functions with a single minima, the median and threshold now contain large numbers of local minima. These local minima make the median and threshold methods poor choices

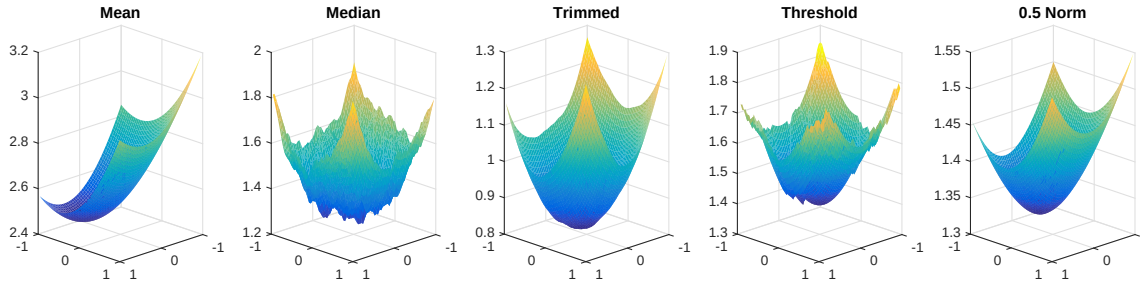


Figure G.2 – The function values over the search space for each metric. Note the jagged shape formed by the median and threshold metric.

when optimisation is required, as while these local minima will disappear as the number of data points approaches infinity, for any practical dataset this will remain an issue.

While it is also theoretically possible to construct problems in which the trimmed means will have local minima (for example discarding all but 1 data point yields the median), the situations are unlikely to occur if sufficient quantities of real data are used. Thus, because of its resilience to biased outliers, simple optimisation and the intuitive nature of the rejection percentage parameter, we have made use of the trimmed means in our outlier rejection process.

In our work we found this simple metric to give good results. We acknowledge that it will not be as efficient as an M-estimator, however, we saw the detailed analysis of all possible M-estimators and their suitability as being beyond the scope of this thesis. Instead, we will note that no part of our method relies on the exact nature of the outlier rejection and any suitable metric could be substituted without impacting the other steps in the approach.

In our work, we have set the trimmed means to reject 25% of the data. However, the method is highly insensitive to the exact value used here as long as it is sufficiently large to remove all outliers. In our testing, any value in the range of 5% to 90% gave reasonable performance, though, as the amount of data rejected increases, the variance of the output will increase due to less data being used in the estimation.