School of Science and Engineering
Department of Mechanical Engineering

A Multisensor SLAM for Dense Maps of Large Scale Environments under Poor Lighting Conditions

Jared R Le Cras

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

December 2012

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: …….

Date: ……...

Abstract

This thesis describes the development and implementation of a multisensor large scale autonomous mapping system for surveying tasks in underground mines. The hazardous nature of the underground mining industry has resulted in a push towards autonomous solutions to the most dangerous operations, including surveying tasks. Many existing autonomous mapping techniques rely on approaches to the Simultaneous Localization and Mapping (SLAM) problem which are not suited to the extreme characteristics of active underground mining environments. Our proposed multisensor system has been designed from the outset to address the unique challenges associated with underground SLAM. The robustness, self-containment and portability of the system maximize the potential applications.

The multisensor mapping solution proposed as a result of this work is based on a fusion of omnidirectional bearing-only vision-based localization and 3D laser point cloud registration. By combining these two SLAM techniques it is possible to achieve some of the advantages of both approaches – the real-time attributes of vision-based SLAM and the dense, high precision maps obtained through 3D lasers. The result is a viable autonomous mapping solution suitable for application in challenging underground mining environments.

A further improvement to the robustness of the proposed multisensor SLAM system is a consequence of incorporating colour information into vision-based localization. Underground mining environments are often dominated by dynamic sources of illumination which can cause inconsistent feature motion during localization. Colour information is utilized to identify and remove features resulting from illumination artefacts and to improve the monochrome based feature matching between frames.

Finally, the proposed multisensor mapping system is implemented and evaluated in both above ground and underground scenarios. The resulting large scale maps contained a maximum offset error of ±30mm for mapping tasks with lengths over 100m.

Acknowledgements

To my amazing wife Hannah whom I fell head over heels for, and married, during the duration of this work, thank you so much for your understanding and loving support!

To my Mother, Father and Sister, for their support and encouragement throughout my life, thank you for being behind me in everything that I pursued.

To my extended family – the Tebbits, thank you for your backing and support when I faced the toughest stretches of this work.

To my supervisor Jon, you have been truly inspirational during this journey, your passion for this discipline has been infectious and it has been a joy to complete this work under your guidance.

To my co-supervisor Andrew, your understanding of the mining industry and decades of experience were invaluable. Thank you for your hospitality in Kalgoorlie and for handling relations with the notoriously fickle mining companies.

To the staff of the Mechatronics department during my time at Curtin – Brad, Euan and Masood. You made mechatronics an exciting and challenging field and no doubt helped shape me into the engineer I am today.

To Jeff and Dave, your technical expertise and support were greatly appreciated. You made everything relating to hardware just that little bit simpler – thank you.

Finally, I have to say thank you to my Lord and Saviour Jesus Christ. It is because of Him that I know there is so much more to this life than academia. May this work be for His glory.

Contents

List of Figures

List of Tables

Glossary

CBM	‘Colour Based Matching’. A technique developed in this work for the identification of incorrect visual feature matches.
Chromaticity Distortion	The difference between two colours, independent of their brightness, found using the chromaticity colour model.
Colour Angles	A colour model that plots RGB colours as vectors in RGB space for comparison.
DoF	‘Degrees of Freedom’. Identifies the number of independent variables affecting the range of state in which a system may exist.
EKF	‘Extended Kalman Filter’ A version of the standard Kalman Filter modified to approximate nonlinear systems.
FAST Corner Detection	‘Features from Accelerated Segment Test Corner Detection’. An image corner detector used for the extraction of visual features.
Fitness Score	A measure of the quality of the alignment of two point clouds, often determined by finding the mean square error of the nearest neighbour offsets.
Fps	‘Frames per second’. A measure of the image production rate of a camera, or the work rate of an image processing algorithm.
Gaussian	Referring to ‘Gaussian Distribution’. A theoretical distribution with known mean and variance represented as a symmetrical bell shaped graph.
GNSS	Global Navigational Satellite System. A global navigation system using positioning satellites and sometimes supplemented with inertial measurements.

GPS	‘Global Positioning System’. A series of space based satellites used for the communication of position and time.
Ground Truth	The known trajectory of an object.
HSV	‘Hue Saturation Value’. A cylindrical coordinate representation of RGB colour. It presents hue on a circular colour chart and then separately defines saturation and ‘darkness’ values.
ICP	‘Iterative Closest Point’. A point cloud registration algorithm which performs iterative transformations to minimize nearest neighbour offset.
IMU	‘Inertial Measurement Unit’. A device which uses a combination of accelerometers, gyroscopes and magnetometers to provide velocity and orientation information.
Jacobian	Referring to ‘Jacobain Matrix’. A matrix of the first order partial derivatives of one vector with respect to another.
Kalman Filter	An algorithm which operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state
MATLAB	‘Matrix Laboratory’. A programming language for numerical computing, with an emphasis on matrix mathematics.
MCL	‘Monte Carlo Localization’. The implementation of a particle filter in a localization task.
O	‘Big O Notation’. In computer science the notation is used to measure an algorithm’s response to changes in input size.
Particle Filter	A discrete estimation technique for the approximation of continuous Bayesian probabilities.
PCL	‘Point Cloud Library’. An open source software project for the processing of 2D and 3D point clouds.

Point Cloud	A set of points in a three dimensional coordinate system.
Quaternion	A technique for representing orientations and rotations in 3D space.
RANSAC	‘Random Sample Consensus’. An iterative method for the approximation of a mathematical model when outlier data is present.
RBPF	‘Rao-Blackwellized Particle Filter’. A particle filter which samples over a subset of the state variables, significantly reducing the number of particles required.
Registration	The process of correctly aligning and concatenating two point clouds.
RGB	‘Red Green Blue’. A colour model which represents a colour via three 8-bit numbers, one each for red, green and blue.
RGB-D	Referring to ‘RGB-D camera’. A device which combines an RGB camera with a structured light or time of flight camera to produce per pixel depth information.
SFM	‘Structure From Motion’. The process of finding the three dimensional structure of an object from motion based input.
SFR	‘Shadow Feature Removal’. A technique developed in this work for the identification of visual features produced by shadows.
SIFT	‘Scale Invariant Feature Transform’. An algorithm for the detection and description of visual features.
SLAM	‘Simultaneous Localization and Mapping’. The task of simultaneously producing a map of an environment while localizing oneself within that map.
TOF	‘Time of Flight’. A technique which measures the time required for a wave to travel through a medium. Often used to determine the distance to reflective objects.

Voxel Reduction	A technique for the simplification of point clouds by averaging points within individual cells of a Euclidean grid.

Chapter 1

Introduction

Despite recent advances in safety protocols, active underground mines are still one of the most dangerous environments frequented by humans. The Western Australian mining industry has the highest mortality rate of any industry in the state, with 50 fatal workplace related incidents in above ground and underground operations reported in the last decade [1]. The inherent risk of underground mining has led to a significant drive towards the implementation of autonomous solutions to the most dangerous tasks. One such task is the surveying of new and existing tunnels to measure the progress of mining and to verify the mine's structural integrity. This potential application has led to the development of our multisensor SLAM system for the autonomous mapping of large scale environments.

The autonomous production of large scale maps in above ground applications is a research topic which has seen many successful implementations in recent years. However, underground environments contain unique and challenging characteristics that make the direct application of many existing above ground techniques impossible. These characteristics include complex geometry (requiring movement in six degrees of freedom), high levels of electromagnetic shielding (preventing transmission of data with the surface), harsh environmental conditions (requiring robust specialized and certified equipment) and poor lighting conditions (often produced entirely by dynamic sources).

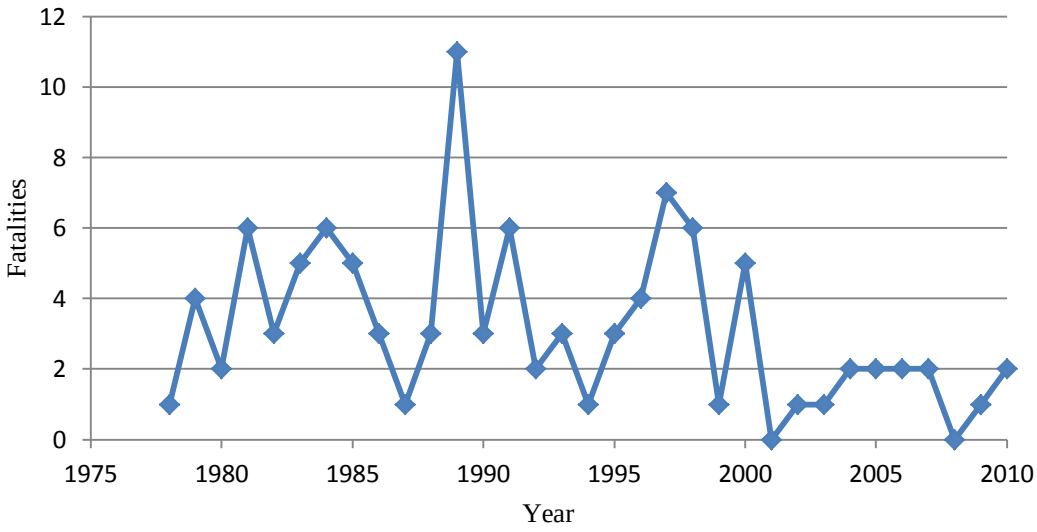


Figure 1.1 – Fatalities from workplace related incidents in the underground mining industry of Western Australia [2].

Our proposed solution to autonomous large scale mapping in active underground mining environments is a hybrid fusion of omnidirectional bearing-only vision-based localization and 3D laser point cloud registration. Combining the high precision nature of laser mapping with the real-time, six degree of freedom localization ability of vision-based SLAM results in a system capable of overcoming the previously mentioned challenges.

1.1 Self-Containment and Portability

Large scale 3D mapping has been successfully accomplished by many research groups through the implementation of complex, specialized systems. A system may be deemed specialized due to a number of design decisions such as the use of externally instrumented localization systems (e.g. GPS), external physical odometry readings, *a priori* information or a simplified motion model that removes the comprehension of higher degrees of freedom. These design choices, while appropriate for the given domains, are not well suited to underground mining applications.

The traditional commercial technique for registering multiple 3D laser scans uses artificial markers or targets placed throughout the scene. These markers are used in

post-processing to semi-automatically align the individual scans. There are many shortcomings to this approach: the scene must be artificially manipulated before scanning can commence, all registration must be completed in a post-processing phase and there is significant manual intervention required. The resulting process is slow and labour intensive.

GNSS (Global Navigational Satellite System including GPS) is often used in outdoor mobile mapping systems to provide an initial pose estimate for fine scale registration [3]. However, GPS based systems are not suitable for indoor or underground applications and can suffer from reliability problems in dense urban areas due to obstruction of satellite signals. The primary application for our own work on large scale 3D mapping is the underground mining industry [4] where there is no access to GPS positioning.

The typical response to the absence of a GPS signal in indoor environments is to make use of physical odometry readings to provide a rough pose estimate prior to fine scale registration [5]. These odometry readings are usually supplied by encoders attached to the wheels of the host platform and are prone to considerable error in environments with significant slip (e.g. gravel, sand or ice). Odometry can be used in conjunction with an Inertial Measurement Unit (IMU) or continuous scan registration to improve robustness to odometry errors and aid in the extrapolation of odometry readings to a six degree of freedom environment [6], [7]. Reliance on physical odometry precludes a truly portable and flexible mapping system due to dependence on the instrumentation of the vehicle itself. It is not possible to produce a single vehicle that is perfectly suited to all potential 3D mapping environments; therefore a self-contained solution is desirable. A specialized vehicle would also require unique certification upon every entry to an active mine site and would greatly increase deployment costs. A self-contained and portable mapping system can be deployed on any existing vehicle, including a mine site certified vehicle, without integration into the vehicle's on-board systems.

Inertial Measurement Units are self-contained; however, they rely on external odometry or magnetometers to prevent significant inherent drift. Magnetometers are unreliable underground due to the presence of large volumes of metallic ore that

interfere with the Earth's magnetic field. As mentioned before, it is highly desirable to avoid physical odometry, so IMU drift cannot be compensated. The sensor is therefore rendered inappropriate for our application.

One method to remove the dependency on physical odometry is to instead use *a priori* information. Früh and Zakhor [8] use aerial photographs or Digital Surface Models to correct 2D laser based localization over large distances. Again, this is only possible in uncluttered outdoor environments and would not be feasible for the continually evolving profile of an underground mine where *a priori* information may not be available or accurate.

The mapping of indoor environments can often be simplified to a three degree of freedom problem due to the flat, level nature of many man-made environments. By removing roll, pitch and vertical translation, the six degree of freedom localization problem is reduced to three degrees of freedom. This allows localization to be produced by simple 2D laser scan registration where the reduced dimensionality considerably simplifies the computations required. The effectiveness of this approach is demonstrated by Thrun *et al.* [9] and Hähnel *et al.* [10]. We are interested in full six degree of freedom localization, so that the results can be extended to geometrically complex non-flat environments.

There are several key motivations for producing a hybrid system that can overcome the unique challenges associated with underground mining environments without resorting to simplifications. Maintaining a six degree of freedom motion model allows the mapping of geometrically complex environments. Removing the dependency on external equipment such as global positioning system (GPS) satellites or targeting systems allows the device to be operated in electromagnetically shielded and unmonitored environments. Eliminating physical odometry measurements removes the dependence on specialized mobile platforms or vehicles which would require unique certification for every mine site application. And finally, avoiding the use of *a priori* information allows the mapping of completely unknown environments. The system should also be capable of operating in poor lighting to ensure the widest range of applicable environments. The production of survey quality results is essential, as the goal application for this system is surveying tasks.

Reflectorless surveying has an accuracy of ±6mm, therefore the highly accurate registration (alignment) of dense point clouds is desirable in order to approach these tolerances.

1.2 Multisensor Solution

Previous successfully implemented examples of large scale 3D mapping have relied on GPS, physical odometry, *a priori* information or a simplified approximation. The autonomous mapping solution presented in this work demonstrates a self-contained, portable multisensor SLAM system capable of large scale 3D mapping based on the use of a 3D laser and omnidirectional camera. Here, large scale mapping refers to the mapping of any environment requiring multiple 3D laser scans for satisfactory coverage. The fusion of real-time bearing-only vision-based localization and intermittent, single point of origin, 3D laser scans has resulted in a system that can accurately perform localization while building a detailed 3D map of the surrounding environment. Multiple 2D laser scanners could also be considered as an appropriate approach to localization; however, vision-based localization was selected due to the ability to provide real-time 3D information and long term feature tracking.

The map building process begins with an initial stationary 3D laser scan. This is followed by motion tracking during transit to the next scan location performed through the implementation of a six degree of freedom bearing-only vision-based localization algorithm. The bearing-only localization is enhanced by the incorporation of range information at the discrete 3D scan positions. This enables correct scaling of the resulting feature map, and improves the general performance of the localization algorithm. Once the target location has been reached and the subsequent 3D scan produced, an iterative closest point (ICP) algorithm [11] is applied to register the new laser data to the existing map. The resulting transformation then compensates the trajectory estimate of the vehicle to accurately reflect the real world location. This process is repeated to iteratively build a large scale 3D map of the environment.

Vision-based localization is shown to be prone to drift caused by the dynamic sources of illumination frequently found in underground mining environments. A technique has therefore been developed to supplement the multisensor SLAM system by filtering visual features during the localization stage of map building. By removing the sources of drift caused by illumination artefacts, localization estimation is significantly improved.

1.3 Chapter Summary

The contents of the chapters comprising this thesis are as follows:

Chapter 2 Background

The background chapter begins with an overview of the history and development of the Simultaneous Localization and Mapping (SLAM) problem. The most popular solutions to the SLAM problem are then presented. Finally, vision-based bearing-only SLAM is discussed including filter selection and feature extraction techniques.

Chapter 3 SLAM Implementations

Chapter 3 examines current implementations of SLAM systems categorized by the desired application. These applications range from basic 2D mapping to survey quality 3D mapping.

Chapter 4 Implementing Bearing-Only SLAM for the Multisensor System

Bearing-only SLAM is investigated as an approach to modular mapping using a single camera only. The associated camera models are discussed, ranging from pinhole to omnidirectional, and the techniques used for the measurement and tracking of visual features are discussed. The chapter finishes with a summary of the advantages of combining bearing-only SLAM with intermittent metric depth data, resulting in a hybrid system.

Chapter 5 Large Scale Mapping from Point Clouds

Here the acquisition and registration of laser point clouds is examined. A summary of existing sensors capable of point cloud capture is presented, followed by a detailed

analysis of the techniques used for registration and map building from point clouds. Finally, the benefits of the integration of real-time localization are investigated, resulting in the continued development of a multisensor system as well as a novel registration technique for laser scans with large offsets.

Chapter 6 Hybrid Integration of Vision-Based SLAM and Point Clouds

Chapter 6 formalizes the proposed multisensor SLAM system by investigating the beneficial interactions of the omnidirectional camera and 3D laser scanner. These benefits include localization scaling using depth information, the provision of initial pose estimates for registration and path correction via long distance cloud correlation.

Chapter 7 Vision-Based SLAM under Dynamic Illumination

This chapter introduces the use of colour information to reduce the detrimental effects of dynamic illumination on vision-based localization. Two novel techniques are derived to improve robustness – Shadow Feature Removal (SFR) and Colour Based Matching (CBM). The techniques are then investigated under a range of experimental scenarios. Testing begins with computer generated scenes and evolves to real world deployment. Finally, improvements to modularity are discussed and results are evaluated.

Chapter 8 Multisensor SLAM Results

The results chapter presents the performance of the multisensor system firstly in an above ground built environment, then in an active underground mining environment. In both scenarios large scale maps are successfully constructed and then carefully scrutinized.

Chapter 9 Conclusions

Here the work described in the thesis is summarized and conclusions are drawn. The main contributions of the thesis are then outlined and recommendations for the directions of future work are made.

1.4 Published Work

The following publications on the work presented in this thesis are also available:

Le Cras, J., Paxman, J., Saracik, B. 2013. Improving Robustness of Vision Based Localization under Dynamic Illumination. Recent Advances in Robotics and Automation, Springer series on Studies in Computational Intelligence. Vol. 480.

Le Cras, J., Paxman, J. 2012. A Modular Hybrid SLAM for the 3D Mapping of Large Scale Environments. In Proceedings of the 12th International Conference on Control, Automation, Robotics and Vision (ICARCV). pp. 1036-1041.

Le Cras, J., Paxman, J., & Saracik, B. 2011. Vision based localization under dynamic illumination. In Proceedings of the 5th International Conference on Automation, Robotics and Applications 2011 (ICARA2011). pp. 453-458.

The following publication is based on related work:

Le Cras, J., Paxman, J., Saracik, B. 2009. An Inspection and Surveying System for Vertical Shafts. In Proceedings of the Australasian Conference on Robotics and Automation 2009 (ACRA2009).

Chapter 2

Background

2.1 Introduction

The Simultaneous Localization and Mapping problem can be summarized as follows: Can an autonomous vehicle be placed in an environment with no *a priori* knowledge and incrementally build a map of that environment while simultaneously determining its current location within that map? Both the localization and mapping elements of SLAM can be successfully determined if the other is known. It is possible to calculate localization if sensed landmarks can be associated with an *a priori* map and a map can be built from sensed landmarks if accurate pose is provided by an independent source. However, if neither the pose nor the map is known, the problem is considerably more complex.

The joint solution of localization and mapping came to the forefront of robotics research at the International Conference on Robotics and Automation in 1986 (ICRA86). Several researchers including Peter Cheeseman, Jim Crowley, Raja Chatila, Olivier Faugeras and Hugh Durrant-Whyte examined the application of estimation-theoretic methods to mapping and localization problems. In 1990, a paper by Smith, Self and Cheeseman demonstrated that estimates of landmarks observed by a mobile robot moving through an unknown environment were correlated with one another due to the common error in vehicle location estimation [12]. Therefore, a

solution could be formed by combining the vehicle pose with the recorded landmark positions in a single state. Unfortunately it also meant that any filter based estimator would require a computation time that scaled with the square of the number of landmarks. It was later shown that the SLAM problem was convergent [13]. Assuming landmarks are stationary; it can be shown that at the theoretical limit, relative localization accuracy becomes equal to the localization accuracy achievable with an *a priori* map.

2.2 Localization and Mapping Techniques

In the time since the definition and proof of convergence of SLAM, many techniques have been developed to solve real world SLAM problems. The following sections will summarize these techniques, starting with the earliest Bayesian estimation techniques and leading up to a review of the state of the art. For the full derivations of each of these techniques please see the comprehensive survey by Thrun *et al.* [14].

2.2.1 Bayesian Estimation

Bayesian estimation is an approach to optimal state estimation involving the minimization of the posterior expected value of a cost function. The approach is therefore well suited to the probabilistic nature of the simultaneous localization and mapping problem. The goal of each step in the Bayesian estimation is to evaluate the current state of the system (x_k), based on all previous observations ($\mathrm{z}_{0:k}$), i.e.

$$p(\mathrm{x}_k|z_{0:k}). \tag{1}$$

The Bayesian estimation of the SLAM problem makes the assumption that the process being observed is Markovian. To be Markovian, the current measurement must be independent from previous measurements, given the current state, i.e.

$$p(\mathrm{z}_k|\mathrm{z}_{0:k-1}, \mathrm{x}_k) = p(\mathrm{z}_k|\mathrm{x}_k). \tag{2}$$

For the Markov assumption to be valid in a SLAM problem, the environment must either be static, or the state of every moving object must be included in the estimator (such as in the work by Kundu *et al.* [15]). The current state distribution can then be determined as follows.

$$\begin{aligned} p(\mathrm{x}_k|\mathrm{z}_{0:k}) &= \frac{p(\mathrm{z}_k|\mathrm{x}_k)p(\mathrm{x}_k|\mathrm{z}_{0:k-1})}{p(\mathrm{z}_k|\mathrm{z}_{0:k-1})} \\ &= \frac{p(\mathrm{z}_k|\mathrm{x}_k)\int p(\mathrm{x}_k|\mathrm{x}_{k-1})p(\mathrm{x}_{k-1}|\mathrm{z}_{0:k-1})d\mathrm{x}_{k-1}}{p(\mathrm{z}_k|\mathrm{z}_{0:k-1})} \\ &= \frac{p(\mathrm{z}_k|\mathrm{x}_k)\int p(\mathrm{x}_k|\mathrm{x}_{k-1})p(\mathrm{x}_{k-1}|\mathrm{z}_{0:k-1})d\mathrm{x}_k}{\int p(\mathrm{z}_k|\mathrm{z}_{0:k-1},\mathrm{x}_k)\,p(\mathrm{x}_k|\mathrm{z}_{0:k-1})d\mathrm{x}_k} \\ &= \eta p(\mathrm{z}_k|\mathrm{x}_k)\int p(\mathrm{x}_k|\mathrm{x}_{k-1})p(\mathrm{x}_{k-1}|\mathrm{z}_{0:k-1})d\mathrm{x}_{k-1} \end{aligned} \quad (3)$$

Where $\eta = \frac{1}{\int p(\mathrm{z}_k|\mathrm{z}_{0:k-1},\mathrm{x}_k)p(\mathrm{x}_k|\mathrm{z}_{0:k-1})d\mathrm{x}_k}$ is a normalization factor to ensure correct probability distribution.

In SLAM problems, the current state is determined in a two stage process. Firstly the prediction step predicts the current state based on the dynamic model of the state (*f*). In SLAM the dynamic model takes the form of a motion model and uses the previous vehicle pose (x_{k-1}) and input controls (u_k) to determine the current state prediction. The prediction stage of equation (3) is contained within the evaluation of the integral term. The second step updates the predicted current state by incorporating the current observations (measurements). The update stage of equation (3) is performed by weighting the predicted current state with the current observations using the normalization factor η.

Bayesian estimation is limited in SLAM applications due to the need to integrate over the entire state space. The state space in a fully specified SLAM problem contains the vehicle (agent) pose history ($x_{0:k-1}$) and all observed landmarks ($\mathrm{m}_{1:N}$). In the following example state space, the current pose is predicted by the motion model (*f*).

$$\mathrm{x}_k = \begin{bmatrix} f(x_{k-1}, \mathrm{u}_k) \\ x_{k-1} \\ \vdots \\ x_0 \\ \mathrm{m}_1 \\ \vdots \\ \mathrm{m}_N \end{bmatrix} \tag{4}$$

SLAM state spaces may contain hundreds or thousands of vehicle poses and landmarks, making the implementation of a Bayes filter impossible due to the prohibitive processing times. To overcome this issue, approximation techniques are used, examples of which are described in the following sections.

2.2.2 Kalman Filtering

The Kalman filter is an exact representation of the Bayesian estimation in equation (3). Kalman filters are effective as long as the system is linear and the noise is represented with a Gaussian distribution. The system process model is defined as:

$$\mathrm{x}_k = F\mathrm{x}_{k-1} + G\mathrm{u}_{k-1} + \omega_{k-1}. \tag{5}$$

Where F is the system transition matrix, G is the gain of the input control and ω_{k-1} is the zero-mean Gaussian process noise vector with a covariance matrix Q. The observation model for the system is defined as:

$$\mathrm{z}_k = H\mathrm{x}_k + v_k. \tag{6}$$

Where H is the observation matrix and v_k is the zero-mean Gaussian observation noise with variance R.

The prediction step of the Kalman filter uses the process model of the system.

$$\bar{\mathrm{x}}_k = F\mathrm{x}_{k-1} + G\mathrm{u}_{k-1} \tag{7}$$

$$\bar{P}_k = FP_{k-1}F^T + Q \tag{8}$$

Where $\bar{\mathrm{x}}_k$ and $\bar{P}_k$ are the state estimate and covariance respectively, after the k^{th} prediction step but before the observation step.

The update step then combines the prediction with the observation model. The covariance of the process model is denoted P_k.

$$\mathrm{x}_k = \bar{\mathrm{x}}_k + K(\mathrm{z}_k - H\bar{\mathrm{x}}_k) \tag{9}$$

$$P_k = \bar{P}_k - KSK^T \tag{10}$$

Where $\mathrm{z}_k - H\bar{\mathrm{x}}_k$ is referred to as the innovation. The innovation covariance is denoted *S* and is defined below along with the Kalman filter gain (*K*).

$$S = H\bar{P}_k H^T + R \tag{11}$$

$$K = \bar{P}_k H^T S^{-1} \tag{12}$$

2.2.3 Extended Kalman Filter

The Extended Kalman Filter (EKF) has the ability to model nonlinear systems by using a linear approximation of the system. The approximation is determined by linearizing about an estimate of the current state and covariance. The nonlinear process model and observation model are denoted:

$$\mathrm{x}_k = f(\mathrm{x}_{k-1}, \mathrm{u}_{k-1}) + \omega_{k-1} \tag{13}$$

$$\mathrm{z}_k = \mathrm{h}(\mathrm{x}_k) + v_k. \tag{14}$$

The Extended Kalman Filter is similar to the standard Kalman filter in that the three main components are the states (x_k) which include vehicle pose and landmark locations, the controls ($\mathrm{u}_{0:k}$) which contain the input to the system, and the

observations ($z_{0:k}$) consisting of the current landmark measurements. There is also zero-mean Gaussian process noise (ω_{k-1}) with covariance Q and observation noise (v_k) with variance R.

The basic EKF algorithm also consists of two steps. The first step updates the probability distribution function from step k–1 to step k based on the dynamic model of the system (f). This step is referred to as the prediction step and determines the predicted state estimation ($\bar{x}$) and covariance ($\bar{P}$) of the system.

$$\bar{x}_k = f(x_{k-1}, u_{k-1}) \tag{15}$$

$$\bar{P}_k = F P_{k-1} F^T + Q \tag{16}$$

Since the system is nonlinear, the function f used to predict the estimate of the current state cannot be directly applied to predict the covariance. Instead, a matrix F of the partial derivatives of the dynamic model with respect to the state vector is calculated (known as a Jacobian matrix). This Jacobian is calculated at every time step, therefore linearizing the system about the current estimate.

The second step of the EKF algorithm updates the probability distribution function at step k based on the observations made at step k. This step is referred to as the update step and again determines the expected value (x_k) and covariance (P_k) of the state x.

$$x_k = \bar{x}_k + K\left(z_k - h(\bar{x}_k)\right) \tag{17}$$

$$P_k = \bar{P}_k - KSK^T \tag{18}$$

Here z_k are the observations produced at step k. The function h defines the sensor measurement model. Since the system is nonlinear, the covariance update requires the Jacobian matrix H_k which contains the derivatives of the measurement function (h) with respect to the state vector. Finally, S is the covariance of the innovation term ($z_k - h(\bar{x}_k)$) and K is the Kalman filter gain:

$$S = H\bar{P}_k H^T + R \tag{19}$$

$$K_k = \bar{P}_k H^T S^{-1}. \tag{20}$$

Extended Kalman Filters are not the only solution to the modelling of nonlinear SLAM systems. However, they were the first solution and are still the most popular due to their relatively low computation costs. The following sections investigate the most popular alternative techniques.

2.2.4 Particle Filters

Particle filters such as Monte Carlo Localization (MCL) [16] attempt to model a continuous nonlinear state space by representing the probability $p(\mathrm{x}_k|z_{0:k})$, produced by Bayesian estimation, by a set of *m* weighted samples.

$$p(\mathrm{x}_k|\mathrm{z}_{0:k}) \approx \{x^i, w^i\}_{i=1,\dots,m} \tag{21}$$

Where each x^i is a sample of the true state x_k. The values of *w* are the importance factors, which provide weighting to each of the samples. The set of samples therefore define a discrete probability function approximating the true continuous probability.

The initial set of samples is based on the knowledge of the vehicle's initial pose. Generally in SLAM implementations the initial vehicle pose is assumed to be the global origin and is known with absolute certainty. Therefore, the single sample that represents the initial pose will have a weighting value of 1 and all other samples will have a weighting of 0.

The iterative update of the particle filter has three stages. The first stage is to produce the weighted samples, distributed based on the probability of the vehicle pose determined in the previous iteration:

$$x_{k-1}^i \sim p(\mathrm{x}_{k-1}|z_{0:k-1}). \tag{22}$$

The second stage predicts the new pose of each sample based on the expected motion inferred from input controls (u_k). The resulting distribution of samples is referred to as the proposal distribution (q_k).

$$x_k^i \sim p\left(x_k | x_{k-1}^i, \mathrm{u}_k\right) \tag{23}$$

$$q_k \coloneqq p(x_k | x_{k-1}, \mathrm{u}_k) p(\mathrm{x}_{k-1} | z_{0:k-1}) \tag{24}$$

The third and final stage is to update the proposal distribution by weighting the importance factor of each sample based on the current observations. This is then scaled by a constant normalization factor (η).

$$\begin{aligned} w^i &= \eta p\left(\mathrm{z}_k | x_k^i\right) \\ &= \frac{\eta p\left(\mathrm{z}_k | x_k^i\right) p\left(x_k^i | x_{k-1}^i, \mathrm{u}_k\right) p(\mathrm{x}_{k-1} | z_{0:k-1})}{p\left(x_k^i | x_{k-1}^i, \mathrm{u}_k\right) p(\mathrm{x}_{k-1} | z_{0:k-1})} \end{aligned} \tag{25}$$

The importance factors are then normalized so that their sum is 1 such that they define a discrete probability distribution.

Particle filters are a simple and efficient technique for approximating the nonlinear nature of SLAM problems. They can also overcome non-Gaussian noise, a known limitation of the EKF approach. However, there are problems associated with the implementation of the basic particle filter algorithm described here. Firstly, particle filters are unable to estimate posteriors for highly accurate observations. When a feature is observed precisely, none of the samples may be close enough to be correctly weighted. This results in the incorrect removal of the closest samples and subsequent localization degradation [17]. Secondly, large sample sizes with high computational cost are required for the effective approximation of state distributions, making real-time implementations difficult. And finally, basic particle filters do not recover well from unexpected large state changes due to their tendency to assign large importance factors to a small number of particles over extended periods of time.

Modifications to the basic particle filter algorithm have helped alleviate some of these issues, yet generally at the cost of simplicity and processing speed. Thrun and Fox *et al.* present a modified Monte Carlo Localization algorithm called Mixture-MCL [17]. Their modifications to the basic particle filter algorithm improve the robustness to the susceptibilities mentioned earlier by combining a regular MCL with a 'dual' MCL which inverts the regular MCL sampling process. Sim and Griffin overcome the large number of particles needed to approximate large state by using a Rao-Blackwellized Particle Filter (RBPF) [18]. RBPFs only sample over a subset of the state variables, significantly reducing the number of particles required and improving processing speed, allowing large scale implementations.

2.2.5 GraphSLAM

GraphSLAM is a graphical network interpretation and solution to the SLAM problem originally reported by Thrun and Montermerlo [19]. An illustrative scheme demonstrating the graph-based approach to a 2D SLAM problem is shown in Figure 2.1. Vehicle poses (*x*) and landmarks (m) are placed in the graphical environment as nodes. Arcs connecting nodes come in two types: motion arcs and measurement arcs. Motion arcs represent the motion of the vehicle and link any two consecutive vehicle poses. Measurement arcs represent sensor measurements and link vehicle poses to landmarks. Motion arcs can be thought of as 'springs' between two vehicle poses, the spring value is determined through the motion arc constraint:

$$[x_k - f(\mathrm{u}_k, x_{k-1})]^T R_k^{-1} [x_k - f(\mathrm{u}_k, x_{k-1})]. \tag{26}$$

Where *f* is the motion model of the vehicle and u_k represents the input controls at the current step. The noise in the vehicle movement is represented by the residual uncertainty *R*. Measurement arcs are also considered to be springs. The spring value is determined through the measurement arc constraint:

$$\left[\mathrm{z}_k^i - h(x_k, \mathrm{m}_j, i)\right]^T Q_k^{-1} \left[\mathrm{z}_k^i - h(x_k, \mathrm{m}_j, i)\right]. \tag{27}$$

Where h is the sensor measurement model and z_k^i is the observation measurement of the i^{th} feature at step k. The covariance of the measurement noise is denoted as Q.

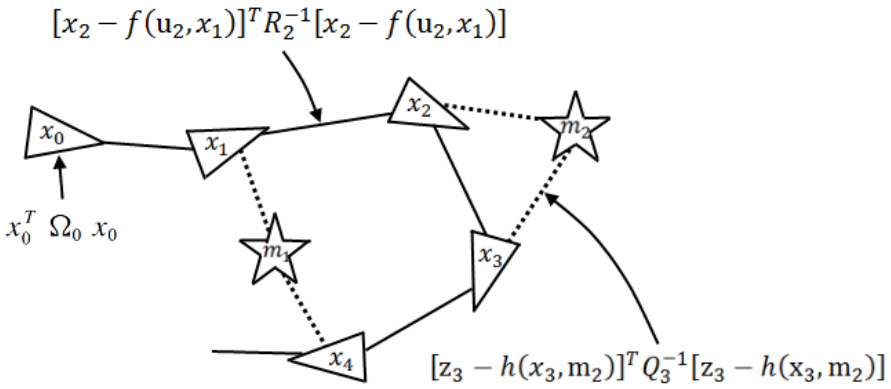
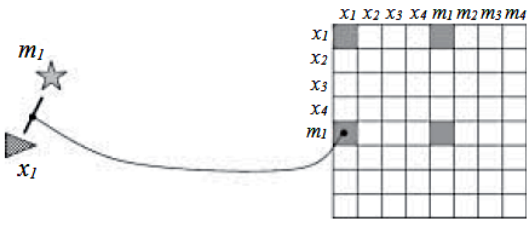
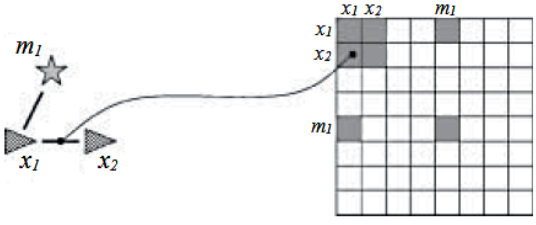


Figure 2.1 – A 2D SLAM problem represented in the GraphSLAM graphical environment. Figure taken from Thrun and Montermerlo [19].

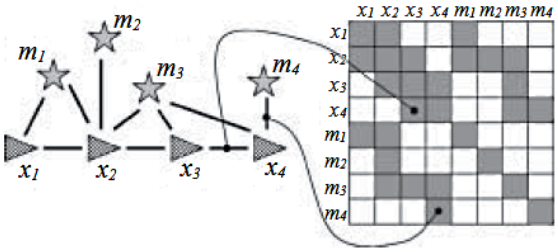
Once spring values are determined, they are used to update the information matrix Ω and the information vector ξ. The initial pose of the vehicle at k=0 is anchored as $(0 \quad 0 \quad 0)^T$ by using the anchoring constraint $x_0^T \Omega_0 x_0$. Figure 2.2 shows the addition of new elements to the information matrix.



(a) Observation of landmark m_1.



(b) Robot motion from x_1 to x_2.



(c) SLAM graph and information matrix after several steps.

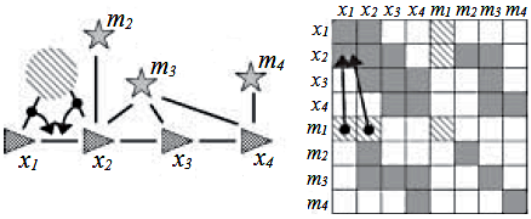
Figure 2.2 – The information matrix filling process. Each motion and measurement arc is entered into the matrix. Figure taken from Thrun and Montermerlo [19].

In order to recover the map and the path from the information matrix and information vector, the following equation is used to determine the map estimate:

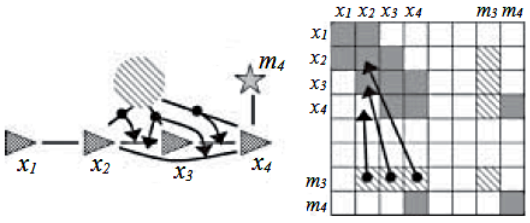
$$\mu = \Omega^{-1}\xi. \tag{28}$$

For real world implementations of GraphSLAM, the information matrix quickly becomes large and complex due to the likelihood of the same feature being observed in significantly different time steps. The complexity of the information matrix makes the matrix inversion step, required to recover the map, difficult and slow. To overcome this problem, a factorization step is used to remove landmarks from the information matrix and information vector. When a landmark is removed, any pose estimate pairs which are linked through common observations of the landmark have new arcs introduced to maintain the equivalent relationship between them, despite the removal of the landmark. This process is illustrated in Figure 2.3. Although the resultant information matrix ($\widetilde{\Omega}$) and information vector ($\tilde{\xi}$) are smaller, they are equivalent to their original forms, significantly simplifying the inversion process and allowing an optimization technique such as conjugate gradient to recover the vehicle path.

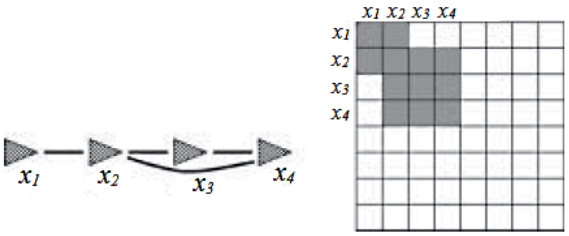
The feature map can also be recovered through the production of a set of information matrices (Ω_j) and vectors (ξ_j) each containing a single landmark (m_j) removed from the original information matrix (Ω). Each new information matrix and vector contains a single landmark as well as every vehicle pose at which the landmark was observed. They also contain the original links between the landmark (m_j) and each vehicle pose (x_k). The vehicle poses are linked with the motion arcs calculated in the simplified information matrix ($\widetilde{\Omega}$), without uncertainty. From this information, the location of the landmark can be easily determined via matrix inversion and an optimization technique such as conjugate gradient. The inversion is linear in the number of vehicle poses, keeping processing time down.



(a) Removing m_1 changes the link between x_1 and x_2.



(b) Removing m_3 introduces a new link between x_2 and x_4.



(c) The result after removing all map features.

Figure 2.3 – The information matrix reduction process. Landmarks are removed, leaving only vehicle poses. Figure taken from Thrun and Montermerlo [19].

The primary benefits of GraphSLAM include its ability to generate maps from state spaces with a number of features up to 10^8 or more and its low computational costs due to the sparse nature of the information matrix and the ability to perform optimization at a frequency lower than frame rate. Unfortunately, the algorithm is only convergent when executed for short periods of time, therefore requiring longer mapping tasks to be broken into pieces so that SLAM can be performed in shortened segments. The SLAM results can then be reconstructed using a higher order algorithm. A detailed comparison of EKF and GraphSLAM can be found in [20].

2.3 Bearing-Only Localization and Mapping

Bearing only SLAM is a mapping and localization task with sensor information that provides the relative bearings to features, but not the corresponding distances. Bearing-only SLAM problems most often arise when a single camera is used for localization. The position of features in the camera image conveys bearing information, but no real world range information without using *a priori* knowledge of the environment or vehicle motion. Bearing-only SLAM with a single camera is often referred to as vision-based bearing-only SLAM or monocular SLAM. There are

several adjustments that must be made when shifting from range and bearing SLAM to bearing-only SLAM. These adjustments are discussed in the following section.

2.3.1 The Approach to Bearing-Only SLAM

The lack of real world scale in bearing-only sensor information requires some adjustments to the general approach to simultaneous localization and mapping in order to accommodate this information. Generally, to produce a map with only bearing information, the trajectory of the host sensor relative to environmental features must be estimated such that the motion satisfies the successive bearing-only sensor observations of those features. However, without *a priori* information about the environment or vehicle motion, the scale of the map cannot be recovered. This means that the first set of feature measurements are usually established with an arbitrary depth value and all future points will have their positions initialised with respect to those first points. Also, points that are observed only once will have a completely unknown depth estimate and so must be handled separately in the SLAM system unless a compensating algorithm is applied (such as inverse depth parameterization by Montiel *et al.* [21]). The approach implemented in our mapping system to handle these aspects of bearing-only SLAM are based on the work of Civera *et al.* [22] and is examined in detail in Section 4.2.3.

Since the motion of the sensor cannot be directly measured in pure bearing-only SLAM, the motion must be approximated. A motion model is used to predict the next location of the sensor based on the previous location and some motion noise. These motion models can vary greatly depending on the known constraints or characteristics of the system. For example, a hand-held sensor would require a motion model that predicts movement in all six degrees of freedom; however the motion can be expected to be relatively smooth so a constant velocity model might be used. Alternatively, if the sensor is mounted on a vehicle, the motion model will be simplified to reflect the range of movements that the vehicle is capable of. The sensor used in our multisensor system will be mounted on a vehicle and so this is taken into account during the derivation of the motion model in Section 4.2.1.

The bearing-only SLAM system has to be able to ‘recognize’ a feature when it is measured multiple times in order to retrieve relevant mapping information. The approach to feature recognition varies depending on the sensor used for bearing-only SLAM; however, most systems use some form of camera. Camera data is analysed as a grid of pixel colours or intensities and theoretically every pixel could be mapped. Unfortunately, there is a large amount of repetition of pixel colours or intensities within camera images and so points with a more unique description are needed for mapping. Feature extraction algorithms are used to extract unique clusters of pixels that can be identified if the same scene is observed again. There are several powerful algorithms available for this task, each with a different approach to feature identification and description. Two popular examples will be discussed in Section 2.3.3.

Finally, extracted features must also be matched between images in order for localization to be possible. This process is known as the correspondence problem. Outlier correspondences must also be identified and rejected to prevent false matches from corrupting the localization result. The approaches to matching and two common techniques for outlier detection are discussed in Section 2.3.4.

2.3.2 Filter Selection for Bearing-Only SLAM

Vision-based bearing-only SLAM is similar to all other forms of SLAM in that the general problem is nonlinear and therefore requires a linearized approximation in order for a practical solution to be possible. In Section 2.2, three principal approaches to nonlinear SLAM problems were presented: Extended Kalman Filters (EKFs), particle filters and graph-based smoothing. Vision-based bearing-only SLAM can be implemented using any one of these techniques; however, each approach has unique characteristics which must be considered when selecting the appropriate solution for a specific application.

Extended Kalman Filtering has historically been the most popular approach to vision-based bearing-only SLAM due to two key characteristics: real-time processing and long term performance. The processing time for an EKF is directly proportional to the size of the state vector squared. For many bearing-only SLAM problems, the

state vector contains only the most recent vehicle pose and all of the recently observed features. Since vision-based SLAM generally has low numbers of features (often less than 30 per frame) the state vector remains small, making EKF implementations highly efficient. However, if only recent features are maintained by the state vector, a higher order process will be required to search for loop closure opportunities. EKFs also have the ability to perform accurate localization over large sequences, a weakness of particle filters. Civera *et al.* report successful localization over trajectories of up to 650m with an average error of around 1% [23]. Although recent Graph-SLAM based techniques have demonstrated real-time performance [24] [25], they rely on high numbers of visual features, which cannot be guaranteed in the difficult underground mining environment.

EKFs perform well on systems that are not severely nonlinear and non-Gaussian. In most SLAM implementations, EKFs will produce a quality result; however, the results produced by particle filters and graph-based smoothing can outperform those produced via EKF. Therefore, to improve the quality of the results obtained through EKF, modifications to the standard EKF SLAM approach must be made to reduce linearization error.

Since many EKF based bearing-only SLAM implementations maintain only the recently observed features in the state vector to improve processing speed, there is no ability to perform loop closure. This means that the uncertainty in the camera location will always continue to grow, with respect to the world reference, as the camera moves away from the origin. Increasing uncertainty in vehicle pose quickly leads to linearization errors. To overcome this problem, Civera *et al.* present camera-centred estimation which locks the frame of reference to the current camera location, significantly reducing linearization errors [23].

An improved measurement model is also proposed by Montiel, Civera and Davison which can handle distant features and features with only a single observation by encoding them using inverse depth [21]. A detailed examination of the inverse depth measurement model can be found in Section 4.2.3. The secondary advantage of this measurement model is that it linearizes well at low parallax, again reducing the linearization error common to EKF SLAM implementations.

Particle filters have also been successfully implemented for vision-based bearing-only SLAM tasks. Particle filters are capable of producing high accuracy results due to accuracy 'scalability' being possible through variation in the number of particles estimating the continuous probabilities. By reducing the number of particles, real-time performance is possible, as demonstrated by Eade and Drummond [26]. However, particle filter accuracy is limited to short sequences. During longer sequences, particle filters suffer from a problem known as 'sample impoverishment' where samples tend to converge to a confined region in the solution space, resulting in state estimations being trapped in local optima.

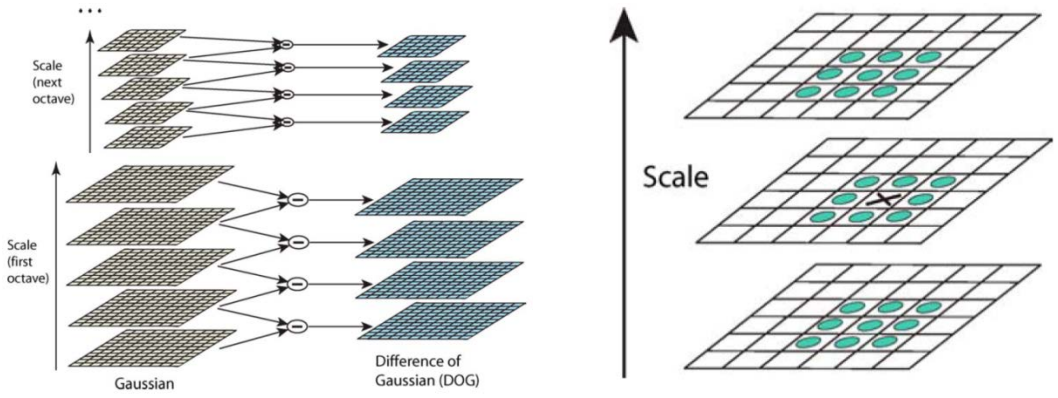
Graph-based bearing-only SLAM has been demonstrated by Eade and Drummond as a technique for optimizing the localization path between groups of features known as 'nodes' [27]. This higher order optimization of local mapping results allows the graph-based approach to perform localization over long sequences that would normally prohibit real-time processing. The use of feature clusters does not lend itself well to real world scaling via sensor fusion. The sparse layout of the nodes could easily result in the application of a scale that suits those few nodes, but does not accurately reflect the scale of the environment. Graph-based bearing-only SLAM has also been demonstrated by Klein and Murray [24] to perform high accuracy localization and mapping in a small desktop environment.

