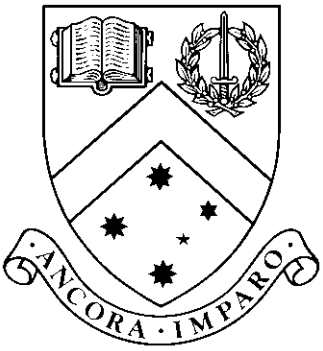
Autonomous Robot Navigation: Appearance Based Topological SLAM

by

Wen Lik Dennis Lui

BE(HONs) Mechatronics Engineering (2006)



Thesis

Submitted by Wen Lik Dennis Lui

for fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Supervisor: Professor Ray Jarvis

Associate Supervisor: Associate Professor R. Andrew Russell

Intelligent Robotics Research Centre

Department of Electrical and Computer Systems Engineering

Monash University, Australia

February, 2011

For my wife, Ming Lih, and my family, who stood by me throughout the course of this thesis

Contents

List of Tables

List of Figures

List of Algorithms

Autonomous Robot Navigation: Appearance Based Topological SLAM

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Wen Lik Dennis Lui
February 14, 2011

Autonomous Robot Navigation: Appearance Based Topological SLAM

Wen Lik Dennis Lui
[redacted]
Monash University, Australia, 2011

Supervisor: Professor Ray Jarvis
[redacted]
Associate Supervisor: Associate Professor R. Andrew Russell
[redacted]

Abstract

The main focus of this research is to develop an autonomous robot capable of self-navigation in an unknown environment. The proposed system performs autonomous navigation primarily based on the following visually perceived information:

- **Range Estimation**: A novel variable single/multi baseline omnidirectional stereovision system with an option to automatically select the baseline that is adjusted to the environment with the establishment of stereo correspondences and triangulation offloaded to the Graphics Processing Unit (GPU). Additionally, as a safety measure, a low level reactive obstacle avoidance system using the disparity maps returned from a Bumblebee stereo camera range system provides a secondary source of pseudo range estimation to steer the mobile robot away from obstacles which the primary system has failed to detect. The two sensors complement one another due to the vertical stereo setup for the primary sensor and the horizontal stereo setup for the secondary sensor, where the former is more sensitive towards horizontal features whereas the latter is better for vertical features.

- **Motion Estimation**: A 3DoF visual odometry system combining distance travelled estimated by a ground plane optical flow tracking system, with bearing estimated by the panoramic visual compass system.

- **Place Recognition**: An appearance-based place recognition system using image signatures created from Haar decomposed omnidirectional images for loop closure detection.

These components were integrated together into the mobile robot's navigation system which balances its effort amongst loop closing and exploration, decides its next course of action, performs path planning and executing the selected path. As the mobile robot engages the environment, the positional drift associated to the mobile robot's estimated

location increases over time, thus, making it necessary to perform loop closing regularly by detecting it via the place recognition system and maintaining the global consistency of its internal representation of the environment (in the form of a topological map) by employing a relaxation technique. Due to the importance of performing loop closing regularly, an active loop closure detection and validation system, that enables the mobile robot to actively search for loop closures and to validate ambiguous loop closures, was proposed, developed and validated.

A wide variety of experiments were conducted to verify and evaluate the performance of the entire system at both the system and subsystem levels. All experimental results were compared against ground truth where possible. Fully autonomous experiments combining all the above were conducted in indoor, semi-outdoor and outdoor environments. In addition, semi-autonomous experiments were conducted where the mobile robot, provided with *a priori* information in the form of a topological map built on a separate occasion in an offline manner, was required to reach a user specified destination (goal oriented). Finally, the proposed place recognition system was applied to the map merging problem where experimental results showed the improved robustness of loop closure and map merging detection when fused with a laser-based metric SLAM system.