2.3.3 Feature Extraction for Bearing-Only SLAM

An important aspect of vision-based bearing-only SLAM is the technique used for the extraction of features from the incoming stream of images. Feature extraction must be robust and repeatable and also must be able to produce unique features that can be identified in subsequent images. Currently the most popular feature extraction techniques are Scale Invariant Feature Transform (SIFT) by Lowe [28], Speeded Up Robust Features (SURF) by Bay, Tuytelaars and Van Gool [29], and Features from Accelerated Segment Test (FAST) Corner Detection by Rosten and Drummond [30]. All of these existing techniques process monochrome images. SIFT and FAST corner detection will be examined in the following sections. SIFT has been selected due to its proven robustness for extracting and matching features across multiple images [31], whereas FAST corner detection has been selected due to its highly efficient

design, allowing integration with applications using real-time video frame rates. SURF is faster than SIFT but less robust and as such may be investigated in future work as an alternative to SIFT if reduction in processing times becomes important. However, at this stage the robustness of SIFT and speed of FAST corner detection incorporate a complete range of feature extraction characteristics upon which to evaluate our work.

SIFT is a feature extraction algorithm designed by David Lowe [28] to produce features that are invariant to scale and rotation. Subsequent matching between these features is robust to affine distortion, change in 3D viewpoint, noise and slight change in illumination. To create a set of image features, a difference-of-Gaussians function is first used to produce incrementally down-scaled, convolved images separated by a constant factor *k* in scale space (Figure 2.4(a)). Maxima and minima keypoints are detected by comparing a pixel from a difference-of-Gaussians image to its nearest 26 neighbours at the current and adjacent scales (Figure 2.4(b)). The use of scale space produces maxima and minima that are invariant to scale.



(a) Creating Difference of Gaussians Image. (b) Comparing a pixel to its 26 nearest neighbours.

Figure 2.4 (a) For each octave of scale space, the initial image is repeatedly convolved with Gaussians which are subtracted to produce difference-of-Gaussian images. After each octave, the image is down-sampled by a factor of 2. (b) Extrema are detected by comparing a pixel to its 26 nearest neighbours in 3x3 regions in the current and adjacent scales. Figure taken from Lowe [28].

Keypoints are then fitted with a 3D Taylor expansion quadratic function [32] to determine the interpolated location of the extrema. The function value at this extrema is then used to reject unstable extrema with low contrast. It is also used to reject keypoints along an edge that are poorly located and prone to noise. An edge based

keypoint will have a large principal curvature across the edge but a small one in the perpendicular direction.

The gradient magnitude and orientation of each keypoint is then determined. The keypoint descriptor is represented relative to the orientation, achieving invariance to image rotation. The keypoint descriptor is designed to be partially robust to change in illumination and 3D viewpoint to complement the previously stated invariance to image scale and rotation. A descriptor is created by first computing the gradient magnitude and direction at each image sample point in a 16x16 region around the keypoint location. These are weighted using a Gaussian window and accumulated into histograms representing 4x4 subregions (Figure 2.5). Descriptors use 16 histograms (4x4). Matches are identified by comparing the accumulative error between the vectors which describe the two features.

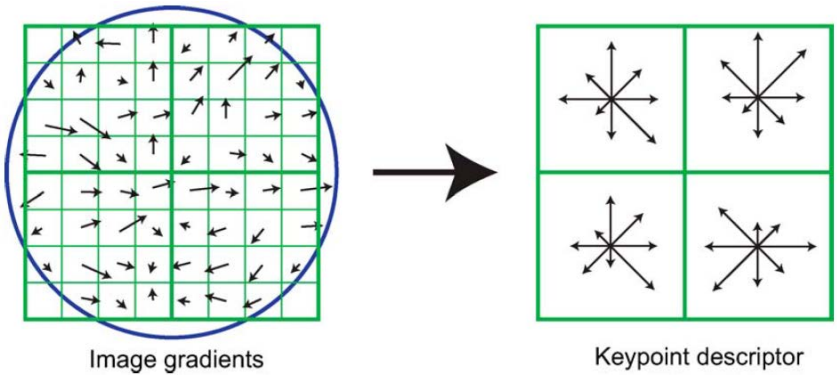


Figure 2.5 - A keypoint descriptor is computed by accumulating the gradient magnitude and direction of each sample point in a region. The example shown here is an 8x8 region reduced to a 2x2 descriptor, the actual SIFT algorithm uses a 16x16 region reduced to a 4x4 descriptor. Figure taken from Lowe [28].

FAST corner detection is a corner extraction algorithm optimized through machine learning. Rosten and Drummond [30] designed the algorithm to be an effective feature extraction algorithm at real-time frame rates. Corner detection is initiated by considering a circle of 16 pixels around a corner candidate *p*. This basic step classifies *p* as a corner if there is a set of n contiguous pixels in the circle which are all brighter or darker than the intensity of the candidate pixel plus a threshold (Figure 2.6).

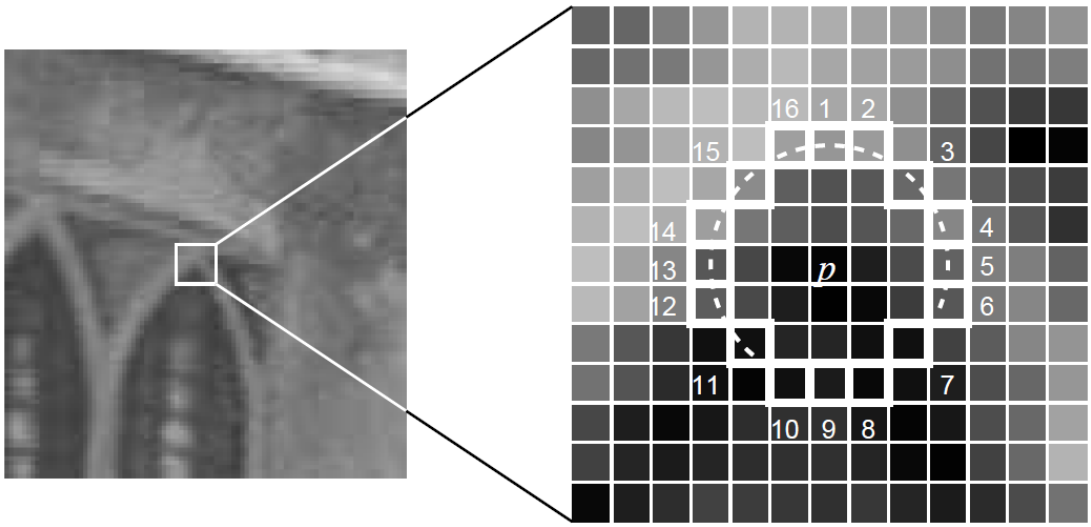


Figure 2.6 – A corner is detected when a circle of 16 pixels about a corner candidate *p* are brighter or darker than a threshold above and below the intensity value of *p*. Figure taken from Rosten and Drummond [30].

Machine learning is then used to improve the classification by using a training dataset. The corners are detected using a slow algorithm which compares each of the 16 circumference pixels to the corner candidate and assigns them a status of brighter, darker or similar based on a convenient threshold. Each pixel location $x \in \{1..16\}$ is then assessed across the set P containing all detected corners. The x that contributes the most information about whether the candidate pixel is a corner is determined using the entropy of a boolean representation of each p. The process is then applied recursively on all three subsets (P_d, P_b, P_s) to determine the x that produces the most information for each subset partition ($P_{d,d}$, $P_{d,b}$, $P_{d,s}$ etc.). This produces a decision tree that can correctly classify all corners in the training dataset. The decision tree can then be hard coded to optimize processing times.

2.3.4 Correspondences in Bearing-Only SLAM

Features can only be used for localization if correspondences can be made between them in subsequent images. The correspondence problem is therefore integral to all vision-based SLAM systems. The correspondence process begins with an initial matching of visual features extracted by feature extraction algorithms. For example, SIFT produces a feature descriptor as part of the extraction process as described in Section 2.3.3. This descriptor uses bins containing normalized vectors to describe the

gradients present in the keypoint region of interest. A simple matching algorithm can compare bin vectors to determine a keypoint's closest match in a subsequent image.

Invariant descriptors such as those produced by SIFT can be computationally expensive to produce, so for real-time algorithms such as FAST corner detection, simple cross-correlation of small image patches around the keypoint can be used for matching. Improvements to the basic image patch matching approach have been suggested to improve invariance to point of view. Chekhlov *et al.* [33] generate patches at various scales when a keypoint is first initialized. The scale change in subsequent images is then predicted in order to select the appropriate patch for matching. Alternatively, Molton *et al.* [34] warp image patches according to predicted motion before the matching process begins.

All visual correspondence techniques are prone to outlier matches due to visual similarities between images and repeated visual features. Outlier rejection is therefore vital to prevent correspondence corruption. Techniques such as Random Sample Consensus (RANSAC) and Joint Compatibility Branch and Bound (JCBB) have therefore been developed to improve the identification and removal of outlier matches.

RANSAC categorizes outliers by firstly fitting a model to a random sample of data. This random sample is considered to consist entirely of inliers (although this may not be the case) and so the model is fitted to all points in the sample. The model is then tested against the entire dataset and the quality of the fit is determined. If the quality of the fit is reasonable, the model is used to reject outliers and is then recalculated based on the remaining inlier points. A fitness score is determined based on the quality of fit to the inlier points and the model with the lowest fitness score after the predetermined number of iterations is used for final outlier rejection. The RANSAC technique was first documented by Fischler and Bolles in 1981 [35]. Since then, much research has been focused on the early detection and elimination of bad model hypotheses to reduce computational cost [36] [37] [38] [39].

JCBB is an outlier detection algorithm by Neira and Tardos [40] which measures the joint compatibility of a set of pairings to reject spurious matches. It does this by

predicting the probability distribution over the measurements and then using a Branch and Bound search technique to identify the maximum set of pairings that is jointly compatible with the prediction. JCBB is more robust to noise than RANSAC but has exponential computational complexity in the number of matches.

Chapter 3

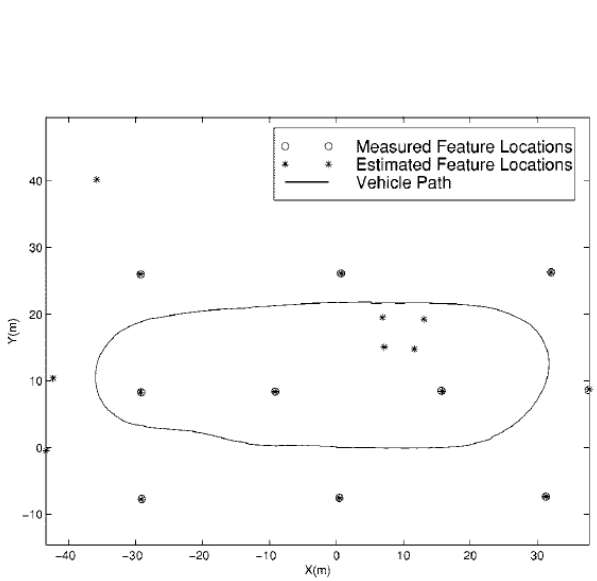
SLAM Implementations

In recent years there have been many techniques and improvements developed in an attempt to solve the problem of large scale environmental mapping. This section will discuss the capabilities of some of the more successful implementations. These implementations will be grouped based on the type and quality of the resulting map. The review will begin with an examination of the initial 2D implementations of SLAM, followed by the first extensions to 3D map building based on 3DoF localization. Vision-based bearing-only SLAM will then be investigated as an approach to 3D mapping with 6DoF localization. Map construction for navigational purposes will be examined, followed by a review of systems designed for high accuracy, survey quality 3D mapping.

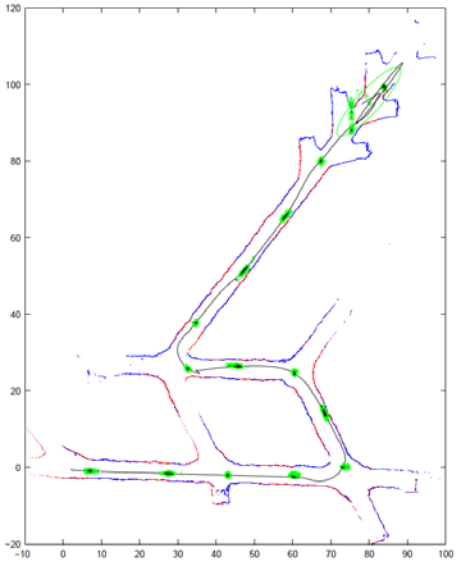
3.1 2D SLAM Implementations

The very first implementations of autonomous mapping systems were based on a simplified 2D approach to the navigation of 3D environments. These early systems were therefore limited to flat, level environments which generally included only building interiors and flat sealed roads. One of the first examples of successful localization and mapping were the results obtained by Dissanayake *et al.* using a combination of radar and wheel odometry mounted on a vehicle [41]. The vehicle repeatedly traversed a flat 160m loop in an outdoor environment, while building a

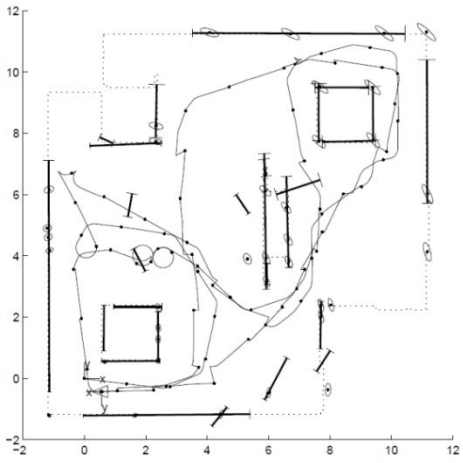
basic map from the observed radar features. The results from that experiment can be seen in Figure 3.1(a). The same vehicle was fitted with a 2D laser scanner in the work by Bailey and demonstrated the ability to produce dense 2D maps using a particle filter approach [42]. The results from a test in a mine tunnel can be seen in Figure 3.1(b). Dense 2D maps were also produced by Tardos *et al.* using a combination of sonar sensors and wheel odometry [43]. The large amounts of noise present in the sonar readings lead to some significant errors in map production as seen in the results in Figure 3.1(c) overlaid on the ground truth map.



(a) Localization and mapping result by Dissanayake *et al.* [41].



(b) Localization and mapping result by Bailey [42].



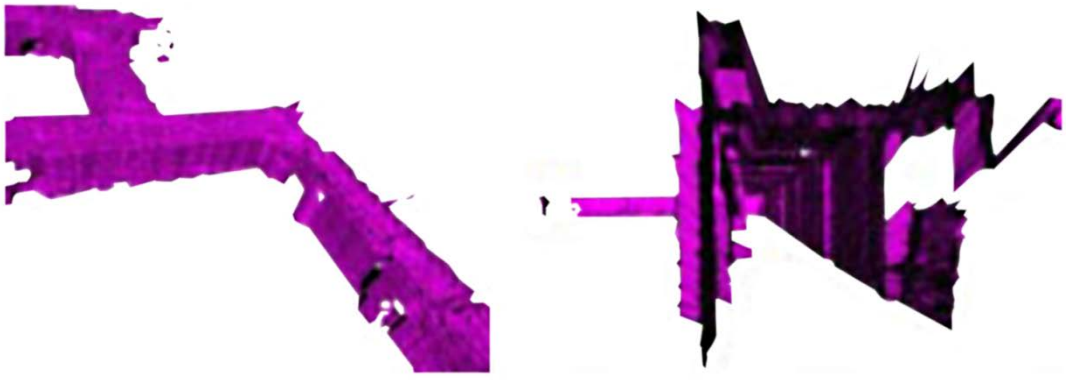
(c) Localization and mapping result by Tardos *et al.* [43].

Figure 3.1 – Localization and mapping results from early implementations using (a) radar, (b) 2D laser and (c) sonar.

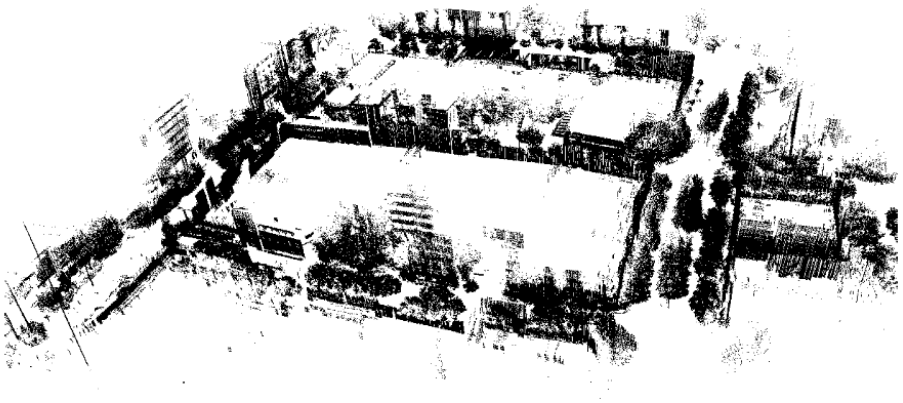
3.2 2D Localization for 3D Mapping

The successful implementations of 2D mapping with 3DoF localization led to the first examples of basic 3D map building systems. Since many 3D mapping tasks are performed in environments that are flat and level and are recorded using vehicles with only three degrees of freedom (X-axis translation, Y-axis translation, yaw rotation), simplifications to the mapping process were possible. The 3D environment could be approximated as a 2D environment by removing Z-axis translation, roll rotation and pitch rotation. 3D sensor observations can therefore be localized using only a 3DoF approximation of motion.

Thrun, Burgard and Fox were another research group to pioneer the use of 2D lasers and wheel odometry for 3DoF localization and 2D mapping [9]. For the production of a 3D map, a second laser was mounted vertically to the vehicle during a localization task to record a series of 2D cross sections. These cross sections could then be compiled using the 3DoF localization results to produce a 3D map. An example 3D mapping result can be seen in Figure 3.2(a). The idea of using two non-parallel 2D laser scanners for 3D mapping was extended to an outdoor application by Howard, Wolf and Sukhatme [3]. The dependency on wheel odometry was exchanged for a dependency on GPS signals which were readily available in the outdoor environment and were far more practical for tasks such as loop closure. The 3D map resulting from a 2km tour of the UCS campus can be seen in Figure 3.2(b).



(a) The indoor 3D mapping results obtained by Thrun, Burgard and Fox [9].



(b) The outdoor 3D mapping results obtained by Howard, Wolf and Sukhatme [3].

Figure 3.2 – Examples of 3D maps built on 3DoF localization results.

Fruh and Zakhor use *a priori* information as an alternative to wheel odometry and GPS signals for drift correction during 3DoF localization [8]. This *a priori* information is in the form of aerial photographs and Digital Surface Models. The 3DoF localization results are used to build a 3D map from a vertically mounted 2D laser scanner, without the need for 3D point cloud registration. *A priori* map information is also used by Oh *et al.* to improve 3DoF localization results by weighting map areas that are more likely to be traversed by a person or vehicle, e.g. a sidewalk in an urban environment [44]. Brenneke, Wulf and Wagner register low density 3D scans by projecting 3D features onto a 2D plane, producing a 'levelled' 2D scan [45]. This 'levelled' 2D scan is then used for 2D registration, significantly reducing processing time. 2D scan registration is also applied by Biber *et al.* who model geometrically simple indoor spaces by the vertical projection and rendering of 2D mapping results [46].

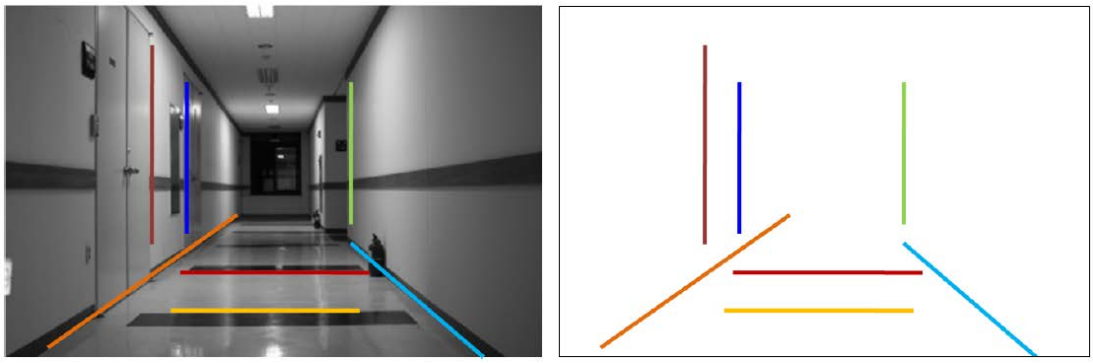
3.3 Bearing-Only 3D Mapping

Bearing-only simultaneous localization and mapping using a single camera (also known as monocular SLAM) has been a popular avenue for research groups due to the ability of a single low cost, simple, passive sensor to provide real-time inferred 3D information in a compact package [47]. Features extracted from the images are used to produce sparse 3D maps. Extended Kalman Filters [23], particle filters [48] [26] and graphical optimization [27] [24] have all been investigated as potential

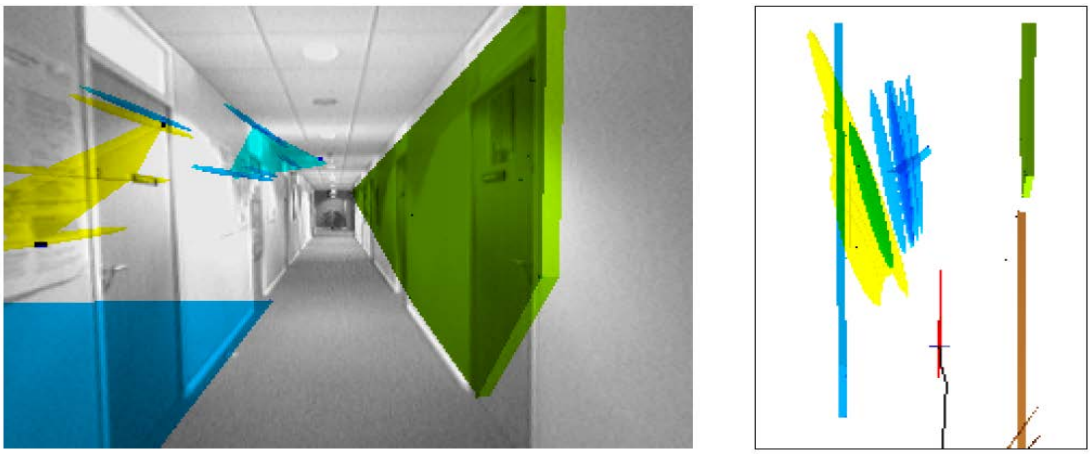
solutions to the monocular SLAM problem. Results from several research groups have shown that vision-based bearing-only SLAM can also produce high accuracy results when combined with some form of sensor fusion to provide real world scaling [23], [26], [27].

A popular sensor choice when implementing sensor fusion is an Inertial Measurement Unit (IMU). Roussillon *et al.* [49] and Nutzi *et al.* [50] both demonstrate the ability to recover scale using IMU data. Other fusion options include GPS signals [51], laser data [52] and wheel odometry [51]. Strasdat *et al.* [53] have shown that map scale also tends to drift over time and so scale compensation during loop closure is required during longer localization paths. Scale drift in the localization path requires scene recognition for loop closure as the pose estimate cannot reliably predict loop closure environments [54].

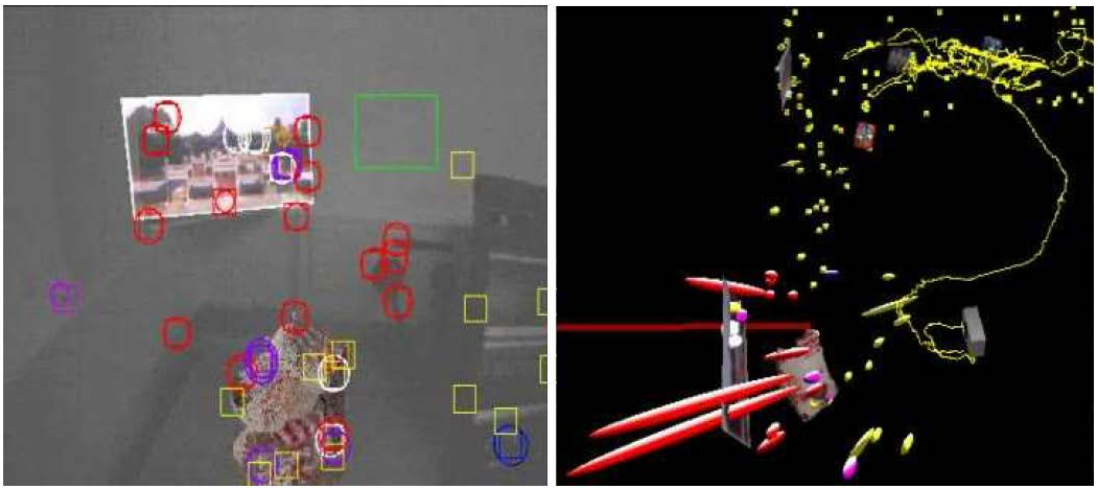
Maps comprising of sparse point features are useful for localization but pose difficulties for navigation tasks where high density maps are required. To address this issue many research groups have turned to the extraction of features other than points (see Figure 3.3). Line features are used by Zhang and Sun [55] to produce a traversable map of indoor environments. A combination of point features and plane features are used to similar effect by Martinez-Carranza and Calway [56]. Hwang and Song use visual corners and light sources with an upwards facing camera [57]. Finally, Civera *et al.* combine monocular SLAM, Structure from Motion (SFM) and object recognition to produce a map containing both feature points and recognizable objects, hence improving the semantics of SLAM [58].



(a) Monocular SLAM map built from line features. Figure taken from Zhang and Sun [55].



(b) Monocular SLAM map built from plane and point features. Figure taken from Martinez-Carranze and Calway [56].



(c) Monocular SLAM map built from point features and recognized objects. Figure taken from Civera *et al.* [58].

Figure 3.3 – Example of the various features used to build maps from bearing-only visual data.

3.4 3D Mapping and Localization for Navigation

Autonomous vehicular navigation in outdoor environments cannot be robustly approximated using 3DoF localization as there is usually significant movement in all six spatial degrees of freedom. Maps produced for robust autonomous navigation must therefore be compatible with movement in six degrees of freedom, while still maintaining real-time processing ability. To achieve these goals, simplifications must sometimes be made to the mapping process. These simplifications can take many forms, including the use of lower density maps created by tracking only a small number of key environmental features, geometrically simplified maps reconstructed

from sensor data, or even traversability maps created by categorizing dense sensor information.

Sensors used for robust outdoor navigation generally require the ability to provide 3D bearing information. The 3D bearings can then be supplied with depth information from the same sensor or a complementary sensor in order to localize both the vehicle and the landmarks in a complex 3D environment. Stereo vision is often used for navigation tasks due to the ability to extract adequate range information from the camera images for obstacle avoidance and basic mapping while maintaining real-time processing. Konolige, Agrawal and Sola apply stereo vision fused with an Inertial Measurement Unit (IMU) to a large scale outdoor navigation and mapping task [59]. By tracking key features in the stereo images and filtering the localization using the IMU, they report a localization error of only 0.1% over distances of up to 10km. Pinies, Tardos and Neira also apply stereo vision to a mapping task, but with no sensor fusion [60]. Their system can produce accurate low density maps over distances up to 220m while maintaining near real-time processing.

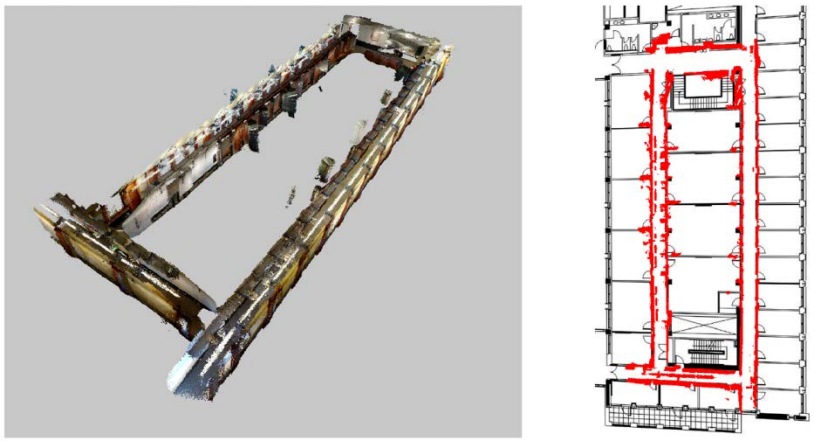
Automated mapping from a series of images produced by a single monocular camera has been a long term goal of the computer vision community. This problem has become known as Structure from Motion (SFM). The objective of SFM is to automatically detect salient features within an image and produce a set of feature correspondences to the other images in the series. Projective geometry is then used to determine the geometric relationship between feature correspondences, these relationships are analysed in order to produce a relative transformation between pairs of images [61]. The transformation solution is often refined via a non-linear optimization stage known as Bundle Adjustment [62]. Feature tracking in SFM can also be extended to produce a simple geometric representation of the environment as seen in the reconstruction of architectural structures by Xiao *et al.* [63]. Sinha *et al.* have produced a similar system with improved geometric complexity, at the cost of autonomy [64]. Example results from both of these approaches can be seen in Figure 3.4(a) and (b).

The reconstruction of environmental structure through the use of RGB cameras with per pixel depth information (known as RGB-D cameras) has also been investigated

by Henry *et al.* [65]. RGB-D cameras provide depth information that is more accurate than the measurements inferred by stereovision and SFM, but it is also significantly less accurate than laser scanners. Henry *et al.* designed a system that combines sparse visual feature mapping and dense point cloud registration to correctly align frames from the camera [65]. The resulting localization is high quality; however, the accuracy of the 3D map is inconsistent when compared to results achieved using laser scanners. Results from an example indoor mapping task can be seen in Figure 3.4(c).

(a) Environmental geometry reconstruction by Xiao *et al.* [63].

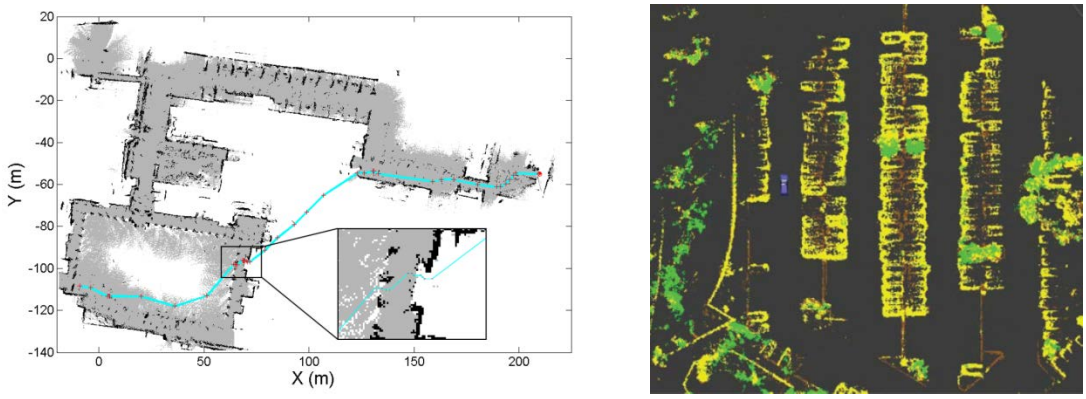
(b) Environmental geometry reconstruction by Sinha *et al.* [64].



(c) Example 3D mapping results using an RGB-D camera, produced by Henry *et al.* [65].

Figure 3.4 – Medium accuracy 3D mapping resulting from (a) autonomous SFM, (b) semi-autonomous SFM, (c) localization and mapping using an RGB-D camera.

Laser scanners can be used for real-time navigation tasks through the construction of traversability maps rather than dense maps. Traversability maps are 2.5D representations of a 3D environment which exclude complex 3D information and retain only data relevant to vehicular movement in three primary degrees of freedom. The maps are divided into regions that are classified as traversable or non-traversable based on laser range finder data. Whitty *et al.* use rotating 2D SICK laser scanners to produce a large scale indoor and outdoor traversability map in real-time [66]. An example map from the navigation system is shown in Figure 3.5(a). Montemerlo *et al.* also produce real-time traversability maps in their vehicle 'Junior' which was entered in the 2007 DARPA Urban Challenge [67]. The vehicle uses a Velodyne 3D laser to produce the real-time traversability map required for the various autonomous driving challenges in the competition. An example of their mapping results can be seen in Figure 3.5(b).



(a) Real-time map produced by Whitty *et al.* [66].

(b) Real-time map produced by Montemerlo [67].

Figure 3.5 – The real-time maps produced by two research groups demonstrate that while laser based mapping is possible in real-time, the results must be simplified to a traversability map.

3.5 Survey Quality 3D Mapping

Survey quality 3D mapping is currently only possible through the use of 3D laser scanners. 3D lasers have high angular resolution (up to 0.06°), a large field of view (360° horizontal and 270° vertical), long range (300m) and high accuracy (±6mm), making them perfectly suited to 3D surveying tasks. Consequently, they also have scan times of up to 30 minutes depending on the desired resolution. A comparison of 3D laser scanners to other available 3D sensors is presented in Table 5.1.

To apply 3D lasers to localization and mapping problems, sacrifices to system self-containment and portability must usually be made. A common technique for the registration of dense 3D laser scans is the use of wheel odometry as an initial pose estimate for a registration algorithm. Surmann, Nuchter and Hertzberg use odometry in this way and also include next best view estimation and path planning during the digitalization of 3D environments [5]. Cole and Newman use odometry, but also include registration uncertainty in an Extended Kalman Filter (EKF) to allow the back-propagation of registration correction upon loop closure [68]. Wheel odometry is again applied by Nuchter *et al.* as an initial pose estimate for a high speed octree based coarse registration step [6]. When combined with low density 3D laser scans, mapping time is significantly reduced.

Wheel odometry is the overwhelming choice for current survey quality 3D mapping solutions. However, there are notable limitations to the use of this form of measurement as detailed in Section 1.1. Therefore, a suitable alternative needs to be found in order to robustly produce survey quality 3D maps of complex environments such as active underground mines.

3.6 Existing Underground Mapping Solutions

Underground mining environments have many characteristics that pose significant challenges to localization and mapping. These challenges go beyond the difficulties associated with indoor and outdoor mapping. Problems include the tendency for underground vehicles to experience wheel slip therefore limiting the effectiveness of wheel based odometry, the absence of static illumination causing difficulties for uncompensated vision-based localization, the isolation of the environment preventing the communication with external sensory equipment such as GPS, the presence of large quantities of metallic ore that interfere with the Earth’s magnetic field preventing IMU drift compensation, and the continually changing level of the tunnels requiring full six degree of freedom compensation. Despite these difficulties, mapping systems have been documented by several research groups as potential solutions to the underground SLAM problem.

Underground mapping in its simplest form is reported by Artan, Marshall and Lavigne where the combination of a 2D laser scanner and wheel odometry is used to produce a 2D map of a mine tunnel [69]. This approach is obviously extremely limited due to a lack of 3D information and the inability to handle movement in six degrees of freedom, despite the frequent occurrence of complex movement in mining environments. An extension of this approach to three dimensions and six degrees of freedom is detailed by Nuchter *et al.* [7]. They rely on accurate wheel odometry for an initial pose estimate for the registration of discrete laser scans collected by a 'nodding' 2D laser scanner. The scans are collected in a stop-and-go fashion and are registered using a customized implementation of ICP. The system is capable of mapping complex environments which require movement in six degrees of freedom by the extrapolation of 2D odometry data to 3D via the transformations resulting from ICP registration. However, the maps produced are not of survey quality and the need for a specialized vehicle limits the applications to decommissioned mines.

Alternatively, if the underground mining environment is known to be relatively level, motion model simplification may allow 2D scan registration to be used for localization, as seen in some mapping examples in the previous section. Thrun *et al.* apply this approach by using a vehicle with horizontal and vertical lasers to map an abandoned mine tunnel [70]. The horizontal laser is used for real-time localization and does not require any additional odometry information. The vertical laser is used to build the 3D map of the environment based on the localization results from the horizontal laser. The ability to perform loop closure further improves the mapping results.

To maintain 3D localization and 6DoF movement without resorting to wheel odometry, Huber and Vandapel detail a system which does not require an initial pose estimate for point cloud registration [71]. Instead, a large number of small, low density scans are collected and then registration is performed between every possible scan pair. The map building step is approached as an optimization problem across all possible registrations simultaneously. The use of simplified surfaces for registration rather than individual laser points reduces processing time to a level that makes this technique plausible. Quality mapping results are reported, but are limited in size to

the area covered by 50 small scans before the $O(N^2)$ processing time becomes prohibitive.

It can be seen that although there have been several specialized implementations of underground mapping systems, currently there is no comprehensive solution to this problem. Each implementation requires some level of compromise which reduces the robustness and prevents application to a wide range of mining environments.

Chapter 4

Implementing Bearing-Only SLAM for the Multisensor System

4.1 Introduction

Bearing-only SLAM performed on a stream of images from a single camera is an effective technique for producing six degree of freedom localization in real-time. Single cameras are inexpensive, compact, self-contained and require little calibration, making them a desirable sensor for a self-contained, portable mapping system. However, there are shortcomings which limit the use of monocular SLAM for mapping applications; these include the inability to recover scale and the sparse maps produced. Sensor fusion with a 3D laser scanner is therefore implemented to overcome these issues. The integration of bearing-only SLAM with high precision laser point clouds is proposed in Chapter 6. Prior to the presentation of sensor fusion for large scale mapping, this chapter examines the implementation of a purely bearing-only vision-based SLAM system using an omnidirectional camera.

4.2 Omnidirectional Bearing-Only SLAM

The mathematical foundations of the Simultaneous Localization and Mapping (SLAM) problem were examined in Section 2.2 and the adaptation of the SLAM solution to a bearing-only vision-based localization problem was then discussed in Section 2.3. The platform selected for the implementation of the mathematical solution to vision-based localization was the Matlab software package produced by MathWorks. Matlab (Matrix Laboratory) is a numerical computing environment designed for the effective execution of matrix based computations. Its ability to implement algorithms, plotting functions and user interfaces makes it perfectly suited to the coding of vision-based localization. Matlab is the primary software package for image processing in academia and is well supported by the image processing community. Matlab's proficiency with matrix based mathematics allows it to efficiently handle the large scale matrices used during vision-based localization for the storage of image data, state vectors, feature properties, covariance values and mapping results. Matlab was also the platform selected for the implementation of Civera's monocular SLAM algorithm [51], upon which our modified algorithm is based.

The monocular SLAM algorithm by Civera *et al.* is the basis for our own implementation of omnidirectional vision-based localization. Civera's algorithm uses an Extended Kalman Filter (EKF) to store and update state information and is based on a constant velocity motion model and a pinhole camera model. An example of the results produced in the Matlab environment can be seen in Figure 4.1. The left results window contains the current image with red, blue and magenta dots representing the locations of matched features, unmatched features and rejected outlier features respectively. The red, blue and magenta ellipsoids represent the uncertainty of their associated feature types and the green crosses mark the predicted locations of previously matched features. The right results window contains a 2D projection of the 3D mapping results with a triangle representing the current vehicle pose and a line from the origin (at coordinates (0,0,0)), to the current pose, representing the localization path. The red dots represent established feature positions and the red ellipsoids represent their locational uncertainty.

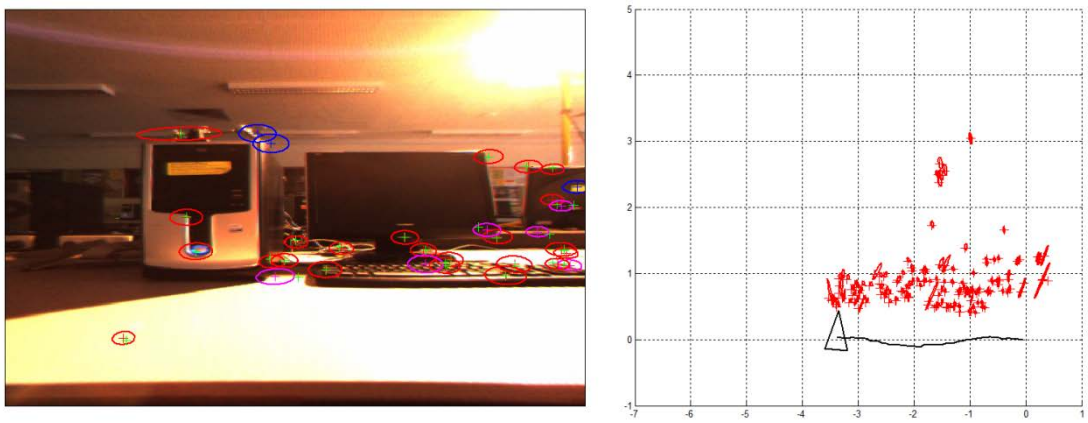
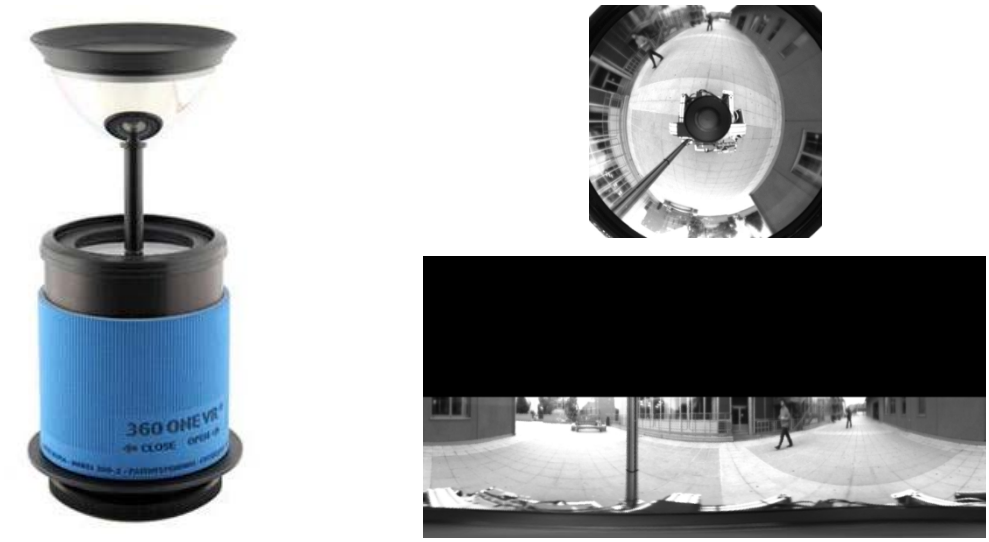


Figure 4.1 – An example of the results screen produced by the Matlab implementation of Civera's monocular SLAM algorithm.

Feature identification and recognition is a significant component of bearing-only SLAM systems and is often the aspect that is most susceptible to failure in environments with poor visual feature quality such as underground mines. The type of camera used for vision-based implementations of bearing-only SLAM can considerably influence the ability of the system to identify and track features. Cameras with low resolution will produce low feature numbers and low feature quality, while cameras with a diminished field of view will have difficulty tracking features over extended periods of time. To improve robustness in environments with poor feature quality, high resolution cameras with large fields of view are highly desirable. Therefore, high resolution omnidirectional cameras were selected as the sensor about which our bearing-only SLAM algorithm would be written.

There are three main types of camera which can be classified as spherical or panoramic. Mirror based panoramic cameras produce a panoramic image by using a single camera to observe an environment which has been reflected by the surface of a parabolic or hyperbolic mirror. Fisheye lenses can also be used to capture panoramic images by capturing light in a field of view often greater than 180°. Finally, multi-camera omnidirectional cameras can produce panoramic images via the stitching of several standard field of view cameras. This type of omnidirectional camera benefits from the least amount of distortion and the greatest spherical coverage but requires the calibration of multiple cameras. Examples of each camera type and their resulting images can be seen in Figure 4.2. The Point Grey Ladybug 2 omnidirectional camera was selected for our vision-based SLAM task as it has the highest spherical

coverage, allowing the tracking of a greater range of visual features. The Ladybug 2 also produces images that are fully rectified based on high quality camera calibration performed during the construction of the camera by Point Grey.



(a) Mirror based panoramic lens and images. Figure taken from Rituerto *et al.* [72].

(b) Fisheye lens and images. Figure taken from Roda *et al.* [73].

(c) Omnidirectional camera and images.

Figure 4.2 – The three major spherical imaging camera types. (a) The Kaidan 360 VR mirror based panoramic camera lens with example original raw image and undistorted panoramic image. (b) The Canon EF 8-15mm f/4 L USM fisheye lens with example original raw fisheye image and panoramic image. (c) The Point Grey Ladybug 2 omnidirectional camera with raw unstitched images and resulting panoramic image. Each panoramic image represents the full visual sphere (180° vertical, 360° horizontal) and demonstrates the coverage of each camera type.

The goal of the EKF vision-based, bearing-only SLAM algorithm is to produce a map of features observed within the stream of images supplied by the omnidirectional camera and to localize the sensor within that map. The movement of features across the image sphere is simultaneously used to produce a map and localize the camera. The state of the system is stored in standard EKF form, with the state vector (x) containing the current camera pose (x) and all of the observed features ($y_{1\ldots n}$).

$$\mathrm{x} = (x^T, \mathrm{y}_1^T, \mathrm{y}_2^T, \ldots, \mathrm{y}_n^T)^T \tag{29}$$

In order for the system to pass through the prediction and update steps of the Extended Kalman Filter, several definitions must first be made. These definitions include the motion model, the sensor model, the feature initialization process and the correspondence process. These definitions will be described in the following subsections.

4.2.1 Camera Motion Model

The camera motion model is a mathematical representation of the expected movement of the camera through its environment. The motion model is used to predict the location of the camera at the next time step, which in turn is used to predict the visual feature locations and then to finally compare the feature predictions to the feature measurements. The camera motion model developed for our multisensor SLAM system is based on a constant velocity model but has been modified by combining it with a vehicle motion model. This hybrid motion model best represents the expected motion of our camera in its environment: we expect movement in all six degrees of freedom, but we also expect the movement to be limited to the physical characteristics of vehicle.

A constant velocity model represents the camera state (x_v) as a vector containing the pose term for the camera's optical centre position (r^{WC}), the quaternion defining the orientation (q^{WC}), and the linear and angular velocity (v^{W} and ω^{C}) relative to the world frame of reference (W) and camera frame of reference (C) respectively. Constant velocity motion models assume that the linear and angular velocity of the camera in the next time step will be equal to the current time step, with the only addition being a velocity 'impulse' representing the uncertainty of the future state. The velocity impulse is represented via linear and angular accelerations a^{W} and α^{C}, with zero mean and known Gaussian distribution, at each time step Δt. The velocity impulses are therefore defined as follows: linear velocity $V^W = \mathrm{a}^W \Delta t$ and angular velocity $\Omega^C = \alpha^C \Delta t$.

The state update for the camera motion is then defined as:

$$f_v = \begin{pmatrix} \mathrm{r}_{k+1}^{WC} \\ \mathrm{q}_{k+1}^{WC} \\ \mathrm{v}_{k+1}^{W} \\ \omega_{k+1}^{C} \end{pmatrix} = \begin{pmatrix} \mathrm{r}_{k}^{WC} + (\mathrm{v}_{k}^{W} + \mathrm{V}_{k}^{W})\Delta t \\ \mathrm{q}_{k}^{WC} \times \mathrm{q}\left((\omega_{k}^{C} + \Omega^{C})\Delta t\right) \\ \mathrm{v}_{k}^{W} + V^{W} \\ \omega_{k}^{C} + \Omega^{C} \end{pmatrix}. \tag{30}$$

Where $\mathrm{q}\left((\omega_k^C + \Omega^C)\Delta t\right)$ is the quaternion defined by the rotation vector $(\omega_k^C + \Omega^C)\Delta t$.

A vehicle motion model is very similar to a constant velocity model, except that it keeps the z-axis translation, roll rotation and pitch rotation at zero. This vehicular motion assumption is too limiting for an underground mine environment, so rather than setting those parameters to zero, they are given a weighting in order to balance the motion model between vehicular and full six degree of freedom. The weighting matrices η_1 and η_2 can be used to adjust the predicted pose and orientation of the camera's optical centre to accommodate this expected vehicular motion as follows:

$$V^W = \eta_1 \mathrm{a}^W \Delta t \tag{31}$$

$$\Omega^C = \eta_2 a^C \Delta t. \tag{32}$$

For our motion model implementation, weightings of $\eta_1 = \mathrm{diag}(1,1,0.1)$ and $\eta_2 = \mathrm{diag}(0.1,0.1,1)$ were applied, reflecting the approximate expected uncertainty in each of the degrees of freedom. These weights performed well in the range of SLAM testing performed on the motion model.

To produce the covariance update step for the system, the derivatives of the dynamic motion model with respect to the state (*F*) and with respect to the Gaussian noise of the model (*G*) are required. These derivatives are defined as:

$$F = \frac{\partial f_v}{\partial \mathrm{x}_v} \tag{33}$$

$$G = \frac{\partial f_v}{\partial \mathrm{n}}. \tag{34}$$

The full expansion of these terms can be found in Appendix A.

4.2.2 Sensor Model

Civera's original monocular SLAM algorithm was based on a pinhole camera model. The pinhole camera model is characterized by a flat, 2D image plane upon which an image is projected. By supplying the focal length of the camera to the pinhole model, the locational bearings of features on the image plane can be determined (see Figure

4.3). The movement of tracked features across the image plane can be interpreted by the motion model as equivalent camera movement

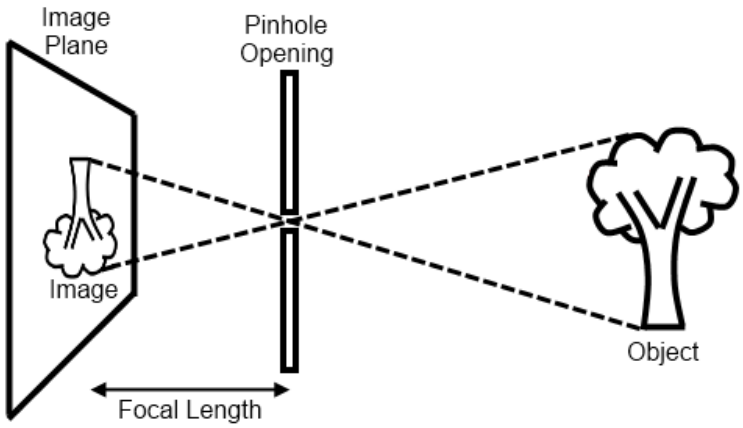


Figure 4.3 – The pinhole camera model. The location of the object (bearing only) can be determined from the image plane if the focal length is known.

The omnidirectional camera required for our system is not compatible with the pinhole camera model and instead requires the use of a spherical camera model. Civera's monocular SLAM Matlab implementation was converted by Rituerto, Puig and Guerrero to incorporate a spherical camera model in place of the pinhole model in order to be used with a mirror based omnidirectional camera [72]. The major change to the software was in the interpretation of feature locations in the world relative to the imaging plane. The projection method used for the omnidirectional camera can be seen in Figure 4.4. This technique is significantly different to the pinhole model shown in Figure 4.3. A feature (x'') in the image supplied by the mirror based omnidirectional camera (see example image in Figure 4.5) is transformed to a new location in a rectified image (x') through the use of H_C which compensates for the distortions caused by the camera and the mirror. The rectified image location is then aligned with the virtual projection centre (C_P) to determine the point at which the feature projection intersects the surface of the unit sphere (x_+). This intersection point contains the bearings of the true feature location in space (X) relative to the centre of the sphere (O).

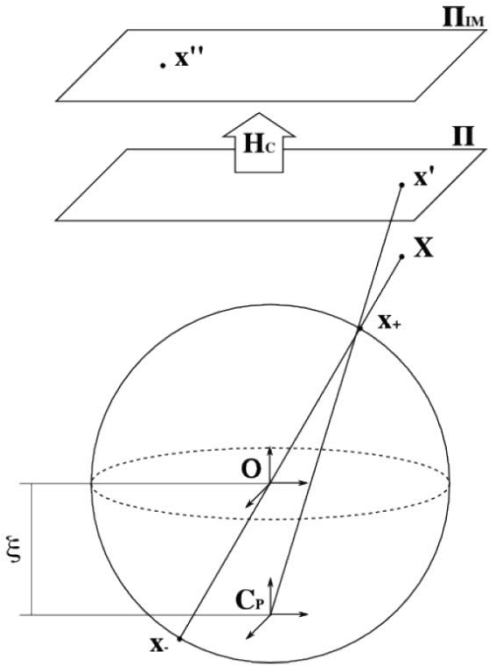


Figure 4.4 – The spherical camera model. Feature locations in the original image are rectified, then projected onto the unit sphere. Figure taken from Rituerto *et al.* [72].

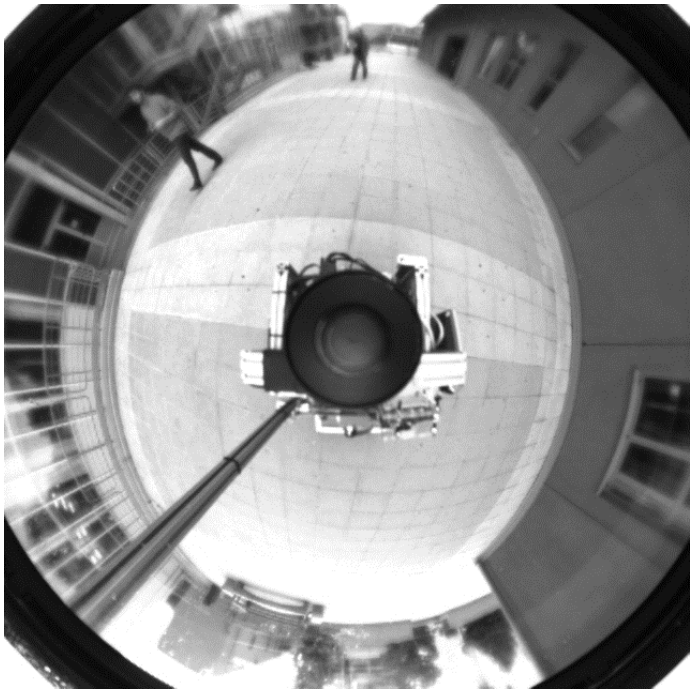


Figure 4.5 – An example image produced by a mirror based omnidirectional camera. Figure taken from Rituerto *et al.* [72].

The Ladybug omnidirectional camera is not a mirror based omnidirectional camera and therefore the spherical camera model of Rituerto *et al.* needs some modification. The images streamed by the Ladybug camera are fully rectified and cover the entire

visual sphere (the portion of the sphere not covered by cameras is masked, see Figure 4.7). Therefore, the projection technique is simplified to the equation below.

$$\begin{pmatrix}\theta_i \\ \phi_i\end{pmatrix} = \begin{pmatrix}\left(\frac{u_i}{u_{max}}\right)2\pi - \pi \\ \left(\frac{v_i}{v_{max}}\right)\pi - \frac{\pi}{2}\end{pmatrix} \tag{35}$$

The unit sphere is still used to project the features into space and to predict the location of existing features in newly acquired images. Each feature encoded by the Extended Kalman Filter (y_i) is recorded as a ray from the camera position at which the feature was first observed (see Figure 4.6).

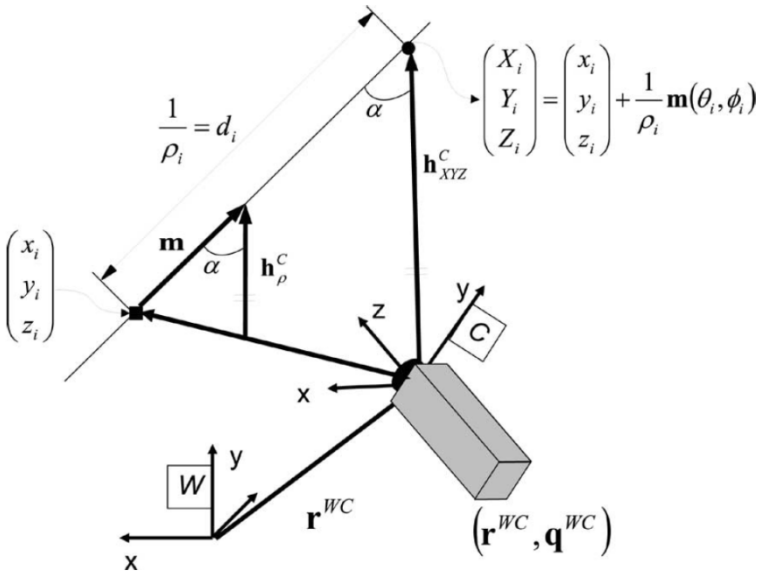


Figure 4.6 – A feature first observed at (x_i, y_i, z_i) is observed again at the current time step. The location of the feature relative to the current camera pose is defined as the sum of the vector representing the pose where the first observation occurred (x_i, y_i, z_i) and the vector representing the angular location and depth of the feature at the first observation ($\frac{1}{\rho_i}\mathrm{m}(\theta_i, \phi_i)$). Figure taken from Civera *et al.* [22].

In order to encode the 3D location of the feature in the state vector, the feature's image based coordinates ($(u_i \quad v_i)^T$) must first be converted to an elevation and azimuth angle with respect to the current camera location. The conversion is straightforward as the image produced by the Point Grey Ladybug 2 omnidirectional camera is a fully rectified panoramic image containing the entire visual sphere.

$$\theta_i^C = \frac{u_i - \left(\frac{u_{max}}{2}\right)}{\left(\frac{u_{max}}{2}\right)}\pi \tag{36}$$

$$\phi_i^C = \frac{v_i - \left(\frac{v_{max}}{2}\right)}{\left(\frac{v_{max}}{2}\right)}\left(\frac{\pi}{2}\right) \tag{37}$$

The image resolution ($u_{max} \times v_{max}$) contains the entire visual sphere; However, the cameras in the Ladybug 2 do not cover the entire visual sphere due to a horizontal field of view of 360° and a vertical field of view of 145° (-55° to +90°). Therefore, the bottom section of the image, not covered by the cameras, is masked (see Figure 4.7).

Figure 4.7 – Example panoramic image produced by the Point Grey Ladybug 2 omnidirectional camera. Note the masked section at the bottom of the image due to the cameras not covering the entire visual sphere.

The azimuth and elevation angles are currently in the camera frame of reference and so must be converted to the world frame of reference before being stored in the state vector. The conversion is produced using the quaternion representing the current camera orientation (q^{WC}).

$$\begin{pmatrix} \theta_i^W \\ \phi_i^W \end{pmatrix} = \begin{pmatrix} \arcsin\left(2(q_0 q_2 - q_3 q_1)\right) \\ atan2\left(2(q_0 q_1 + q_2 q_3), 1 - 2(q_1^2 + q_2^2)\right) \end{pmatrix} + \begin{pmatrix} \theta_i^C \\ \phi_i^C \end{pmatrix} \tag{38}$$

The resulting azimuth and elevation angles are encoded in the state vector. At the next time step, these stored feature coordinates will need to be converted back to image based coordinates in order to predict the location of each feature in the newly acquired image. The conversion for the stored feature points is

$$\mathrm{h}^C = \mathrm{R}^{CW}\left(\rho_i\left(\begin{pmatrix}x_i\\ y_i\\ z_i\end{pmatrix} - \mathrm{r}^{WC}\right) + \mathrm{m}(\theta_i, \phi_i)\right), \tag{39}$$

where R^{CW} is the rotation matrix based on the camera orientation q^{WC} and where $\mathrm{m} = [\cos\phi_i \sin\theta_i \quad -\sin\phi_i \quad \cos\phi_i \cos\theta_i]^T$.

The vector h^C is a non-unit directional vector. The angles representing this directional vector (azimuth and elevation) are calculated as:

$$\begin{pmatrix}\theta_i^C\\ \phi_i^C\end{pmatrix} = \begin{pmatrix}\arctan(\mathrm{h}_x^C, \mathrm{h}_z^C)\\ \arctan\left(-\mathrm{h}_y^C, \sqrt{{\mathrm{h}_x^C}^2 + {\mathrm{h}_z^C}^2}\right)\end{pmatrix}. \tag{40}$$

The azimuth and elevation angles can then be used to find the position of the predicted feature in the image at the current camera location. Again, this conversion is straightforward due to the Ladybug 2 image being fully rectified and containing the entire visual sphere. Note that Equation (41) is the inverse of Equation (35).

$$\mathrm{h}_u = \begin{pmatrix}u_i\\ v_i\end{pmatrix} = \begin{pmatrix}\left(\frac{\theta_i}{\pi}\right)u_{max} + \left(\frac{u_{max}}{2}\right)\\ \left(\frac{\phi_i}{\pi/2}\right)v_{max} + \left(\frac{v_{max}}{2}\right)\end{pmatrix} \tag{41}$$

4.2.3 Feature Initialization and Normalization

The defining difference between bearing-only SLAM and bearing-and-range SLAM is the lack of a depth measurement in feature observations. One consequence is that the resulting maps and localization are dimensionless – the real world scale is not

determined unless other information is incorporated. To represent observed features without range information, the feature locations are initialized with an infinite range value defined using a six dimensional state vector. This vector encodes the feature location based on the ray from the camera pose at which the feature was first observed. This concept is depicted in Figure 4.6.

The six dimensional vector describing the location of the observed feature is

$$\mathrm{y}_i^W = (x_i \quad y_i \quad z_i \quad \theta_i \quad \phi_i \quad \rho_i)^T. \tag{42}$$

This vector describes a point located at

$$\mathrm{x}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i} \mathrm{m}(\theta_i, \phi_i), \tag{43}$$

where:

$$\mathrm{m}(\theta_i, \phi_i) = [\cos\phi_i \sin\theta_i \quad -\sin\phi_i \quad \cos\phi_i \cos\theta_i]^T. \tag{44}$$

The y_i^W vector contains the camera position at which the feature was first observed (x_i, y_i, z_i) as well as the azimuth and elevation angles which define the unit directional vector $\mathrm{m}(\theta_i, \phi_i)$. The range to the feature (d_i) is stored as the inverse depth $\rho_i = 1/d_i$, this allows the encoding of features at an infinite depth, i.e. $d_i = \infty$, $\rho_i = 0$. The ability to encode features with an infinite range allows a feature to be added to the filter as soon as it is observed and without any specialized initialization technique. As the estimate of the feature position improves during camera motion, the inverse depth value is refined to reflect the increased positional information. However, this depth value is arbitrary and only reflects the depth of the feature relative to the depth of other observed features; the overall map has no known scale. For more information on encoding feature range as inverse depth and the proof that inverse depth parameterization can be linearized for use in an Extended Kalman Filter, see [22].

In order to initialize a new feature in the filter, the real world elevation and azimuth angles must first be extracted from the image produced by the camera. This conversion from image coordinates to a world reference azimuth and elevation angle is examined in Section 4.2.2. The position of the optical centre is also required for feature initialization and is extracted directly from the current camera position.

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \mathrm{r}^{WC} \tag{45}$$

This information, along with an infinite initial depth estimate, is used to encode the new feature in the following format:

$$\mathrm{y}^{NEW} = (x_i \quad y_i \quad z_i \quad \theta \quad \phi \quad \rho_0)^T. \tag{46}$$

The new feature is then added to the state vector.

$$\mathrm{x}^{NEW} = \begin{pmatrix} \mathrm{x}^{OLD} \\ \mathrm{y}^{NEW} \end{pmatrix} \tag{47}$$

The state covariance is updated after feature initialization using:

$$\mathrm{P}^{NEW} = \mathrm{J} \begin{pmatrix} \mathrm{P}^{OLD} & 0 & 0 \\ 0 & \mathrm{R} & 0 \\ 0 & 0 & \sigma_{\rho_0} \end{pmatrix} \mathrm{J}^T. \tag{48}$$

Where R is the image noise covariance and J is the Jacobian of the initialization function:

$$\mathrm{J} = \begin{pmatrix} & \mathrm{I} & & 0 \\ & & & \vdots \\ & & & 0 \\ \dfrac{\partial \mathrm{y}}{\partial \mathrm{x}_{cam}} & 0 \quad \ldots \quad 0 & & \dfrac{\partial \mathrm{y}}{\partial \mathrm{h}} \end{pmatrix}. \tag{49}$$

The full expansion of the terms used in the Jacobian can be found in Appendix A.

Finally, the state vector must undergo quaternion normalization to ensure that its normal is equal to one, i.e. $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. To accomplish this, the state vector is updated as follows:

$$\mathrm{x}^{norm} = \begin{pmatrix} \mathrm{r}^{WC} \\ \mathrm{q}^{WC} \\ \hline |\mathrm{q}^{WC}| \\ \mathrm{v}^{W} \\ \omega^{C} \\ \mathrm{x}_{map} \end{pmatrix}. \tag{50}$$

The covariance should also be updated with the Jacobian of the transformation.

$$\mathrm{P}^{norm} = \mathrm{J}_{norm}\mathrm{P}\mathrm{J}_{norm}^{T} \tag{51}$$

$$\mathrm{J}_{norm} = \begin{pmatrix} \mathrm{I} & 0 & 0 \\ 0 & \dfrac{\partial \mathrm{q}^{norm}}{\partial \mathrm{q}} & 0 \\ 0 & 0 & \mathrm{I} \end{pmatrix} \tag{52}$$

The full expansion of the terms used in the Jacobian can be found in Appendix A.

4.2.4 Correspondence Estimation

Determining correspondences in Civera's EKF monocular SLAM algorithm is a two stage process. The first stage occurs during the prediction step of EKF and uses the motion model to predict the location of existing features in the next observation. Every feature in the current state vector is predicted to narrow the search region during matching and improve real-time processing.

In the second stage, visual feature matching is achieved by comparing image patches around existing and new features within the search region. Image patches with a high level of cross-correlation are identified as initial matches and are added to the set of individually compatible matches (z) for processing by a correspondence rejection algorithm. In the case of Civera's monocular SLAM, correspondence rejection is handled by a 1-point RANSAC algorithm [51]. The name of this algorithm refers to

the single match pair required to generate a hypothesis for evaluation. The location of the matched feature is used to perform an update of the state vector alone (not the covariance matrix) which is then used to predict the match locations for the other features in the set z. Matches in the set z that are within the 99% probability region of the prediction are classified as low innovation inliers and support the current hypothesis. Once the most supported hypothesis is determined, the inliers are used to recalculate the state vector update and also determine the appropriate covariance matrix update. The updated covariance matrix is then used to identify high innovation inliers that were initially classified as outliers. The set of high and low innovation inliers are retained and used to perform the update step of the system.

4.3 EKF Bearing-Only SLAM Algorithm

The mathematical definitions required to perform EKF based bearing-only SLAM were outlined in the previous section. Now that these definitions have been made, the Extended Kalman Filter algorithm can be presented. This algorithm is based on the omnidirectional monocular SLAM algorithm by Rituerto, Puig and Guerrero [72] which is a modified version of the original pinhole algorithm by Civera *et al.* [51].

```
Input:  camera_images
Output: camera_pose
        feature_map

image = get_next_image
filter = initialize_features(filter, image)

For i=1 to (num_images – 1)
    image = get_next_image
    filter = ekf_prediction(filter)
    features = extract_features(image)
    matches = match_features(filter, features)
    filter = ekf_update(filter, matches)
End For

camera_pose = filter.camera_pose
feature_map = filter.feature_map
```

Algorithm 4.1 – The EKF bearing-only vision-based SLAM algorithm.

The EKF based algorithm requires only a stream of images from a camera as the input. The output from the algorithm is a vector containing the camera pose at each time step and another vector containing the position of each of the observed features in the feature map. The algorithm begins by extracting a set of features from the first image in the sequence. The technique used for feature extraction is the Scale Invariant Feature Transform (SIFT) by Lowe [28], which is discussed in detail in Section 2.3.3. The image is divided into a user defined grid space and only one feature from each grid square on the image is stored in the state vector to reduce computation times (see Figure 4.8). A numerical description of each feature is stored along with the position of the feature. The feature initialization process is described in detail in Section 4.2.3. The initialization process must occur once before the main loop in the program begins so that there are existing features in the state vector to predict locations for, and match with, during the first iteration of the loop.

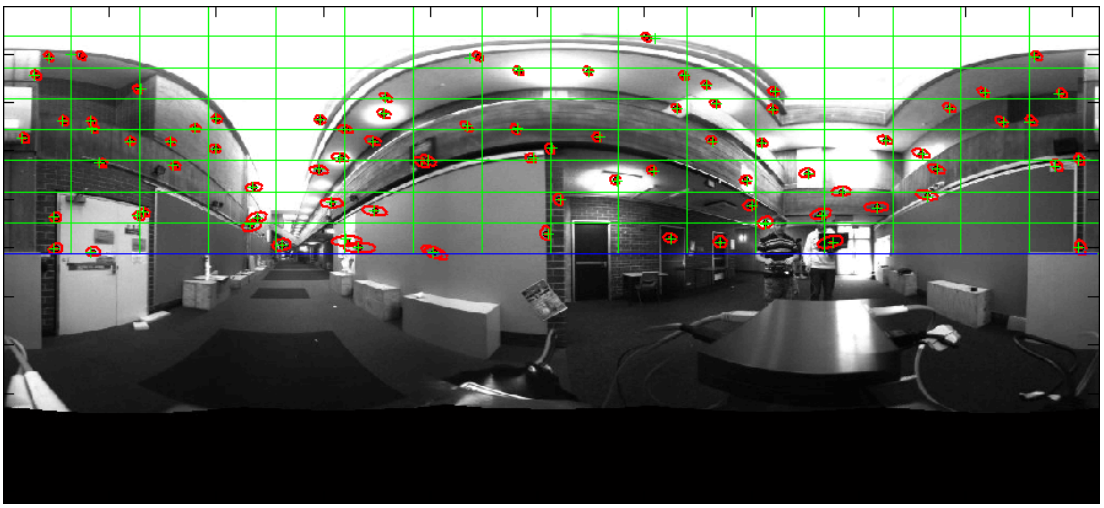


Figure 4.8 – The Ladybug image is covered by a user defined grid allowing only one feature per grid square to be stored in the state vector. The green crosses are the feature position predictions. The red dots are the matched feature locations. Each red dot is surrounded by a red ellipsoid representing the uncertainty of the feature location. Note that only the top half of the image is used for feature extraction as the bottom half contains parts of the mobile platform which would conflict with the static environment in the filter.

The program then enters the main loop. The first step in the main loop is to acquire the next image from the camera stream. The EKF prediction step is then performed. The prediction step predicts the new camera position and orientation based on the motion model defined in Section 4.2.1. The features already stored within the state vector can then have their locations in the camera's predicted visual sphere calculated. This calculation is based on the process described in Section 4.2.2.

Features are then extracted from the current image using SIFT. The entire set of extracted features is then matched with the existing features in the state vector using the numerical descriptors. A set of matched features is produced. Features in the state vector which are not matched for a user defined number of consecutive time steps are replaced with a new feature from the current image located in the same grid square.

Finally, the algorithm enters the update step. In this step the location of predicted features and matched features are compared. If the offset between the two falls outside a user defined threshold, the match is rejected and not used for the state update (see Figure 4.9). The features passing the discrepancy test are then used to update the camera pose and feature map.

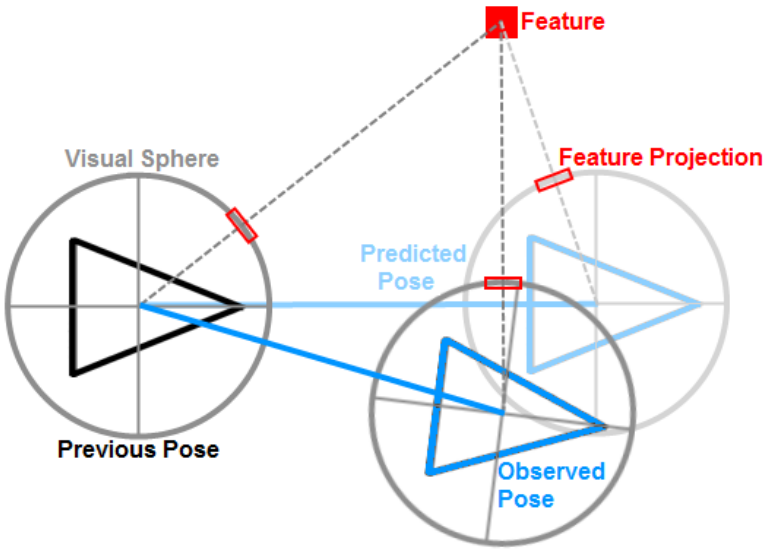


Figure 4.9 – An offset between predicted and observed pose results in an offset in the feature projection in the visual sphere. The feature projection offset can be used to reject outlier feature matches.

An example of the results produced in the Matlab environment can be seen in Figure 4.10. The left window contains the current image supplied by the Ladybug camera. The green grid represents the division of the image into ‘windows’ which allow only one tracked feature each. The window based feature limitations are implemented to maintain an even spread of features across the image when there is a low maximum number of maintained features. The green crosses are the predicted feature locations and the red dots are the measured feature positions. The red ellipsoid around each red

dot represents the uncertainty of the feature location. The right window displays a 2D projection of the resulting 3D map. The vehicle pose is represented by a black triangle. The vehicle always begins at the origin coordinates (0,0,0). The blue line from the origin to the current vehicle pose represents the localization path. The position and uncertainty of older established features that are no longer being predicted are shown in magenta. Features currently being observed are shown in red. The red lines represent features that have been observed in only a single image frame and therefore have no depth estimate. The scale of the right side window is purely arbitrary as there is no depth information supplied by other sources at this stage.

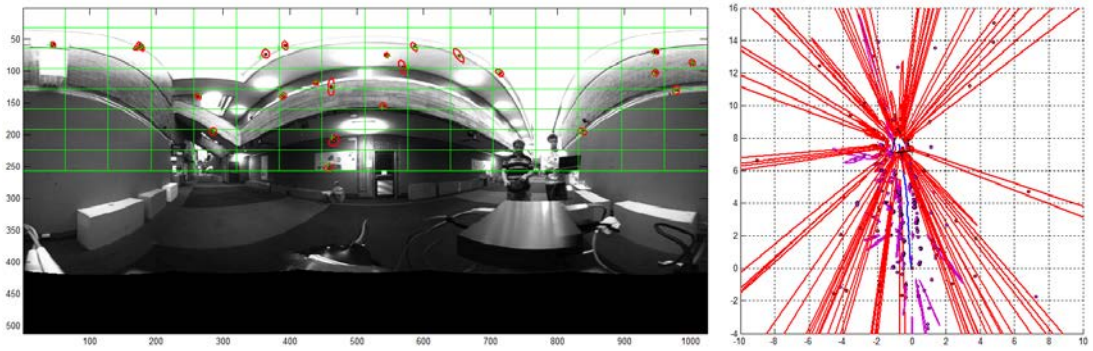


Figure 4.10 – An example of the results screen produced by the Matlab implementation of vision-based localization modified to work with the Ladybug omnidirectional camera.

4.4 Sensor Fusion and Multisensor SLAM

The major issue with bearing-only SLAM is that only feature bearing information can be measured, it is therefore impossible to retrieve real world scale without utilizing additional information. Although the vehicle and each feature in the state vector are positioned relative to other features, the scale is arbitrary. The scaling factor between the map and the real world is unknown, and therefore the information is incomplete for surveying applications.

To provide scaling information to the state vector and to also provide the high accuracy, dense, large scale maps required for surveying applications in underground mines, 3D depth information from a 3D laser scanner is fused with the vision-based bearing-only SLAM. Chapter 5 discusses the use of a 3D laser for producing extensive, dense, survey quality maps and Chapter 6 examines the sensor fusion

relationships between vision-based bearing-only localization and large scale mapping from point clouds.

Chapter 5

Large Scale Mapping from Point Clouds

5.1 Introduction

Building survey quality three dimensional maps is a task that is most commonly performed through the use of a scanning laser system. The resulting sets of laser data points, or ‘point clouds’, are combined to form large scale environmental maps. However, the alignment and concatenation of two point clouds (hereafter referred to as registration) is not a trivial task, even with significant manual input. Survey quality maps are accurate to ±2mm for reflector-based surveying and ±6mm for reflectorless surveying. Therefore, to build a large scale survey quality map, accurate alignment of point clouds is required to maintain the quality of the registered map.

The discussion of large scale mapping from point clouds will begin with a break-down of the current technologies and techniques being applied to the task of 3D mapping. The advantages and disadvantages of competing approaches are examined in detail to ensure that the appropriate approach is selected for the task of large scale 3D mapping in underground mining environments.

The need for an autonomous solution to the alignment of point clouds is also examined and compared to the current industry standard semi-autonomous techniques. This is followed by an in-depth discussion of the Iterative Closest Point (ICP) registration technique implemented in our own mapping system. Algorithms related to registration and data reduction processes are discussed, along with an approach that allows an extension to the maximum point cloud origin offset during registration.

Finally, the requirement for an initial point cloud location (pose) estimate from an external source is investigated. A brief overview of the solution to the autonomous production of this initial pose estimate is provided prior to a full discussion of the proposed multisensor technique in the following chapter.

5.2 Acquisition of Depth Information

The production of accurate three dimensional maps of large scale environments is a task with applications in many disciplines. These applications include, but are not limited to, the mapping of underground mine tunnels (both new and abandoned), dense urban areas, archaeological excavations, forensic crime scenes and natural disaster areas. There are many ways to approach this task, as demonstrated by the extensive literature on unique mapping systems. There are also several distinctive sensor types that can be used for the purpose of mapping. These include stereo vision cameras, time of flight cameras, structured light cameras, 2D laser scanners and 3D laser scanners. There are many implementations of each sensor type produced by a variety of manufacturers. Table 5.1 contains a summary of example characteristics for each ranging sensor type. It can be seen that only laser scanners provide the range and accuracy required for the survey quality mapping of large scale environments.

Sensor	Range	Precision	Angular Resolution	Field of View	Freq.
Stereo Vision Point Grey Bumblebee 2	Infinite	Variable	H = 0.15° V = 0.15°	H = 97° V = 73°	48Hz
Time of Flight SwissRanger SR4000	0.8-5.0m	±10mm	H = 4.09° V = 4.24°	H = 43° V = 34°	50Hz
Structured Light Microsoft Kinect	0.6-4.6m	±25mm	H = 0.09° V = 0.09°	H = 58° V = 44°	30Hz
2D Laser SICK LMS511	0-80m	±40mm	H = 0.25° V = n/a	H = 190° V = n/a	35Hz
Multi 2D Laser Velodyne HDL-64E	0.1-120m	±20mm	H = 0.09° V = 0.4°	H = 360° V = 26°	15Hz
3D Laser Leica ScanStation C10	0.1-300m	±6mm	H = 0.06° V = 0.06°	H = 360° V = 270°	0.0018Hz

Table 5.1 – Comparison of ranging sensor types. Average error values are provided for precision; however, these values are dependent on the depth of the scene.

The range and accuracy advantages of laser scanners are offset by some significant disadvantages. The most notable of which is the lack of real-time information from 3D laser scanners and the limitations of 2D laser scanners to a single sensing plane. All of the range sensing alternatives mentioned in Table 5.1 aim to address these shortcomings by providing 3D range information in real-time; however, the resulting loss of range and accuracy makes large scale 3D mapping unproductive. The solution then is to minimize the limiting effects of laser scanners.

The underlying technology in most 2D laser systems, multiple 2D laser systems and 3D laser systems is practically identical. All three systems are based on the rotation of a single laser beam in a 2D plane. The emitted laser beam is reflected by the

environment and the reflected light is captured by a photodiode receiver and the time of flight is calculated (see Figure 5.1). Since the speed of light is known, time of flight can be used to determine the distance to the sensed environmental object.

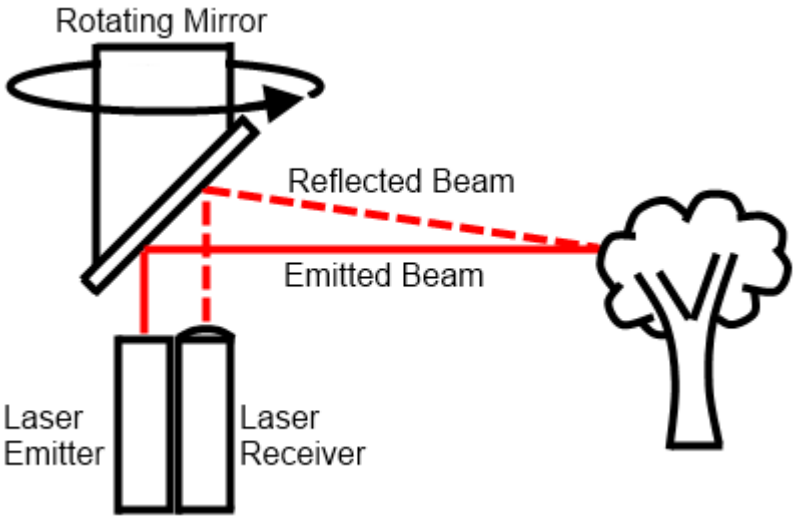


Figure 5.1 – The technology behind 2D laser scanners.

The technology used in 2D laser scanners is limited to a two dimensional plane. Multiple 2D laser scanner systems and 3D laser scanner systems provide two alternative approaches for adapting 2D laser scanning technology to 3D applications. Multiple 2D laser scanner systems such as Velodyne's HDL-64E model scanner use a large number of 2D laser scanners (64 for the HDL-64E) to cover a portion of the vertical field of view (see Figure 5.2). The whole system is then rotated to cover the horizontal field of view as well. This approach can produce real-time 3D sensing, but has a strictly limited vertical field of view and poor vertical angular resolution (see Table 5.1). 3D laser scanners such as Leica's ScanStation produce a 3D scan by rotating the scan plane through a perpendicular axis. This allows the laser to periodically scan the majority of the 3D field of view. This approach is limited by a poor scan rate (see Table 5.1) and therefore the sensor must remain stationary for extended periods of time while a scan is completed. For this reason, such 3D scanners are not suitable for highly dynamic environments.

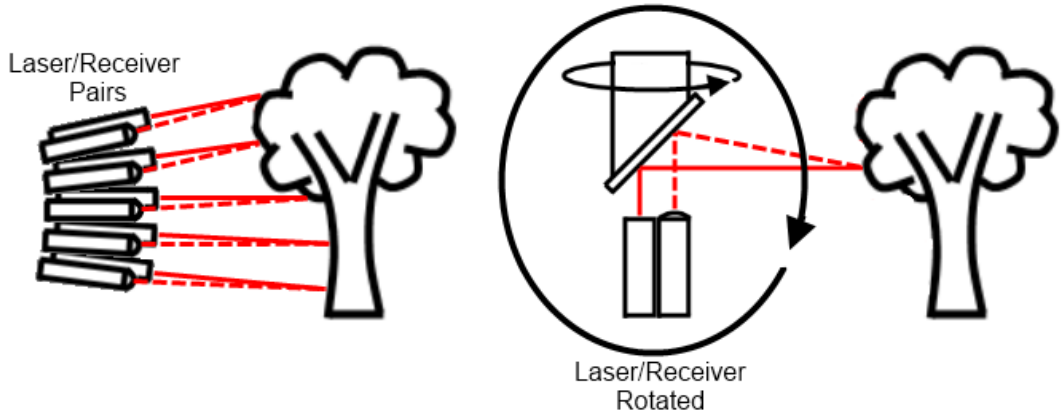


Figure 5.2 – (Left) Multiple 2D laser scanners are used to cover a portion of the vertical field of view. (Right) A single 2D laser is rotated about a secondary perpendicular axis to allow scanning of the majority of the 3D field of view.

5.3 3D Registration for Large Scale Mapping

Section 5.2 concluded that 3D laser scanners were the only sensor capable of providing the range and accuracy required for survey quality map building applications. However, the resulting latency limits the sensor to the stationary scanning of mostly static environments. Therefore, to produce a large scale 3D map, discrete scans must be obtained at a sparse interval to reduce overall mapping time to be within acceptable limits. These discrete scans must then be co-registered to form the overall map.

The distance separating the discrete scan locations is an important variable that strongly influences several key mapping characteristics. The scan spacing should therefore be selected based on the parameters of the large scale mapping task. The point density of the final map is directly related to the discrete scan spacing. By increasing the distance between scans, the overlap between scans will be reduced and this in turn will reduce the likelihood of successful registration. Increasing scan spacing also increases the impact of occlusions on the final map. The advantage of increasing scan spacing is found in the reduction in scan time and the fewer number of registration steps required to complete the map (assuming successful registration). These factors are examined in Figure 5.3.

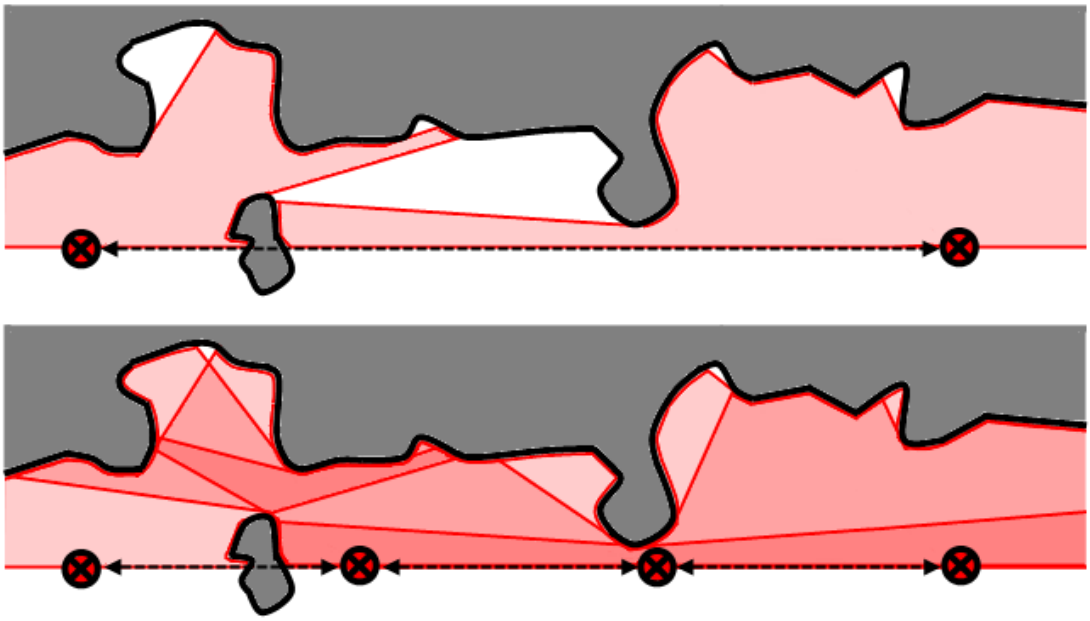


Figure 5.3 – The characteristics caused by the spacing of discrete laser scans. Large spacing (top) results in low point density, reduced overlap and problems with occlusions, yet benefits from a significant reduction in scanning and processing time (assuming successful registration). Smaller spacing (bottom) results in higher point density, increased overlap, and robustness to occlusions, yet suffers from a significant increase in scanning and processing time.

The difficult task of accurately co-registering discrete 3D laser scans has seen many techniques put forward by various research groups. However, it is the most basic technique that still dominates industrial use of 3D laser scanners. The simplest way to align 3D laser point clouds is to insert key markers into the scene before the scan is performed and then manually (or semi-automatically) align these markers in post-processing. This basic technique has grown over the years so that most commercial registration software packages now perform semi-autonomous alignment; however, there is still the need to manually place markers in the scene and tag the markers in post-processing. The resulting process is slow and labour intensive.

The alternative to the commercial registration process is a fully autonomous registration system. If the environment to be mapped is outdoors and free from major occlusions, Global Positioning System (GPS) satellites may be used to provide the global position of each scan location, eliminating the use of manually placed markers. Unfortunately, GPS positioning produces an average error of around 20 metres without the use of external ground based equipment or expensive subscription-based commercial systems. To achieve the accuracy necessary for scan registration, external instrumentation techniques such as static baseline surveying or

differential GPS are required. Successful registration of a 3D point cloud to an existing Digital Terrain Model (DTM) was achieved by Du and Teng through the use of static baseline surveying to calculate the earthwork volume of a landslide [74]. The required external instrumentation may improve global consistency; however the process is still only semi-autonomous and therefore contains limitations that should be avoided when designing a fully autonomous, modular mapping system.

Indoor and underground environments usually do not have access to a reliable GPS signal resulting in alternative techniques being developed for point cloud registration. Wulf *et al.* use Inertial Measurement Unit (IMU) data combined with wheel odometry readings and 2D localization to register 3D laser scans that are produced on the move [75]. Again, the simplification of a six degree of freedom localization task to a three degree of freedom localization task does not translate well to complex environments such as those found underground. IMU and odometry data is prone to error and drift and prevents the system being self-contained and portable.

The most effective way to combine 3D point clouds is through the use of a fine scale registration algorithm such as Iterative Closest Point (ICP) by Besl and McKay [11] (discussed in detail in Section 5.4) or 3D Normal Distribution Transform (3D NDT) by Magnusson and Ducket [76]. These registration algorithms require an approximate initial scan pose estimate to produce an accurate alignment result. The multisensor SLAM system detailed in our own work uses ICP for final registration. The technique for providing initial scan pose estimates without resorting to the use of external equipment is discussed in Section 5.5.4.

5.4 Registration Techniques

5.4.1 Iterative Closest Point

The Iterative Closest Point (ICP) algorithm was first published by Paul Besl and Neil McKay in 1992 [11]. The algorithm is designed to first determine the distance between each point in a set of data and a ‘model shape’. The mean squared distances between the nearest neighbouring points in the data set and the model shape are determined, then the optimum transformation is calculated to minimize these

distances. The optimum transformation is applied to the data set and the new nearest neighbours are then determined. The process is repeated until the average mean squared nearest neighbour distance falls below a selected threshold. Each step of this process will now be examined in detail.

A data set (P) and a model shape (X) are supplied to the algorithm. If the data set and model shape are not supplied in point cloud format, they must be converted. If curves or surfaces are supplied, they must first be converted to a line set or triangle set respectively using a simplex-based approximation. For a parametric space curve $C = \{\vec{r}(u)\}$, the equivalent polyline can be computed $L(C, \delta)$ such that the piece-wise linear approximation never deviates from the space curve by more than a specified distance δ. Similarly, for a parametric surface $S = \{\vec{r}(u, v)\}$, the equivalent triangular set can be computed $T(S, \delta)$ such that the piece-wise triangular approximation never deviates from the surface by more than a specified distance δ. Finally, the resultant polyline or triangular set is converted to point cloud form. Polylines are converted to point clouds using their endpoints and triangle sets are converted to point clouds using their vertices.

The minimum distance d between a single point $\vec{p}$ from the data set P and a point $\vec{x}$ from the model shape X is determined by:

$$d(\vec{p}, X) = \min_{\vec{x} \in X} \|\vec{x} - \vec{p}\|. \tag{53}$$

The closest point in X that yields the minimum distance is denoted $\vec{y}$ such that $d(\vec{p}, \vec{y}) = d(\vec{p}, X)$, where $\vec{y} \in X$. The resulting set of closest points is denoted Y. To determine the rotation (as a unit quaternion $\vec{q}_R = [q_0 q_1 q_2 q_3]^t$) and translation ($\vec{q}_T = [q_4 q_5 q_6]^t$) that minimize the distance between the data set P and the corresponding closest points Y, a least squares registration step is performed. The mean square objective function to be minimized is:

$$f(\vec{q}) = \frac{1}{N_p}\sum_{i=1}^{N_p} \|\vec{x}_i - \boldsymbol{R}(\vec{q}_R)\vec{p}_i - \vec{q}_T\|^2, \tag{54}$$

$$\text{where, } \boldsymbol{R} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2+q_2^2-q_1^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2+q_3^2-q_1^2-q_2^2 \end{bmatrix}. \tag{55}$$

The mean values of set P and X are given by

$$\vec{\mu}_p = \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{p}_i \qquad \text{and} \qquad \vec{\mu}_x = \frac{1}{N_x}\sum_{i=1}^{N_x}\vec{x}_i. \tag{56}$$

The cross-covariance matrix Σ_{px} of the sets P and X is given by

$$\Sigma_{px} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[(\vec{p}_i - \vec{u}_p)(\vec{x}_i - \vec{\mu}_x)^t\right] = \frac{1}{N_p}\sum_{i=1}^{N_p}[\vec{p}_i\vec{x}_i^t] - \vec{\mu}_p\vec{\mu}_x^t. \tag{57}$$

The cyclic components of the anti-symmetric matrix $A_{ij} = \left(\Sigma_{px} - \Sigma_{px}^T\right)_{ij}$ are used to form the column vector $\Delta = [A_{23} \quad A_{31} \quad A_{12}]^T$. This vector is then used to form the symmetric 4x4 matrix $Q(\Sigma_{px})$ where $\mathbf{I}_3$ is the 3x3 identity matrix and *tr* (trace) is the sum of the matrix diagonal.

$$Q(\Sigma_{px}) = \begin{bmatrix} tr(\Sigma_{px}) & \Delta^T \\ \Delta & \Sigma_{px} + \Sigma_{px}^T - tr(\Sigma_{px})\boldsymbol{I}_3 \end{bmatrix} \tag{58}$$

The unit eigenvector $\vec{q}_R = [q_0 \quad q_1 \quad q_2 \quad q_3]^t$ corresponding to the maximum eigenvalue of the matrix $Q(\Sigma_{px})$ is selected as the optimal rotation. The optimal translation vector is given by

$$\vec{q}_T = \vec{\mu}_x - \boldsymbol{R}(\vec{q}_R)\vec{\mu}_p. \tag{59}$$

The optimal translation and rotation is then applied to the point set P. If the mean square error falls below a predetermined threshold, the iteration is terminated, otherwise the process is repeated. The overall iterative closest point process is summarized in the following algorithm:

```
INPUT:  P (Initial data set)
        X (Model shape)
OUTPUT: P(k) (Aligned data set)

k = 0
P(k) = P

While Error < T (Tolerance)
    Y = find_nearest_points(P(k), X)
    (Trans, Rotn, Error) = mean_square_error(P(k), Y)
    P(k+1) = transform(P(k), Trans, Rotn)
End While
```

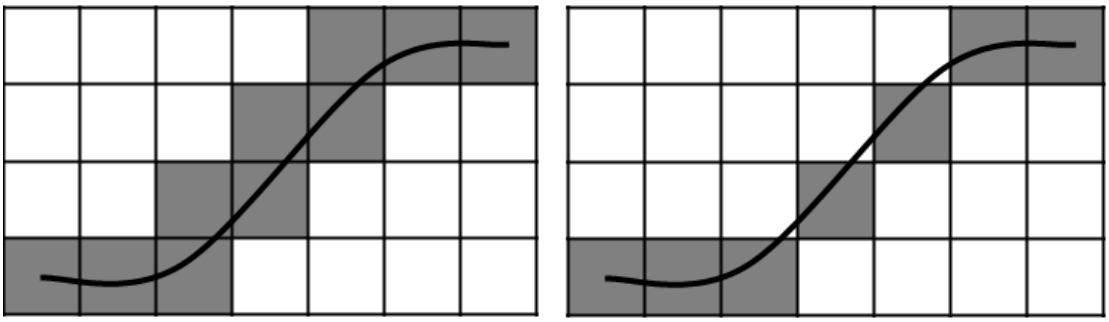
Algorithm 5.1 – The Iterative Closest Point (ICP) algorithm [11].

5.4.2 Voxel Based Reduction

The processing cost for the Iterative Closest Point (ICP) algorithm is $O(N_p \log N_x)$ for the closest point association step and $O(N_p)$ for both the computation and application of the resultant transformation. Therefore, the most effective way to reduce processing time in order to improve performance in field deployment is to reduce the number of points used for ICP. An effective technique for the reduction of point cloud density without the sacrifice of shape complexity can be found in voxel based reduction.

Voxels are unit squares (in the case of 2D applications) or unit cubes (in 3D applications) with a user defined scale. Voxel based reduction has its roots in volume graphics where geometric objects were converted from their continuous geometric representation into a set of voxels that provided the best approximation of the continuous object [77]. This set of voxels did not simply contain all voxels that were intersected by the continuous object body, as this would often result in an unsatisfactorily coarse result as seen in the 2D representation in Figure 5.4(a). The

2D solution is to rasterise (convert to pixels) the continuous line while maintaining separation. Here separation refers to the maintained independence of the two sides of the line as seen in Figure 5.4(b).