Acknowledgments

It is a pleasure to thank those who made this thesis possible. First and foremost, I would like to thank my supervisor, Professor Ray Jarvis. It is difficult to overstate my gratitude to him. Throughout the course of this thesis, he provided me with inspirational guidance, motivation and constructive comments. Many thanks to my associate supervisor, A/Prof. Andy Russell, who regularly shares his knowledge on biologically inspired mechanisms and has loaned me his brand new ActivMedia P3-AT mobile robot over a period of time. Thanks goes to A/Prof. Lindsay Kleeman for the many invaluable discussions in robotics and his support in the collaboration between his PhD student, Fredy Tungadi, and myself. I am also grateful to A/Prof. Lindsay Kleeman and Dr. Wai Ho Li for organizing the *Intelligent Robotics Research Centre Seminars* which have been a great place for practice presentations and also a place where the staff and postgraduate students can share their many interesting stories from home and abroad.

It was a great pleasure being a member of the *Intelligent Robotics Research Centre*. I am indebted to my many student colleagues and labmates, both past and present, for making the journey an intellectually stimulating and enjoyable one. Dr. Wai Ho Li provided me with a constant supply of latest news and development in science and technology. Together with Damien Browne, they made themselves available for lengthy discussions which normally lead to interesting debates on any topic in general. Fredy Tungadi regularly shared his insights on autonomous exploration and mapping and was a very supportive research partner. Alan Zhang, Jay Chakravarty, Nghia Ho, Zhi Li, Anh Tuan Phan, Fahmi Miskon, Om Gupta, Rahul Walia, Sutono Effendi and others who were available for discussions, lunches and sports. I also acknowledge the contribution of Tony Brosinski, from the mechanical workshop, for sharing his expertise in hardware fabrication.

The completion of this thesis would not be possible without the support from my family. To my parents, James and Soo Lan, thank you for encouraging me to pursue my dreams and the unconditional love you provided me with throughout all these years. To my siblings, Albert and Vincent, thank you for your moral support. Special thanks to Ming Lih, my best friend and wife, for her love and support, and for her patience putting up with my long working hours that extended to weekends at times. Last but not least, I would like to express my gratitude to my family-in-law who entrusted me with their beloved daughter and sister.

This research was supported by the International Postgraduate Research Scholarship (IPRS), Monash Graduate Scholarship (MGS) and the *Intelligent Robotics Research Centre* at Monash University. Funding for travelling to both international and local conferences were also provided by the *Monash Research Graduate School* and the *Department of Electrical and Computer Systems Engineering*. The financial support is gratefully acknowledged.

1

Introduction

According to Oxford Dictionaries (2010), the word robot originated from the Czech noun *robota* meaning "forced labour". A robot is defined as "a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer" and the word was coined in the play, *Rossum's Universal Robots*, by Čapek (1920). It was further popularized through the writings of Isaac Asimov's Robot Series (a series of short stories and novels) and has become the standard word to describe what is known as today's robots.

Robots today are classified into two categories; industrial and service robots. Industrial robots (robotic arms/manipulators with varying degrees of freedom, maximum payload, etc) are currently employed in various industries ranging from the manufacturing of motor vehicles, electronics, food, etc, due to its robustness, repeatability and accuracy to operate in a tightly controlled environment. In addition, it has recently gained success and widespread use as teleoperated surgical robots (e.g. da Vinci multipurpose surgical robotic system (Intuitive Surgical, 2010)) and is moving towards the direction of semi-autonomous systems (e.g. guiding anesthetic shot to patients from a remote location (Tighe et al., 2010)). Milestones achieved by surgical robots are available in (Gomes, 2010) whereas milestones achieved by industrial robots can be found in (The Japan Society of Mechanical Engineers, 2007).

Service robots can be further sub-classified as to whether they are meant for professional or personal use. Professional uses of service robots range from construction and demolition applications, rescue and security applications to research platforms used in private or university laboratories across the world. Personal uses of robots are mostly found in domestic and entertainment/leisure applications such as the popular home vacuum cleaners and lawn mowing robots for the former and toy and hobby systems for the latter. According to the report by the International Federation of Robotics (2010), there was strong investments in industrial robots from 2004-2005 but installations slowed down from 2006-2008 with sales nosediving in 2009 due to the recent global economic meltdown.

However, the report suggests strong recovery in 2010 and projected growth from 2011-2013. Despite the slump, the projected number of industrial robots will increase from 1 million in 2009 to 1.2 million in 2013 with 30% of these industrial robots operating in Japan. The total number of service robots is expected to increase from 8.7 million in 2009 to approximately 20.1 million units in 2013. In addition, it is predicted that the personal use of robots, such as having a personal domestic assistant or personal transportation, is expected to rise in the future. Robots are best suited for tasks which fall into any of the following categories; dull, repetitive, dangerous or inaccessble to human beings. They can also be used to provide assistance to the aging population, thus, overall, improving our quality of life. Based on the report from the Australian Bureau of Statistics (2008), people aged 65 years and over made up 13% of Australia's population in 2007. This proportion is projected to increase to 23-25% in 2056 based on the current trends in fertility, life expectancy at birth, net overseas migration and net interstate migrations. On the global stage, there are almost 500 million worldwide aged 65 years and above in 2006, and this is projected to increase to 1 billion in year 2030 (1 in every 8 of the global population), with the most significant increases taking place in developing nations (National Institute on Aging et al., 2007). The global growth in the aging population is mainly due to the increasing average lifespan as a direct result of better accessibility to healthcare, food and basic necessities.

For robots to transit from operating in tightly controlled environments (e.g. factory floor) to unknown environments without human supervision, more intelligent robots are required to cope with unstructuredness and ambiguity. For example, a humanoid robot targeted to domestic applications has to navigate safely in a dynamic environment, while performing complex actions such as object recognition and manipulation, as it attempts to fetch the daily newspaper from the driveway. In the case of large environments, swarm robotics may be used to expedite the exploration of the environment through coordination and cooperation. For long term operations, robots are required to learn and adapt to its surroundings. Nevertheless, a fundamental requirement to drive the next generation of robots to a wider range of applications is mobility.

The first major effort undertaken to develop an intelligent mobile robot was carried out by Nilsson (1969). From there on, some key developments on mobile robots up to 1995 were documented and discussed in Gage (1995). Mobile robots can be applied to many applications such as courier services in hospitals (Takashi et al., 2010) and offices, autonomous vehicles (Montemerlo et al., 2008; Thrun, 2010) (where the technology described has been recently used by Google's fleet of autonomous cars logging over 140k miles at the time of writing), landmine detection (Toko et al., 2004), planetary explorations (Bajracharya et al., 2008) and search and rescue operations (Nagatani et al., 2009). Applications that are more relevant locally in Australia are such as bush fire fighting (Jarvis and Marzouqi, 2005; Jarvis, 2008), mining operations (Nebot, 2007), underwater surveying (Williams et al., 2010) and domestic/personal robots (Jarvis, 2003; Jarvis et al., 2009). Despite the progress made over the past decades, there are still problems yet to be resolved and issues to be addressed before a fully autonomous mobile robot can be deployed into real world

applications, which may be made up of large, dynamic and unstructured environments. As a result, most mobile robots nowadays are prototypes (e.g. service robots for professional use or entertainment robots with limited use) mainly found in laboratories. More effort is required before this next generation of robots can be employed by the industry and for personal use (e.g. domestic or personal transportation). As such, the primary focus of this thesis is to propose, develop and validate methods for developing an intelligent mobile robot that is capable of self navigation in unknown environments. The following section outlines the various challenges and motivation behind the proposed system.

1.1 Challenges and Motivation

The three main challenges in intelligent robotics are (a) control systems, (b) perception systems and (c) advanced mechanical design. It is generally accepted that perception is the main bottleneck in robotics. A robot without a reliable perception of its surroundings makes advanced control systems and mechanical designs ineffective since actions carried out by a robot with an ambiguous representation of its environment is less likely to produce the desired results. Depending on the task at hand, there are varying degrees of difficulty when it comes to perception. For some tasks, one might argue that basic structural information of the environment should suffice. However, in many cases, where a robot with a higher degree of intelligence is required or desirable, the robot may require more than just basic structural information, such as shape, object classification, place recognition, face recognition, etc. To better equip robots to challenges present in the real world, many have turned to visual information for answers.