(a) Voxel reduction using all intersections. (b) Voxel reduction after being rasterized.

Figure 5.4(a) The voxel reduction result when using all voxels containing part of the continuous object. (b) The voxel reduction result after being rasterized. Note the continued separation of the two sides of the voxel approximation of the continuous line.

The 2D voxel solution does not translate to 3D applications as the 2D definition of separation cannot be applied to 3D surfaces. The reason for needing a different definition for 3D separation is that voxel based surfaces cannot be defined as an ordered sequence of voxels and there is also no specific number of adjacent voxels for each surface voxel (see Figure 5.5). To alleviate the shortcomings of the 2D definition, 3D discrete topology is used to define 3D separation. 3D discrete topology uses a set of 3D Euclidean grid points known as 3D discrete space. A voxel is a unit cubic which is centred at each grid point. Voxels containing part of the continuous 3D surface model are said to be ‘black’ while all others are said to be ‘white’. The adjacent relations of the black voxels are then classified into one of three categories: 6-adjacent, 18-adjacent or 26-adjacent. Two voxels are 26-adjacent if they share a vertex, edge or face, they are further classified as 18-adjacent if they share only an edge or a face, and finally they are still further classified as 6-adjacent if they share only a face (see Figure 5.5).

It can then be said that a sequence of black or white voxels having the same adjacency (N-adjacent) are an N-path if all consecutive voxel pairs are N-adjacent. A set of voxels is defined as N-connected if an N-path exists between every voxel pair. To determine if 3D separation is occurring, assume there is a voxel space denoted Σ which includes a single subset of black voxels denoted S. If the complementary set of

white voxels ($\Sigma - S$) is not N-connected, i.e. there are two or more white N-connected components, then S is N-separating in Σ. The voxel approximation can then be simplified (by converting black voxels to white) and retested to confirm continued separation. This approach allows for the successful reduction of the voxel grid approximation while maintaining 3D separation.

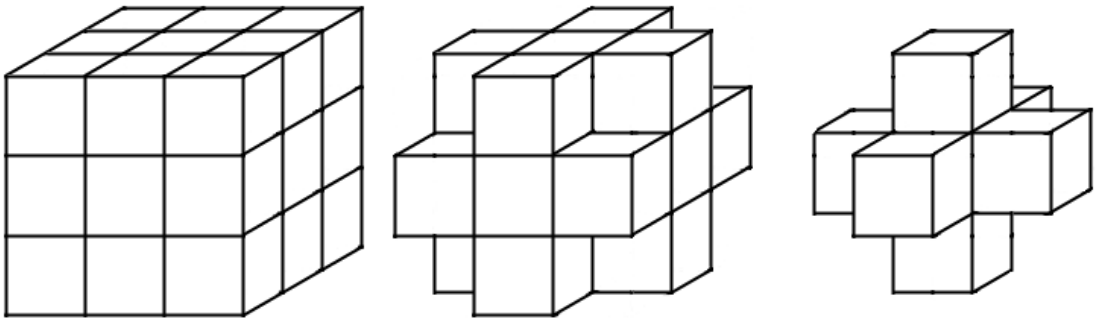


Figure 5.5 – From left to right, voxels that are 26-adjacent to the centre voxel, voxels that are 18-adjacent to the centre voxel and voxels that are 6-adjacent to the centre voxel.

For 3D point clouds the voxel reduction process is often simplified as there is less dependency on successful separation. Implementations may simply replace a set of points occurring within the same voxel cube by a single point representing their average position. The algorithm below is an example approach to this simplistic point cloud voxel reduction.

```
INPUT:  voxel_size (Size of voxel unit length)
        P (Point cloud)
OUTPUT: Q (Voxel reduced point cloud)

For i=1 to size(P)
    vox_pos = find_current_voxel(P(i),voxel_size)
    voxel_grid(vox_pos).values += P(i)
    voxel_grid(vox_pos).count += 1
End For

Q = voxel_grid(:).values/voxel_grid(:).count
```

Algorithm 5.2 – Simple voxel based reduction of a point cloud.

This approach is crude yet has benefits in processing cost. This style of voxel based reduction is implemented in the Point Cloud Library (PCL) open source project which was utilized in our own work and is discussed further in Section 5.4.3.

5.4.3 Point Cloud Library

The Point Cloud Library (PCL) Project is an open source, standalone project for the processing of 3D point clouds [78]. The PCL framework contains leading edge algorithms for filtering, feature estimation, surface reconstruction, registration, model fitting and segmentation. These are combined to form higher order tools for tasks such as mapping and object recognition. PCL is released under the terms of the Berkeley Software Distribution (BSD) license and can be accessed at http://pointclouds.org.

The PCL voxel grid reduction and iterative closest point algorithms were incorporated into our own mapping software. Voxel grid reduction is detailed in Section 5.4.2 and iterative closest point is detailed in Section 5.4.1.

5.5 Multisensor SLAM registration

The reliable registration of point clouds using the Iterative Closest Point (ICP) algorithm is only possible if an initial pose estimate is supplied. Without a sufficiently accurate initial pose estimate, the optimization step of ICP is likely to reach a local, rather than global, minimum. The algorithm may even fail completely if there are not enough points within the user defined point association radius (the maximum distance allowed for a nearest neighbour search). Our approach to providing an initial pose estimate is introduced in Section 5.5.4 and is explored in detail in Chapter 6. Our implementation of point cloud registration is examined in Sections 5.5.1 through 5.5.3. These sections include the optimization of the algorithm with respect to reliability, precision and processing time.

5.5.1 Implementation

The standard Iterative Closest Point (ICP) algorithm is single cycle, is applied to every point and has hard coded values for parameters such as point association radius, convergence threshold and maximum number of iterations in the optimization step. This rigid approach requires precise tuning for every new application and is therefore not well suited to autonomous implementation. To improve the applicability and robustness of our own implementation of ICP, several modifications were made.

The first modification to the ICP implementation was the introduction of a two stage registration process. During initial experimentation it was found that the parameters required to reliably align two point clouds had to be relaxed in order to accommodate the possibility of notable error in the initial pose estimate. This mainly involved increasing the point association radius (used as a threshold for nearest neighbour searches) and the RANSAC rejection threshold (discussed later in this section). The result of relaxing these parameters was reliable 'rough' point cloud alignment, but poor fine scale alignment. To produce a survey quality map, this result was not acceptable. A two stage registration process was therefore introduced to address the lack of fine scale precision.

A suitable point association radius for coarse alignment in the initial ICP stage was found through experimentation using laser data collected form the interior of Curtin University's architecture building. It was found that a point association radius that was about 25% of the size of the estimated distance between scan locations worked well. This value was reduced to about 1.5% for the fine registration stage. These values were used throughout the experimental results reported in Chapter 8. The pose estimate supplied by the coarse registration stage is of sufficient accuracy to allow a reliable fine registration result despite the significantly reduced point association radius. The registration results in Figure 5.6 and Figure 5.7 demonstrate the effectiveness of the two stage registration technique. The results show the alignment using only the initial pose estimate (left), then coarse registration (middle) and finally fine registration (right). The alignment results are scored based on the average mean squared nearest neighbour offset between the two point clouds. The fitness scores for each of the results are $0.675m^2$, $0.178m^2$ and $0.165m^2$ respectively (i.e. an average

neighbour offset of 0.822m, 0.422m and 0.406m respectively). It is worth noting that these fitness scores are averaged across all point pairs and as such are only useful for comparing subsequent registration results for the same point cloud. The average is heavily influenced by scan specific outliers and is therefore not an accurate indicator of the quality of the fit compared to other point cloud results.

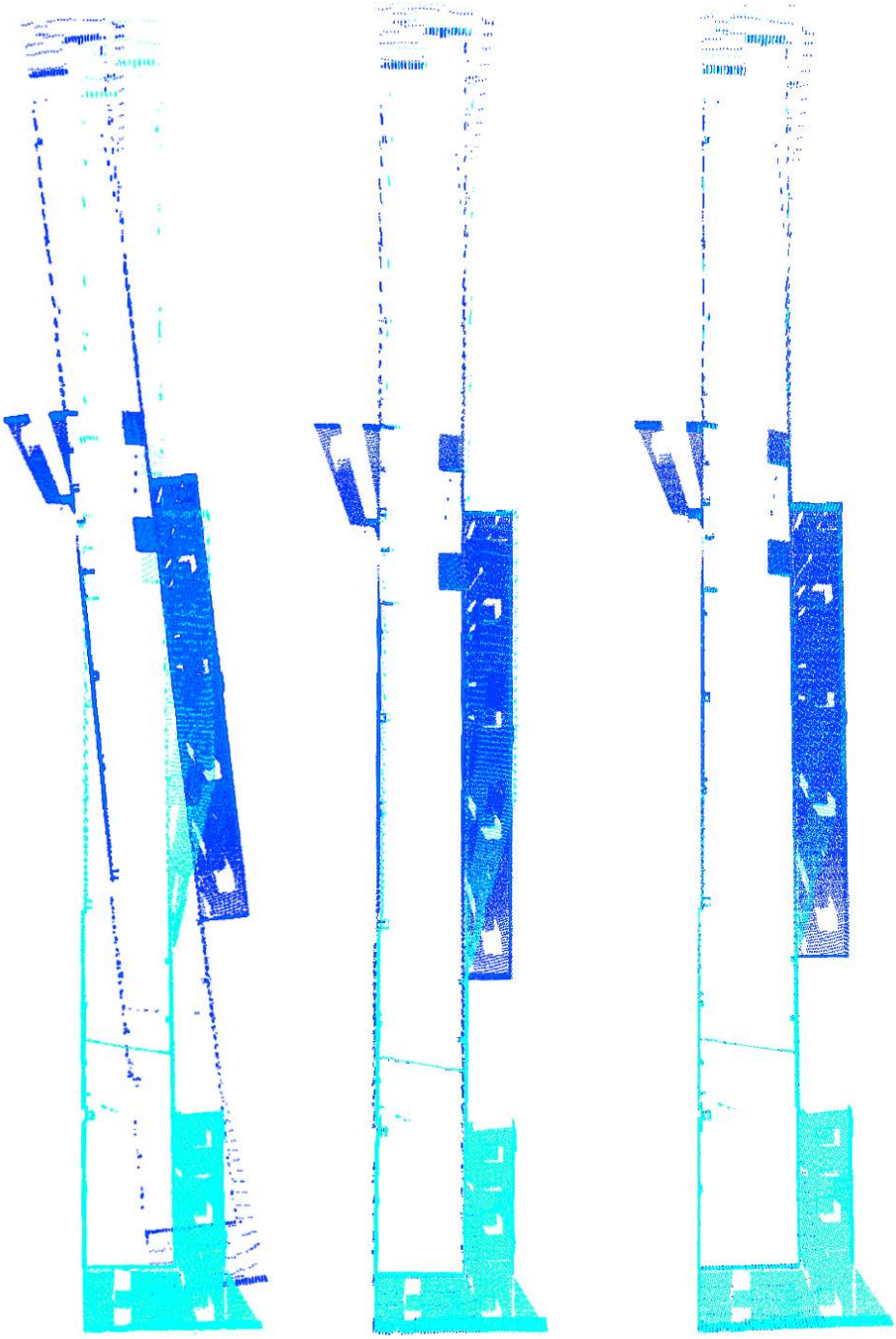


Figure 5.6 – Alignment of scans 1 (light blue) and 2 (dark blue) from the architecture building dataset. (Left) Scan alignment based on initial pose estimate, fitness score of 0.675m^2. (Middle) Scan alignment after coarse registration, fitness score of 0.178m^2. (Right) Scan alignment after fine registration, fitness score of 0.165m^2.

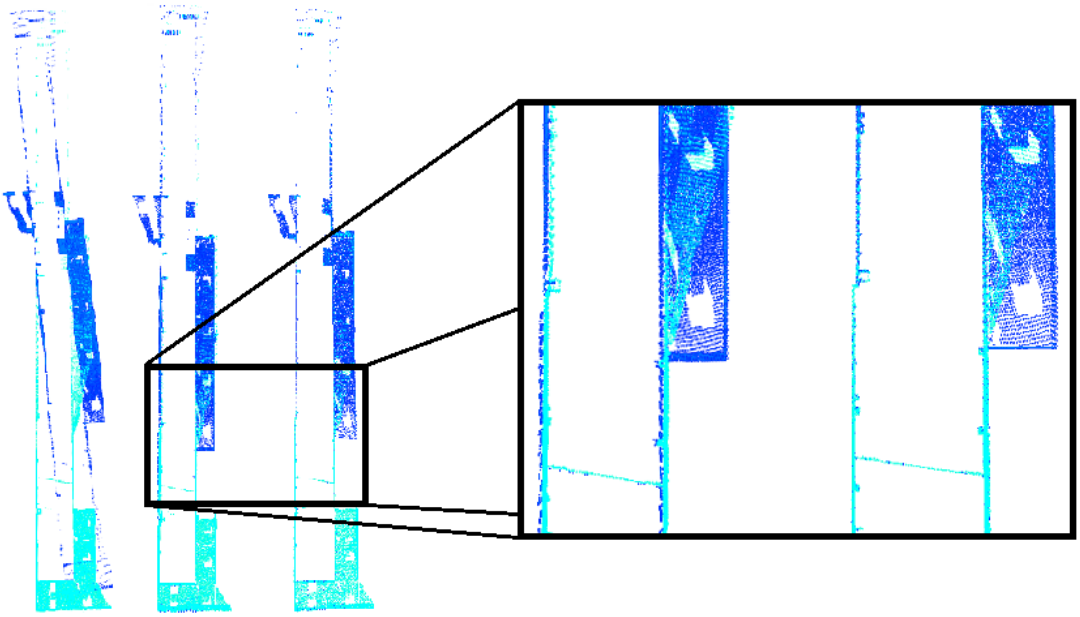


Figure 5.7 – Detailed look at the comparison between coarse and fine registration results from Figure 5.6.

Another improvement to the generic ICP algorithm is the inclusion of Random Sample Consensus (RANSAC) point rejection. The RANSAC algorithm in its generic form was proposed by Fischer and Bolles [35]. This point filtering technique is included as part of the Point Cloud Library default implementation of ICP and is used to reject points that are considered to be outliers. A point is considered to be an outlier if the distance between the point and its associated point in the existing map is greater than a set threshold after a transformation has occurred (as opposed to the point association radius which filters before a transformation). Again, through experimentation, it was found that an effective RANSAC rejection threshold value is generally around half the size of the point association radius. This rejection threshold was used throughout the experimental results in Chapter 8.

To further improve the robustness of our ICP implementation, random seeding was introduced to reduce the accuracy constraints on the initial pose estimate. The coarse registration step is executed 20 times and each execution is provided with a variation of the initial pose estimate. The pose variation is provided by a random number generator that produces a scaling factor between -30% and +30%. A different scaling factor is applied to the pose estimate for each of the six degrees of freedom. Registration is performed and the process is repeated 20 times. The result with the

lowest fitness score (the mean squared nearest neighbour distance) is passed to the fine registration step.

Finally, to improve processing time so that it was feasible to run 20 iterations of ICP in a real world deployment, voxel based reduction was also implemented (see Section 5.4.2). The size of the voxel reduction is larger for the coarse registration than for the fine registration. For a dataset collected within the Curtin University architecture building with an average point depth of ~10m (see Figure 5.9), it was found that a voxel grid of 30cm worked well during coarse registration (i.e. a grid of about 3% of the average point depth). This reduces the number of points in an average point cloud by about 99.4%, yet still provides sufficient structure for consistent registration. For the fine registration step, a voxel grid of 5cm is used, reducing the number of points by about 88.4% while maintaining the cloud density required for fine registration. These voxel grid sizes represented the optimum compromise between processing speed, robustness and accuracy during experimentation on the first three point clouds. The values were used throughout the experimental results reported in Chapter 8. The reductions occur on both the newly acquired data and the existing map. However, the final transformation is performed on the full point cloud which is then concatenated to the full existing map, this maintains data integrity.

The result of voxel grid reduction is a significant reduction in processing time. The cost of each iteration during ICP is:

$$O\left(N_p \log N_x + 2N_p\right), \tag{60}$$

where N_p is the number of points in the new data and N_x is the number of points in the existing map. The use of the previously mentioned voxel grid dimensions results in a reduction in processing time of 99.5% for coarse registration and 89.7% for fine registration.

5.5.2 Algorithm

The implementation of the modifications to generic ICP detailed in the previous section can be seen in the following algorithm. The algorithm is supplied with a new

point cloud, an existing map and a pose estimate. The voxel grid reduction is then performed on the new data and existing map to reduce processing time during the subsequent steps. A randomly generated pose based on the initial pose estimate is used to transform the new data before coarse registration is performed. This step is repeated 20 times with different random poses. The registration result with the lowest fitness score is passed to the fine registration step. Upon registration convergence, the transformations produced during the execution of the algorithm (random pose, coarse registration and fine registration) are all applied to the full data point cloud (i.e. without any voxel reduction). This transformed data is then concatenated with the full existing map (also with no voxel reduction).

```
INPUT:  P_data (New data point cloud)
        P_map (Existing map point cloud)
        Trans_pose (Pose estimate transformation)
OUTPUT: Q (Updated map)

P_data_30cm = voxel_reduction(P_data, size_30cm)
P_data_5cm = voxel_reduction(P_data, size_5cm)
P_map_5cm = voxel_reduction(P_map, size_5cm)

For i=1 to 20
   rand = random_number(-1, +1)
   Rand_pose = Trans_pose * rand * 0.3
   P_data_pose = transform(P_data_30cm, Rand_pose)
   (Fit, Trans_coarse) = ICP_coarse(P_data_pose, P_map_5cm)
   If Fit < Min_fit
      Trans_rand = Rand_pose
   End If
End For

P_data_5cm = transform(P_data_5cm, Trans_rand, Trans_coarse)
Trans_fine = ICP_fine(P_data_5cm, P_map_5cm)

Q = P_map
Q += transform(P_data, Trans_rand, Trans_coarse, Trans_fine)
```

Algorithm 5.3 – Modified ICP algorithm.

5.5.3 Registration of Datasets with Large Offset

The iterative closest point (ICP) registration of two point clouds which are offset by a large distance (in rotation and/or translation) is susceptible to error due to the high probability of ICP convergence to a local minimum. Common environments for large scale 3D mapping are elongated or corridor-like in shape, resulting in point clouds containing non-uniform data point distribution (as seen in Figure 5.8). A large offset between point cloud origins results in a reduced overlap between the model shape and the newly acquired dataset. This reduced overlap, combined with non-uniform point density distribution, produces the phenomenon seen in Figure 5.8 where the sparse section of the newly acquired dataset overlaps the dense section of the model shape. The sparse section of the model shape, in turn, overlaps with the dense section of the dataset.

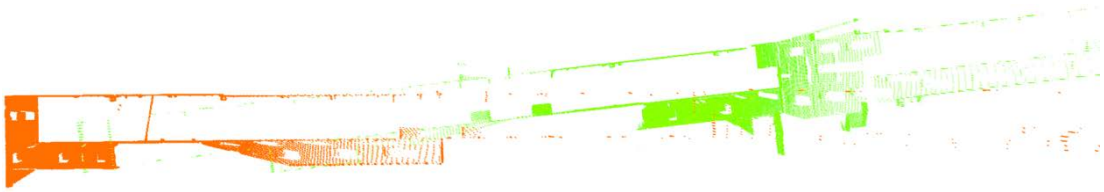


Figure 5.8 – Two point clouds from the Curtin University Architecture building with an offset of ~40 metres. Alignment before registration based on an average pose estimate. Note the sparse section of the data set (green) overlapping the dense section of the existing model shape (orange) and the sparse section of the model shape overlapping the dense section of the data set.

During the ICP registration process, the dense section of the dataset carries significantly more weight than the sparse section due to the quantity based preference of mean square optimization. This weighting is applied despite the fact that the sparse section (usually containing less than 20% of the points) contains about half of the scanned surfaces (see Figure 5.9). The overlap produced by large point cloud origin offset causes the ICP algorithm to focus on the alignment of the dense section of the dataset with the poorly distinguished sparse section of the existing model shape. The low number of target points during alignment with the model shape causes many dataset points to be associated with the same model shape point. This sharing of target points reduces the efficiency of the registration and makes it more prone to falling into local minima that satisfy a small, dense section of the dataset (see Figure 5.10). This phenomenon was experienced during our own work with ICP and point clouds with large offsets (greater than 20 metres point of origin separation).

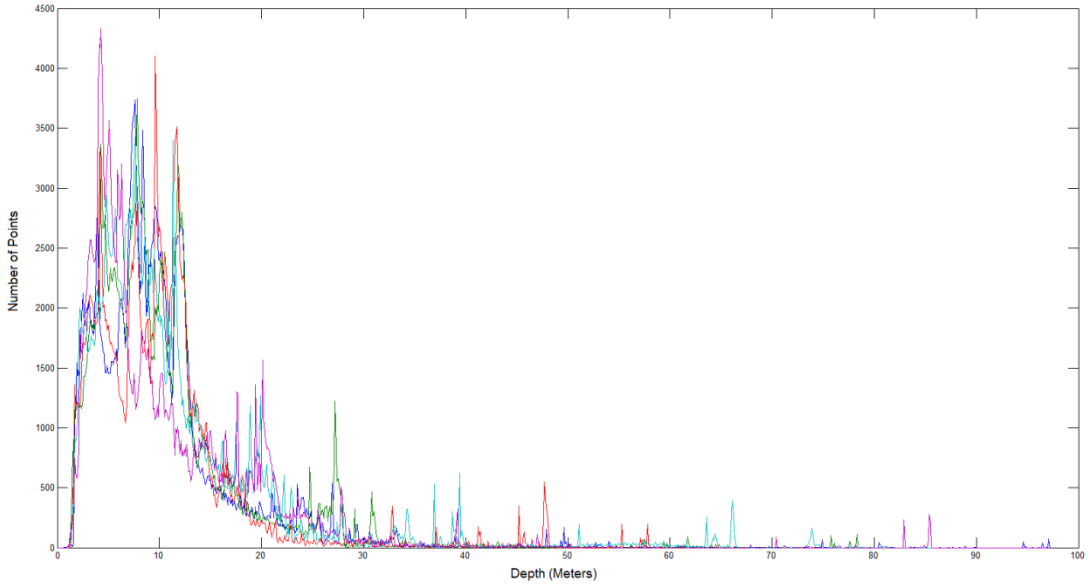


Figure 5.9 – The distribution of points across a range of depths for the five point clouds (voxel reduced) of the Architecture building dataset. Depth values are in reference to the location of each scan (i.e. not global coordinates). Note the number of points that occur in the 2 – 20 meter range, despite the 100 meter length of the building.

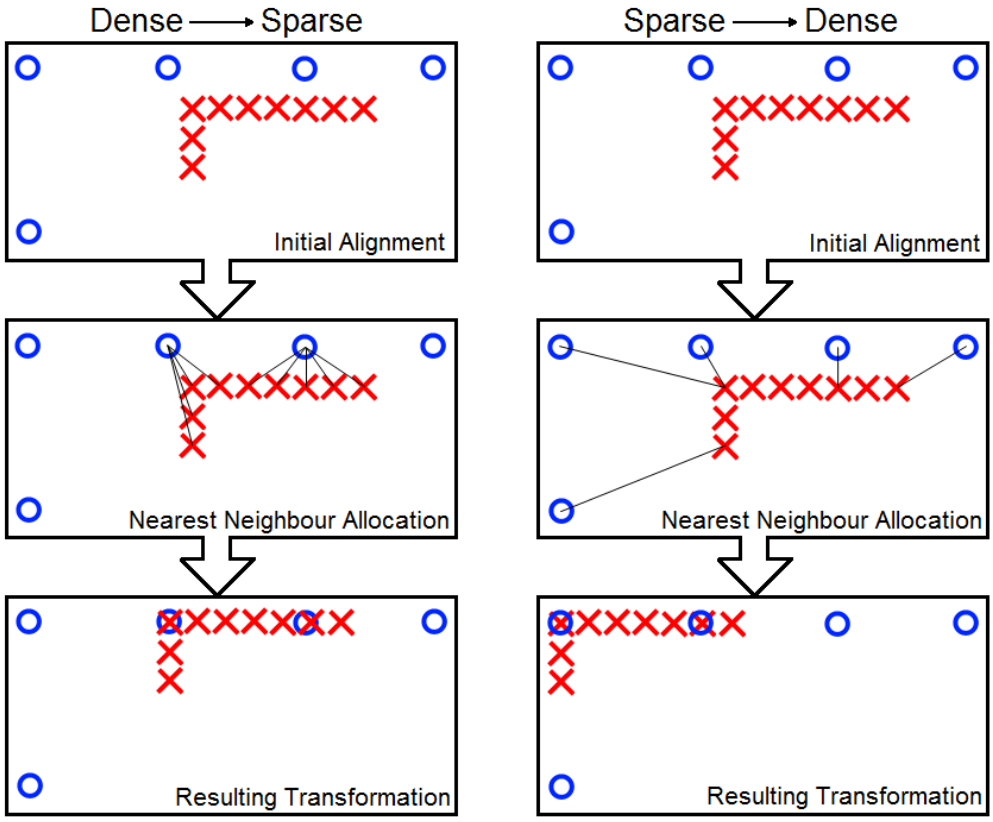


Figure 5.10 – Comparison of registering a dense point cloud onto a sparse point cloud (left) and a sparse point cloud onto a dense point cloud (right). The registration of dense point clouds onto sparse point clouds is prone to reaching local optima.

For scans recorded at a distance of up to 20 metres apart, a two stage registration process was sufficient to successfully reach the global minimum, as described in detail in Section 5.5.1. However, to correctly register point clouds with a large offset between scan origins, a revised technique was required. The newly acquired dataset is segmented into a dense section and a sparse section. The sparse section of the dataset is then used for registration onto the overlapping dense section of the existing model shape, significantly improving the result. The process is repeated to segment the existing model shape into a sparse section and a dense section so that the sparse section of the model shape can be registered onto the dense section of the newly acquired dataset. This process ensures that the registration process is always a sparse cloud onto a dense cloud, maximizing the opportunity to find the global optimum alignment. The processing time is also significantly reduced since the sparse segment is now used for N_p and the ICP registration step cost is $O(N_p \log N_x + 2N_p)$.

To successfully segregate the dense and sparse sections of a point cloud, a dividing plane called the density transition plane is used. This plane is located at the midpoint of the straight line distance between the points of origin of the most recent two scans (see Figure 5.11). The plane is perpendicular to the straight line path and roughly represents the point at which the newly acquired laser data becomes more dense than the existing model. The algorithm below details the segregation process.

```
INPUT:  Orig1 (Position of most recent model origin)
        Orig2 (Position estimate of data origin)
        P (Point cloud)
OUTPUT: Q (Sparse segment of point cloud)

norm = Orig2 – Orig1
m = Orig2 + (norm / 2)

For i=1 to size(P)
    If norm•(P(i)-m)<0 (Point in sparse segment)
        Count += 1
        Q(count) = P(i)
    End If
End For
```

Algorithm 5.4 – Segmentation of the sparse section of new point cloud data.

The sparse segment of the new data is then used for coarse registration onto the existing dense model. Upon coarse registration convergence, a fine ICP algorithm is run to provide the final alignment. The reverse process (sparse model onto dense data) is also implemented to confirm the results. The process can also be combined with random seeding to improve robustness to error in initial pose estimate. Superior results are obtained via this cropping technique and processing time is also significantly reduced. The improved registration robustness led to the application of this technique for all registration tasks, not just for datasets with a large offset.

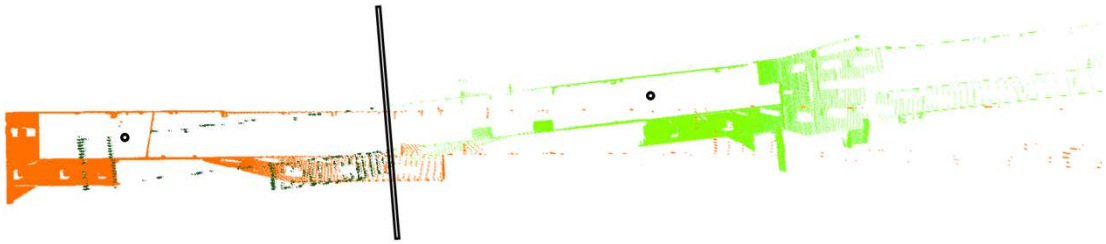


Figure 5.11 – The density transition plane, pictured here as a black rectangle, is located halfway between, and perpendicular to, the points of origin estimates of the two scans, pictured here as black circles. The plane segregates the newly acquired data into a sparse section (dark green) and a dense section (light green). The sparse section is then used for registration onto the existing map (orange).

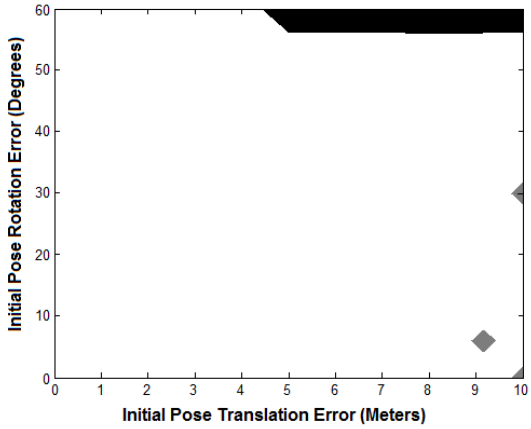
Occasionally a new dataset will be difficult to register due to the shape of the data itself. In these cases the registration algorithm will experience difficulties regardless of the quality of the initial pose estimate. A modified version of the segmentation technique can be applied in these cases to improve the likelihood of successful registration. The modified technique further segments the sparse sections of both the model and data, resulting in multiple segments and multiple registration attempts. Registration is then performed between each of the sparse segments and the entirety of the other scan. If multiple segments produce the same final transformations during registration, that transformation is considered the correct alignment and is applied to the entire dataset. If this does not occur, the segment with the lowest fitness score is considered the correct alignment and the resulting transformation is applied to the entire dataset. This technique was highly effective for the registration of the underground dataset discussed in Section 8.2. This modified technique can also be combined with random seeding to further improve robustness at the cost of increased processing times.

5.5.4 Bearing-only SLAM for Initial Pose Estimate

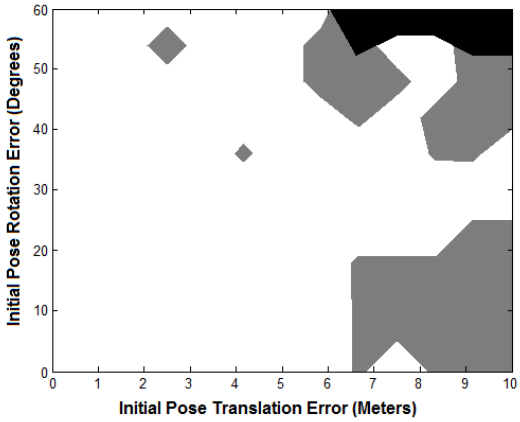
It has been demonstrated that an Iterative Closest Point (ICP) algorithm can successfully co-register two point clouds, even with a significant offset between their points of origin (see Section 5.5.3). However, successful registration is only possible if the algorithm is supplied with a sufficiently accurate initial pose estimate, otherwise finding a local minimum or complete failure of the registration is likely.

To investigate the effect that initial pose estimate quality has on the ability of ICP to find the global minimum solution, a series of tests was run to examine the limits of successful registration. The test started with two correctly aligned point clouds from the architecture building dataset, one representing the existing map and one representing newly acquired data. The new data point cloud was then transformed through a series of specified translation and rotation combinations in order to investigate whether a correct match could be found under the prescribed conditions. The range of translation tested was 0 to 10 meters in 1 meter increments and the range of rotation tested was 0 to 60 degrees in 5 degree increments. Each translation and rotation was positive and applied to all three axes equally in order to visualize the results in a two dimensional plot. The results were then categorized based on fitness score into one of three categories: global minimum (white), local minimum (grey) and algorithm failure (black). These results are shown in Figure 5.12.

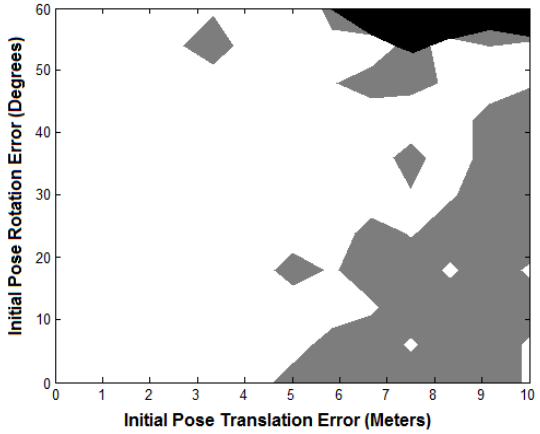
To produce a fully automated mapping system, the production of initial pose estimates must also be autonomous and of sufficient accuracy to ensure the correct optimization during co-registration. Figure 5.12(e) overlays the results from the four ICP tests (Figure 5.12(a)-(b)) and demonstrates the range of translation and rotation error allowable in the initial pose estimate for consistent successful registration.



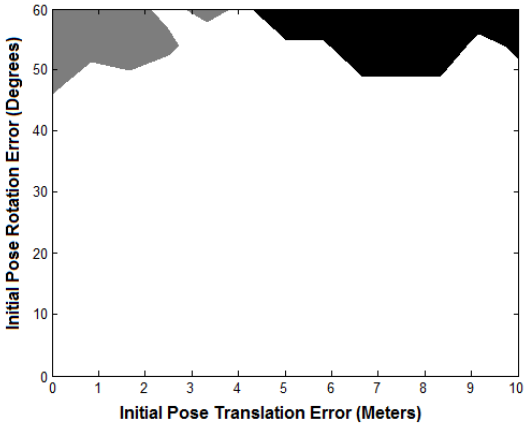
(a) Heat map of cloud 2 ICP Registration with cloud 1.



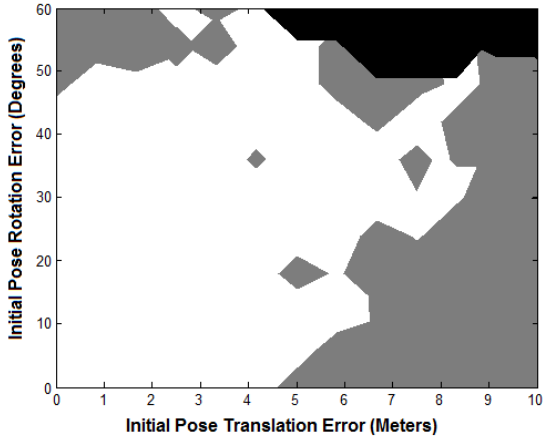
(b) Heat map of cloud 3 ICP Registration with cloud 2.



(c) Heat map of cloud 4 ICP Registration with cloud 3.



(d) Heat map of cloud 5 ICP Registration with cloud 4.



(e) Combination of architecture building dataset heat maps.

Figure 5.12 – Heat maps showing error in initial pose translation and rotation. White areas are correct registration (global minimum), grey areas are local minimum and black areas are failures. Individual registration results are shown for point clouds (a) 1 and 2, (b) 2 and 3, (c) 3 and 4, (d) 4 and 5. (e) A combination of all heat maps shows the required accuracy for initial pose estimates when registering an entire dataset.

Our approach to obtain the initial pose estimate is based on vision-based localization augmented by information from the discrete range-and-bearing laser scans. This technique is examined in detail in the following chapter. Bearing-only localization produces localization results in a dimensionless coordinate system, so associated ranges from the 3D laser scans are used to produce a correctly scaled localization result. The combination of two sensors and two approaches to map building and localization has resulted in a hybrid Simultaneous Localization and Mapping (SLAM) system. The initial pose estimates supplied by the hybrid SLAM system are prone to drift due to errors in bearing-only localization, yet they still provide the ICP algorithm with enough information to correctly perform co-registration.

Chapter 6

Hybrid Integration of Vision-Based SLAM and Point Clouds

6.1 Introduction

Producing high quality 3D maps of large scale environments is a task which cannot currently be completed autonomously through the use of a single sensor. Laser range finders can provide high accuracy, dense 3D information about large scale environments, yet the inability to robustly and autonomously combine multiple laser scan point clouds without additional information (e.g. pose estimation) has prevented the development of a 3D laser-only mapping system. Our solution to this problem is the combination of two sensors – a 3D laser range finder and an omnidirectional camera.

Simultaneous Localization and Mapping (SLAM) is a task performed by an autonomous agent to produce a map of the environment while simultaneously localizing its pose within that map. The mathematical background of SLAM was discussed in Chapter 2. Monocular SLAM is a mapping and localization problem where a single camera is the only source of information about the environment. The

task is also referred to as bearing-only localization due to the nature of a single camera only being able to supply bearing information about visual features and not range information. Monocular SLAM has the ability to perform six degree of freedom localization and mapping in real-time, yet is limited by the inability to recover real world scale and the production of sparse mapping results. There are examples of dense scene reconstruction in real-time for small scale scenes [79], although the scale still cannot be recovered and the results are prone to failure under dynamic lighting conditions (common in underground mining environments). Dense large scale scenes can also be reconstructed [80], but with excessive processing times and the same problems with illumination change and scale recovery.

The requirements for survey quality mapping results in underground environments can therefore only be achieved through the use of laser scanners as explained in Chapter 5. Our solution to the large scale 3D mapping problem is to combine the dense data of laser mapping with the speed and flexibility of vision-based bearing-only localization. The result is a hybrid SLAM system capable of producing survey quality maps in difficult environments such as underground mines. Although monocular SLAM can handle minor dynamic elements in an environment, dynamic elements will corrupt the discrete 3D laser scans rendering them inappropriate for survey use. Therefore, the effect of dynamic environments is considered outside of the scope of this work and only primarily static environments will be considered.

This chapter will investigate the implementation of the proposed multisensor mapping system. The interaction between the two sensors will be examined, including the use of the 3D laser for localization scale and the omnidirectional camera for initial pose estimate. The processing times will then be determined and compared to the requirements for a productive real world deployment. Finally, the suitability of the system for underground mining environments will be discussed.

6.2 Localization Scaling from Depth Information

Vision-based bearing-only localization and mapping algorithms have the ability to produce a three dimensional map of landmarks and provide pose estimates for the

current and previous camera positions. Although this technique is capable of producing reasonable mapping results, there is an inability to recover real world scale as discussed in Chapter 4. To meet the specifications for self-containment and portability listed in Section 1.1, a series of sensors capable of supplying range information were evaluated in Section 5.2. 3D laser scanners were determined to be the only sensor capable of producing the dense, large scale, survey quality mapping results required for our application. A 3D laser scanner can also provide the range information required to improve localization accuracy.

To produce a correctly scaled localization result, information from the 3D laser scanner has to be integrated into the vision-based localization algorithm. Since the 3D laser scanning process requires significant time (refer to Table 5.1), the sensor must be stationary during the scanning process to prevent scan distortion. Therefore, depth information is not available to the localization algorithm while the camera is moving. There will only be depth information at the localization path origin and destination.

There are two possible approaches to integrate the discrete depth information provided by the 3D laser into the localization algorithm. The first approach is to establish some visual features observed from the origin with known depth values. These features can then be adjusted in the state vector to have a certain 3D position provided by the laser. The covariance matrix can also be edited to reflect the high accuracy range data provided for these features. As the camera begins to move and new features are observed, their positional information will be encoded in the state vector and covariance matrix relative to the highly accurate original features and so overall accuracy will be improved. The depth information provided at the destination can also be incorporated and back propagated to improve the accuracy of the localization path leading up to the destination.

The problem with this approach is that features established at the origin, before camera movement begins, were not well maintained by the feature extraction and matching algorithms. Within an average of only 10 – 20 frames (of a 400 frame sequence) virtually all of the original features were dropped. Since the features were held for such a brief period of time, their high accuracy location information was not

effectively passed during the process of establishing new features. Without the direct influence of real world scale, the localization results quickly revert to an arbitrary scale based on the acceleration noise of the motion model [81]. The localization results were distorted accordingly. An example of this problem is shown in Figure 6.1. The left image shows the initial frame from the video sequence with the approximate outline of the corridor shown by the feature locations which were established using laser range information. The right image shows the video sequence at frame 400 (of 464). Note that the original features were not well established and so are no longer present on the map (established features are marked with magenta dots). Instead the algorithm has reverted to an arbitrary scale shown by the significantly reduced width of the corridor. An investigation into the possibility of scaling acceleration noise based on the laser data to reduce the impact of this effect is included in the future work discussion in Section 9.2.

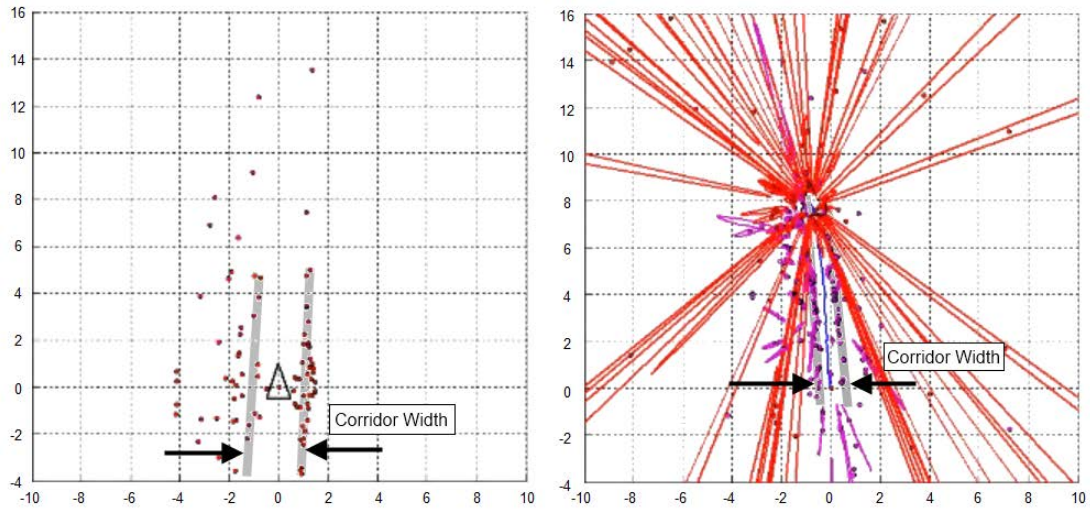


Figure 6.1 – Frames 1 (left) and 400 (right) from a 464 frame video sequence where features were initialised using laser based depth information. The depth information decays quickly, reverting to an arbitrary scale, resulting in the apparent reduction in the size of the main corridor being mapped. The units of all axes are meters.

The second approach is to allow the vision-based localization algorithm to run to completion without any integration of depth information from the 3D laser scanner. It has been shown that the inverse depth based monocular SLAM algorithm does not require any dimensioned information for successful execution [82]. The initial scale of the localization result is based on the acceleration noise in the motion model [81], so even when there is a low number of maintained features, the constant velocity

motion model maintains a reasonably consistent scale (although there will always be some level of scale drift [53]). When the camera has reached the destination, a combination of the scales from the laser scans at the origin and destination are used to scale the overall localization result. The two scales are combined based on the number of established features at the origin and destination using the following algorithm:

```
INPUT:  orig_feats (Established origin features)
        dest_feats (Established destination features)
        orig_depth (Depth image produced at origin)
        dest_depth (Depth image produced at destination)
OUTPUT: scale

For i=1 to size(orig_feats)
   meas_depth = get_depth(orig_feats(i).uv, orig_depth)
   scale_list_orig(i) = meas_depth/orig_feats.depth
End For

For j=1 to size(dest_feats)
   meas_depth = get_depth(dest_feats(i).uv, dest_depth)
   scale_list_dest(i) = meas_depth/dest_feats.depth
End For

scale_orig = mean(scale_list_orig)
scale_dest = mean(scale_list_dest)

combo_feat = size(orig_feats) * size(dest_feats)

scale = (scale_orig * size(orig_feats) / combo_feat)
        + (scale_dest * size(dest_feats) / combo_feat)
```

Algorithm 6.1 – Calculation of scale from laser scan data at origin and destination.

The algorithm calculates the scale by first determining the average scale at the origin and destination, based on the well-established features (observed in at least 10 frames) at each location. The arbitrary depth of each feature is compared to the depth value assigned to the feature from the laser data and the average is calculated. The average scales from the origin and destination are then combined through a weighting based on the number of well-established features at each location.

This approach works well and provides a consistently accurate localization result. The requirements for localization accuracy are examined in Section 6.3. This algorithm was implemented within our vision-based localization algorithm and was used during the production of all of the final results reported in Chapter 8.

To allow the vision-based localization algorithm to scale the results effectively, the depth information from the 3D laser scan must be supplied in a format which can be associated with an image from the omnidirectional camera. The solution to this compatibility problem is to produce an omnidirectional depth image from the laser data. A depth image is a matrix with the same dimensions as an image from the Ladybug camera; however, instead of containing 8 bit RGB values, it contains 16 bit scaled depth values. The density of the 3D laser scan allows every pixel in the Ladybug image to have an associated depth value. The depth image can therefore be used as a look-up table for determining the depth values of visual features. The image is produced using the following algorithm:

```
INPUT:  laser_data (Laser point cloud)
        image_dim  (Final image dimensions)
OUTPUT: depth_image

For i=1 to size(laser_data)
   theta = get_theta(laser_data(i))
   phi = get_phi(laser_data(i))

   [u, v] = convert_coords(theta, phi, image_dim)
   depth_image(u,v) = laser_data(i).depth/max_depth*2^16
End For
```

Algorithm 6.2 – Algorithm used to produce a depth image from a laser point cloud.

The algorithm converts a laser point cloud to a depth image by determining the bearings of each point. These bearings then provide the point with a (u, v) coordinate within the output image dimensions. The distance from the point to the origin (its depth) is then scaled to a 16 bit number based on the maximum depth found in the point cloud and is stored in the depth image at the calculated (u, v) coordinates. An

example depth image compared to the equivalent image from the Ladybug camera can be seen in Figure 6.2. This process can be performed during the laser scanning process since only one new laser point at a time is required. The 16 bit number can quickly be produced at the end of the scanning process by scaling all of the pixels in the depth image based on the maximum depth value found in the depth image. Calibration is currently handled manually due to variable alignment during data collection in the architecture building. However, once an appropriate mounting platform for the camera and laser is fabricated, a full spherical mapping will be undertaken to optimize alignment between the camera and laser data.

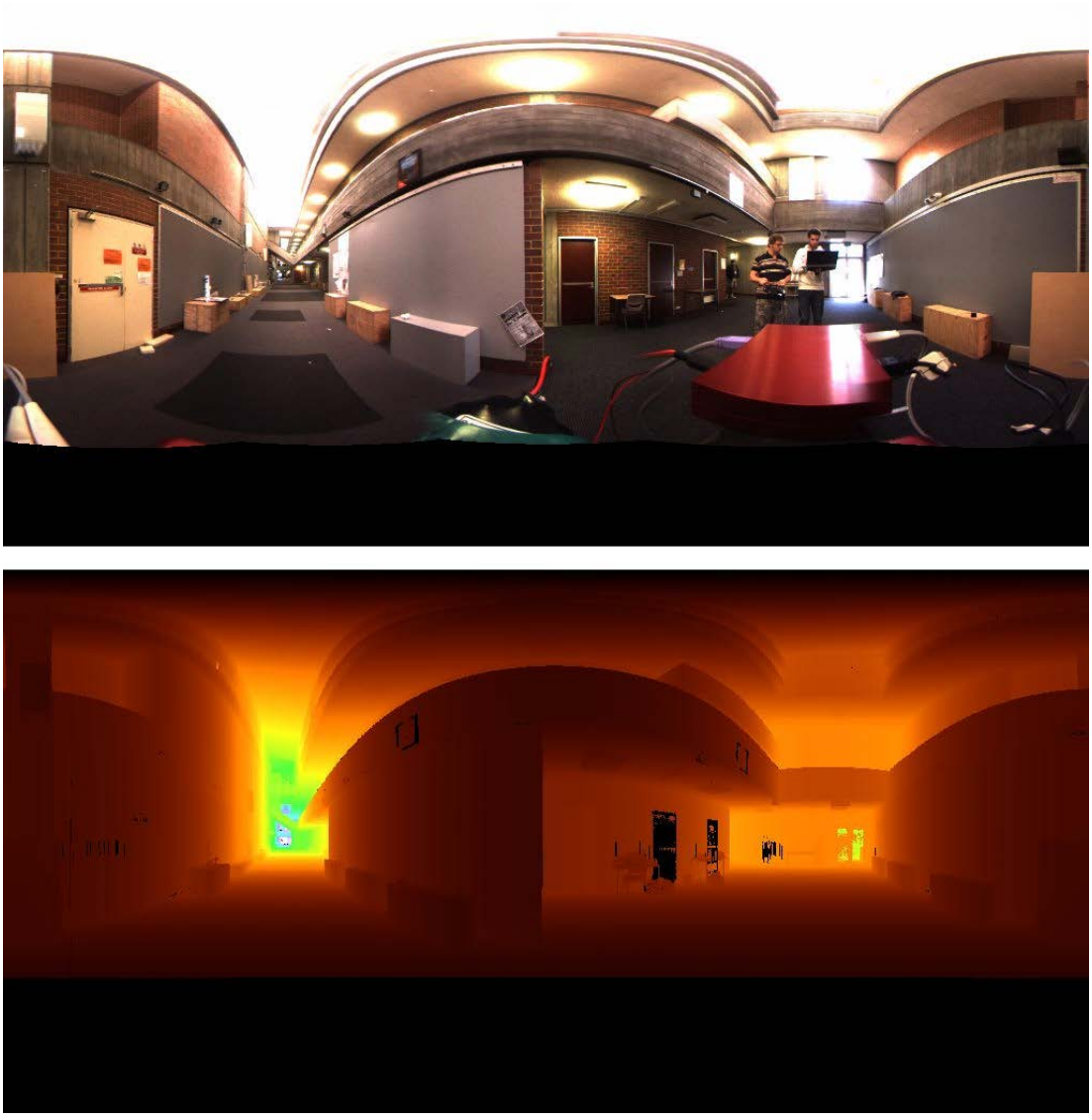


Figure 6.2 – Comparison of the image produced by the Ladybug omnidirectional camera (top) and the equivalent depth image produced from the associated laser point cloud.

6.3 ICP with Pose Estimate

For the successful registration of two point clouds, a sufficiently accurate initial pose estimate is required. The need for this pose estimate is discussed in detail in Section 5.5.4. Without a high quality initial pose estimate, the optimization step of registration algorithms such as Iterative Closest Point (ICP) will likely find a local minimum or fail completely rather than finding the desired global minimum.

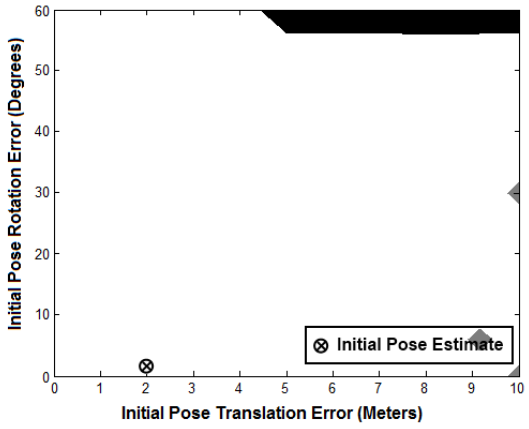
The initial pose estimates supplied by the scaled localization results from vision-based localization in the Curtin University architecture building were tested to determine if they were of sufficient accuracy to allow successful localization. The four localization results from the architecture building provided four initial pose estimates for the registration of four laser scans (scans 2 – 5) with the existing map (scan 1). Using the ICP technique described in Section 5.5.1, the initial pose estimates were of sufficient accuracy to produce successful ICP registration. The detailed results from the architecture building can be found in Section 8.1.

		Localization Result		**ICP Correction**		**Final Position**	
Dataset		**Trans.**	**Rotation**	**Trans.**	**Rotation**	**Trans.**	**Rotation**
	x,ϕ	-2.193m	2.704°	1.512m	-2.693°	-0.752m	0.009°
1	y,θ	16.88m	0.362°	1.954m	-0.327°	18.80m	0.047°
	z,ψ	-0.160m	0.278°	-0.549m	0.246°	-1.514m	0.539°
	x,ϕ	-2.989m	3.312°	0.497m	-3.294°	-4.603m	0.111°
2	y,θ	19.34m	-1.031°	0.976m	1.071°	19.83m	0.408°
	z,ψ	-0.182m	-5.487°	-1.009m	6.314°	-2.181m	0.753°
	x,ϕ	-4.453m	2.777°	-1.144m	-2.859°	-10.20m	0.014°
3	y,θ	17.46m	-0.740°	0.452m	0.516°	16.01m	0.056°
	z,ψ	-0.107m	-16.09°	-0.880m	15.88°	-1.682m	-0.255°
	x,ϕ	-1.681m	3.534°	-0.047m	-3.512°	-2.832m	0.054°
4	y,θ	17.96m	-0.634°	1.301m	0.596°	19.08m	0.179°
	z,ψ	-0.181m	-3.811°	-1.159m	3.529°	-2.402m	-0.326°

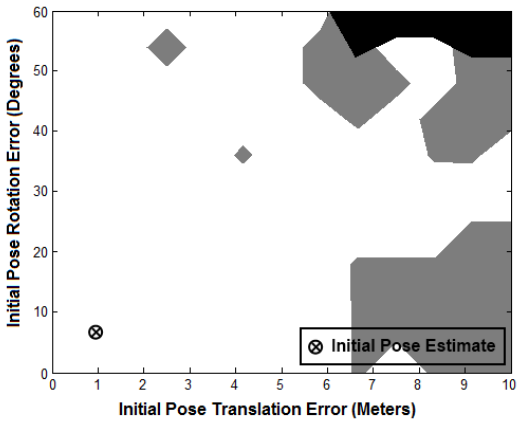
Table 6.1 – The localization result, ICP correction and adjusted scan location for each of the four datasets from the Curtin University architecture building.

Table 6.1 shows each localization result, the ICP correction and the final scan location for each of the four localization datasets. The unusual z-axis displacement is caused by a small uncompensated angular offset (~5°) between the camera and laser. This will be resolved in future deployments when a proper mounting platform will be fabricated for the camera and laser, and a high accuracy calibration procedure is performed.

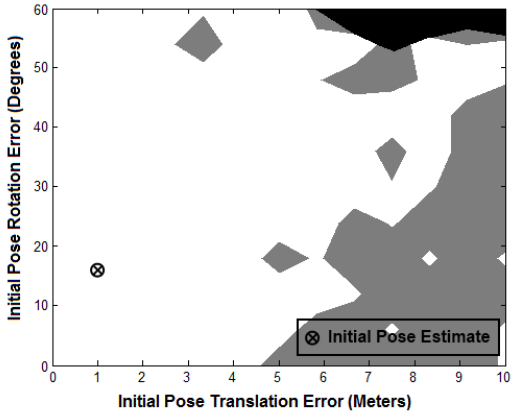
To determine the robustness of the multisensor SLAM system, a test needed to be formulated which would provide some form of measurement for the 'quality' of the initial pose estimate. A measurement for quality can be determined if the initial pose estimate supplied by vision-based localization is compared to the failure conditions for ICP (i.e. where the optimization step finds a local minimum or fails completely). The failure conditions for ICP had been roughly determined in the testing performed in Section 5.5.4. 'Heat maps' were produced, showing the offset values for translation and rotation that resulted in ICP failure (results in Figure 5.12). The initial pose estimates supplied by vision-based localization can be plotted on these heat maps by using the largest translation and rotation offset corrected for by ICP. The offset errors are used as the coordinates for the initial pose estimate on the heat map. The plots for each of the datasets from the architecture building with the initial pose estimate overlayed can be seen in Figure 6.3.



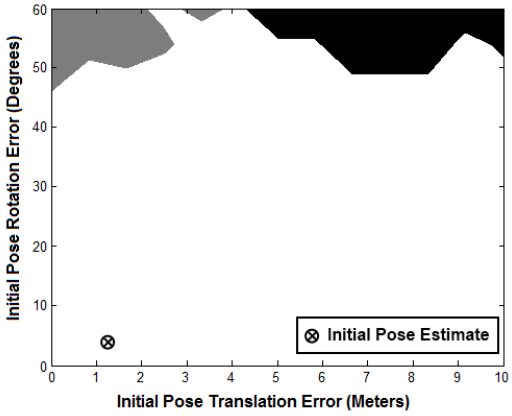
(a) Heat map of cloud 2 ICP registration with cloud 1.



(b) Heat map of cloud 3 ICP registration with cloud 2.



(c) Heat map of cloud 4 ICP registration with cloud 3.



(d) Heat map of cloud 5 ICP registration with cloud 4.

Figure 6.3 – Heat maps showing error in initial pose translation and rotation. White areas are correct registration (global minimum), grey areas are local minimum and black areas are failures. Individual registration results are shown for point clouds (a) 2, (b) 3, (c) 4, and (d) 5. Each heat map contains an indicator representing the maximum offset error in the initial pose estimate provided by vision-based localization.

All of the heat maps demonstrate that the initial pose estimate supplied by the scaled vision-based bearing-only SLAM algorithm falls easily within the region of successful global optimization. However, the heat maps also demonstrate that there are limitations to the robustness of the ICP algorithm to pose offset error. The additions to the ICP registration process discussed in Section 5.5 significantly improve the robustness to offset error. Random seeding has the effect of moving the pose estimate within the heat map, increasing the chance of it falling within the successful global optimization region. The segmentation technique also improves robustness by producing multiple registration scenarios, again improving the chance of the pose estimate falling within the successful global optimization region.

To further improve the long term localization results produced by the scaled vision-based bearing-only SLAM algorithm, a technique referred to as ‘localization path refinement’ is implemented. The technique is possible due to the long distance measurements provided by the 3D laser scanner. The system begins by taking an initial stationary 3D scan. The vehicle then moves to the subsequent scan location while tracking its motion using vision-based localization. Vision-based localization, like all forms of SLAM, is prone to drift and so features further from the origin increase in locational uncertainty. Upon reaching the destination, a second 3D laser scan is performed and the rough position of this scan is supplied by the localization

result. The rough initial pose estimate is used as the starting point for ICP, resulting in convergence and producing a highly correlated point cloud registration. After registration, the position of the second scan relative to the first scan is known to be accurate to a matter of millimetres. This accuracy is a vast improvement over the initial pose estimate which can vary by as much as 4.5 meters and 16° over a 20 meter traverse. The ICP registration accuracy can therefore be passed to the vision-based localization to improve the current position estimate significantly, resulting in the effect we refer to as 'localization path refinement'.

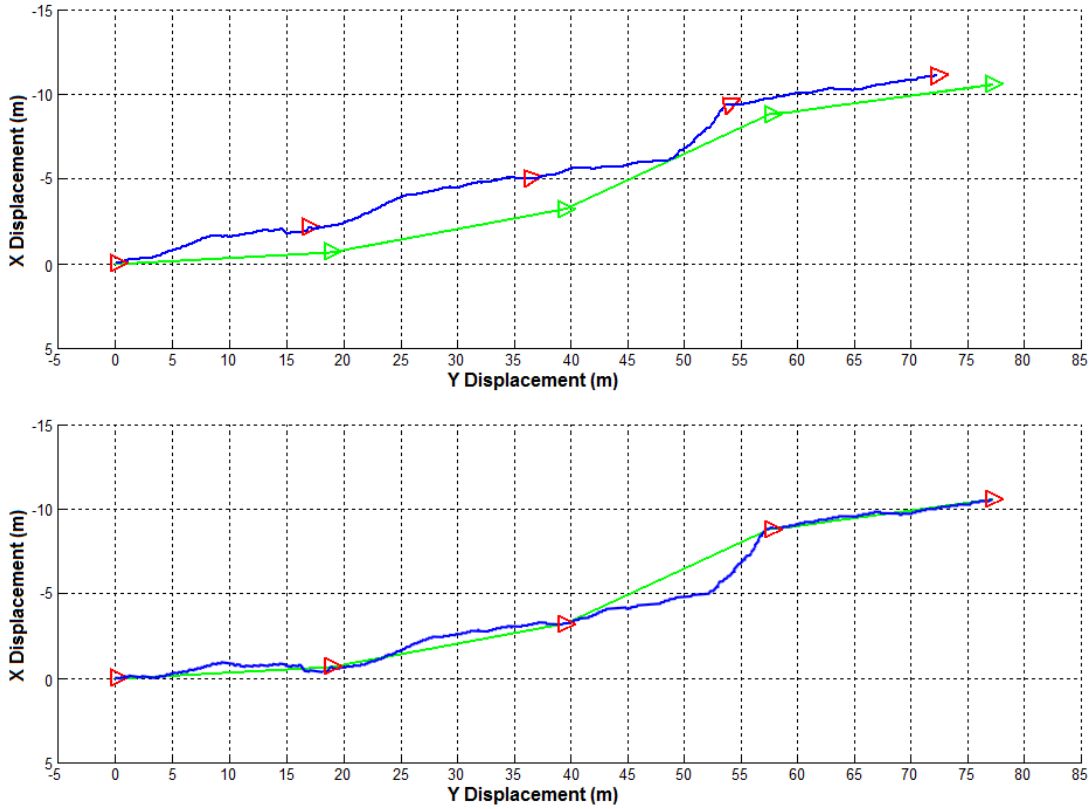


Figure 6.4 – The localization result (blue) from vision-based localization without ICP path refinement (top) and with localization path refinement (bottom). The approximated ground truth is also shown (green) and triangles represent the expected and actual scan locations.

The effects of localization path refinement in the multisensor SLAM system can be seen in Figure 6.4. The top image shows the measured localization path relative to the ground truth when no refinement is performed. The ground truth was determined from survey results of the scan locations and the approximately straight line trajectory of the vehicle between scans. The bottom image shows the effect that passing the positional information produced by registration to the localization

algorithm has on the localization result. There is a significant improvement when compared to the ground truth. The most significant deviation occurs between scans 3 and 4. This area of ground was paved in exposed brickwork and the vehicle used to transport the sensors had no suspension, resulting in image blur due to the vibration of the vehicle as it traversed this area. The image blur degraded the quality of the localization result, yet the quality of the pose estimate was still sufficient to perform successful registration.

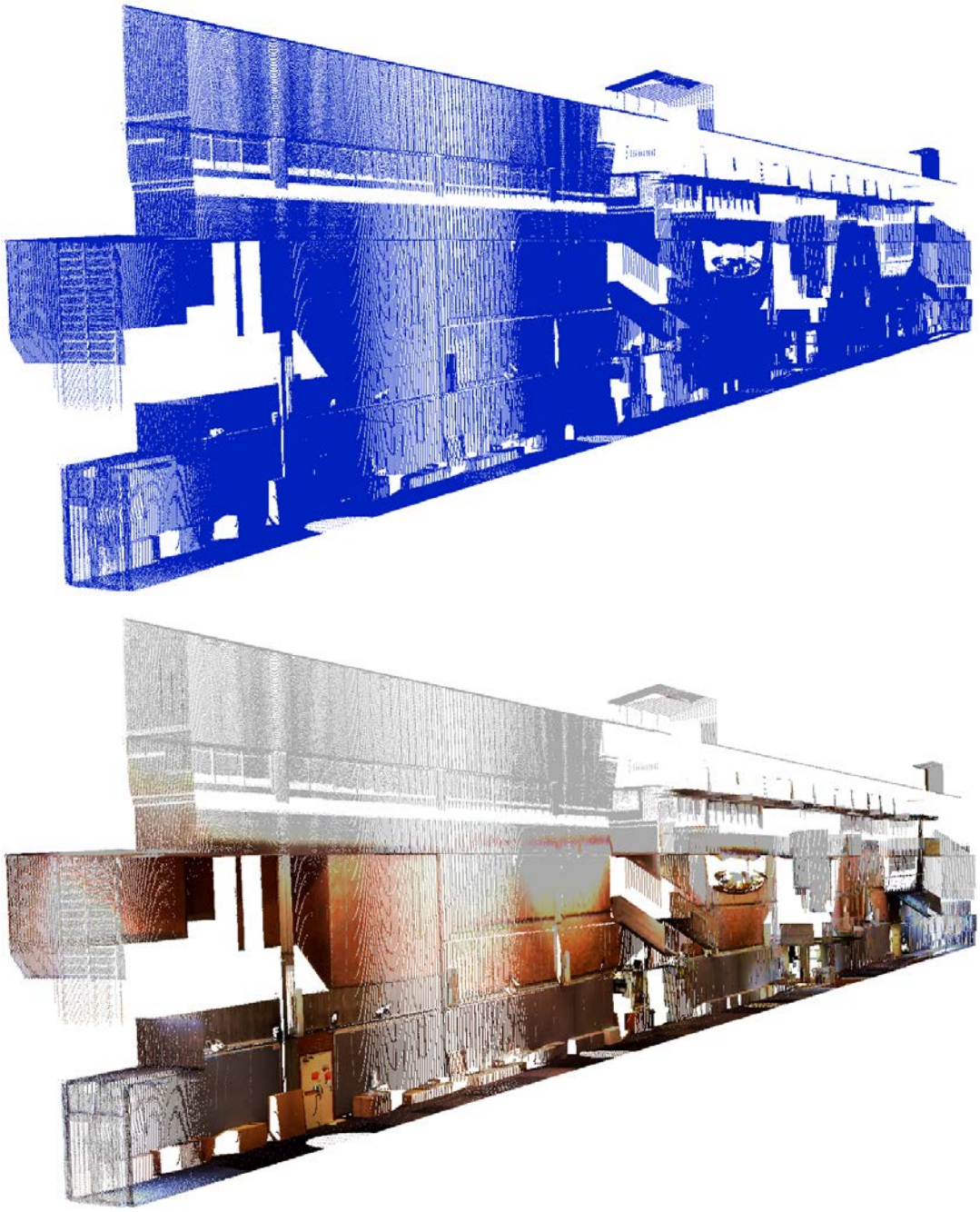


Figure 6.5 – A cross section of the registered architecture building dataset without any rendering (top) and with rendering from the omnidirectional camera (bottom).

Another advantage of the fusion of 3D laser scanner and omnidirectional camera for map building is the ability to render the point cloud using information from the omnidirectional image. Since every scan location has an omnidirectional image recorded at the same point, the bearing information of each point in the point cloud can be used to find the associated RGB pixel colour from the image. Rendering a point cloud can provide a host of additional information during the map building process and makes the resulting large scale map look familiar to an observer. An example of a point cloud with and without rendering can be seen in Figure 6.5.

6.4 Processing Time

For the proposed multisensor SLAM system to be a genuine option for the large scale mapping of underground mining systems, there are constraints on the execution times which must be adhered to. If the system is to be competitive with the currently available semi-autonomous techniques for laser based 3D mapping, the processing time for data collection and map building must be similar or even faster. This section will therefore examine the amount of time required for the full mapping process to take place, from initial environment preparation, to final scan alignment. An approximate value will be produced for the amount of time per scan; this will allow time calculations for projects of any size.

The conventional map registration technique will be examined first and used as a baseline comparison for the results obtained later in the section for the fully autonomous multisensor SLAM system. The semi-autonomous conventional process begins with the initial preparation of the environment to be scanned, this includes setting up the targets used during scan alignment and mounting and levelling the laser scanner. Targets may be reused in multiple scans; however, the laser levelling process must occur for every individual scan. An estimate of four minutes per scan will be used to cover time needed for laser levelling, transport to the next scan location and part of the overall target setup time.

The scan times for the 3D laser itself will be based upon the times required for a Leica C10 to perform a full 3D scan. Table 6.2 shows the scan times based on the

density of the resulting point cloud. For the following calculations it will be assumed that a medium density scan is performed, requiring a scanning time of 9.8 minutes. Finally, the scans must be registered. The registration process begins with the manual identification and scanning of the targets within the environment, this process requires approximately two minutes per target. On average, three targets are required to perform alignment, so the target identification and scanning time totals about six minutes per scan. The time taken for the actual registration step in the accompanying Leica Cyclone software is negligible due to the extremely accurate initial pose supplied by the targets. Therefore, the total time required to semi-autonomously acquire and register a scan is approximately 20 minutes.

Density	Horizontal Angular Res.	Vertical Angular Res.	Maximum Scan Rate	Time
Low	0.115°	0.115°	50,000 points/s	2.4 min
Medium	0.057°	0.057°	50,000 points/s	9.8 min
High	0.029°	0.029°	50,000 points/s	39.2 min
Highest	0.012°	0.012°	50,000 points/s	245.0 min

Table 6.2 – Scan times for the Leica C10 3D laser scanner.

The generation of a large scale 3D map of an underground mining environment using our multisensor mapping system is a different process to that required for the conventional mapping technique. There is no environment preparation required and so no time is lost prior to laser scanning commencement. The time required for a single full 3D scan will again be based on the Leica C10 medium density scan processing time, i.e. 9.8 minutes. The time required to move between scan locations is based on the speed of the vehicle carrying the equipment. Assuming a vehicle speed of approximately 5km/h (around walking speed) the time taken to traverse a scan offset distance of 20 meters would be 14.4 seconds.

The monocular SLAM algorithm by Civera *et al*. is designed to work in real-time and has a reported processing time of 21ms per frame when tracking 12 features at 30 frames per second (fps) and with a state vector containing 300 elements on a 1.8GHz Pentium laptop [83]. Our own implementation of Civera's algorithm has not

yet been optimized for real world deployment. However, based on Civera's processing time calculations, a predicted processing time for our own algorithm may be approximated. Our vision-based localization algorithm tracks 50 features on average, at 15 fps and has a state vector containing 550 elements. Using Civera's calculations, the approximate expected times would be: image acquisition and feature extraction: 7ms, EKF prediction: 4ms, feature matching: 17ms, EKF update: 31ms. The total processing time per frame would therefore be 59ms which is the equivalent of about 17 fps. The use of SIFT for feature extraction and the spherical camera model conversion by Rituerto *et al.* will slow processing times, but on a more powerful modern PC running at 2.8GHz, an optimized version of our algorithm should run in real-time using a camera recording at 15 fps. This means that no additional time would be required to perform localization apart from the time already required to traverse the distance between scan locations.

To correctly scale the localization result, a depth image must be produced, as described in Section 6.2. The current Matlab implementation of the depth image production algorithm can compute and store cloud data at up to 90,000 points per second. This means that if the algorithm could have access to the laser points as the scan is occurring, the depth image could be produced during the acquisition of the laser data. Since the laser scan rate is only 50,000 points per second, no additional time would be required for the production of the depth image.

The final stage of the multisensor SLAM system process is the registration of the newly acquired data to the existing map. Since path planning is not an aspect of this SLAM system, registration does not need to be performed as soon as the new laser data is collected. The registration step can therefore be performed with a one scan lag: the registration of a new scan will occur while the next scan is being recorded by the 3D laser. The average time required to register a new point cloud from the architecture dataset, based on the ICP implementation mentioned in Section 5.5.1, is 5.5 minutes. Since the registration time is less than the laser scanning time of 9.8 minutes, no additional processing time is required. The total time required to autonomously acquire and register a point cloud is equal to the scan time plus the travel time to the next scan location. Since the scan time and travel time are fixed, the SLAM system can operate as fast as possible with current sensor limitations. This

is a notable improvement over the conventional registration technique which requires the same scan and travel times plus an additional 10 minutes to register each new scan.

6.5 The Mining Environment and Dynamic Shadows

The target application for our multisensor SLAM system is the autonomous mapping of underground environments. There are unique characteristics of underground environments that greatly increase the difficulty of performing robust vision-based localization. The detrimental effects experienced when collecting an underground dataset for the evaluation of the multisensor SLAM system included large amounts of visual occlusion, large regions of complete darkness and oversaturated brightness, lens flares from vehicle lighting and many dynamic shadows. These issues can be traced back to two main problems: visual occlusion or dynamic lighting.

Visual occlusion can be addressed simply by an alteration in hardware as discussed in Section 8.2. However, poor lighting is a serious and common problem in underground environments. Regions of complete darkness and oversaturation reduce the useful portion of the visual sphere, yet have no other detrimental effect on the vision-based localization process. Conversely, lens flares and dynamic shadows can cause major problems due to the feature detection algorithms identifying features on dynamic illumination artefacts, including shadows and lens flares. The motion of these features can contradict the tracking of the scene, resulting in the corruption of the localization path. In order to perform robust vision-based localization in these situations, a technique is required to remove the negative influence of these lighting conditions.

Colour information is available in most vision-based localization tasks, yet is rarely utilized due to the use of monochrome images by popular feature extraction algorithms. Chapter 7 investigates the incorporation of colour information for the detection and removal of harmful dynamic lighting effects and then applies the derived techniques in vision-based localization scenarios.

Chapter 7

Vision-Based SLAM under Dynamic Illumination

7.1 Introduction

The use of supplementary information in feature extraction and matching is an important addition to vision-based SLAM in environments with poor lighting conditions. Underground mining tunnels are prone to illumination artefacts such as lens flares and dynamic shadows caused by the regular use of dynamic light sources (see Figure 7.1). These lighting characteristics are vastly different to the conditions used for the evaluation of most feature extraction algorithms [32], [28], [30], [29]. To overcome the detrimental effects of these lighting conditions, additional information is required to distinguish reliable visual features from illumination artefacts.

There is currently no complete solution to the problem of vision-based localization under dynamic illumination; however, many research groups have attempted to improve robustness. To improve illumination robustness in a vision-based object recognition task, Burghouts and Geusebroek integrate the Gaussian opponent colour model into SIFT [84]. The model consists of intensity, red-green and yellow-blue

channels. The separation of intensity enhances illumination invariance and results in improved matching when compared to standard SIFT over a range of lighting conditions. This work was aimed at a general object recognition problem where the constraints on computation time are less strict than those involved in real-time applications such as the real-time localization in our multisensor SLAM system.

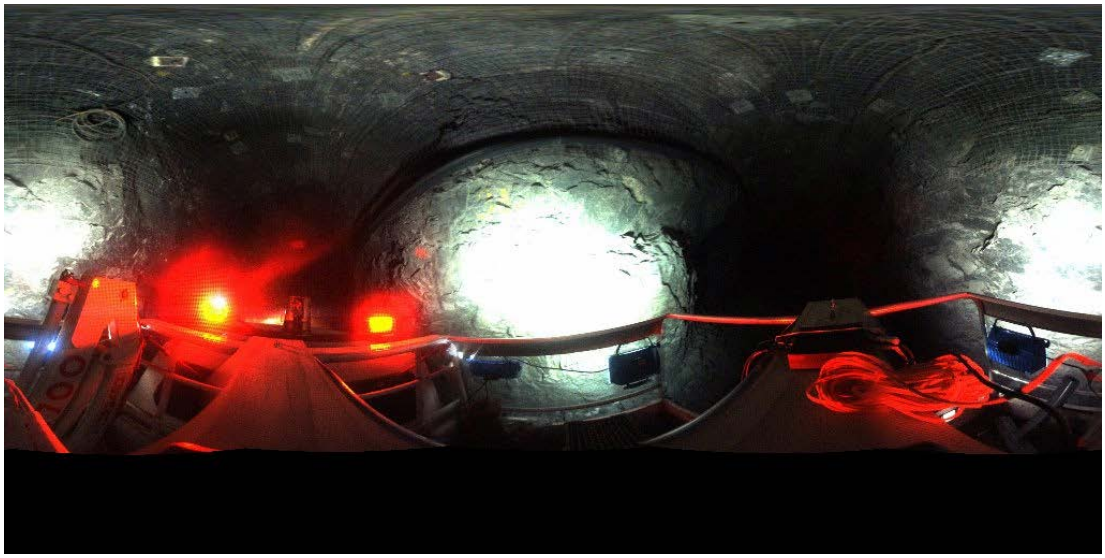


Figure 7.1 – An example of the lighting conditions in an underground mining environment. This image is from the sequence used for evaluation in Chapter 8.

Another approach to improving illumination robustness in localization is demonstrated by the promising results achieved by Silveira and Malis by modelling illumination change as a surface that can evolve over time [85] [86]. The model is combined with projective geometry to produce basic planar visual tracking. This approach requires no *a priori* knowledge about the light source(s) or the subject surface materials. A planar feature in an image is 'warped' to match the same feature seen from a different viewpoint in a subsequent image. The 'warp' required to match the features is then used to perform basic localization. The need for high quality planar features is the limiting factor of this technique, preventing application in underground mining environments despite its illumination robustness.

Visual sensors can also be supplemented with information from other external sensors to reduce the impact of dynamic illumination. Sunghwan *et al.* fuse stereo vision with sonar to produce a hybrid indoor SLAM system that is partially invariant to lighting conditions [87] [88]. Errors in vision-only localization occur when

stationary features appear to ‘move’ due to a change in lighting conditions. EKF based slam compensates for this movement by unnecessarily adjusting the location estimation of the feature. The fusion with depth data negates this compensation by keeping an accurate depth estimate of the feature, resulting in the apparent feature movement being ignored. This is an effective approach to illumination invariance, but is limited by the poor range and accuracy of sonar, as well as the need for a secondary sensor and sensor fusion algorithms.

Finally, *a priori* information can be used to improve localization robustness to changing lighting conditions. Bischof *et al.* [89] use an illumination insensitive eigenspace approach to robustly recognize objects under varying lighting conditions. The recognition algorithm is trained using multiple images of an object under various lighting conditions, resulting in an eigenspace representation that is combined with a randomized voting algorithm. Steinbauer and Bischof used this approach on images from an omnidirectional camera to improve localization [90]. A set of training images with known locations were combined with robot orientation odometry data to aid matching and localization. Since our applications involve unstructured and unknown environments, it is not appropriate to depend on *a priori* information.

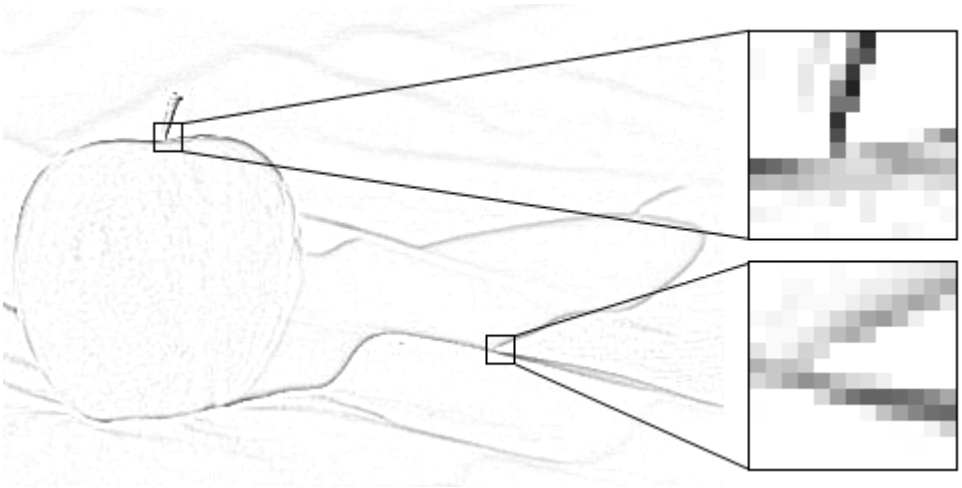
The lack of a robust vision-based localization technique for dynamically illuminated underground environments led to the development of our own technique. This chapter will begin with a report on the difficulties of using monochrome images for vision-based localization in environments with dynamic illumination. This is followed by an investigation into the use of colour information to improve robustness to dynamic illumination in localization tasks. The development of two novel techniques to improve illumination robustness based on Horprasert’s chromaticity colour model [91] is then discussed. Finally, the performance of the two techniques is examined through a series of computer generated and real world evaluations.

7.2 Monochrome Localization

The feature detection approaches used by both SIFT and FAST corner detection (as discussed in Section 2.3.3) are highly unreliable when implemented on datasets with

significant illumination variation. The difference of Gaussian approach used by SIFT to extract scale invariant features cannot differentiate between object based features and illumination based artefacts. A shadow boundary will produce a difference of Gaussian feature similar to that produced by the edge of an object. This is highlighted in Figure 7.2(b) where the stalk of the apple produces a response similar to that produced by an intersection of shadows projected on the background (see Figure 7.2(a) for original reference image). FAST corner detection uses the monochrome version of the image and encounters the same difficulties as SIFT when differentiating shadow based artefacts from object features. The features shown in Figure 7.2(c) both contain strong instances of FAST corners yet are the same example features noted in the SIFT image, one feature represents an object corner and the other an intersection of shadows.

(a) The unaltered reference image.



(b) Features on the Difference of Gaussians image used by SIFT.

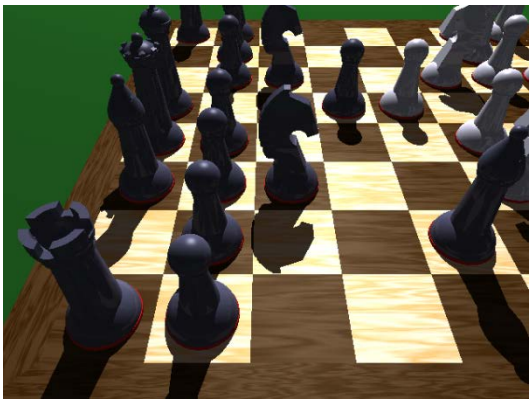
(c) Features on the monochrome image used by FAST corner detection.

Figure 7.2 (a) The original unaltered image containing a dominant object and shadow used for feature extraction by SIFT and FAST corner detection. (b) Two potential features identified from the Difference of Gaussians image, both contain strong criteria for feature detection; however, one is object based and one is shadow based. (c) The same two features identified in the Difference of Gaussians image are shown here to also produce strong selection criteria during FAST corner detection.

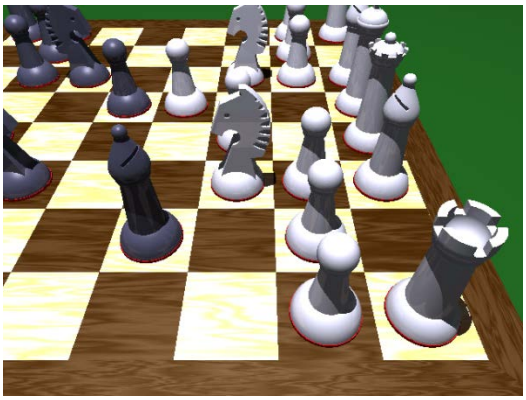
The inability to differentiate object features from shadow features results in erroneous localization in environments with dynamic illumination. In monocular vision-based localization, a stream of images from a single camera is the only information used for localization. Features are extracted from each image, matched between frames and tracked while within the camera's field of view. Feature movement that is contradictory to the actual movement of the camera will impair the resulting localization. Inconsistent feature movement occurs when shadow features move independently to object features due to a dynamic source of light.

The effect that dynamic illumination has on localization can be seen in Figure 7.3. A camera moving in a straight line, from left to right, observes a computer generated scene containing a chess set that is lit via a dynamic light source. The use of computer generated scenes for evaluation is discussed in Section 7.4.2. Two images from the computer generated sequence can be seen in Figure 7.3(a) and (b). The light source moves from behind the scene to directly overhead while casting dynamic shadows as illustrated in Figure 7.3(c). The ground truth for the camera movement can be seen in Figure 7.3(d). The resulting localization path contains significant distortion due to the movement of the primary light source (Figure 7.3(e)). The result is significantly different to the localization path produced under static lighting

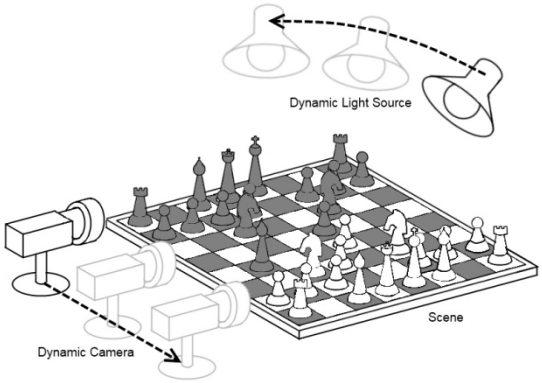
conditions which can be seen in Figure 7.3(f). The scale in Figure 7.3 (d) – (f) is dimensionless as the scene is computer generated. Consistent scaling was achieved by identifying common features (represented as ellipsoids) within each localization result.



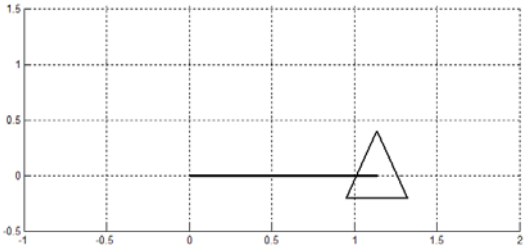
(a) First image from the computer generated sequence.



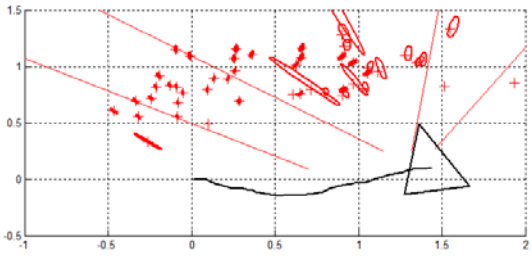
(b) Last image from the computer generated sequence.



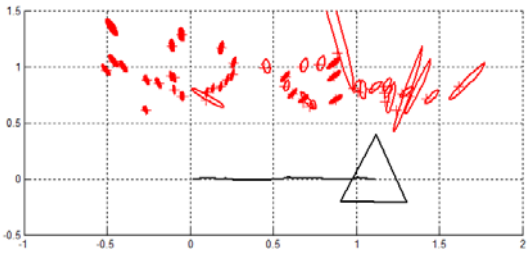
(c) Depiction of scene.



(d) Camera movement ground truth.



(e) Dynamic illumination localization path.



(f) Static illumination localization path.

Figure 7.3 (a) The first image from the computer generated sequence. (b) Last image from the sequence. (c) A depiction of the scene, including the camera motion path and the path of the moving primary light source. (d) The ground truth of the camera motion. (e) The localization path produced when the scene has dynamic illumination. (f) The localization path produced when the scene has static illumination.

The erroneous localization seen in Figure 7.3(e) is a result of the monocular SLAM algorithm assuming that it is operating within a primarily static environment. The algorithm uses a RANSAC based filter to remove features that it considers to be based on a dynamic source. However, dynamic features created by a moving light source are uniformly distributed across the scene and move in a consistent manner, making them impossible to isolate using traditional motion model based filtering. The algorithm resolves the two types of motion by combining them into the curved path seen in Figure 7.3(e). To correct the localization, an alternative approach that differentiates object based features from illumination artefacts is required.

7.3 Colour Information for Shadow Detection

Colour information is readily available and can be easily integrated into feature descriptors to increase the constraints on feature extraction and matching to improve robustness to dynamic illumination. Colour information can be utilized to determine if a feature is based on a variation in illumination or the edge of an object. A feature that is extracted from the edge of a shadow is likely to have a variation in perceived intensity only, that is, the colour difference between pixels on either side of the feature 'edge' are separated by a shift in intensity only. Alternatively, features extracted from the edges of objects will contain a variation in true colour – a colour change that is independent of intensity levels. Therefore, in order to correctly distinguish between physical features and illumination artifacts, it is desirable to measure the true colour of the features.

Swain and Ballard were the first to use a three dimensional RGB histogram to describe pixel colour values for recognition, known as 'colour indexing' [92]. Finlayson, Chatterjee and Funt expanded upon this approach to produce basic, lighting invariant, object recognition using 'colour angles' [93]. They demonstrated improved efficiency over colour indexing, at the cost of number of correct recognitions. An alternative approach trialled by Geusebroek *et al.* estimated the original colour of an object mathematically given known lighting conditions [94]. This technique was robust to viewing direction, surface orientation, highlights, illumination direction, illumination intensity, illumination colour and inter-reflection,

but was limited to a range of known materials and strictly controlled lighting conditions.

Successful shadow detection in a dynamic environment was achieved by Horprasert, Hardwood and Davis to detect moving objects against a static background [91]. An RGB histogram compared successive pixel colours between frames to produce a colour model based on two components: brightness and chromaticity. These two components were used to examine a change in pixel colour over time such that a differentiation could be made between static background, shadow, and dynamic foreground. The concept behind the technique is that a shadow cast on an object will result in a large change in brightness, whereas the chromaticity difference between pixels will remain small. Figure 7.4 demonstrates that shadow and object features extracted from an image can have similar monochrome responses, yet the shadow feature can be easily distinguished as being a variation in colour intensity alone.

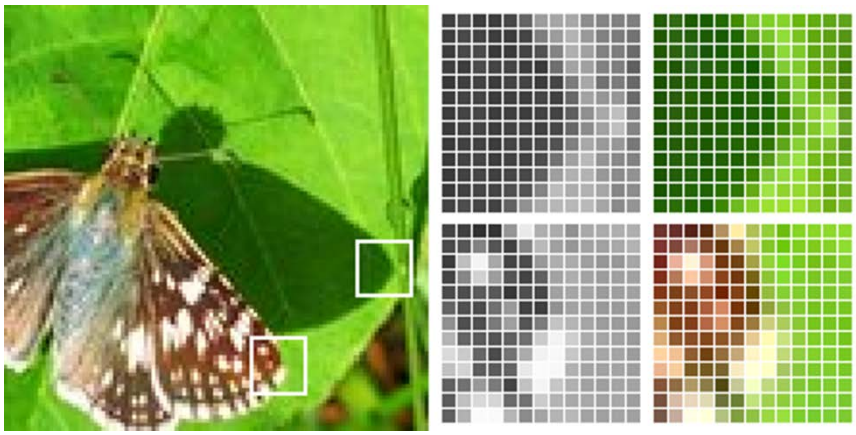


Figure 7.4 – Two features are extracted from the image, one based on the edge of a shadow and the other on the edge of an object. The response in monochrome looks similar, however the colour image clearly shows that the shadow based feature has a variation in colour intensity only. The object based feature contains a variety of different colours.

The chromaticity model proposed by Horprasert *et al.* [91] compares the distortion between two pixel colours in RGB space to determine the difference in 'true' colour. A line OE passing through the origin and the first pixel colour ($E = [E_R, E_G, E_B]$) is called the expected chromaticity line and is used to determine the distortion in chromaticity and brightness. If the second pixel colour ($I = [I_R, I_G, I_B]$) is on this line, then there is a distortion in brightness only (α), otherwise there is also a distortion in chromaticity (CD), see Figure 7.5.

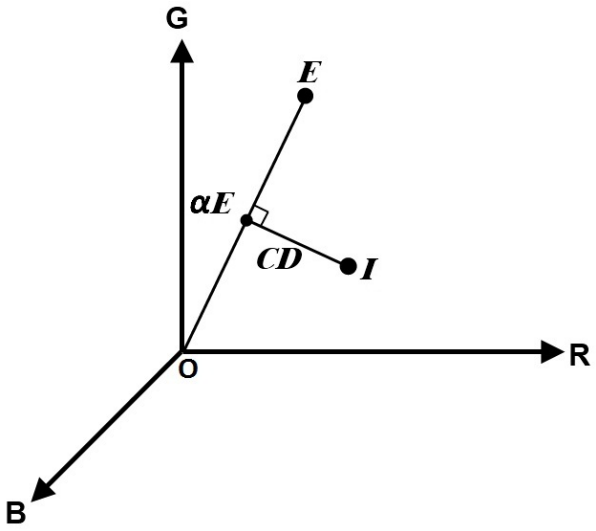


Figure 7.5 – Determining the brightness and chromaticity distortion between two colours (E, I) [91].

The brightness distortion (α) is a scalar value which represents the point on the expected chromaticity line that is closest to the comparative colour (I). It is defined as:

$$\alpha(E,I) := \arg\min_{\alpha} \|I - \alpha E\| . \tag{61}$$

Brightness distortion will be equal to 1 if the comparative colour has the same brightness, less than 1 if it is darker and greater than 1 if it is brighter. Chromaticity distortion (CD) is the orthogonal distance between the expected chromaticity line and the comparative colour (I). The chromaticity distortion of the second pixel is given by:

$$CD := \min_{\alpha} \|I - \alpha E\| = \|I - \alpha E\| . \tag{62}$$

Our approach applies this colour model to features extracted from images rather than individual pixels. To make the technique compatible with multiple feature extraction algorithms such as SIFT and FAST corner detection, a 3x3 grid is used to represent the colour information of extracted features. The 3x3 grid is scaled to the magnitude of the extracted feature, allowing for the use of scale invariant features such as those produced by SIFT. The feature is divided into 9 grid squares (3x3) and the mean colour of all of the pixels contained within the grid square is stored as a single RGB representative. Figure 7.6 demonstrates this process.

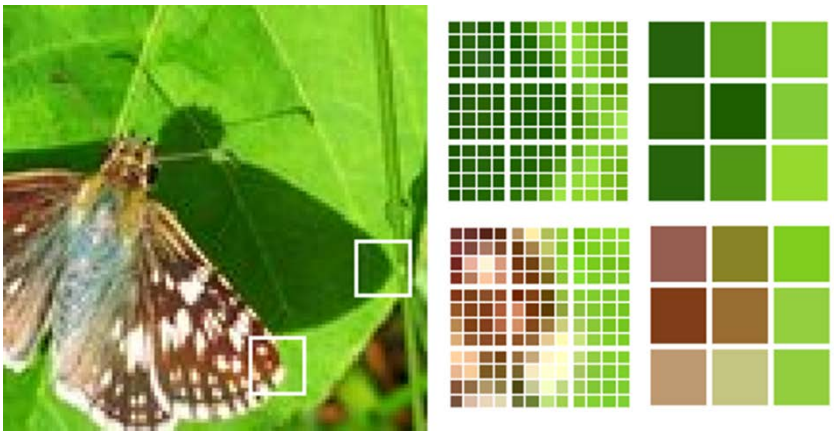


Figure 7.6 – Extracted features are broken into a 3x3 grid; the pixel colours within each grid square are then averaged to produce a 3x3 colour representation of the feature.

The colour of a feature (E) is represented by the mean RGB values of the 3x3 grid:

$$E = \begin{bmatrix} E_{R0} & E_{R1} & \dots & E_{Rn-1} \\ E_{G0} & E_{G1} & \dots & E_{Gn-1} \\ E_{B0} & E_{B1} & \dots & E_{Bn-1} \end{bmatrix}. \tag{63}$$

The mean colour of the feature is characterized by:

$$\mu(E) = \begin{bmatrix} \mu(E_R) \\ \mu(E_G) \\ \mu(E_B) \end{bmatrix} = \begin{bmatrix} \mu\{E_{R0} & \dots & E_{Rn-1}\} \\ \mu\{E_{G0} & \dots & E_{Gn-1}\} \\ \mu\{E_{B0} & \dots & E_{Bn-1}\} \end{bmatrix}. \tag{64}$$

The standard deviation of the RGB values for each feature is given by finding the standard deviation of each row of E.

$$\sigma(E) = \begin{bmatrix} \sigma(E_R) \\ \sigma(E_G) \\ \sigma(E_B) \end{bmatrix} = \begin{bmatrix} \sigma\{E_{R0} & \dots & E_{Rn-1}\} \\ \sigma\{E_{G0} & \cdots & E_{Gn-1}\} \\ \sigma\{E_{B0} & \dots & E_{Bn-1}\} \end{bmatrix} \tag{65}$$

Dividing each pixel value by the standard deviation of the entire feature produces an output scaling that emphasizes the difference between changes in brightness and changes in chromaticity. This emphasis simplifies the thresholding process.

7.3.1 Shadow Feature Removal

The chromaticity distortion colour model can be used to differentiate between object based features and illumination artefacts. This information can be applied to localization tasks to improve robustness by identifying and removing features that are the result of illumination alone. Dynamically illuminated environments contain dynamic shadows. The tracking of dynamic shadows will lead to erroneous localization; therefore, the removal of features based on dynamic lighting artefacts will improve the robustness of the localization system.

Both SIFT and FAST corner detection extract features from the ‘edges’ of objects in the image. These edges are therefore the focus of identifying a feature as shadow-based or object-based. To determine the likelihood of a feature being extracted from a shadow, the chromaticity distortion of the feature is calculated. The RGB value of each square in the 3x3 grid representing feature colour is compared to the mean colour of the entire feature. The greatest chromaticity distortion is then compared to a threshold to identify the feature as shadow-based or object-based. A feature extracted from the edge of a shadow will have a low chromaticity distortion due to the ‘edge’ containing a variation in illumination only. A feature extracted from the edge of an object will usually have a higher chromaticity distortion due to the edge containing a variation in colour, as seen in Figure 7.7.