The way humans see the world is primarily dependent on visual information. Visual sensors have become an attractive source of information on robots as they are cheap, passive and widely available. This growth is further enhanced by the availability of cheap computing power and off-the-shelf cameras with better image quality and higher frame rates. In the past two decades, researchers have demonstrated mild success in using visual information for tasks such as general object recognition, ego-motion estimation, scene geometry, etc, which facilitates a wide variety of higher level tasks such as learning, object manipulation, navigation, etc. However, passive vision is plagued with problems such as the effect of lighting variation, dependency on textured environments, higher computational requirements, colour constancy, etc. For mobile robots, the list of issues extends to motion blurring (motion and vibration of the mobile robot), the *windowing* problem suffered on perspective camera systems (e.g. the need to register different viewpoints of the same location and problems with minimal or non-overlapping regions) due to its limited field of view, the thirst for computational power resulting in higher power consumption which affects the total operational time of the mobile platform due to the current limitations in energy storage and the need to satisfy real-time constraints.

Due to the promising potential of visual information, the author was motivated to develop a mobile robot using vision as its primary sensing mode. Instead of using a perspective camera system as its primary vision system, the author explored how an omnidirectional vision system can be used to alleviate the previously described *windowing*

problem. A multi-level system is envisioned here, from a low level reactive obstacle avoidance system to a high level central decision making system that takes into account the various sensory information available at the point in time in order for the mobile robot to make the best informed decision. Ideally, the system should have multiple complementary sensors such as the combination of laser range information, visual information, Global Positioning System (GPS), Inertial Measurement Unit (IMU), etc. Nevertheless, in this work, the author's primary focus is to demonstrate how visual information alone can be used by a fully autonomous mobile robot to navigate in an unknown environment, which combines various vision-based technologies to perform range estimation, visual odometry, place recognition, etc. Despite the primary objective to develop a pure vision-based mobile robot, the author also investigated on how laser range information and GPS can be incorporated into the system. Besides the challenges described in the previous paragraphs, it is also crucial to ensure the proper means of evaluating the performance of the various system modules and the system as a whole. Benchmarking and performance evaluation have recently been a very popular topic in robotics. However, this is not a simple matter, given the complexity of robotic systems. The following sections describe the key research contributions made and the organization of the thesis.

1.2 Key Contributions

The work described in this thesis has produced a total of 4 conference papers and 1 journal article. Additionally, a second journal article will be submitted to *The International Journal of Robotics Research* for review. Details of all publications are as follows:

- W.L.D. Lui and R. Jarvis, "An Omnidirectional Vision System for Outdoor Mobile Robots", in *Workshop on Omnidirectional Robot Vision* (Workshop Proceedings of SIMPAR 2008), Venice, Italy, 2008, pp. 273-284. (Lui and Jarvis, 2008)

- W.L.D. Lui and R. Jarvis, "Eye-Full Tower: A GPU-based Variable Multibaseline Omnidirectional Stereovision System with Automatic Baseline Selection for Outdoor Mobile Robot Navigation", *Robotics and Autonomous Systems*, Vol. 58, No. 6, 2010, pp. 747-761. (Lui and Jarvis, 2010b)

- F. Tungadi, W.L.D. Lui, L. Kleeman and R. Jarvis, "Robust Online Map Merging System using Laser Scan Matching and Omnidirectional Vision", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 7-14. (Tungadi et al., 2010)

- W.L.D. Lui and R. Jarvis, "A Pure Vision-based Approach to Topological SLAM", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 3784-3791. (Lui and Jarvis, 2010c)

- W.L.D. Lui and R. Jarvis, "An Active Visual Loop Closure Detection and Validation System for Topological SLAM", *Australasian Conference on Robotics and Automation*, Brisbane, Australia, 2010. (Lui and Jarvis, 2010a)

- W.L.D. Lui and R. Jarvis, "A Pure Vision-based Topological SLAM System", *The International Journal of Robotics Research*. (To be Submitted)

The key research contributions made are summarized in the following subsections.