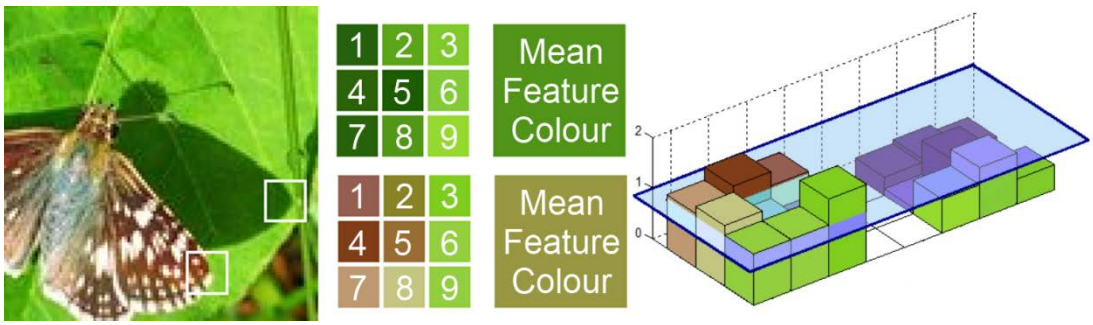


Figure 7.7 – The 9 squares from the 3x3 grids representing 2 different features are compared to the mean colour of each feature. The chromaticity distortion between each pixel and the mean feature colour is then compared to a threshold (shown in blue). If the feature has pixel values above the threshold, the magnitude of the chromaticity distortion suggests that the feature is object based.

Brightness and chromaticity distortion are determined by comparing each of the 9 squares in the 3x3 feature grid (E_i) to the mean colour value of the overall feature.

$$\alpha_i = \frac{\frac{E_{Ri}\mu(E_R)}{\sigma^2(E_R)} + \frac{E_{Gi}\mu(E_G)}{\sigma^2(E_G)} + \frac{E_{Bi}\mu(E_B)}{\sigma^2(E_B)}}{\left(\frac{\mu(E_R)}{\sigma(E_R)}\right)^2 + \left(\frac{\mu(E_G)}{\sigma(E_G)}\right)^2 + \left(\frac{\mu(E_B)}{\sigma(E_B)}\right)^2} \tag{66}$$

$$CD_i = \sqrt{\left(\frac{E_{Ri} - \alpha\mu(E_R)}{\sigma(E_R)}\right)^2 + \left(\frac{E_{Gi} - \alpha\mu(E_G)}{\sigma(E_G)}\right)^2 + \left(\frac{E_{Bi} - \alpha\mu(E_B)}{\sigma(E_B)}\right)^2} \tag{67}$$

The standard deviation of the feature is used here to emphasize small changes in colour across the feature. Even a minor colour deviation in a feature with only a small range of colour will produce a large chromaticity distortion vector. Therefore, only features that are truly illumination artefacts will have a chromaticity distortion range below a selected threshold (τ_{SFR}). These features are removed as they are not considered to be robust to changes in illumination. The remaining features can then be used for matching, increasing the probability of illumination robustness during localization.

7.3.2 Colour Based Matching

Colour information can also be used to improve the robustness of feature matching between images. Both SIFT and FAST corner detection use monochrome images for feature matching and therefore can be misled by lighting conditions that cause different objects to look similar in grey scale, as in Figure 7.8. A correctly matched feature will exhibit a small difference in average true colour, as opposed to an incorrectly matched feature which will have a significant difference in average true colour.

Through the use of the chromaticity distortion model to determine the true colour of a feature, mismatches can be identified and removed. The feature filtering occurs after the monochrome feature matching algorithm has finished producing matches; since the filtering is a post-processing task and not directly integrated into the

matching algorithm, the technique can be easily applied to any matching algorithm with little modification.

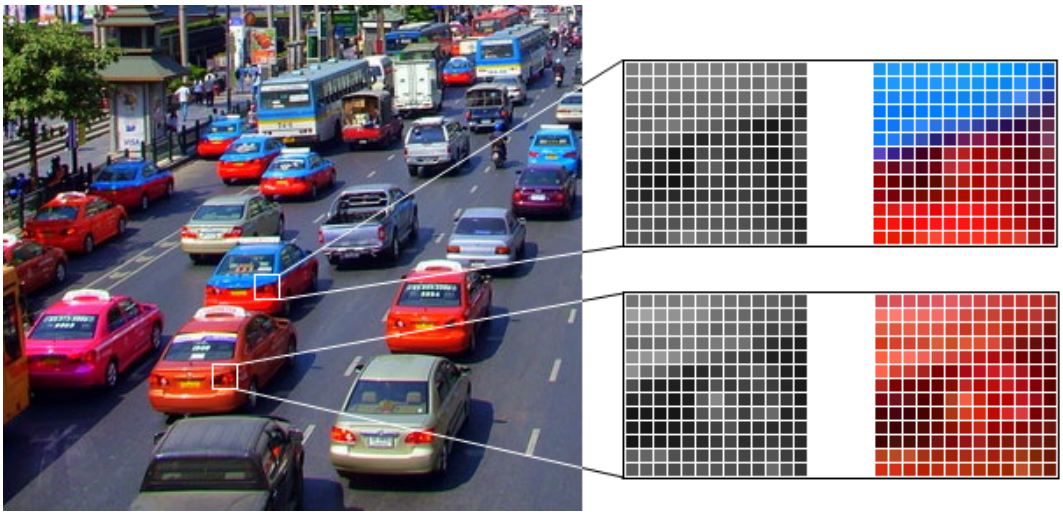


Figure 7.8 – Two different features can look similar in monochrome, leading to mismatches being produced by the standard matching algorithms employed by SIFT and Fast corner detection. Comparing the colour information of matched features can reduce the number of mismatches.

To identify a match as being of similar average true colour, the chromaticity distortion between the matched features is calculated and compared to a threshold. The distortion value should be low, regardless of lighting conditions, if a correct match has been made. A standard feature matching algorithm is used to produce a set of matched features. The brightness and chromaticity distortion is determined by comparing the colour of one feature (E) to the colour of the matched feature (I).

$$\alpha = \frac{\frac{\mu(I_R)\mu(E_R)}{\sigma(I_R)\sigma(E_R)} + \frac{\mu(I_G)\mu(E_G)}{\sigma(I_G)\sigma(E_G)} + \frac{\mu(I_B)\mu(E_B)}{\sigma(I_B)\sigma(E_B)}}{\left(\frac{\mu(E_R)}{\sigma(E_R)}\right)^2 + \left(\frac{\mu(E_G)}{\sigma(E_G)}\right)^2 + \left(\frac{\mu(E_B)}{\sigma(E_B)}\right)^2} \tag{68}$$

$$CD = \sqrt{\frac{(\mu(I_R) - \alpha\mu(E_R))^2}{\sigma(I_R)\sigma(E_R)} + \frac{(\mu(I_G) - \alpha\mu(E_G))^2}{\sigma(I_G)\sigma(E_G)} + \frac{(\mu(I_B) - \alpha\mu(E_B))^2}{\sigma(I_B)\sigma(E_B)}} \tag{69}$$

The standard deviation of colour across each of the features is again used to emphasize minor differences in colour. This is particularly powerful in scenes with recurring similar features as even slight differences in colour result in large

chromaticity distortion vectors. Conversely, matched features that have a chromaticity distortion above a selected threshold (τ_{CBM}) are discarded as they are considered likely to be a false match due to lighting conditions.

7.3.3 Comparison of Colour Models

The chromaticity distortion model has never before been applied to a localization task. To confirm that it was an appropriate choice of colour model, it was compared to two other models which separate colour from intensity – HSV and colour angles. Hue Saturation Value (HSV) is a cylindrical coordinate representation of RGB colour. It presents hue on a circular colour chart and then separately defines saturation and 'darkness' values. Alternatively, colour angles describe the difference between two colours in RGB space as the angle between the two RGB vectors. The colour models are evaluated based on their ability to distinguish shadow based features and object based features. The image of a handle casting a shadow on a door (Figure 7.9(a)) was used to compare the three techniques.

To commence the evaluation, SIFT is used to identify about 3000 features in the image. Each colour model is then tuned to reject around 1000 features based on a comparison of each square in the 3x3 grid to the mean feature colour, as described in Section 7.3.1. The results in Figure 7.9(b-d) show that the chromaticity distortion model is the only colour model to correctly reject the features on the edge of the shadow and on the highlight. HSV and colour angles produced similar results, both poorly identifying the shadow and highlight. The chromaticity distortion technique was trialed on other images to test the repeatability of the results and similar outcomes were achieved.

Without the addition of the standard deviation seen in Equation (66) and (67) chromaticity distortion produced similar results to colour angles and HSV. The reason for standard deviation being applied to chromaticity distortion rather than either of the other models came down to processing time. On average chromaticity distortion was processing three times faster than colour angles. The conversion of the RGB image to HSV using a lookup table significantly exceeded the chromaticity distortion processing time before any calculations using HSV values could be

produced. Since the focus of this work was on real-time applications for localization, chromaticity distortion was the obvious choice.

(a) Original Image.

(b) Chromaticity based classification.

(c) HSV based classification.

(d) Colour angles based classification.

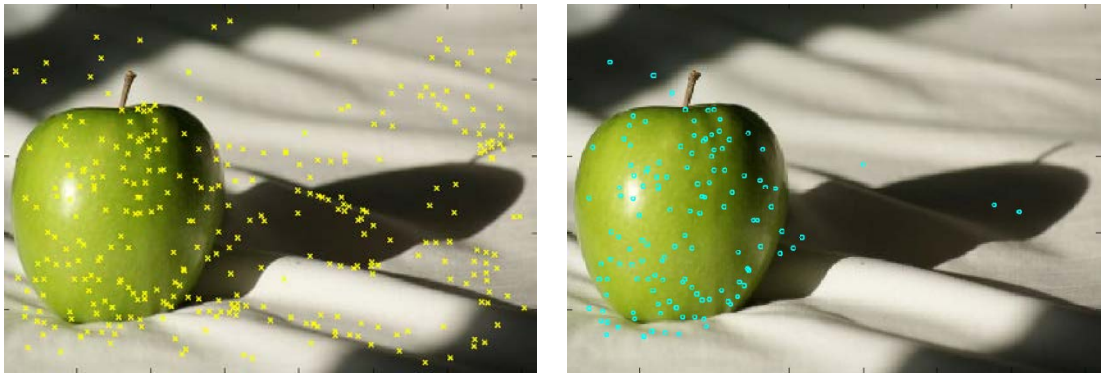
Figure 7.9 (a) An image with a shadow and highlight used for colour model comparison. Around 3000 SIFT features were extracted from the image, then 1000 of those features were removed that were identified as a distortion in illumination alone based on (b) the chromaticity distortion colour model, (c) the HSV colour model and (d) the colour angles colour model.

7.4 Initial Technique Evaluation

The chromaticity distortion colour model proved to be an ideal choice for the identification of features caused by illumination artefacts. The next stage of the evaluation is to apply the two techniques derived in Sections 7.3.1 and 7.3.2 to a simple feature extraction and matching task. Shadow Feature Removal (SFR) and Colour Based Matching (CBM) are first applied individually and then in tangent to fully assess the performance of both techniques.

7.4.1 Evaluating Shadow Feature Removal

The first level of testing for Shadow Feature Removal (SFR) is the removal of shadow based features from a single image. The chromaticity distortion between each square in the 3x3 feature colour grid and the mean feature colour was determined and compared to a threshold, as detailed in Section 7.3.1. Features with a chromaticity distortion below the threshold were removed. Figure 7.10(a) contains all of the features extracted by SIFT while Figure 7.10(b) shows the SIFT features retained based on their chromaticity distortion values. Features are mainly retained on the edges and patterning of the main object (the apple), whereas features are mainly removed from the edges and body of the shadows.



(a) All features identified by SIFT. (b) Features retained by chromaticity distortion filter.

Figure 7.10 (a) SIFT is used to extract features from the image, the feature centres are shown on the image while magnitude and orientation are hidden to reduce clutter. (b) Features that pass through the chromaticity distortion filter and are classified as object based.

7.4.2 Evaluating Colour Based Matching

To analyze the use of colour information to improve the robustness of standard monochrome feature matching algorithms, a computer generated dynamic illumination scenario was developed in POV-Ray (the Persistence of Vision Raytracer http://www.povray.org). This allowed explicit control over the light source properties and position. By controlling the camera and light source, two scenes could be produced that were identical apart from lighting conditions (see Figure 7.11). This could be used to assess the effectiveness of the technique, as correct feature matches would occur at the same x and y coordinates in both images.

Figure 7.11 – The two images produced by POV-Ray used to assess colour based matching. The only difference between the images is the position of the light source.

SIFT features were extracted from both images. The standard matching algorithm was executed first to produce a benchmark for testing. Colour Based Matching (CBM) was then employed to determine the true colour difference between matched features. If the difference was above a threshold the features were reclassified as mismatches. This technique does not produce any additional matches; rather it only removes false positives produced by standard matching algorithms. This post-processing characteristic makes the technique applicable to any matching algorithm that produces a list of matched features.

Threshold	**Total Matches**	**Mismatches**	**% Error**
No Threshold	954	168	17.6
$\tau_{CBM} = 0.1$	373	24	6.4
$\tau_{CBM} = 0.05$	250	6	2.4

Table 7.1 – Matched features that have a chromaticity distortion below the threshold τ_{CBM} are retained. Tighter thresholds reduce the total number of matches but also reduce the percentage error.

The results in Table 7.1 demonstrate that despite the reduction in the number of total matches, the percentage error is greatly reduced. This evidence shows that standard monochrome feature matching algorithms can often be misled by colours that appear similar in grey scale. In fact, 96% of the mismatches between the two images seen in Figure 7.11 can be correctly identified through the use of colour information. The total number of feature matches remaining after the filtering is still significant and would actually need to be reduced further for real-time localization applications.

7.4.3 Combining Both Techniques

Shadow feature removal and colour based matching can work together to further improve the effectiveness of feature matching between images with variations in illumination. To test the effect that a combination of both techniques has on the feature matching accuracy, POV-Ray was again used to produce a series of identical images with variations in lighting (see Figure 7.12). SIFT was used to extract features from the images which were then filtered using SFR to remove shadow based features. The remaining features were then matched using the standard feature matching algorithm. The list of feature matches was then analyzed using CBM to further reduce the number of mismatches. The results from combining both techniques can be seen in Table 7.2.

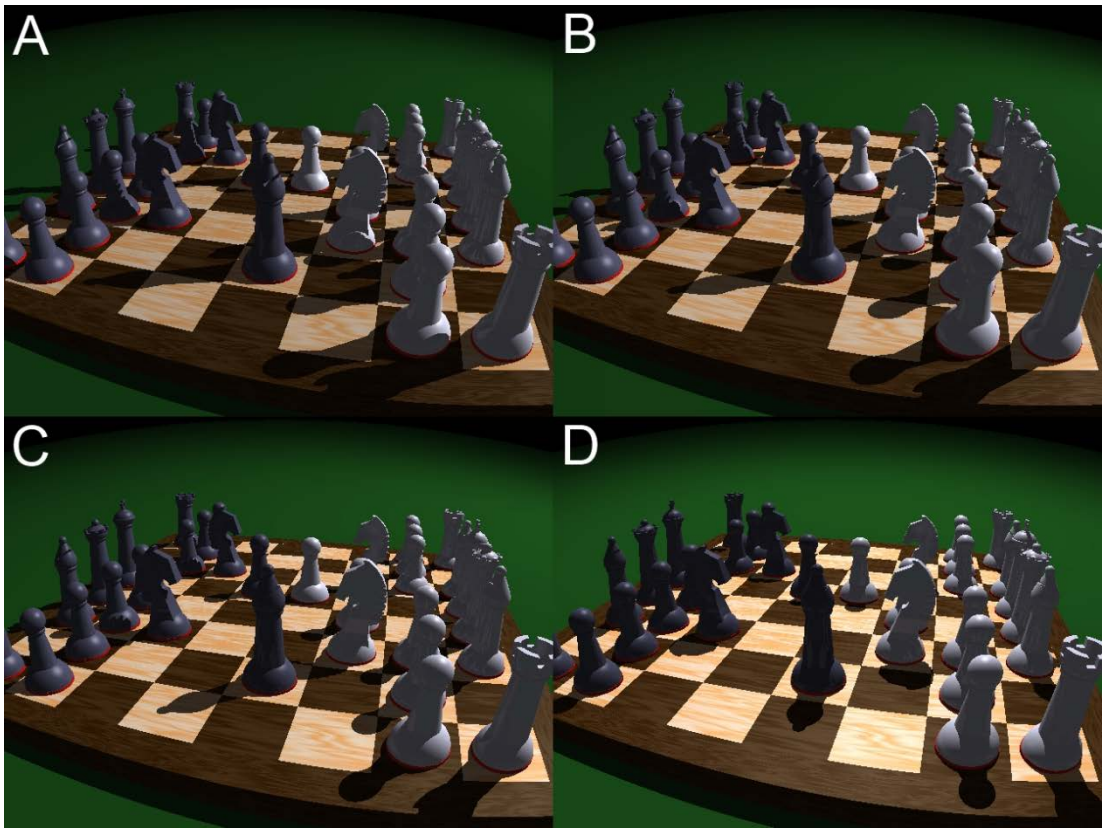


Figure 7.12 – The four images used to assess a combination of both chromaticity distortion based techniques. Each image is identical apart from variations in illumination. Correctly matched features will therefore occur in the same x and y coordinates in each image.

Images	Threshold	Total Matches	Mismatches	% Error
A – B	No Threshold	954	168	17.6
	τ_{SFR}=1.4, τ_{CBM}=0.1	105	6	5.7
	τ_{SFR}=1.4, τ_{CBM}=0.05	71	1	1.4
A – C	No Threshold	657	147	22.4
	τ_{SFR}=1.4, τ_{CBM}=0.1	75	5	6.7
	τ_{SFR}=1.4, τ_{CBM}=0.05	58	0	0
A – D	No Threshold	484	145	29.9
	τ_{SFR}=1.4, τ_{CBM}=0.1	48	3	6.2
	τ_{SFR}=1.4, τ_{CBM}=0.05	33	1	3.0

Table 7.2 – Features are removed initially if they are below the shadow feature removal threshold (τ_{SFR}). After initial matching features are also removed if they have a difference in chromaticity above the colour based matching threshold (τ_{CBM}).

The results show a dual improvement: a reduction in the percentage of incorrect matches and an output containing a small number of high quality matches rather than a large number of low quality matches. The percentage of mismatches dropped from 17.6% to 1.4% for the images with the smallest change in lighting (A – B) and from 29.9% to 3.0% for the images with the largest change in lighting (A – D). The number of total matches also dropped from 954 to 71 for the images A – B and from 484 down to 33 for images A – D. These numbers of features are far more suited to real-time applications than the total number of matches without filtering. Figure 7.13 contains a plot of percentage error throughout the experiment.

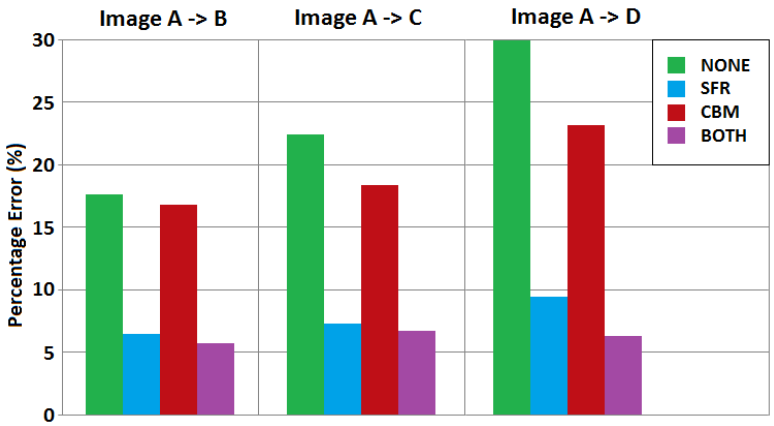


Figure 7.13 – A plot of percentage error across all experiments. The effects of shadow feature removal and colour based matching are analysed individually as well as in unison. The 'No Threshold' results represent the unfiltered percentage error.

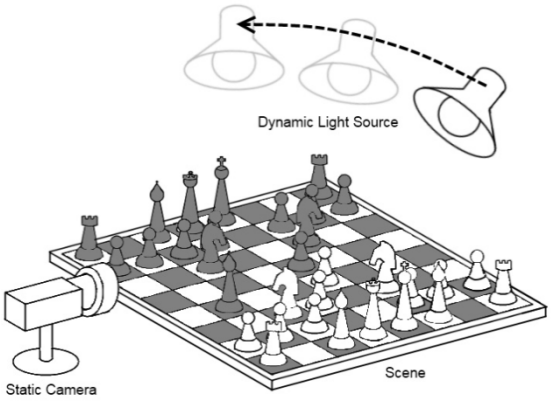
7.5 Simulated Performance

The ability of shadow feature removal and colour based matching to improve feature matching accuracy was demonstrated in Section 7.4. However, the two techniques were designed for a localization task based on a series of monocular images. To adequately assess the two techniques, they must be proven to be effective in a dynamically illuminated localization task. POV-Ray was again used to produce a series of evaluation sequences containing different combinations of camera movement and environmental lighting conditions. These sequences were used to assess the performance of shadow feature removal and colour based matching in a range of localization tasks.

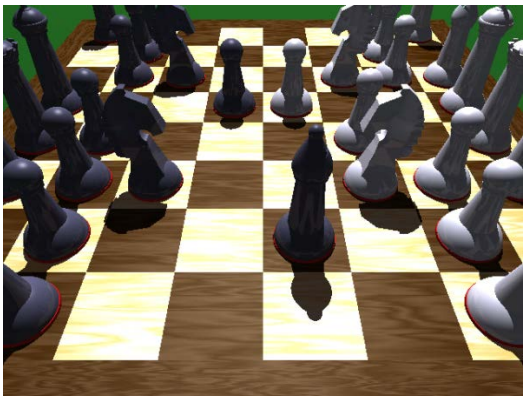
7.5.1 Static Camera with Dynamic Illumination

The simplest way to examine the effect that dynamic illumination has on localization is to test a static scene with a stationary camera and dynamic illumination. The correct localization path is for the camera not to move, despite the movement of shadows within the scene. FAST Corner Detection is used to extract features for the localization applications, so the chromaticity distortion technique will be applied to the features extracted by this algorithm. The SLAM implementation for this experiment was based on the six degree of freedom monocular EKF SLAM algorithm written by Civera *et al.* [51]. The algorithm was only modified by the addition of the shadow feature removal and colour based matching algorithms. The scene analyzed in simulation contains the same chess board as Figure 7.11. The camera remains stationary while the single light source moves from slightly behind the board to directly overhead, casting the dynamic shadows seen in Figure 7.14(b).

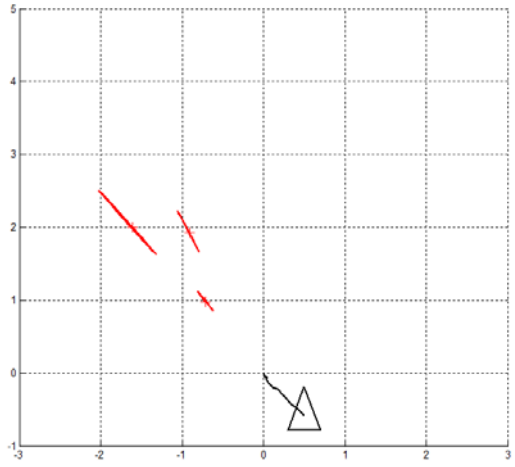
The results shown in Figure 7.14 are a top view representation of the 3D localization produced by the monocular SLAM algorithm. Each image shows the localization path as a line starting at the origin (0, 0) and finishing in the center of the triangle representing the final orientation of the camera. The ellipsoids in each image center on a tracked feature and represent the locational uncertainty of that feature. The depth values are dimensionless as the scene is computer generated; however, the results are scaled uniformly to allow for comparison.



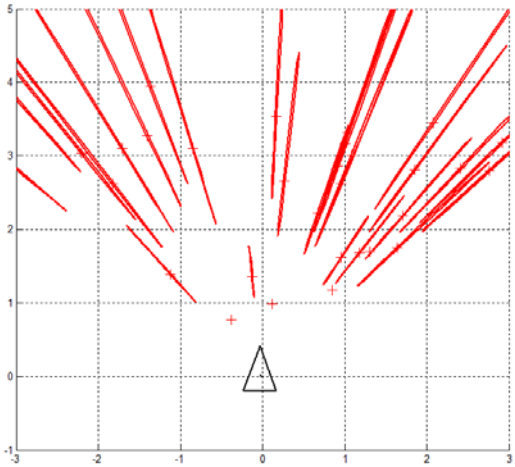
(a) Depiction of scene.



(b) View from camera.



(c) Localization path of standard SLAM algorithm.



(d) Localization path of compensated SLAM algorithm.

Figure 7.14 (a) Layout of the scene used to analyse the performance of the algorithms using a static camera under dynamic lighting. The light source moved from behind and slightly above the scene, to directly overhead. (b) The view of the scene from the static camera . (c) The localization path of the standard SLAM algorithm. The lack of features is due to the combination of tracked static object features and tracked dynamic shadow features causing feature depth estimates to erroneously drift off screen. (d) The localization path produced by the SLAM algorithm incorporating shadow feature removal and colour based matching.

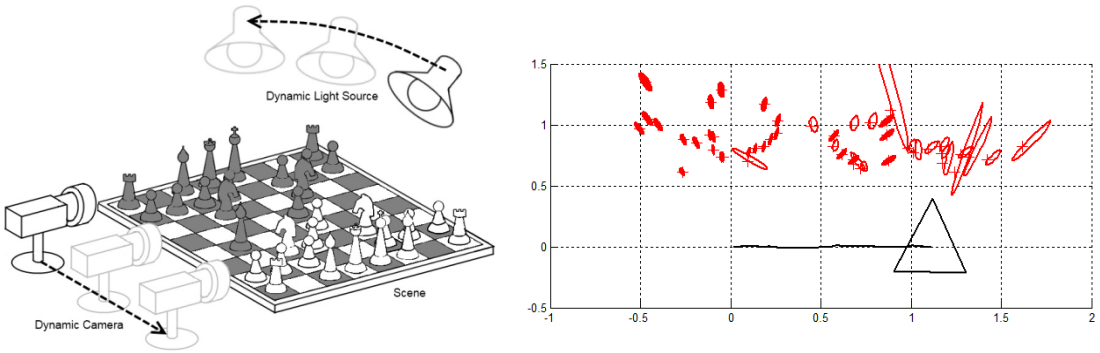
The results of running the uncompensated SLAM algorithm on the scene can be seen in Figure 7.14(c). As the shadows move towards the back of the scene, the camera localization assumes that they are stationary and therefore produces an erroneous localization path which moves the camera away from the features. This behavior resulted in the majority of the features drifting off the screen completely as their depth estimates are continually distorted. The localization path resulting from the addition of the chromaticity distortion based algorithms to the SLAM algorithm can be seen in Figure 7.14(d). Since the moving shadow features are adequately filtered during localization, the camera does not move from the origin. The long uncertainty

ellipsoids are due to the scene only being viewed from a single position, so no depth estimate of the features in the scene is possible.

7.5.2 Dynamic Camera with Dynamic Illumination

To fully assess the effect that chromaticity distortion filtering was having on localization, a dynamic scene with dynamic lighting needed to be evaluated. To complete the assessment accurately, a series of computer generated images was produced (using POV-Ray) to allow full control of camera movement and lighting conditions. The scenario was again based around the chess board in Figure 7.11. The camera moved with a constant velocity in a straight line parallel to the edge of the chess board, while the single light source moved from slightly behind the chess board to directly overhead, causing dynamic shadows.

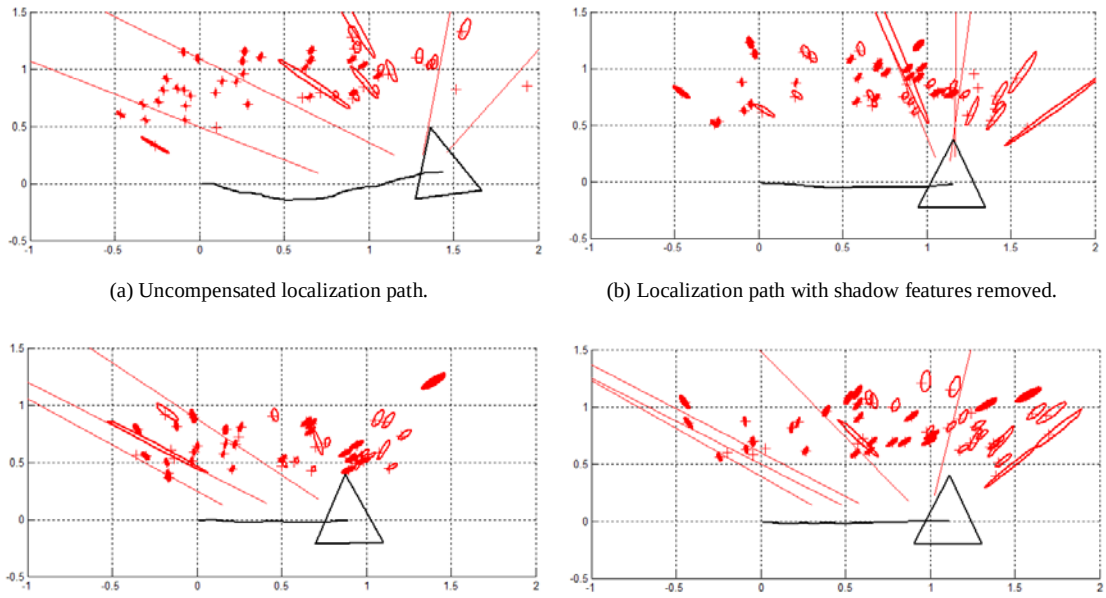
The plots seen in Figure 7.15 and Figure 7.16 are a top view representation of the 3D localization produced by the test algorithms. Each image is in the same format as the results in Section 7.5.1, showing the localization path, feature uncertainty ellipsoids and final camera orientation. The remaining lines point to features that have only been identified in a single image and therefore do not have a depth estimate.



(a) Depiction of scene.

(b) Baseline localization under static illumination.

Figure 7.15 (a) Layout of the scene used to assess the performance of the two algorithms in a localization task with dynamic camera movement. The camera moves from the left of the chess board to the right. Meanwhile, the light source moves from behind and slightly above the board to directly overhead. (b) The localization path used as a baseline for comparison. The path was generated using the standard SLAM algorithm under static illumination.



(a) Uncompensated localization path.

(b) Localization path with shadow features removed.

(c) Localization path with colour based matching.

(d) Localization path with both techniques implemented.

Figure 7.16 (a) The localization path produced by running the uncompensated SLAM algorithm on the dynamically illuminated scene. (b) The localization path with shadow features removed before feature matching. (c) The localization path when feature matching is filtered using colour information. (d) The localization path resulting from a combination of both filtering techniques.

The first test was conducted with a fixed light source to establish a baseline for comparison. The localization path shown in Figure 7.15(b) was in a straight line and there was no change in camera orientation. This was the expected localization path based on the earlier description of the camera movement through the scene. The second test introduced the dynamic light source and was run using the unfiltered SLAM algorithm. The dynamic shadows have a significant influence on the effectiveness of the localization, resulting in the unreliable path seen in Figure 7.16(a).

The third test removed shadow features identified by their chromaticity distortion. This approach produced a significant improvement in localization, yet still contained evidence of the error caused by the dynamic light source as seen in Figure 7.16(b). The fourth test used colour information to filter feature matches based on their chromaticity distortion. The localization path produced by this approach significantly reduced the error caused by the dynamic light source. The consequence of this improvement was a large reduction in the number of maintained matches, resulting in the discrimination between the displacement of this path and the baseline path

(compare Figure 7.16(c) and Figure 7.15(b)). The final test combined both filtering techniques. The resulting localization path seen in Figure 7.16(d) demonstrates the effectiveness of the approach. The distortion produced by the dynamic light source was completely removed and the high level of maintained features produced the correct displacement. These results demonstrate the effectiveness of the two novel techniques as approaches to overcoming the challenges of localization in dynamically illuminated environments.

7.5.3 Noise

Computer generated images are a useful way to quickly assess the performance of filtering based on chromaticity distortion; however, to completely assess the competence of the algorithm, camera defects need to be included that would be found in real world datasets. Images taken by an actual camera contain noise such as lens distortion, unfocused features, lens flare, graininess and poor contrast. These five types of noise were all added to the simulated dataset used in Section 7.5.2 to test the algorithm's robustness. Figure 7.17 compares an image from the dataset before and after the addition of camera defects.

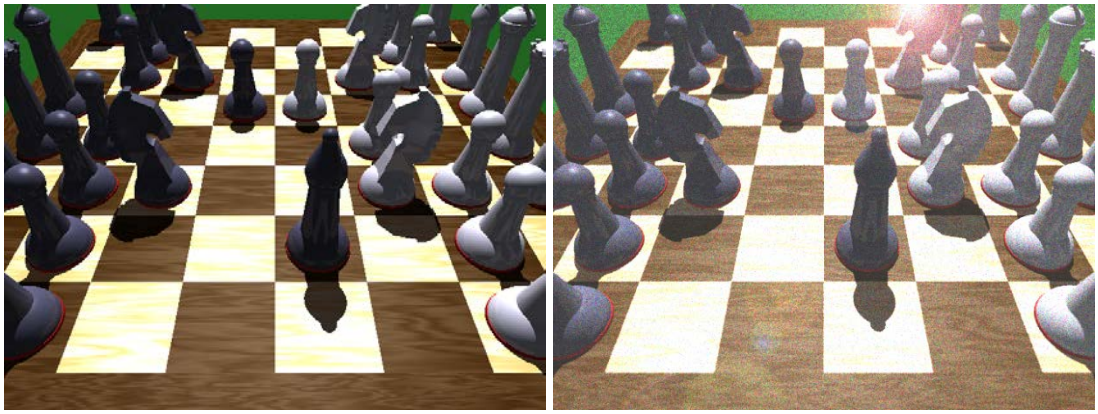
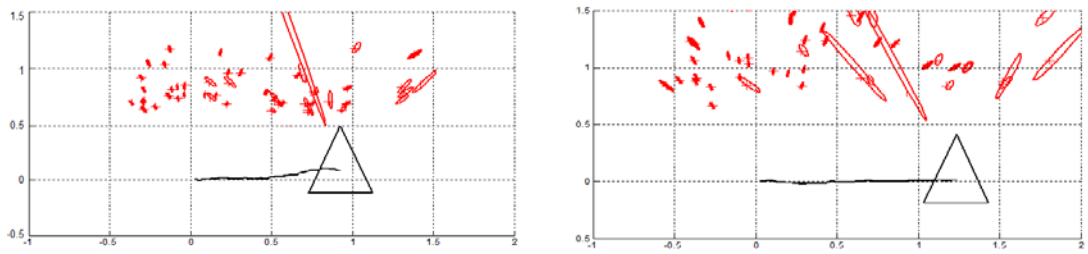


Figure 7.17 – A comparison of an image with no noise (left) and an image with lens distortion, unfocused features, lens flare, graininess and poor contrast (right).

The standard SLAM algorithm was first used to assess the effect that noise had on the uncompensated localization path. The results in Figure 7.18(a) show that the addition of noise has further deteriorated the original uncompensated localization path seen in Figure 7.16(a). However, the addition of noise has only a minor influence on the localization path resulting from the implementation of shadow

feature removal and colour based matching. A comparison of the noisy localization path in Figure 7.18(b) to the noiseless path in Figure 7.16(d) demonstrates the robustness of the two derived compensation algorithms. The next stage of assessment was to trial the algorithms on real world datasets with dynamic illumination.



(a) Noisy localization from standard SLAM algorithm.

(b) Noisy localization from compensated SLAM algorithm.

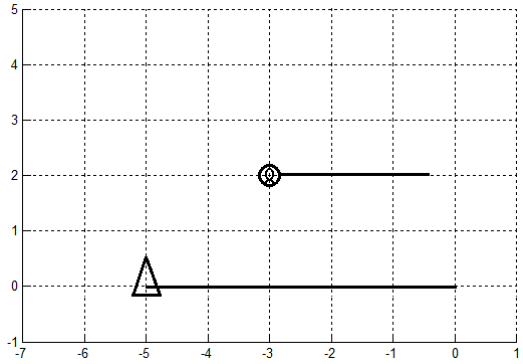
Figure 7.18 (a) The localization path of the uncompensated SLAM algorithm on the noisy data sequence. It is substantially different to the data sequence without noise. (b) The localization path of the SLAM algorithm with shadow features removed and colour based matching on the noisy dataset. There is little difference when compared to the sequence without noise, demonstrating the robustness of the two techniques.

7.6 Real World Performance

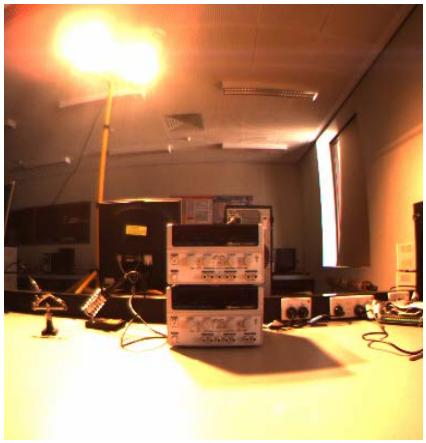
Shadow feature removal and colour based matching were shown, through the use of simulation, to be highly robust to dynamic illumination during a localization task. The addition of simulated noise also did not impede the quality of the localization results. However, there is no substitute for real world environments, so the next level of testing was based on actual images recorded by a single camera from a Point Grey Ladybug 2. The images were heavily distorted and contained significant noise, allowing the full assessment of the robustness of the two techniques.

7.6.1 Dynamic Camera with Dynamic Illumination

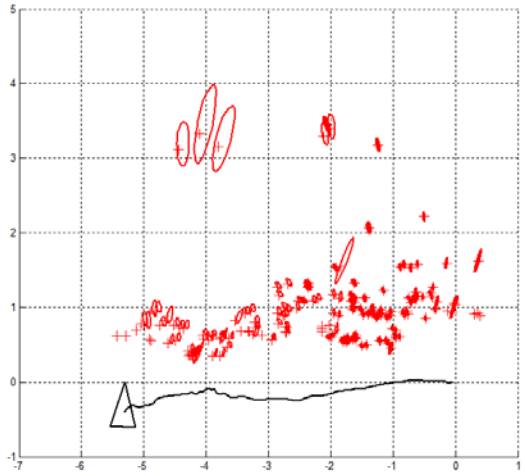
To analyze real world performance, a dataset was recorded of movement through a scene with dynamic lighting as seen in Figure 7.19(b). The camera moves through the scene sideways in controlled laboratory conditions so that the ground truth could be accurately measured. The camera moves five meters to the left in a straight line, while the light source moves parallel to the camera yet at a slower speed producing shadows that drift away from the direction of camera movement (Figure 7.19(a)).



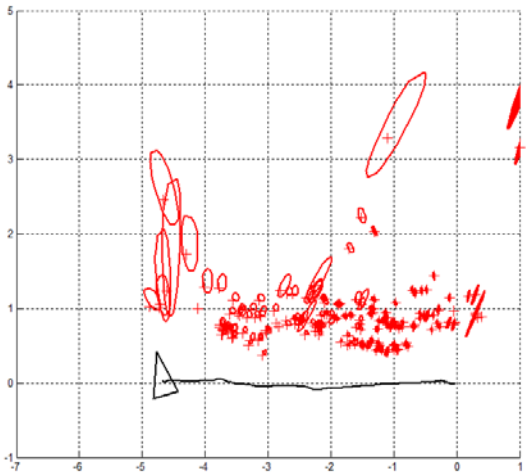
(a) The ground truth path of the camera and the primary light.



(b) An image from the dynamic camera dataset.



(c) Localization path from standard SLAM algorithm.



(d) Localization path from compensated SLAM algorithm.

Figure 7.19 (a) The ground truth for the scenario, the triangle represents the camera (with the orientation representing the camera orientation) which travels from the origin (0,0), 5 meters to the left (-5,0). The circle represents the light source which moves from half a meter to the left (-0.5,0) to 3 meters to the left (-3,0). (b) An example image from the test dataset of dynamic camera movement through a real world scene with dynamic illumination. (c) The localization path of the standard SLAM algorithm shows significant drift. (d) The localization path of the SLAM algorithm using shadow feature removal and colour based matching has minor drift.

This dataset was first run through the uncompensated monocular SLAM algorithm and then the compensated algorithm. The localization results are again presented in the same format as previous results. Applying the uncompensated SLAM algorithm to the dataset produced accurate localization for a brief window of time, as expected by the slight illumination invariance characteristic of most feature extraction techniques. As the dataset continues, however, the tracking of dynamic shadow features leads to the incorrect localization path seen in Figure 7.19(c). Far superior results are produced by the algorithm incorporating shadow feature removal and

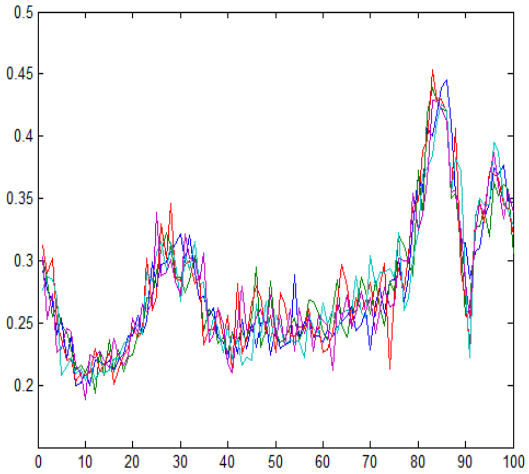
colour based matching (Figure 7.19(d)) where the localization path is highly correlated to the ground truth. The resulting map was manually scaled based on the real world distances of well-established visual features to enable a clear metric comparison with the ground truth.

7.7 Algorithm Features

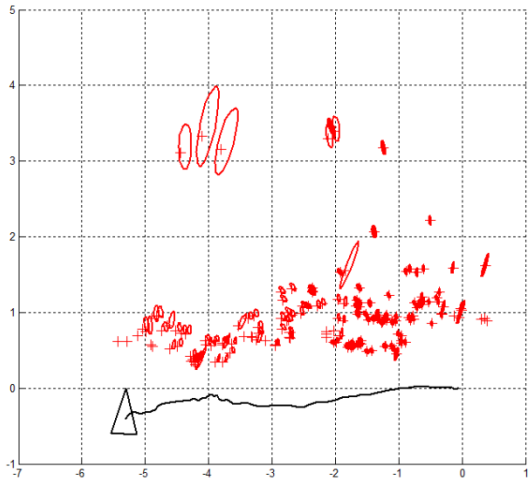
Throughout the testing of shadow feature removal and colour based matching, many improvements were made to optimize performance. Automatically scaling thresholds were included to improve localization results and the code structure was optimized to increase real-time performance. An additional feature yet to be investigated is the ability to localize the primary light source(s) based on the positions and movement of shadow based features. This feature will be relinquished as future work as discussed in Section 9.2.

7.7.1 Automatically Scaling Thresholds

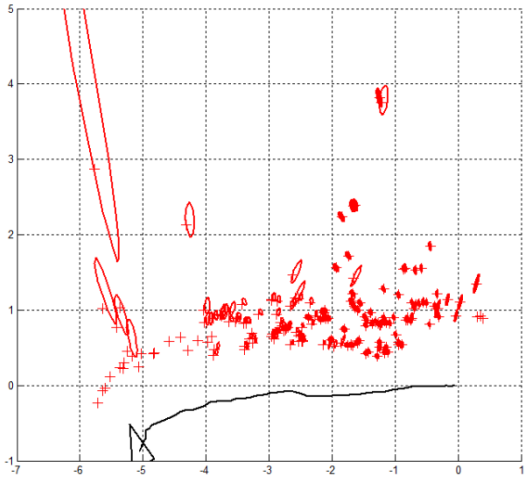
The thresholds required for improved localization using either shadow feature removal or colour based matching are dependent upon the colour range of the image. Images that are low in colour contrast need to have the shadow feature removal threshold relaxed, or else the majority of genuine features may be classified as shadows. Conversely, the colour based matching threshold must be stringent otherwise the colour difference between mismatched features may pass through the chromaticity distortion filter. To overcome this discrimination, automatically scaling thresholds were implemented based on the colour content of an image. The chromaticity distortion colour model was again utilized to determine the colour range of an image. The chromaticity distortion of a sample of random pixels was compared to the mean colour of the entire image. The standard deviation of the sample correlated with the colour range present in each image and was used to scale the threshold.



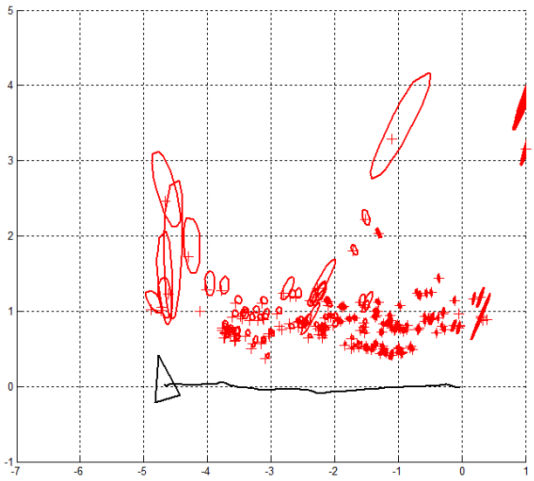
(a) Standard deviation of random pixel chromaticity distortions.



(b) Path from standard SLAM.



(c) Path from compensated SLAM with set threshold.



(d) Path from compensated SLAM with auto-scaling threshold.

Figure 7.20 (a) The colour range of each image in a sequence is determined by finding the standard deviation of the chromaticity distortion value of a random sample of pixels. (b) The localization path produced by the standard SLAM algorithm. (c) The localization path produced by the compensated SLAM algorithm with a set threshold. (d) The localization path produced by the compensated SLAM algorithm with an auto-scaling threshold.

The standard deviation of 400 random pixels per image over a 100 image sequence can be seen in Figure 7.20(a). The five lines in the plot represent five executions of the random sampling algorithm to check for consistency. The localization path of the 100 image dataset without any compensation can be seen in Figure 7.20(b), this is the same dataset as used in Section 7.6.1. The localization path of the sequence with a fixed shadow feature removal threshold of 1.0 can be seen in Figure 7.20(c). The fixed threshold cannot adapt to the range of colour contrast experienced during the sequence (illustrated in Figure 7.20(a)) and fails to robustly filter the shadow based

features. To improve this result, an automatically scaling threshold was applied to the dataset based on the standard deviation range seen in Figure 7.20(a). The standard deviation of each frame was scaled such that the resultant threshold was always between 0.5 and 1.5. These vales represent the extreme operating conditions of the filter. Below this limit, the majority of features are passed, making the filter redundant. Above the limit, a substantial number of features are classified as shadow features and are rejected, significantly increasing the processing time or even arresting the localization altogether. The localization path resulting from automatically scaling the threshold can be seen in Figure 7.20(d). The same approach can be applied to the colour based matching threshold, also improving localization precision. The thresholds required for the chromaticity distortion based filter can therefore be determined in real-time without prior knowledge of the dataset or lighting conditions.

7.7.2 Real-time performance

Vision-based localization is a task that must be capable of real-time implementation, otherwise it becomes redundant. In order to demonstrate the capabilities of the two chromaticity distortion techniques as genuine approaches to improve vision-based localization, they too had to be executable in real-time. The implementation of the two techniques was coded in Matlab. Since Matlab is designed for research and not for real-time deployment, significant improvements could be made to the execution times of the algorithms by converting them to C code.

The results from the processing time tests can be seen in Table 7.3. Initially, the series of operations were executed in an unoptimized for-loop resulting in large processing times. The code was then modified so that all of the repetitive math work was completed using matrix operations. This significantly improved the resulting processing times. The total time required to run both shadow feature removal, colour based matching and to determine the scaling factor for automatically scaling thresholds is 0.00778s per frame. This accounts for 23.34% of the available processing time available when operating at real-time frame rates of 30Hz. The processing time is also consistent, as shown by the plot in Figure 7.21 where the optimized shadow feature removal algorithm was executed over 100 iterations.

Although there are some spikes in processing time, the variance is low and the spikes may be accounted for by running the test on a non-dedicated laptop. The laptop used for the evaluation had a 2.67GHz dual core processor.

Code	Processed	Time	Frequency	%Resources @30Hz
Unoptimized				
SFR	2873 Features	5.873s	0.17Hz	17619%
CBM	506 Matches	0.2299s	4.35Hz	690%
Optimized				
SFR	2873 Features	0.00641s	156Hz	19.23%
CBM	506 Matches	0.00025s	3986Hz	0.75%
Auto Threshold	400 Samples	0.00112s	890Hz	3.37%
Combined	All	0.00778s	128Hz	23.34%

Table 7.3 – The processing time for both of the chromaticity distortion based filtering techniques. Both the optimized and unoptimized execution times are recorded. Shadow feature removal was executed on the 2873 features extracted from a standard test image. Colour based matching was executed on the 506 matches produced between two test images. The final column shows the percentage of processing resources consumed while running at 30Hz.

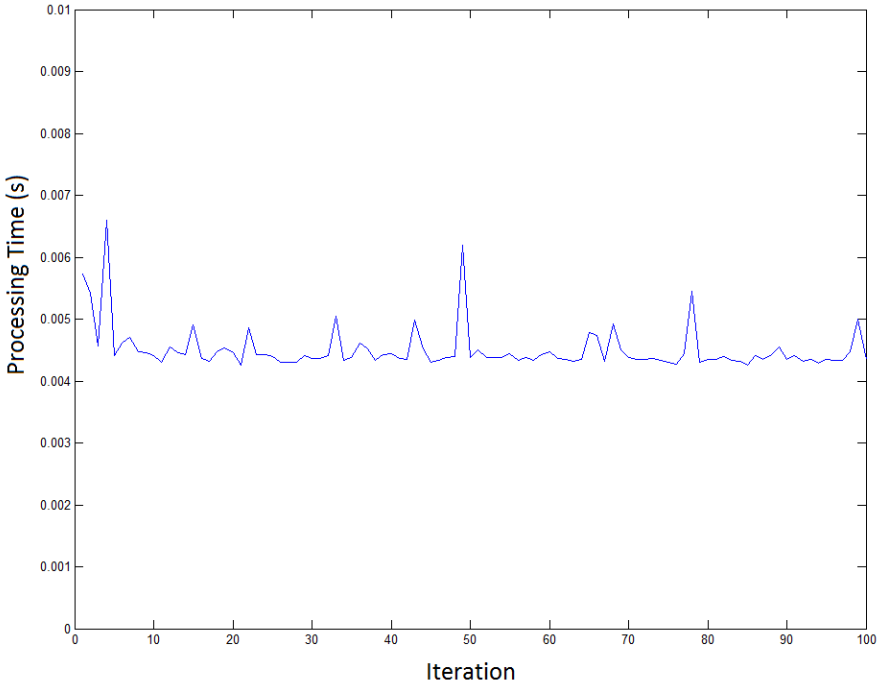


Figure 7.21 – The processing time for the optimized shadow feature removal algorithm over 100 iterations.

Chapter 8

Multisensor SLAM Results

8.1 Large Scale Indoor Results

For an initial evaluation of the effectiveness of our large scale 3D mapping system, a dataset was collected within the architecture building on the Curtin University Bentley campus. A Leica ScanStation 3D laser scanner was used to record a total of five 3D scans along the internal length of the building, approximately 20 meters apart. A cross-sectional view of each of the scans is shown in Figure 8.1. A Point Grey Ladybug 2 omnidirectional camera was also used to record a continuous stream of images as the sensor payload was moved between laser scan locations.

This first test of our multisensor SLAM system was to successfully co-register the five laser scans from the architecture building, producing a cohesive large scale map. Initially, the laser scans have no known position or orientation information. The scans are provided with an approximate initial pose generated by the monocular SLAM algorithm described in Section 4.3 and scaled using the technique in Section 6.2. The individual scans are then registered and rendered using the approach in Section 6.3 to produce a large scale 3D map.

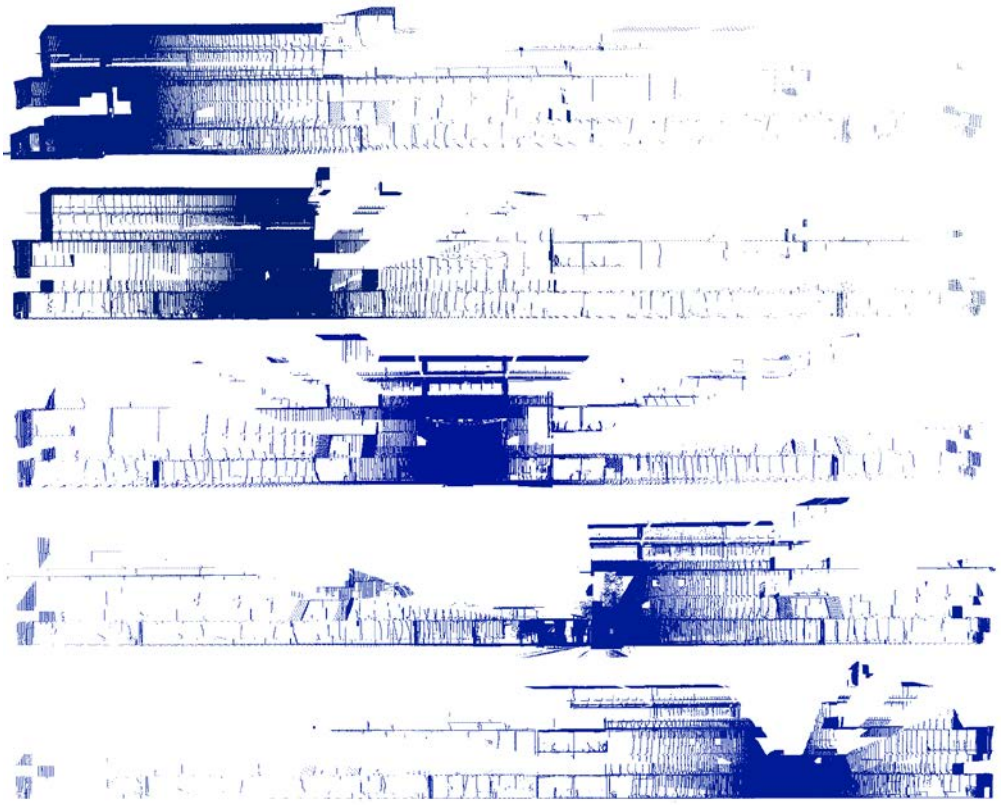
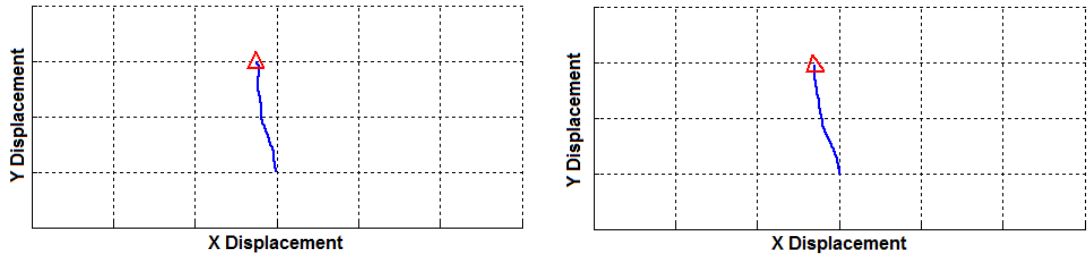
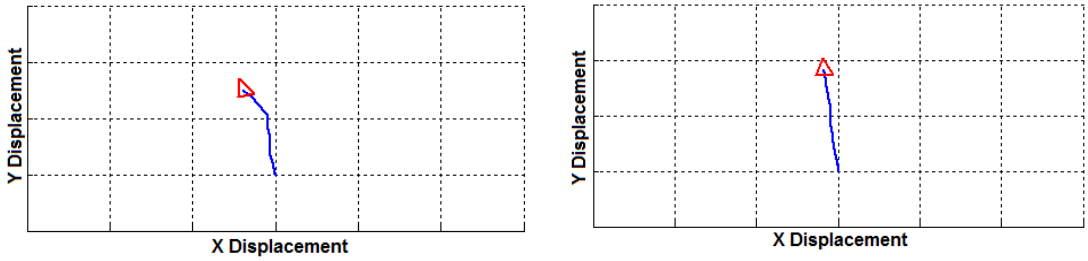


Figure 8.1 – The five 3D laser scans recorded in the architecture building. The order of scans is 1 (top) to 5 (bottom).



(a) Raw localization result of scan 1 → 2.

(b) Raw localization result of scan 2 → 3.



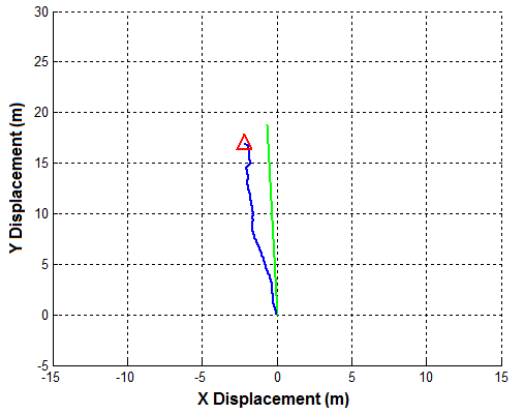
(c) Raw localization result of scan 3 → 4.

(d) Raw localization result of scan 4 → 5.

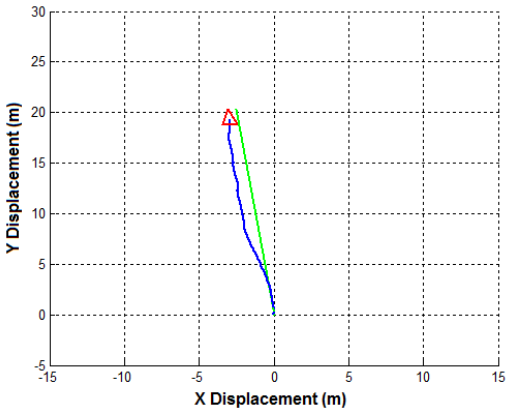
Figure 8.2 – The raw localization results from the four datasets collected by the Ladybug camera. The results have no scale at this stage as they are yet to be combined with the depth data recorded by the laser.

The position of the first scan is considered the global origin point for both the map of the building and the localization. The stream of images provided by the Ladybug camera during transit between scans is used for localization. Initially the localization result has no scale due to the dimensionless nature of bearing-only SLAM. The four dimensionless localization results can be seen in Figure 8.2.

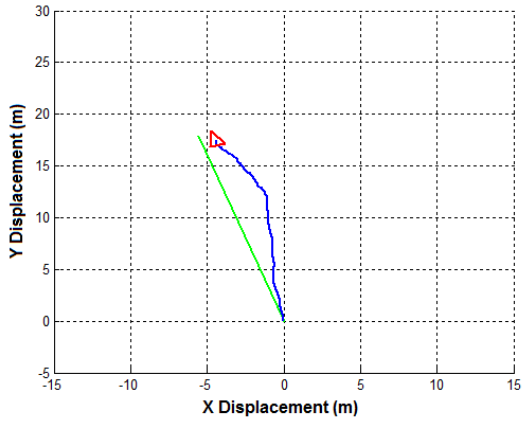
The laser information is then used to scale the localization results based on the depth measurements of well-established features, as described in Section 6.2. The correctly scaled localization results are shown in Figure 8.3. The final position and orientation information produced by the vision-based localization can be found in Table 6.1 in Section 6.3.



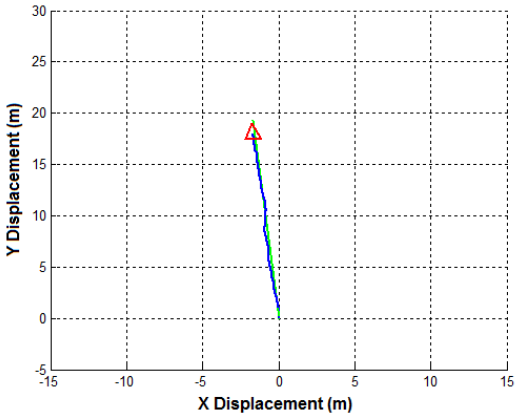
(a) Scaled localization 1 → 2 (blue) vs ground truth (green).



(b) Scaled localization 2 → 3 (blue) vs ground truth (green).



(c) Scaled localization 3 → 4 (blue) vs ground truth (green).



(d) Scaled localization 4 → 5 (blue) vs ground truth (green).

Figure 8.3 – The localization results from Figure 8.2 after scaling based on the laser range data of well-established features. The ground truth is an approximately straight line path between surveyed scan locations.

The position and orientation information supplied by vision-based localization is used as the initial pose estimate for the Iterative Closest Point (ICP) algorithm, as described in Section 6.3. The resulting fitness scores for each of the five laser scans in recorded in Table 8.1. The fitness scores are the mean squared error of the nearest neighbour distance between point clouds. As such, the scores can only be compared between subsequent results from the same two point clouds. The large fitness score for scan 4 is due to a significant number of outliers with no correct alignment points in the other scans (primarily produced by trees scanned through windows in the building). These points could have been discarded; however we decided to include every point in every scan to maintain consistency.

Registration	**2 → 1**	**3 → 1/2**	**4 → 1/2/3**	**5 → 1/2/3/4**
Fitness Score (m^2)	0.5746	1.0888	226.95	3.2754
Avg. Neighbour Offset (m)	0.7580	1.0435	15.064	1.8098

Table 8.1 – Fitness scores and average nearest neighbour offset distances for each of the four registration results from the architecture building dataset.

The transformation resulting from the ICP algorithm is then applied to the localization result to reduce the impact of long term drift. A visual representation of the improvement to the localization result is available in Figure 6.4 in Section 6.3. To provide an estimated measurement of the localization improvement, the area enclosed between the approximated ground truth and the localization result is estimated before and after ICP based correction, as shown in Figure 8.4. The ground truth is known to be an approximately straight line path between surveyed scan locations. The offset error for each localization result is recorded in Table 8.2. The large offset errors before ICP correction are caused by a gradual z-axis drift not observable in the 2D (x-y) representation of the results in Figure 8.3.

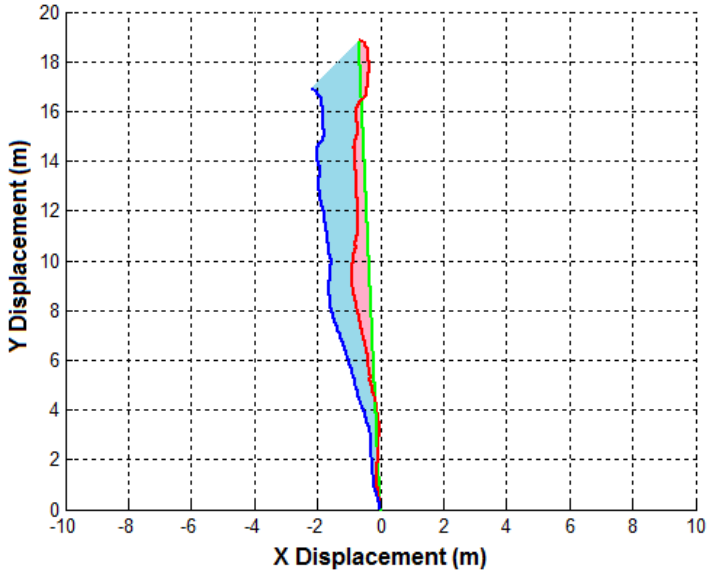


Figure 8.4 – To determine the offset error before (blue) and after (red) ICP correction, the area between each localization path and the ground truth is found.

Dataset	**1 → 2**	**2 → 3**	**3 → 4**	**4 → 5**
Original Offset (m^2)	32.12	21.72	30.43	19.57
Corrected Offset (m^2)	4.76	7.34	18.69	2.48

Table 8.2 – Offset error for each localization dataset before and after ICP based correction.

The final large scale mapping result is shown in detail in Figure 8.5, Figure 8.6 and Figure 8.7. A cross section of the rendered map is available in Figure 6.5 in Section 6.3. The final map has an average nearest neighbour offset distance of 1.49cm when compared to the full map produced by the semi-autonomous commercial technique described in Section 6.4. This does not necessarily mean that the map produced by our multisensor SLAM system has an average error of 1.49cm, as there are many possible sources of error in the conventional technique and it therefore cannot be guaranteed to be an inerrant ground truth. However, the average offset does demonstrate that the map produced by our multisensor SLAM system is highly correlated to the map produced by conventional techniques and can therefore be considered as a convincing alternative.

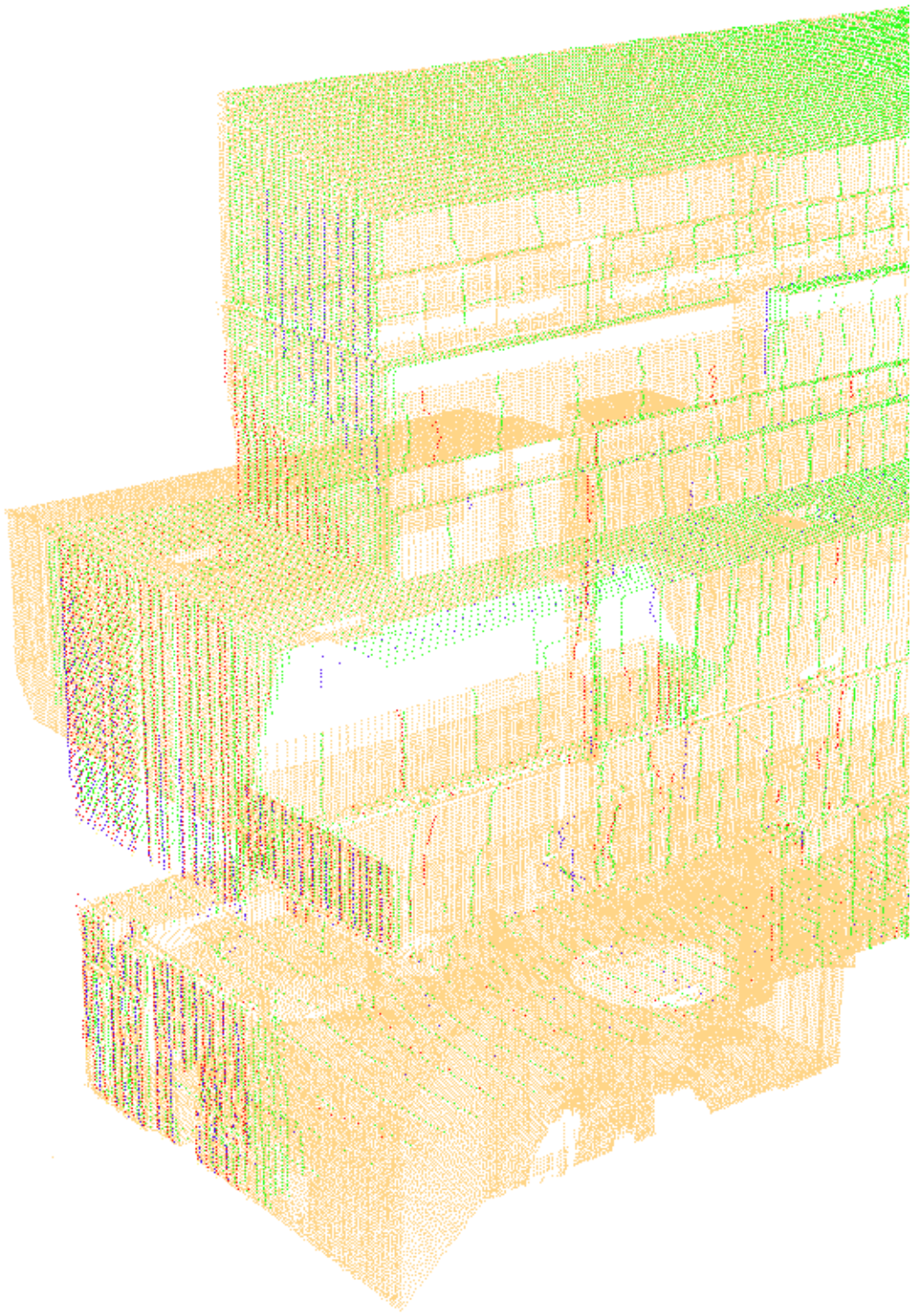


Figure 8.5 – Close up of the architecture building showing the alignment of scans 1 (orange), 2 (green), 3 (blue) and 4 (red).

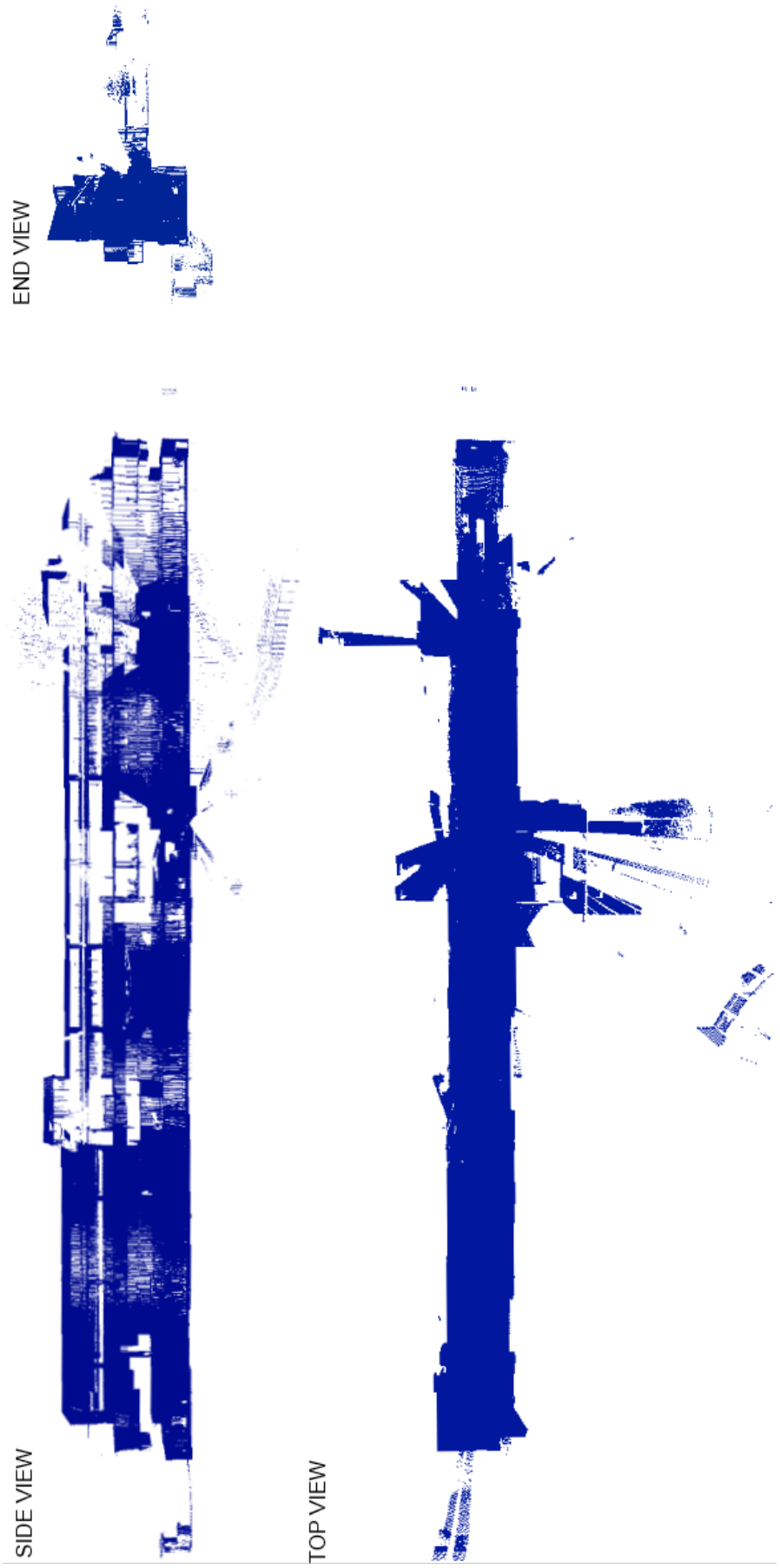


Figure 8.6 – Side, top and end view of the completed map of the architecture building without rendering.

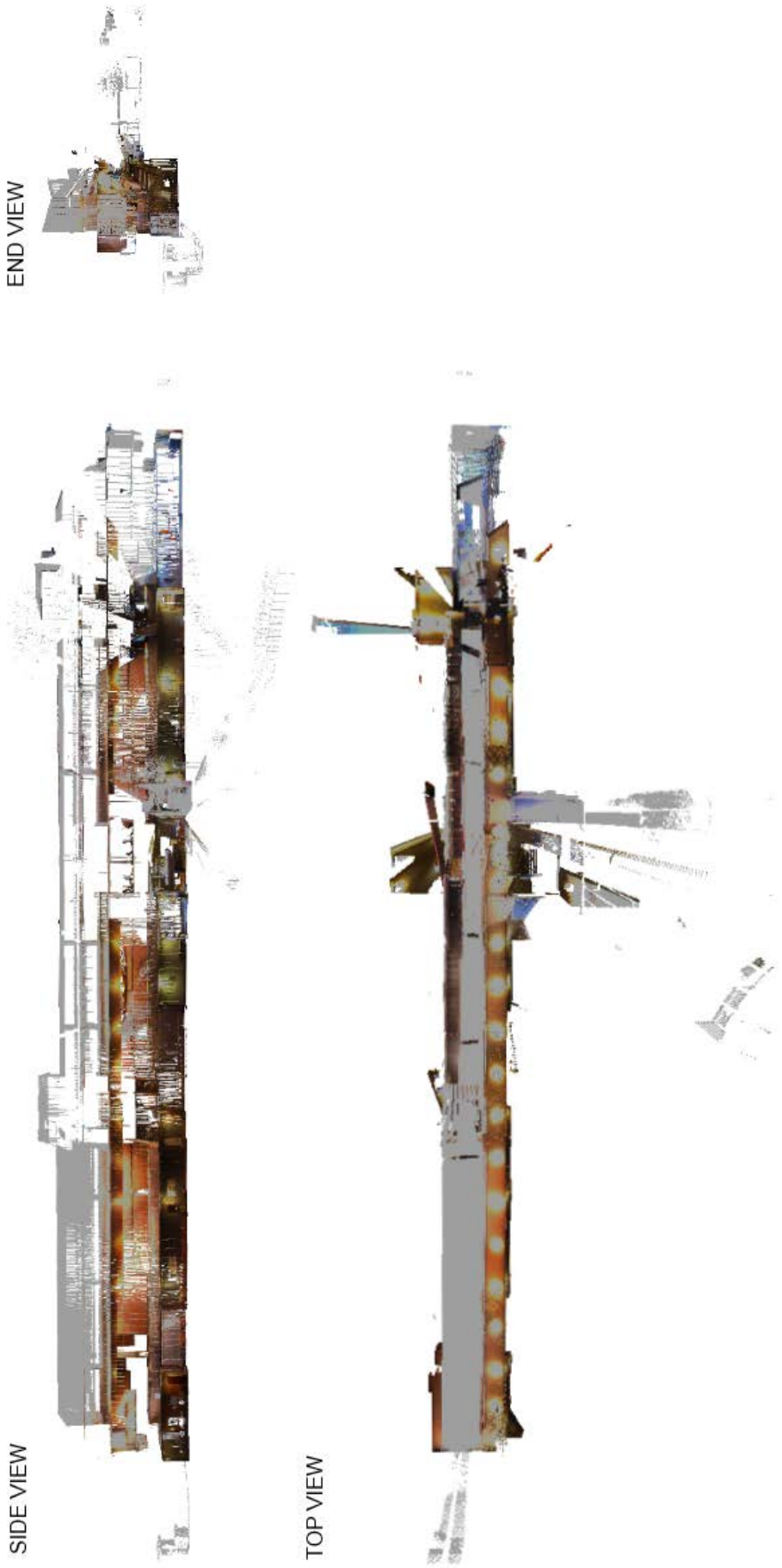


Figure 8.7 – Side, top and end view of the completed map of the architecture building with rendering.

8.2 Large Scale Underground Tunnel Results

To assess the robustness of the hybrid large scale map building system and its applicability to underground environments, another dataset was collected in an active spiral decline tunnel at the Kalgoorlie Consolidated Gold Mines (KCGM) Mt Charlotte Mine in Kalgoorlie, Western Australia. The dataset was collected by a mine surveyor acting on instructions and is therefore a realistic representation of typical data that the system would have to process in an industrial deployment. It also allows evaluation of the robustness of the technique as many degrading conditions were experienced. The majority of these conditions are unique to active underground mines and will be discussed later in the section. A Leica C10 was used to collect four laser scans at roughly 25-30m intervals, often barely line-of-sight, covering a large portion of a single loop of the spiral decline (see Figure 8.8). A stream of images was again recorded by a Point Grey Ladybug 2 omnidirectional camera between scan locations.

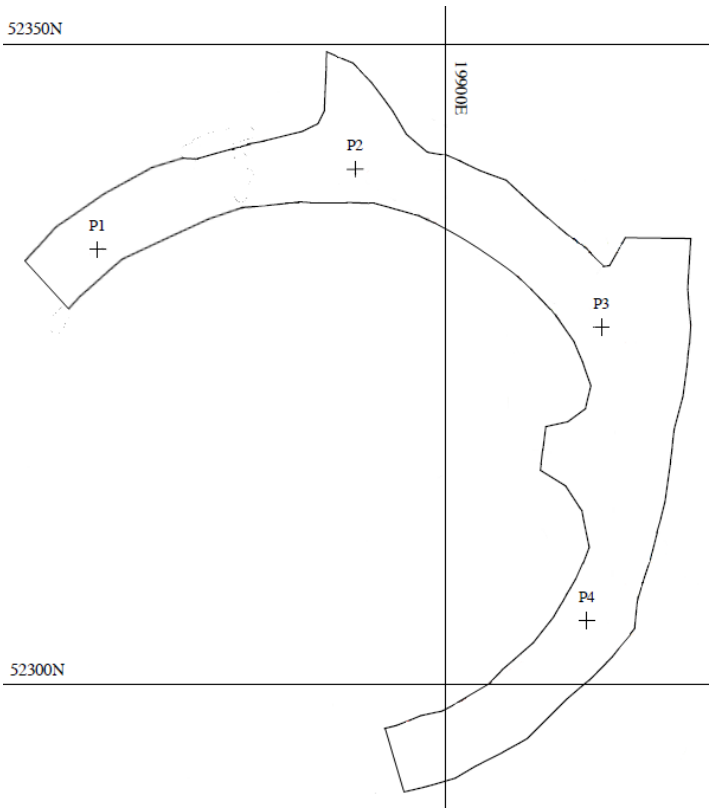


Figure 8.8 – The ground truth for the four laser scans in the spiral decline. Positions were recorded using survey equipment.

The laser and camera were mounted on a mine site certified vehicle for the data collection process (Figure 8.9). The laser was removed during transit as the temporary fixture was deemed inadequate for handling the stresses caused by movement over uneven terrain. This will not be an issue for future deployments, as a high quality fixture will be designed to handle these additional stresses. Once again, the goal of this test is to successfully combine the laser scans into a single large scale 3D map of the environment.

Figure 8.9 – The omnidirectional camera, 3D laser scanner and lighting mounted on a mine site certified vehicle.

There are several factors that make datasets collected in underground mining environments far more difficult for autonomous processing than those collected in above-ground or structured environments. The first of these factors is the poor quality images returned by the Ladybug omnidirectional camera. The underground images contain large areas of occlusion due to the host vehicle, lens flare from vehicle lights, sensor saturation from the additional mounted lights and areas of complete darkness. These factors are shown in Figure 8.10. Figure 8.11 compares the useful image area from an above ground dataset and from an underground dataset.

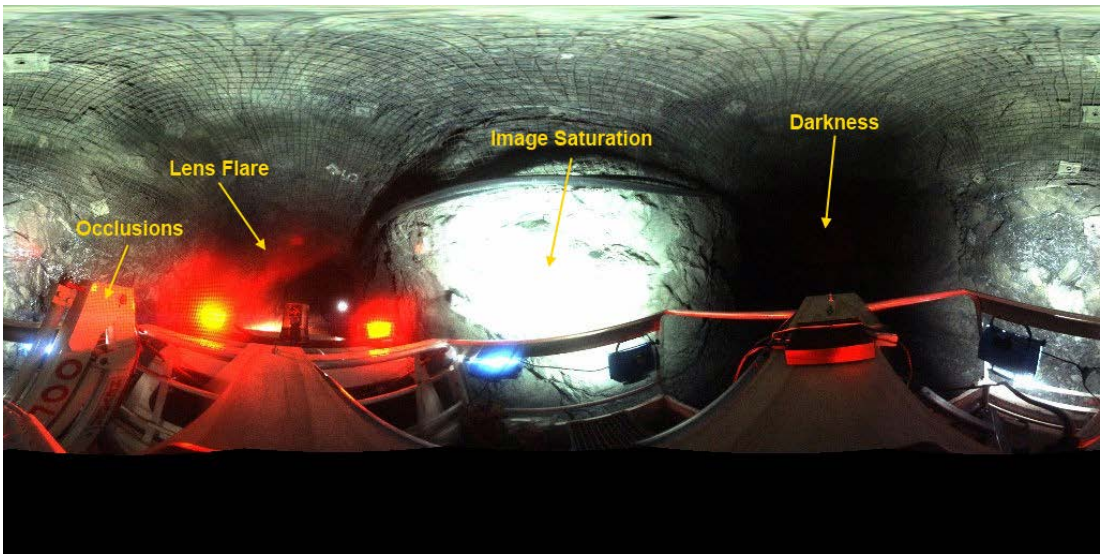


Figure 8.10 – Image factors that degrade autonomous processing performance.

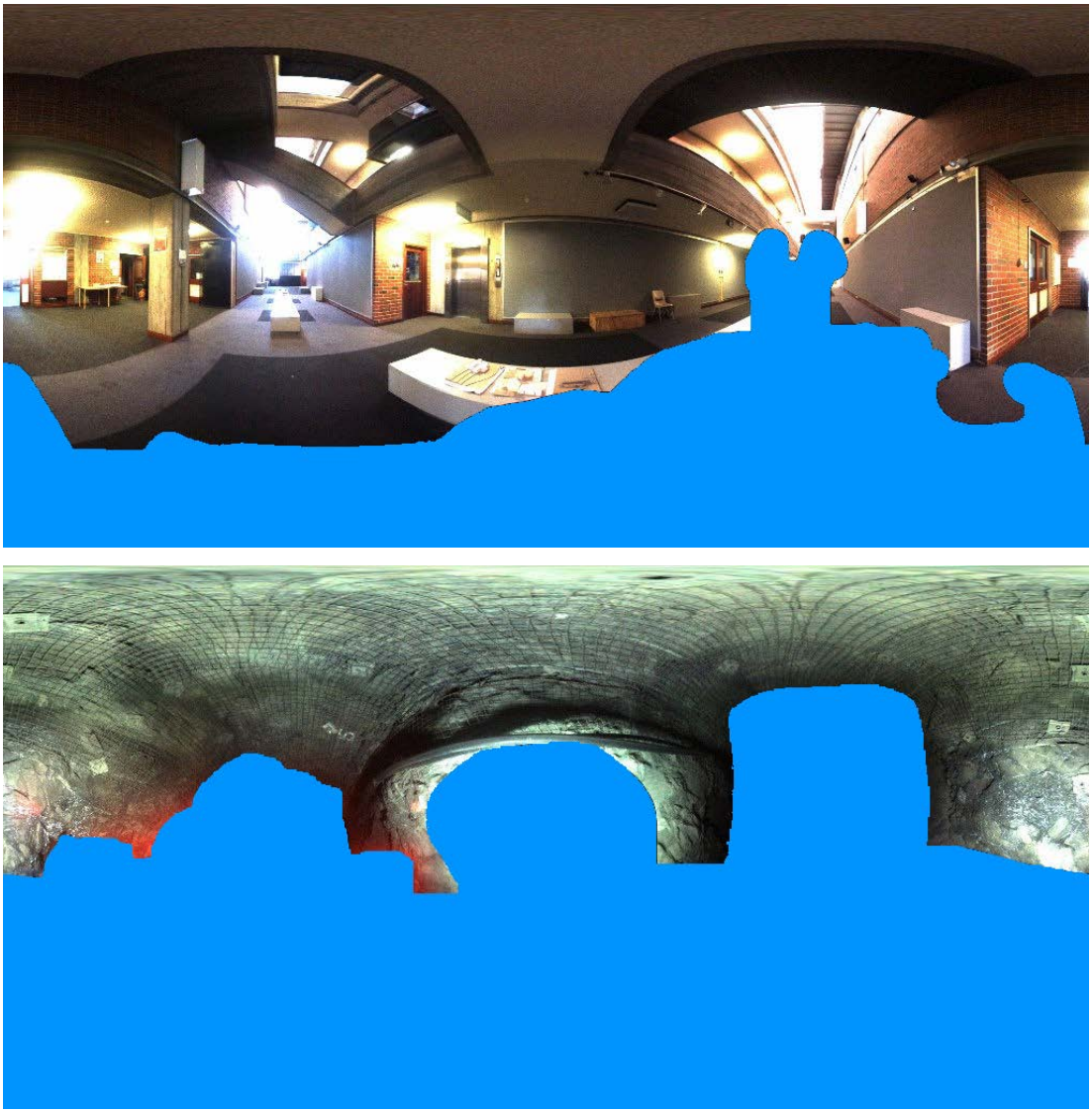
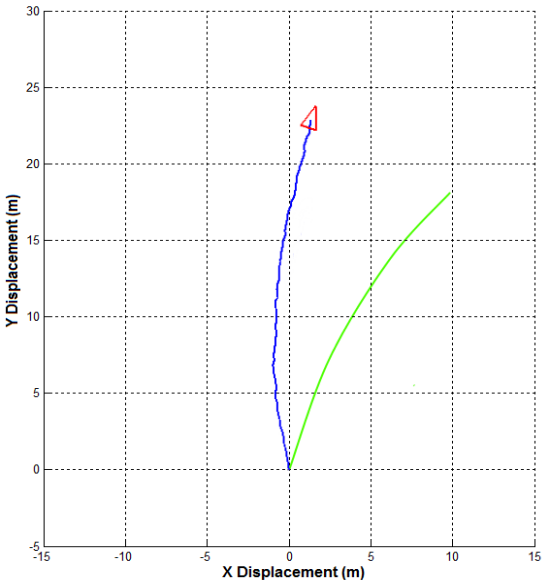
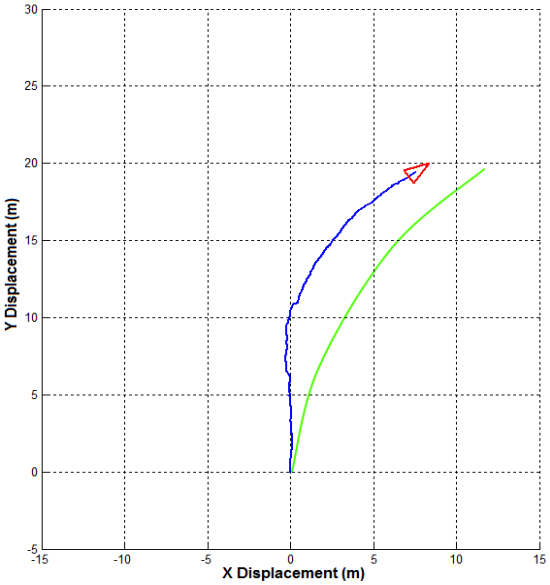


Figure 8.11 – Comparison of useful image region size in the indoor dataset (top) and the underground dataset (bottom). The images represent the entire visual sphere but have the areas that are not useful for vision-based localization masked.

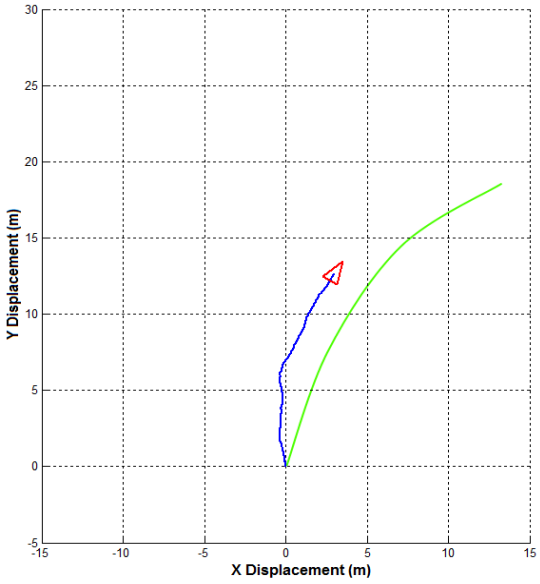
The dataset collected in the underground environment by the omnidirectional camera also experienced full six degree of freedom movement within the spiral decline, as well as rough terrain and uneven road surfaces. By comparison, the indoor test only experienced three degrees of freedom and relatively smooth terrain. The localization algorithm can handle six degrees of freedom; however, the increased movement complexity does reduce the accuracy of the constant velocity motion model.



(a) Localization path (blue) and estimated trajectory (green) for P1 → P2.



(b) Localization path (blue) and estimated trajectory (green) for P2 → P3.



(c) Localization path (blue) and estimated trajectory (green) for P3 → P4.

Figure 8.12 – Plots of the localization results from vision-based SLAM compared to the estimated trajectory interpolated from surveyed scan locations and the curvature of the spiral decline. Distances in meters.

Despite the reduction in quality of the images supplied by the omnidirectional camera, sufficient results were still obtained by the vision-based localization algorithm. The results shown in Figure 8.12 have been scaled using the laser based technique described in Section 6.2. The ground truth for the origin and destination positions was surveyed and is therefore highly accurate. The estimated trajectory between the origin and destination is based on the curvature of the spiral decline and is not of the same accuracy.

The scale for the localization results is based on the laser depth information of features that have been matched at least five times and are therefore considered well-established. The scaling is performed at step 25 of the sequence and again at the final step. Table 8.3 contains the scaling information used for each of the three sequences.

	Sequence 1 (P1→P2)	**Sequence 2 (P2→P3)**	**Sequence 3 (P3→P4)**
Established features at step 25	3	9	8
Average scale at step 25	1.1756	0.8324	1.2335
Established features at final step	7	6	0
Average scale at final step	1.4846	2.2118	0
Combined Scale	1.3919	1.3842	1.2335

Table 8.3 – Scale calculation for underground spiral decline localization results.

The results in Figure 8.12 show that the vision-based localization paths suffer from inaccurate initial yaw estimation, resulting in the final estimation of the destination scan location having a significant offset from the estimated trajectory. This offset occurs despite the reasonable quality tracking during the remainder of the sequence. The likely cause of the poor yaw tracking is the high level of visual occlusion in the panoramic images supplied by the Ladybug camera. Quality yaw features generally occur in a central horizontal band across the entire width of the panoramic image, however, this region of the image is filled with the visual occlusions shown in Figure 8.11 and so is masked during vision-based localization. Yaw tracking would likely be improved in future deployments by the omnidirectional camera being mounted in a higher position, reducing the visual occlusion caused by the host vehicle.

It can also be seen from Figure 8.12 and Table 8.3 that the vision-based localization result in sequence 3 is of lower quality than the results from sequence 1 and 2. Difficult lighting conditions hinder the performance of all underground vision-based localization, so Shadow Feature Removal (SFR) was applied to the dataset as addressed in Section 7.3.1. Unfortunately, since SFR and CBM (Colour Based Matching) are both elimination techniques, they can only be applied mildly to the underground dataset due to the already low number of features available. Figure 8.13 demonstrates the slight improvement produced by the application of SFR. The localization path more accurately reflects the shape of the estimated trajectory. However, it still suffers from the same initial yaw inaccuracy as the other sequences and poor scaling.

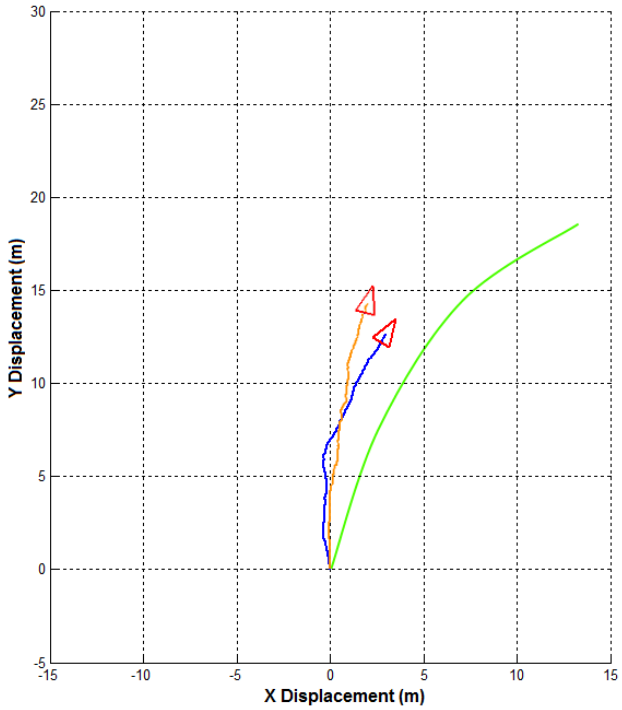


Figure 8.13 – The initial vision-based localization path (blue) compared to the path resulting from the application of SFR (orange). The estimated trajectory can be seen in green.

Laser data collected underground also has characteristics that increase the difficulty of autonomous registration. A factor unique to the spiral decline is the significantly reduced overlap due to the cylindrical nature of the environment. All of the scans from the indoor dataset contained the basic structure of the entire environment due to the open layout of the Architecture building (see Figure 8.1). This resulted in large areas of overlap, improving the robustness of scan registration. The underground

dataset has only small areas of overlap as shown in Figure 8.14. This means that the registration step is actually only performed between the current scan and previous scan, rather than the current scan and the entire existing map. The reduced overlap noticeably degrades the robustness of the registration step of the SLAM system.

Figure 8.14 – The overlap regions of the underground laser dataset. The lightest red represents regions with no overlap, medium red is regions with overlap only between the most recent two scans and dark red shows overlap between the current scan and earlier scans (the existing map).

Occlusions are a significant problem in less structured underground environments such as the spiral decline. The jagged surface of the tunnels reduces the fine scale overlap between laser scans. The surfaces are covered in peaks and valleys, which would normally improve registration due to their irregular, unique shapes. However, the large offset of the scan origins in the spiral decline dataset causes the phenomenon seen in Figure 8.15. Opposing sides of the peaks and valleys are scanned from each origin, resulting in very little overlap in the point clouds. The goal of the ICP algorithm used in our multisensor SLAM system is to minimize the nearest neighbour offset of the two scans; this can easily result in an incorrect

alignment which would actually produce a lower fitness score than the correct alignment. An example of this phenomenon from the underground decline dataset is shown in Figure 8.16.

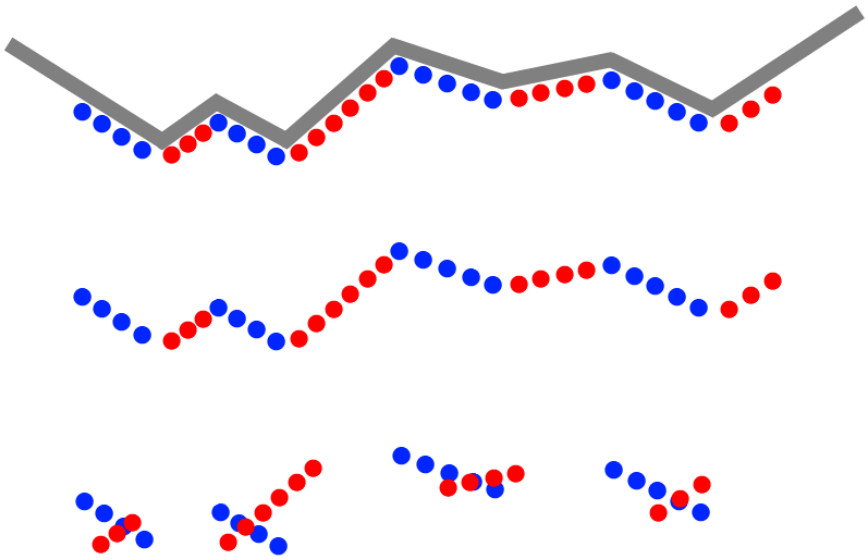


Figure 8.15 – A jagged surface is scanned from two origins with a large offset, resulting in the red and blue point clouds (top). The correct alignment would not produce the lowest average nearest neighbour offset (middle). An incorrect alignment is actually optimum (bottom).