Non-central Omnidirectional Catadioptric Stereovision

To facilitate the development of a non-central omnidirectional catadioptric stereovision system, a suitable camera calibration technique was proposed. Unlike many camera calibration techniques, which estimate the intrinsic and extrinsic parameters of the system, the proposed camera calibration technique computes the discrete relationship between the angles of elevation of the non-central equiangular mirror with respect to the radial distances measured from the centre of the mirror in the image. This was achieved by exploiting the geometry of the catadioptric vision system with respect to the calibration grid on the inner surface of the calibration bin. Using this information, triangulation could be performed to estimate the 3D position of a pair of stereo correspondences established using a local area-based method, producing a dense disparity map. Processing times were significantly reduced as both the establishment of stereo correspondences and triangulation were offloaded to the Graphics Processing Unit (GPU), providing an option to use this as a real-time system.

Initial results of the work, illustrating the use of the proposed camera calibration technique for the estimation of the 3D positions of stereo correspondences established using a local area-based method on a CPU, are available in (Lui and Jarvis, 2008). An updated version of the stereovision system, describing the establishment of stereo correspondences and performing triangulation on the GPU, is available in (Lui and Jarvis, 2010b). Additionally, it also includes a more detailed analysis of the proposed camera calibration technique.

Automatic Baseline Selection

The novelty of the proposed automatic baseline selection technique is its capability to determine a suitable baseline for a given environment. The baseline is adjusted based on the characteristics of the employed equiangular mirror and the histogram of disparity distribution in the omnidirectional image pair. The mobile robot can utilize the selected baseline when a single omnidirectional image pair is required. In other cases where a denser and more accurate representation of the environment is desired, this technique can be utilized to capture multiple stereo pairs at baselines adapted to the environment such that ambiguity in stereo correspondences can be reduced via a multibaseline stereovision algorithm based on voxel voting. The automatic baseline selection technique combined with the non-central omnidirectional catadioptric stereovision (camera calibration, GPU implementation and multibaseline stereovision) is thoroughly described in (Lui and Jarvis, 2010b).

Visual Odometry

There may be many methods to derive the mobile robot's odometry using visual information, with some more accurate than others. However, for it to be useful on mobile robots, the visual odometry system has to operate in real-time. The proposed visual odometry system localizes the mobile robot using distance travelled estimated by a ground plane optical flow tracking system, with bearing estimated by a panoramic visual compass system (appearance-based technique). Necessary modifications and enhancements were made to the original algorithms (Fernandez and Price, 2004; Labrosse, 2007) to deal with issues observed in actual experiments (i.e. vibration induced by the motion of the mobile robot on the omnidirectional catadioptric vision system). The final product is an inexpensive, 3DOF, real-time visual odometry system tested in a wide range of environments. It can also be easily transferred and used on a different mobile robot platform since, as an external system, it is not affected by wheel slippage and is more tolerant to wheel radius changes due to mechanical imperfections and load variations. The visual odometry system is described in (Lui and Jarvis, 2010c).

Probabilistic Place Recognition using Haar Wavelets

It was only recently that a complete probabilistic framework for appearance-based localization and mapping was proposed (Angeli et al., 2008; Cummins and Newman, 2008). In this thesis, a refined probabilistic place recognition system was proposed based on the work of Angeli et al. (2008) using omnidirectional images. It yielded a set of general expressions with parameters that could be utilized to vary the speed of convergence of the system and determine what should be deemed as a discriminative matching score. Due to this refinement, the framework could be conveniently applied to systems using different appearance models and image querying metrics or targeting different applications. In this case, the refined probabilistic framework has been demonstrated to work well with image signatures created using the standard Haar 2D decomposition technique as its appearance model. The performance of the probabilistic framework has been compared to the rank-based framework (based on the ranking of Haar scores and a fixed threshold) and a complete description and analysis of the probabilistic framework have been detailed in a journal article which will be submitted to *The International Journal of Robotics Research* for review.