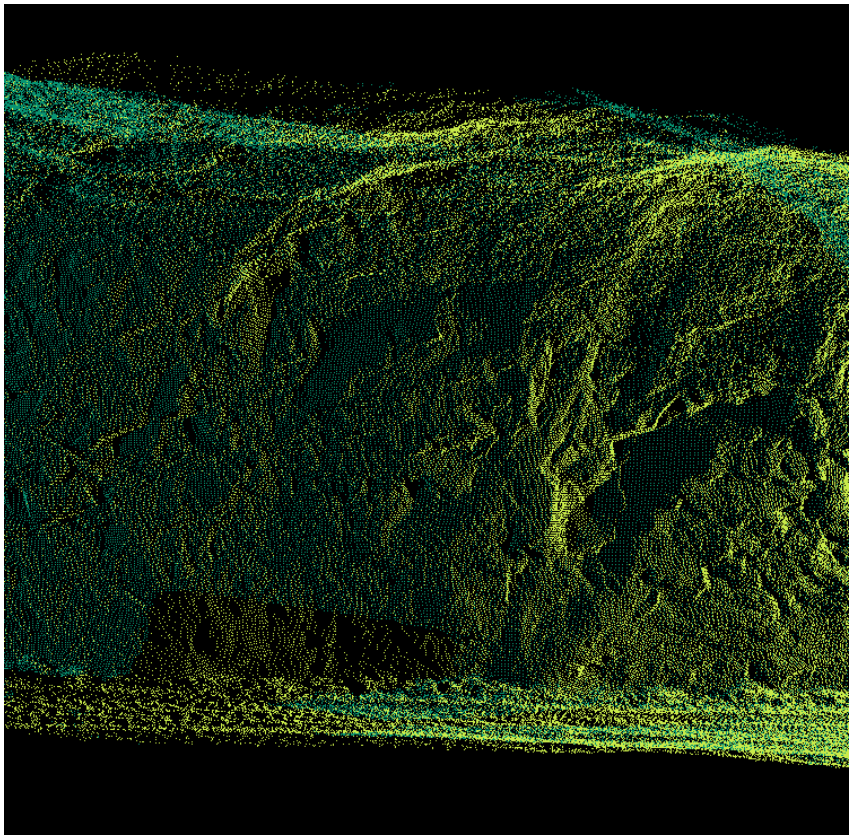


Figure 8.16 – An example of poor fine scale overlap where the two point clouds contain opposite faces of the same surface.

To improve the robustness of the registration step, the technique described in Section 5.5.3 was implemented. The overlapping region between the previous scan origin and the current pose estimate is segmented into six sections and registration is performed on each section. Six separate attempts at registration improves the robustness in a similar way to random seeding, reducing the likelihood of finding a local minimum rather than the desired global minimum. The registration result with the lowest fitness score has its transformation applied to the entire new data cloud, producing an updated map. This approach is well suited to environments where there may be minimal point cloud overlap as it reduces the influence of outlier points through the iterative registration and comparison of point cloud segments.

This approach proved robust for the registration of scans 1, 2 and 3 from the underground spiral decline dataset. Unfortunately, the low quality localization result for the initial pose estimate of scan 4 meant that the technique had to be combined with extensive random seeding for successful registration. To robustly produce the correct registration, approximately 1000 iterations would be required to guarantee a pose close enough to the ground truth for successful alignment to occur. This number of iterations requires a prohibitive amount of processing time and therefore demonstrates the importance of a sufficiently accurate localization result for the system to be effective. A summary of the transformations produced by the registration algorithm is available in Table 8.4.

		Localization Result		**ICP Correction**		**Final Position**	
Dataset		**Trans.**	**Rotation**	**Trans.**	**Rotation**	**Trans.**	**Rotation**
	x,ϕ	1.328m	-1.037°	-2.663m	1.593°	9.743m	0.014°
1	y,θ	22.80m	-1.369°	-0.546m	0.653°	18.63m	0.029°
	z,ψ	-0.033m	-17.04°	-3.437m	-29.57°	-3.079m	-46.59°
	x,ϕ	7.538m	0.527°	3.504m	1.501°	11.93m	0.050°
2	y,θ	19.42m	-2.108°	0.452m	1.662°	19.51m	0.042°
	z,ψ	-0.068m	-53.24°	-3.888m	-2.647°	-3.702m	55.85°
	x,ϕ	2.986m	-0.028°	4.456m	0.951°	13.50m	0.020°
3	y,θ	12.62m	-1.071°	9.259m	0.470°	18.56m	0.018°
	z,ψ	-0.020m	-32.70°	-3.368m	-30.88°	-3.307m	-63.58°

Table 8.4 – Transformations resulting from initial pose estimate and ICP registration.

The transformation produced by the registration step is also applied to the vision-based localization result to reduce the impact of long term drift. The resulting localization paths can be seen in Figure 8.17 and are compared to the expected ground truth of the host vehicle. Again, only the scan locations were accurately surveyed so the ground truth between these points is an extrapolation based on the curvature of the spiral decline. Table 8.5 contains the scan locations as determined by our multisensor SLAM system compared to the surveyed positions. The results show that there is an average offset error of less than 3cm for the entire large scale map.

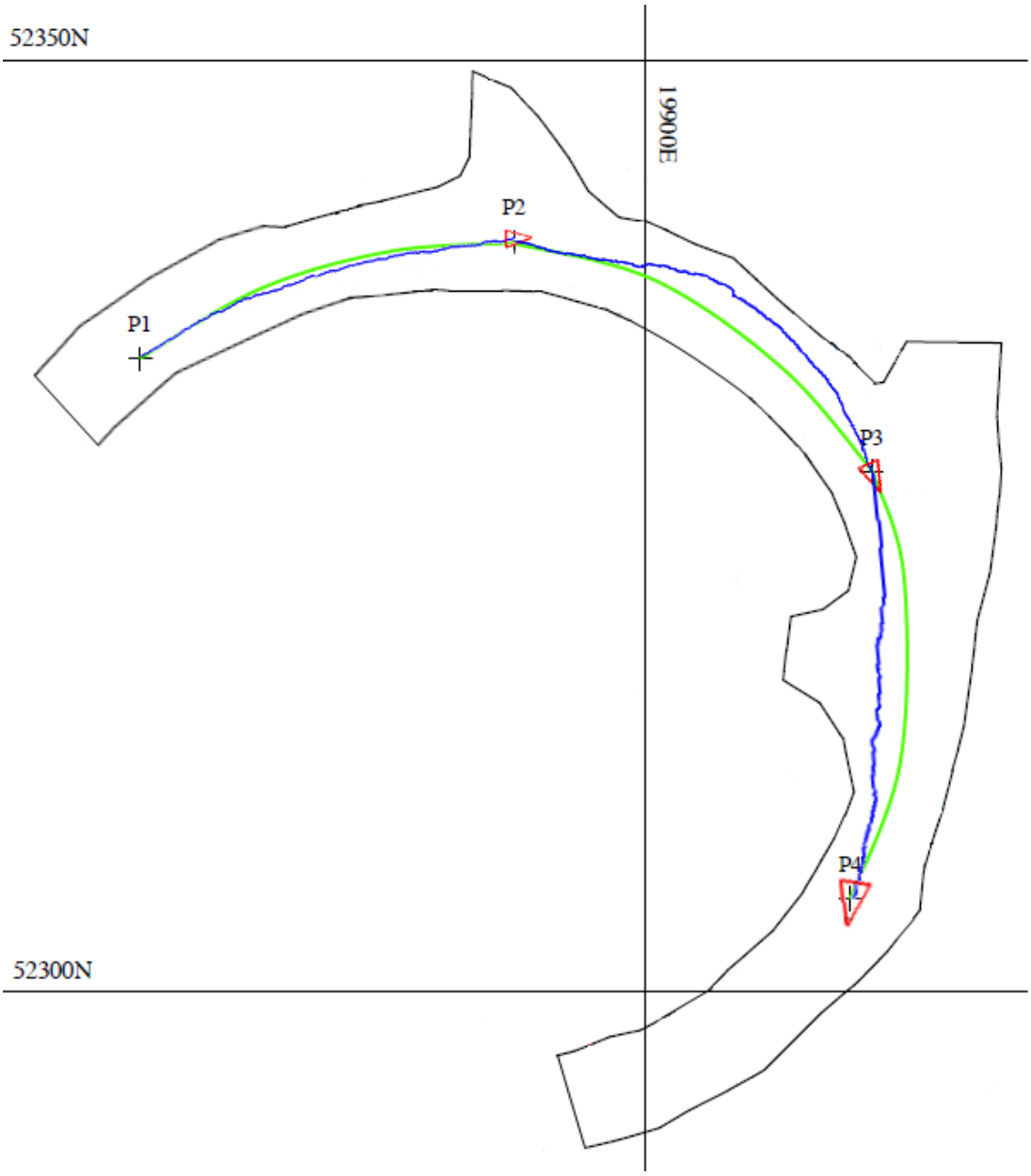


Figure 8.17 – The corrected localization path resulting from scan registration (blue) and the estimated trajectory (green).

		P1	P2	P3	P4
Surveyed Result	**x**	0.0m	20.082m	39.289m	38.157m
	y	0.0m	6.276m	-6.042m	-28.978m
	z	0.0m	-3.104m	-6.807m	-10.114m
Multisensor SLAM Result	**x**	0.0m	20.066m	39.311m	38.179m
	y	0.0m	6.270m	-6.072m	-28.997m
	z	0.0m	-3.079m	-6.781m	-10.088m
Offset Error	**x**	0.0m	0.016m	0.022m	0.022m
	y	0.0m	0.006m	0.030m	0.019m
	z	0.0m	0.025m	0.026m	0.026m

Table 8.5 – Comparison of scan locations resulting from the survey and the multisensor SLAM system.

The four laser scans requiring registration can be seen individually in Figure 8.18. The final large scale fully registered map is shown in Figure 8.19 and a rendered version of this map is in Figure 8.20. Rendering an underground map using our current technique is not particularly effective due to only using images taken at the discrete laser scan locations. This results in large sections of the map lacking sufficient illumination. The rendering results could be improved by using images from the duration of the vehicle motion. However, an improved rendering technique is outside the scope of this work and will be noted in Section 9.2 as future work.

Figure 8.18 – The four laser scans comprising the spiral decline dataset. Alignment has been customized to give an idea of final fit.

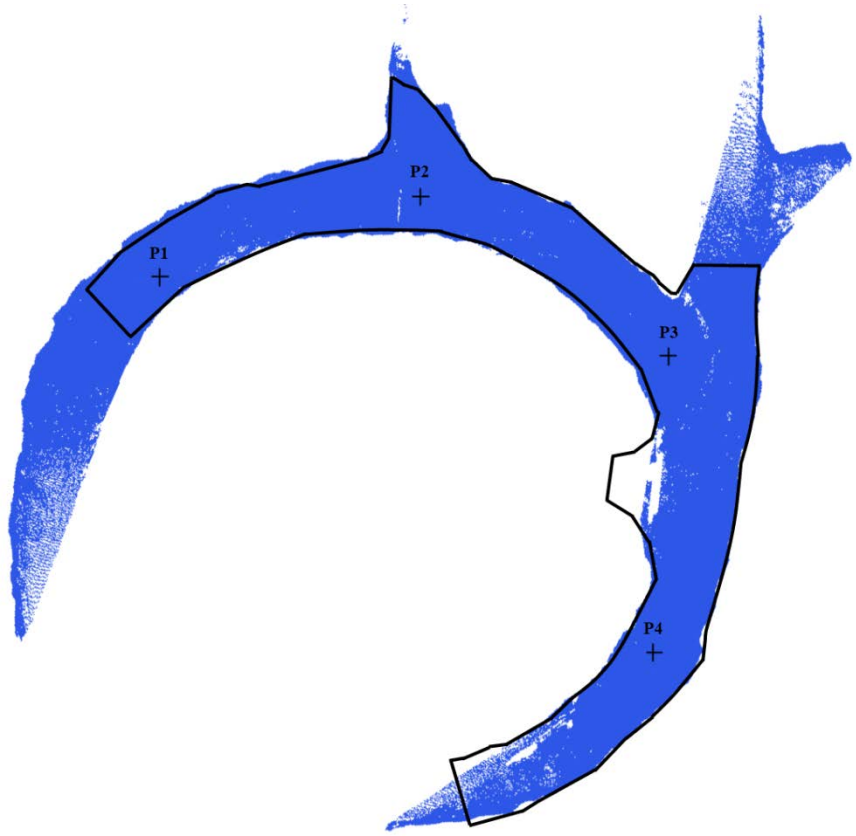


Figure 8.19 – Final registration of all four laser scans overlayed with the surveyed map of the area.

Figure 8.20 – Final registration of all four scans with rendering.

Chapter 9

Conclusions

This thesis has described the development and implementation of an autonomous large scale mapping system designed for the surveying of underground mines and other extensive spaces. Unlike many existing autonomous mapping techniques which rely on specialized approaches to the SLAM problem in order to provide a viable solution, our proposed system is self-contained, portable and robust, making it applicable to a wide range of environments, particularly underground mines.

Our autonomous mapping solution is based on a hybrid fusion of omnidirectional bearing-only vision-based localization and 3D laser point cloud registration. The fusion of these two techniques combines the real-time, six degree of freedom localization attributes of vision-based SLAM with the high precision and dense data of 3D laser scanners to produce a viable solution to autonomous mapping in the difficult active underground mining environment.

To further improve the robustness of our proposed multisensor SLAM system, colour information is utilized in the vision-based localization to improve robustness to the effects of dynamic illumination. Visual features are filtered by identifying and removing features caused by dynamic shadows. Colour information is further used to

supplement the feature matching process through the addition of colour matching to the existing monochrome descriptor matching.
Finally, the multisensor system is implemented and evaluated on both indoor and underground datasets. Large scale survey quality maps were successfully constructed in both environments demonstrating the effectiveness of our proposed system.

9.1 Contributions

The novel contributions of our work are summarized below on a per chapter basis. These contributions are unique to this work and have not been documented by any other source at the time of writing.

Chapter 5 Large Scale Mapping from Point Clouds
The first contribution of this chapter is a unique implementation of the Iterative Closest Point (ICP) algorithm which is customized to suit the registration of dense point clouds with significant offset using an initial pose estimate from vision-based localization. The algorithm includes voxel based reduction to improve processing times and random seeding to improve robustness to poor localization results.

A technique for the registration of point clouds with significant offset is also presented. This technique is designed to successfully register point clouds which would fail when using the standard registration approach. The newly acquired data is divided into dense and sparse sections using the ‘density transition plane’. The sparse section is then used for registration with the existing map, reducing erroneous registration and improving processing time. The sparse sections can be further segmented and individually registered to further improve robustness to difficult point cloud shapes with large offsets.

Finally, a method for the visualization of ICP registration capabilities is presented. The two dimensional ‘heat maps’ plot ICP results over a range of translations and rotations. The results are classified as global minimum obtained (successful registration), local minimum obtained or registration failure.

Chapter 6 Hybrid Integration of Vision-Based SLAM and Point Clouds

In this chapter a technique is presented for the real world scaling of a dimensionless bearing-only localization result. The laser data obtained at the origin and destination of a localization path is used to provide depth measurements for well-established features. The scale is calculated as the average ratio between the arbitrary depth value assigned to each feature by the localization algorithm and the actual depth information provided by the laser. The number of features with depth estimates at the origin and destination is used as a weighting factor to determine the overall scale applied to the localization result.

An approach to localization path refinement is also presented to reduce drift over long distances. The transformation resulting from successful ICP point cloud registration is applied to the localization result, significantly improving positional accuracy. The localization correction improves future pose estimates and therefore increases the likelihood of successful future point cloud registrations.

Chapter 7 Vision-Based SLAM under Dynamic Illumination

Two novel techniques are presented in this chapter as approaches to improve the robustness of vision-based localization to environments with dynamic illumination. The first technique applies the chromaticity distortion colour model to a visual feature to determine if the range of pixel colours present within the feature identifies it as part of an object or part of a shadow. Features that are considered to be part of a shadow are discarded before the feature matching step to prevent their dynamic nature from affecting the localization result.

The chromaticity distortion model is applied again in a second technique to improve the robustness of vision-based localization during the feature matching stage. The colour ranges present in the pixels of two features are compared if they are matched by the monochrome based feature matching algorithm. A significant difference in colour range between the two matched features indicates a likely mismatch, so the feature match is rejected. The removal of mismatched features further improves the localization result.

The two novel techniques are then evaluated in simulated and real world scenarios with static and dynamic camera motion. A technique for automatically selecting thresholds for the Shadow Feature Removal and Colour Based Matching methods is also presented, enabling fully autonomous implementation.

Chapter 8 Multisensor SLAM Results

The final contributions of this thesis are the in-depth performance evaluation of the multisensor large scale mapping system which incorporates all of the developed techniques. The system is evaluated in an above ground built environment and an active underground mining environment.

9.2 Future Work

The design and implementation of our multisensor mapping system has opened several areas of possible research which were considered outside the scope of this thesis. These directions for possible future research work could further improve the robustness and modularity of our mapping system.

Monocular SLAM Acceleration Noise Scaling from Laser Data

Section 6.2 discussed the issue of monocular SLAM with well-established laser based initial features reverting to an arbitrary scale due to insufficient feature correlation. The arbitrary scale is based on the acceleration noise in the motion model. To reduce the impact of poor feature correlation, the acceleration noise could be scaled so that the resulting localization scale generated by the acceleration noise is close in size to the localization scale generated by using features with known depth estimates from laser data. Extensive simulation would likely be required to effectively determine the relationship between the scaled features and the required characteristics of the motion model.

Sliding ICP Optimization

A significant portion of large scale environments which require survey quality mapping consist of ‘tunnel’ or ‘corridor’ like structures. The resulting point clouds collected during the mapping process can be difficult to align due to the presence of a

large region parallel to the length of the tunnel where nearest neighbour distances of the overlapping point clouds are consistently small (see Figure 9.1). ICP will often converge at the first alignment that falls within this region rather than exploring possible higher quality alignments within the extents of the region. To improve registration results, a 'sliding' optimization technique could be developed which refines the initial ICP solution by searching for globally optimum alignments within this region. The implementation could be limited to a one or two degrees of freedom optimization problem, which would not require significant processing time. Evaluation could be performed on the datasets collected during testing of the multisensor SLAM system.

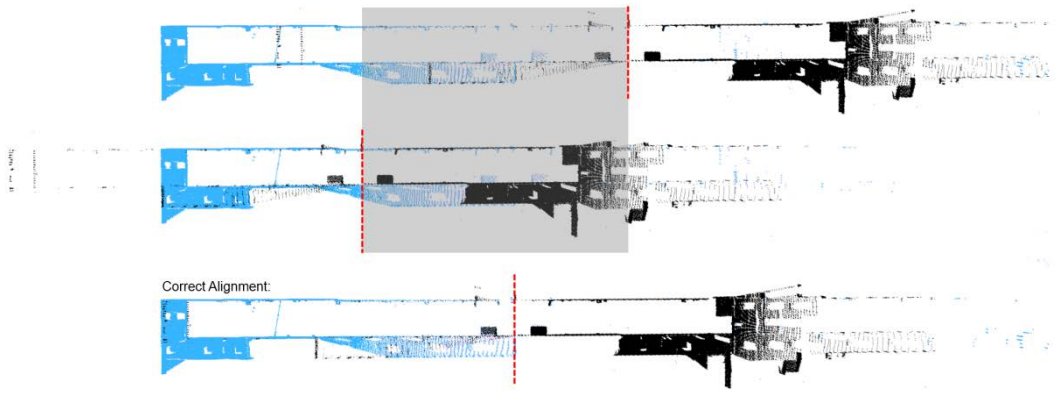


Figure 9.1 – The correct ICP alignment (bottom) falls inside a region of consistently small fitness scores (grey box).

Improved Rendering using Multiple Locations for Omnidirectional Images

The rendered map seen in Figure 8.20 demonstrates the shortfalls of only using images at the scan origins for rendering in underground environments. The short illumination range causes large sections of the map to be rendered using complete darkness. To overcome this problem, a series of images from the duration of camera motion could be stitched together to form a single cohesive image to be used for rendering.

Motion Tracking for Dynamic Illumination Sources

Currently, features caused by dynamic illumination in our vision-based localization algorithm are discarded as their motion is inconsistent with the motion of the camera. Robustness to dynamic illumination could be further improved if these features were instead used to estimate the location of dynamic light sources, which can be

explicitly modelled and included in the EKF state. Assuming the presence of only a small number of dynamic light sources, each source could be tracked independently and their effect on the estimation of camera motion could be more effectively determined. Their motion could even be used to predict the appearance of illumination artefacts in future images.

9.3 Discussion

This thesis has described in detail the design and implementation of a multisensor SLAM system for the dense large scale mapping of underground mines. With the completion of a working system it is appropriate to briefly discuss the consequences of this research and the future directions of the field. Although the system developed in this work successfully demonstrated the ability to autonomously combine point clouds to produce large scale underground mapping results, robustness is an issue that is at the heart of all mining operations and, as such, is the area of this research that would require the most attention in future development. Therefore, thorough testing on a more comprehensive underground dataset would be the first recommendation in the process to optimize the multisensor SLAM system.

Despite recent improvements in the accuracy and density of maps produced by monocular or stereo vision, 3D lasers are still the only option for the survey quality 3D mapping of underground mines. And since the registration of large point clouds with small numbers of overlapping points currently still requires some form of pose estimate, the autonomous production of underground maps will require a multisensor approach for some time to come.

The long term solution to survey quality underground mapping is most likely the development of real-time 3D scanning sensors which would maintain the accuracy of 3D laser scanning, but also allow highly accurate real-time localization to be performed. Until sensor technology reaches that point, the multisensor approach must continue to be refined into a robust, yet somewhat slow (due to scan times) alternative.

Improving localization is the key to improving robustness. Even difficult point clouds will autonomously align quickly and accurately if provided with a high accuracy pose estimate. Localization results can benefit from improvements in feature quality, feature number and correspondence accuracy. There is no shortage of textured surfaces in underground mining environments; however, poor lighting conditions dynamically affect the appearance of features to the camera. Although the work in this thesis towards improving robustness of vision-based localization under dynamic illumination has enhanced the identification of quality features in poor lighting conditions, there is still room for further refinement.

The lack of long distance illumination also prevents the long term tracking of features, as they quickly disappear into the darkness as the host vehicle moves through the tunnel. The lack of long term features prevents the application of keyframe-based localization algorithms and increases the likelihood of significant long term drift. The useful image area in underground mines is also significantly reduced due to these lighting distances as well as occlusions. It is therefore vital to maximize the number of features extracted from the scene in order to maintain quality localization. Future improvements in efficiency and processing power will allow future filtering algorithms to track more features in each frame and this will inevitably improve localization.

Finally, correspondence estimation can be improved to increase the number of correctly tracked features. The work in this thesis has shown the advantage that including colour information in the correspondence process can have on localization. Further improvements may be possible by implementing the light source tracking concept outlined as future work in section 9.2.

References

[1] Department of Mines and Petroleum, “Fatal Accident Reports - By Date,” 2011.

[2] R. Kahler, “Fatalities in the West Australian Mining Industry 1970 - 2006,” The InterSafe Group, 2007.

[3] A. Howard, D. Wolf and G. Sukhatme, “Towards 3D mapping in large urban environments,” in *International Conference on Intelligent Robots and Systems*, 2004.

[4] J. Le Cras, J. Paxman and B. Saracik, “An Inspection and Surveying System for Vertical Shafts,” in *Australasian Conference on Robotics and Automation*, 2009.

[5] H. Surmann, A. Nuchter and J. Hertzberg, “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments,” *Robotics and Autonomous Systems*, pp. 181-198, 2003.

[6] A. Nuchter, K. Lingemann, J. Hertzberg and H. Surmann, “6D SLAM - 3D mapping outdoor environments,” *Journal of Field Robotics*, pp. 699-722, 2007.

[7] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg and S. Thrun, “6D SLAM with an application in autonomous mine mapping,” in *International Conference on Robotics and Automation*, 2004.

[8] C. Fruh and A. Zakhor, “An automated methodfor large-scale ground-based city model acquisition,” *International Journal of Computer Vision*, pp. 5-24, 2004.

[9] S. Thrun, W. Burgard and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping,” in *International Conference on Robotics and Automation*, 2000.

[10] D. Hahnel, W. Burgard, D. Fox and S. Thrun, “An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,” in *International Conferecne on Intellignet Robots and Systems*, 2003.

[11] P. Besl and N. McKay, “A method for registration of 3-D shapes,” *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, pp. 239-256, 1992.

[12] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous robot vehicles,* pp. 167-193, 1990.

[13] M. Csorba, "Simultaneous localisation and map building," *OCLC's Experimental Thesis Catalog,* 1998.

[14] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, MIT Press, 2005.

[15] A. Kundu, K. M. Krishna and C. V. Jawahar, "Realtime multibody visual SLAM with a smoothly moving monocular camera," in *IEEE International Conference on Computer Vision,* 2011.

[16] F. Dellaert, D. Fox, W. Burgard and S. Thrun, "Monte Carlo localization for mobile robots," in *International Conference on Robotics and Automation,* 1999.

[17] S. Thrun, D. Fox, W. Burgard and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robotics," *Artificial Intelligence,* pp. 99-141, 2001.

[18] R. Sim, P. Elinas and M. Griffin, "Vision-based SLAM using the rao-blackwellised particle filter," in *Workshop on Reasoning with Uncertainty in Robotics,* 2005.

[19] S. Thrun and M. Montermerlo, "The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures," *International Journal of Robotics Research,* pp. 403-429, 2006.

[20] H. Strasdat, J. Montiel and A. Davison, "Visual SLAM: Why Filter?," *Image and Vision Computing,* vol. 30, pp. 65-77, 2012.

[21] J. Montiel, J. Civera and A. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Robotics: Science and Systems,* 2006.

[22] J. Civera, A. J. Davison and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics,* pp. 932-945, 2008.

[23] J. Civera, O. Grasa, A. Davison and J. Montiel, "1-point RANSAC for EKF-based Structure from Motion," in *International Conference on Intelligent Robots and Systems,* 2009.

[24] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *International Symposium on Mixed and Aurmented Reality,* 2007.

[25] K. Konolige and M. Agrawal, “FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066-1077, 2008.

[26] E. Eade and T. Drummond, “Scalable Monocular SLAM,” in *Computer Vision and Pattern Recognition*, 2006.

[27] E. Eade and T. Drummond, “Monocular SLAM as a Graph of Coalesced Observations,” in *IEEE International Conference on Computer Vision*, 2007.

[28] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, pp. 91-110, 2003.

[29] H. Bay, T. Tuytelaars and L. Van Gool, “Surf: Speeded up robust features,” in *ECCV*, 2006.

[30] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European Conference on Computer Vision*, 2006.

[31] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1615-1630, 2005.

[32] M. Brown and D. Lowe, “Invariant Features froom Interest Point Groups,” in *British Machine Vision Conference*, 2002.

[33] D. Chekhlov, M. Pupilli, W. W. Mayol and A. Calway, “Robust Real-Time Visual SLAM Using Scale Prediction and Exemplar Based Feature Description,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[34] N. D. Molton, A. J. Davison and I. D. Reid, “Locally Planar Patch Features for Real-Time Structure from Motion,” in *British Machine Vision Conference*, Kingston, 2004.

[35] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, pp. 381-395, 1981.

[36] R. Raguram, J. M. Frahm and M. Polleyeys, “A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus,” in *European Conference on Computer Vision*, 2008.

[37] O. chum and J. Matas, "Optimal Randomized RANSAC," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1472-1482, 2008.

[38] D. Capel, "An effective bail-out test for RANSAC consensus scoring," in *British Machine Vision Conference*, 2005.

[39] D. Nister, "Preemptive RANSAC for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, no. 5, pp. 321-329, 2005.

[40] J. Neira and J. D. Tardos, "Data Association in Stochastic Mapping using the Joint Compatibility Test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890-897, 2001.

[41] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, pp. 229-241, 2001.

[42] T. Bailey, *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*, 2002.

[43] J. Tardos, J. Neira, P. Newman and J. Leonard, "Robust Mapping and Localization in Indoor Environments Using Sonar Data," *International Journal of Robotics*, pp. 311-330, 2002.

[44] S. M. Oh, S. Tariq, B. Walker and F. Dellaert, "Map-based priors for localization," in *International Conference on Intelligent Robots and Systems*, 2004.

[45] C. Brenneke, O. Wulf and B. Wagner, "Using 3D laser range data for SLAM in outdoor environments," in *International Conference on Intelligent Robots and Systems*, 2003.

[46] P. Biber, H. Andreasson, T. Duckett and A. Schilling, "3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera," in *International Conference on Intellignet Robots and Systems*, 2004.

[47] A. Davison, N. Molton, I. Reid and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1052-1067, 2007.

[48] N. Kwok and G. Dissanayake, "Bearing-only slam in indoor environments using a modified particle filter," in *Australasian Conference on Robotics and*

Automation, 2003.

[49] C. Roussillon, A. Gonzalez, J. Sola, J.-M. Codol, N. Mansard, S. Lacroix and M. Devy, "RT-SLAM: A Generic and Real-Time Visual SLAM Implementation," *Lecture Notes in Computer Science,* vol. 6962, pp. 31-40, 2011.

[50] G. Nutzi, S. Weiss, D. Scaramuzza and R. Siegwart, "Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1-4, pp. 287-299, 2011.

[51] J. Civera, A. J. Davidson, O. G. Grasa and J. M. Montiel, "1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry," *Journal of Field Robotics*, pp. 609-631, 2010.

[52] X. Zhang, A. Rad and Y.-K. Wong, "Sensor Fusion of Monocular Cameras and Laser Rangefinders for Line-Based Simultaneous Localization and Mapping (SLAM) Tasks in Autonomous Mobile Robots," *Sensors*, vol. 12, no. 1, pp. 429-452, 2012.

[53] H. Strasdat, J. M. M. Montiel and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Science and Systems*, 2010.

[54] B. Williams, G. Klein and I. Reid, "Automatic Relocalization and Loop Closing for Real-Time Monocular SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, no. 9, pp. 1699-1712, 2011.

[55] G. Zhang and I. H. Sun, "Building a partial 3D line-based map using a monocular SLAM," in *International Conference on Robotics and Automation*, 2011.

[56] J. Martinez-Carranza and A. Calway, "Unifying Planar and Point Mapping in Monocular SLAM," in *British Machine Vision Conference*, 2010.

[57] S. Y. Hwang and J. B. Song, "Monocular Vision-Based SLAM in Indoor Environment Using Corner, Lamp, and Door Features From Upward-Looking Camera," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4804-4812, 2011.

[58] J. Civera, D. Galvez-Lopez, L. Riazuelo, J. Tardos and J. M. M. Montiel, "Towards semantic SLAM using a monocular camera," in *International Conference on Intelligent Robots and Systems*, 2011.

[59] K. Konolige, M. Agrawal and J. Sola, "Large-Scale Visual Odometry for Rough Terrain," *Springer Tracts in Advanced Robotics*, pp. 201-212, 2011.

[60] P. Pinies, J. D. Tardos and J. Neira, "Large-Scale 6-DOF SLAM with Stereo-in-Hand," *IEEE Transactions on Robotics*, pp. 946-957, 2008.

[61] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

[62] B. Triggs, P. McLauchlan and R. Hartley, "Bundle adjustment - A modern synthesis," *Vision Algorithms: Theory and Practice*, pp. 298-375, 2000.

[63] J. Xiao, T. Fang, P. Zhao, M. Lhuillier and L. Quan, "Image-based street-side city modeling," *ACM Transactions on Graphics*, 2009.

[64] S. Sinha, D. Steedly, R. Szeliski, M. Agrawala and M. Pollefeys, "Interactive 3D architectural modeling from unordered photo collections," *ACM Transactions on Graphics*, 2008.

[65] P. Henry, M. Krainin, E. Herbst, X. Ren and D. Fox, "RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments," in *International Symposium on Experimental Robotics*, 2010.

[66] M. Whitty, S. Cossell, K. Son Dang, J. Guivant and J. Katupitiya, "Autonomous Navigation using a Real-Time 3D Point Cloud," in *Australasian Conference on Robotics and Automation*, 2011.

[67] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffman, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt and S. Thrun, "Junior: The Stanford entry in the Urban Challenge," *Journal of Field Robotics*, pp. 569-597, 2008.

[68] D. M. Cole and P. M. Newman, "Using laser range data for 3D SLAM in outdoor environments," in *International Conference on Robotics and Automation*, 2006.

[69] U. Artan, J. Marshall and N. Lavigne, "Robotic mapping of underground mine passageways," *Mining Technology*, pp. 18-24, 2011.

[70] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer and W. Whittaker, "A system for volumetric

robotic mapping of abandoned mines," in *International Conference on Robotics and Automation*, 2003.

[71] D. Huber and N. Vandapel, "Automatic Three-dimensional Underground Mine Mapping," *The International Journal of Robotics Research*, pp. 7-17, 2006.

[72] A. Rituerto, L. Puig and J. J. Guerrero, "Visual SLAM with an Omnidirectional Camera," in *Proceedings of the 2010 International Conference on Pattern Recognition*, 2010.

[73] J. Roda, J. Saez and F. Escolano, "Ceiling mosaics through information-based SLAM," in *International Conference on Intelligent Robots and Systems*, 2007.

[74] J.-C. Du and H.-C. Teng, "3D laser scanning and GPS technology for landslide earthwork volume estimation," *Automation in Construction*, pp. 657-663, 2007.

[75] O. Wulf, K. O. Arras, H. I. Christensen and B. A. Wagner, "2D mapping of cluttered indoor environments by means of 3D perception," in *International Conference on Robotics and Automation*, 2004.

[76] M. Magnusson and T. Ducket, "A comparison of 3D registration algorithms for autonomous underground mining vehicles," in *Second European Conference on Mobile Robotics*, 2005.

[77] A. Kaufman, "Volume Graphics," *Computer*, pp. 51-64, 1993.

[78] R. Rusu and S. Cousins, "3D is here: Point Cloud Library," in *International Conference on Robotics and Automation*, 2011.

[79] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense Tracking and Mapping in Real-Time," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[80] Y. Furukawa, B. Curless, S. M. Seitz and R. Szeliski, "Towards internet-scale multi-view stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2101.

[81] J. Civera, A. J. Davison and M. M. Montiel, "Dimensionless Monocular SLAM," *Lecture Notes in Computer Science*, vol. 4478, pp. 412-419, 2007.

[82] J. Civera, *Real-time ekf-based structure from motion*, 2009.

[83] J. Civera, A. J. Davison and J. M. Martinez Montiel, "Structure from Motion using the Extended Kalman Filter," *Springer Tracts in Advanced Robotics*, pp.

1-172, 2012.

[84] G. Burghouts and J. Geusebroek, "Performance evaluation of local color invariants," *Computer Vision and Image Understanding,* pp. 48-62, 2009.

[85] G. Silveira and E. Malis, "Real time visual tracking under arbitrary illumination changes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[86] G. Silveira, E. Malis and P. Rives, "An Efficient Direct Approach to Visual SLAM," *IEEE Transactions on Robotics,* pp. 969-979, 2008.

[87] A. Sunghwan, C. Jinwoo, C. Minyong and C. Wan Kyun, "Metric SLAM in home environment with visual objects and sonar features," in *IEEE International Conference on Intelligent Robots and Systems*, 2006.

[88] A. Sunghwan, C. Jinwoo, D. Nakju Lett and C. Wan Kyun, "A practical approach for EKF-SLAM in an indoor environment: fusing ultrasonic sensors and stereo camera," *Autonomous Robots,* pp. 315-335, 2008.

[89] H. Bischof, H. Wildenauer and A. Leonardis, "Illumination insensitive recognition using eigenspaces," *Computer Vision and Image Understanding,* pp. 86-104, 2004.

[90] G. Steinbauer and H. Bischof, "Illumination Insensitive Robot Self-Localization Using Panoramic Eigenspaces," *RoboCup 2004: Robot Soccer World Cup VIII*, pp. 84-96, 2005.

[91] T. Horprasert, D. Hardwood and L. S. Davis, "A robust background subtraction and shadow detection," in *Asian Conference on Computer Vision*, 2000.

[92] M. Swain and D. Ballard, "Color Indexing," *International Journal of Computer Vision*, pp. 11-31, 1991.

[93] G. D. Finlayson, S. S. Chatterjee and B. V. Funt, "Color Angular Indexing," in *4th European Conference on Computer Vision*, 1996.

[94] J. M. Geusebroek, R. van den Boomgaard, A. W. Smeulders and H. Geerts, "Color invariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1338-1350, 2001.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Appendix A

Mathematical Definitions

1 Motion Model Derivative Expansion

The covariance update step of the camera motion model requires the derivatives of the dynamic motion model with respect to the state (*F*) and with respect to the Gaussian noise of the model (*G*). These derivatives are defined as:

$$F = \frac{\partial \mathrm{f}_v}{\partial \mathrm{x}_v} = \begin{pmatrix} \mathrm{I} & 0 & \Delta t\mathrm{I} & 0 \\ 0 & \frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \mathrm{q}_k^{WC}} & 0 & \frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \omega_{k+1}^{C}} \\ 0 & 0 & \mathrm{I} & 0 \\ 0 & 0 & 0 & \mathrm{I} \end{pmatrix} \tag{70}$$

$$G = \frac{\partial \mathrm{f}_v}{\partial \mathrm{n}} = \begin{pmatrix} \Delta t\mathrm{I} & 0 \\ 0 & \frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \Omega^{C}} \\ \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{pmatrix}. \tag{71}$$

The partial derivative $\frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \mathrm{q}_k^{WC}}$ is defined as:

$$\frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \mathrm{q}_k^{WC}} = \begin{pmatrix} \mathrm{q}_0^1 & -\mathrm{q}_1^1 & -\mathrm{q}_2^1 & -\mathrm{q}_3^1 \\ \mathrm{q}_1^1 & \mathrm{q}_0^1 & \mathrm{q}_3^1 & -\mathrm{q}_2^1 \\ \mathrm{q}_2^1 & -\mathrm{q}_3^1 & \mathrm{q}_0^1 & \mathrm{q}_1^1 \\ \mathrm{q}_3^1 & \mathrm{q}_2^1 & -\mathrm{q}_1^1 & \mathrm{q}_0^1 \end{pmatrix}. \tag{72}$$

Where q^1, q^2 and q^3 represent the following quaternions: $\mathrm{q}^1 = \mathrm{q}\left((\omega_k^C + \Omega^C)\Delta t\right)$, $\mathrm{q}^2 = \mathrm{q}_k^{WC}$, $\mathrm{q}^3 = \mathrm{q}_{k+1}^{WC}$.

The partial derivative $\frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \omega_{k+1}^{C}}$ is decomposed using the chain rule:

$$\frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \omega_{k+1}^{C}} = \frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \mathrm{q}\left((\omega_k^C + \Omega^C)\Delta t\right)} \frac{\mathrm{q}\left((\omega_k^C + \Omega^C)\Delta t\right)}{\partial \omega_{k+1}^{C}}. \tag{73}$$

Where:

$$\frac{\partial \mathrm{q}_{k+1}^{WC}}{\partial \mathrm{q}\left((\omega_k^C + \Omega^C)\Delta t\right)} = \begin{pmatrix} \mathrm{q}_0^2 & -\mathrm{q}_1^2 & -\mathrm{q}_2^2 & -\mathrm{q}_3^2 \\ \mathrm{q}_1^2 & \mathrm{q}_0^2 & -\mathrm{q}_3^2 & \mathrm{q}_2^2 \\ \mathrm{q}_2^2 & \mathrm{q}_3^2 & \mathrm{q}_0^2 & -\mathrm{q}_1^2 \\ \mathrm{q}_3^2 & -\mathrm{q}_2^2 & \mathrm{q}_1^2 & \mathrm{q}_0^2 \end{pmatrix}. \tag{74}$$

2 Feature Initialization Derivative Expansion

The covariance update of the feature initialization step requires the derivatives of the feature (y) with respect to the camera state (x_{cam}) and the detected image point (h). The derivative with respect to the camera state is defined as:

$$\frac{\partial \mathrm{y}}{\partial \mathrm{x}_{cam}} = \frac{\partial \mathrm{y}}{\partial \mathrm{r}^{WC}} \frac{\partial \mathrm{y}}{\partial \mathrm{q}^{WC}}. \tag{75}$$

The Jacobian with respect to the camera position (r^{WC}) is:

$$\frac{\partial \mathrm{y}}{\partial \mathrm{r}^{WC}} = (\mathrm{I} \quad 0). \tag{76}$$

The derivatives with respect to the camera orientation (q^{WC}) are:

$$\frac{\partial \mathrm{y}}{\partial \mathrm{q}^{WC}} = \begin{pmatrix} 0 & \frac{\partial \theta}{\partial \mathrm{q}^{WC}} & \frac{\partial \phi}{\partial \mathrm{q}^{WC}} & 0 \end{pmatrix}. \tag{77}$$

Where:

$$\frac{\partial \theta}{\partial \mathrm{q}^{WC}} = \frac{\partial \theta}{\partial \mathrm{h}^{W}} \frac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}^{WC}} \tag{78}$$

$$\frac{\partial \phi}{\partial \mathrm{q}^{WC}} = \frac{\partial \phi}{\partial \mathrm{h}^{W}} \frac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}^{WC}} \tag{79}$$

$$\frac{\partial \theta}{\partial \mathrm{h}^{W}} = \begin{pmatrix} \dfrac{\mathrm{h}_z^W}{\mathrm{h}_x^{W\,2} + \mathrm{h}_z^{W\,2}} & -\dfrac{\mathrm{h}_x^W}{\mathrm{h}_x^{W\,2} + \mathrm{h}_z^{W\,2}} \end{pmatrix} \tag{80}$$

$$\frac{\partial \phi}{\partial \mathrm{h}^{W}} = \begin{pmatrix} \dfrac{\mathrm{h}_x^W \mathrm{h}_y^W}{\left(\mathrm{h}_x^{W\,2} + \mathrm{h}_y^{W\,2} + \mathrm{h}_z^{W\,2}\right)\sqrt{\mathrm{h}_x^{W\,2} + \mathrm{h}_z^{W\,2}}} \\ -\dfrac{\sqrt{\mathrm{h}_x^{W\,2} + \mathrm{h}_z^{W\,2}}}{\mathrm{h}_x^{W\,2} + \mathrm{h}_y^{W\,2} + \mathrm{h}_z^{W\,2}} \\ \dfrac{\mathrm{h}_z^W \mathrm{h}_y^W}{\left(\mathrm{h}_x^{W\,2} + \mathrm{h}_y^{W\,2} + \mathrm{h}_z^{W\,2}\right)\sqrt{\mathrm{h}_x^{W\,2} + \mathrm{h}_z^{W\,2}}} \end{pmatrix}^T \tag{81}$$

$$\frac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}^{WC}} = \begin{pmatrix} \dfrac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}_0^{WC}} \dfrac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}_1^{WC}} \dfrac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}_2^{WC}} \dfrac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}_3^{WC}} \end{pmatrix} \tag{82}$$

$$\frac{\partial \mathrm{h}^{W}}{\partial \mathrm{q}_i^{WC}} = \mathrm{h}^{C} \frac{\partial R^{WC}}{\partial \mathrm{q}_i^{WC}}. \tag{83}$$

Where the derivatives for the rotation matrix (R^{WC}) with respect to each quaternion component (q_i^{WC}) are:

$$\frac{\partial R^{WC}}{\partial \mathrm{q}_0^{WC}} = \begin{pmatrix} 2\mathrm{q}_0^{WC} & -2\mathrm{q}_3^{WC} & 2\mathrm{q}_2^{WC} \\ 2\mathrm{q}_3^{WC} & 2\mathrm{q}_0^{WC} & -2\mathrm{q}_1^{WC} \\ -2\mathrm{q}_2^{WC} & 2\mathrm{q}_1^{WC} & 2\mathrm{q}_0^{WC} \end{pmatrix} \tag{84}$$

$$\frac{\partial R^{WC}}{\partial \mathrm{q}_1^{WC}} = \begin{pmatrix} 2\mathrm{q}_1^{WC} & 2\mathrm{q}_2^{WC} & 2\mathrm{q}_3^{WC} \\ 2\mathrm{q}_2^{WC} & -2\mathrm{q}_1^{WC} & 2\mathrm{q}_0^{WC} \\ 2\mathrm{q}_3^{WC} & 2\mathrm{q}_0^{WC} & -2\mathrm{q}_1^{WC} \end{pmatrix} \tag{85}$$

$$\frac{\partial R^{WC}}{\partial \mathrm{q}_2^{WC}} = \begin{pmatrix} -2\mathrm{q}_2^{WC} & 2\mathrm{q}_1^{WC} & 2\mathrm{q}_0^{WC} \\ 2\mathrm{q}_1^{WC} & 2\mathrm{q}_2^{WC} & 2\mathrm{q}_3^{WC} \\ -2\mathrm{q}_0^{WC} & 2\mathrm{q}_3^{WC} & -2\mathrm{q}_2^{WC} \end{pmatrix} \tag{86}$$

$$\frac{\partial R^{WC}}{\partial \mathrm{q}_3^{WC}} = \begin{pmatrix} -2\mathrm{q}_3^{WC} & -2\mathrm{q}_0^{WC} & 2\mathrm{q}_1^{WC} \\ 2\mathrm{q}_0^{WC} & -2\mathrm{q}_3^{WC} & -2\mathrm{q}_2^{WC} \\ 2\mathrm{q}_1^{WC} & 2\mathrm{q}_2^{WC} & 2\mathrm{q}_3^{WC} \end{pmatrix}. \tag{87}$$

The derivative for the feature with respect to the detected image point is:

$$\frac{\partial \mathrm{y}}{\partial \mathrm{h}} = \begin{pmatrix} \frac{\partial \mathrm{y}'}{\partial \mathrm{h}} & 0 \\ 0 & 1 \end{pmatrix}. \tag{88}$$

Where $y' = (x_i \quad y_i \quad z_i \quad \theta_i \quad \phi_i)$, which includes all of the feature parameters except the inverse depth (ρ_0). The derivative $\frac{\partial \mathrm{y}'}{\partial \mathrm{h}}$ is expanded as:

$$\frac{\partial \mathrm{y}'}{\partial \mathrm{h}} = \frac{\partial \mathrm{y}'}{\partial \mathrm{h}^W} \frac{\partial \mathrm{h}^W}{\partial \mathrm{h}}. \tag{89}$$

Where:

$$\frac{\partial \mathrm{y}'}{\partial \mathrm{h}^W} = \begin{pmatrix} 0 & \frac{\partial \theta}{\partial \mathrm{h}^W} & \frac{\partial \phi}{\partial \mathrm{h}^W} \end{pmatrix}. \tag{90}$$

With $\frac{\partial \theta}{\partial \mathrm{h}^W}$ and $\frac{\partial \phi}{\partial \mathrm{h}^W}$ previously defined in Equations (80) and (81) respectively.

$$\frac{\partial \mathrm{h}^W}{\partial \mathrm{h}} = R^{WC} \tag{91}$$

3 Quaternion Normalization Derivative Expansion

The covariance update of the quaternion normalization step requires the derivative of the normalized quaternion (q^{norm}) with respect to the non-normalized quaternion (q).

$$\frac{\partial q^{norm}}{\partial q} = (q_0^2 + q_1^2 + q_2^2 + q_3^2)^{-\frac{2}{3}} \mathcal{Q} \tag{92}$$

Where: $\mathcal{Q} = \begin{pmatrix} q_1^2 + q_2^2 + q_3^2 & -q_0 q_1 & -q_0 q_2 & -q_0 q_3 \\ -q_1 q_0 & q_0^2 + q_2^2 + q_3^2 & -q_1 q_2 & -q_1 q_3 \\ -q_2 q_0 & -q_2 q_1 & q_0^2 + q_1^2 + q_3^2 & -q_2 q_3 \\ -q_3 q_0 & -q_3 q_1 & -q_3 q_2 & q_0^2 + q_1^2 + q_2^2 \end{pmatrix}$.