Active Loop Closure Detection and Validation

Loop closing is a vital component for mobile robot navigation without *a priori* information of the environment since the robot has to explore, build and at the same time maintain a globally consistent map. However, being able to detect loop closures (i.e. place recognition system) without actively searching for it, is simply trying to perform loop closing by chance. In this work, a simple strategy was proposed to ensure that the mobile robot actively searches for loop closures at appropriate times. The proposed strategy uses a number of evaluations based on past and present information to validate the detected

loop closure. If a loop closure was found to be ambiguous, the mobile robot actively validates the loop closure event. System restoration would be performed when the loop closure event could not be validated, thus, reducing the risk of damaging the map built so far by the mobile robot. This has been detailed in (Lui and Jarvis, 2010a)

Topological SLAM using a Pure Vision-based Approach

In this work, a topological SLAM system was developed by combining a variable multi/single baseline omnidirectional catadioptric stereovision system, visual odometry, a complete loop closing system and a reactive obstacle avoidance system. Using the omnidirectional catadioptric stereovision system as its primary sensor, the mobile robot was able to perceive its surroundings in 3D. This information was subsequently used for path planning purposes. As the mobile robot tries to execute its planned path, the trajectory taken by the mobile robot was estimated by the visual odometry system. The complete loop closing system, which consists of an appearance-based place recognition system with active loop closure detection, loop closure validation and system restoration, was developed to enable the mobile robot to deliberately return to previously explored locations such that loop closing could be performed regularly to maintain the global consistency of the topological map and contain the errors accumulated in the mobile robot's localization process. Relative metric information, between the current location of the mobile robot with the matching node in the topological map that raised the loop closure event, was recovered utilizing information from the omnidirectional catadioptric stereovision system. Lastly, a reactive obstacle avoidance system based on the disparity maps returned from a Bumblebee stereovision camera would temporarily override the normal control of the mobile robot when it was found too close to an obstacle (not detected by the omnidirectional catadioptric stereovision system).

The key contribution here is the integration of the individual modules described in the previous paragraph into a complete system that was tested in indoor, semi-outdoor and outdoor environments. The initial integrated system, without active loop closure detection and validation, was tested in an indoor environment with results published in (Lui and Jarvis, 2010c). The final integrated system, tested in a variety of semi-outdoor and outdoor environments, has been detailed in a journal article which will be submitted to *The International Journal of Robotics Research* for review.

Online Map Merging

The proposed map merging system merges maps acquired from different runs on a single mobile robot into a globally consistent map, which can be extended for use in a multi-robot setting. This is the first system which combines a probabilistic Haar-based place recognition system with laser scan matching for map merging. It is fast, enabling it to be used as an online process, merging maps while the mobile robot is traversing the environment and robust due to the combination of visual and laser range information. In addition, it is algorithmically simple, efficient and does not require any offline processing. This has led to the publication in (Tungadi et al., 2010). Nonetheless, a more complete

description and analysis of the probabilistic framework are only available in a recent journal article which will be submitted to *The International Journal of Robotics Research* for review.

1.3 Thesis Organisation

The thesis is organized into the following chapters and appendices. Since each chapter covers a distinct topic, a literature review catering to the specific topic will be presented in each chapter. A multimedia DVD containing videos of experimental results accompanies this thesis. Details of contents included in the multimedia DVD can be found in Appendix A.

Chapter 2: Omnidirectional Catadioptric Stereovision

This chapter describes the development of the non-central omnidirectional catadioptric stereovision system. It is divided into 2 sections. The first section details the camera calibration technique. It covers in detail the process of recovering the discrete relationship between the angles of elevation of the non-central equiangular mirror with respect to the radial distances measured from the centre of the mirror in the image. Further details on the problem formulation are included in Appendix B. Additionally, it also provides an analysis on the impact to the system if the assumptions made were violated.

The second part of this chapter describes the establishment of stereo correspondences via a local area-based method and triangulation on the Graphics Processing Unit (GPU). The visualization of the estimated 3D positions of stereo correspondences in various environments and the comparison between the processing time required by different combinations of stereo parameters on the proposed GPU implementation are both included in this chapter. This chapter concludes by comparing the system to related work in the literature and suggests possible improvements.

Chapter 3: Multibaseline Omnidirectional Stereovision

This chapter describes the advantages of combining multiple stereo pairs taken at different baselines for a stereovision system. Since the mobile robot is equipped with a variable baseline system capable of capturing multiple stereo pairs at different baselines, the mobile robot has to decide on the baselines to use for capturing these multiple stereo pairs. An automatic baseline selection technique based on the characteristics of the employed equiangular mirror and the histogram of disparity distributions is proposed in this chapter for capturing multiple stereo pairs at baselines adapted to the environment. This technique can also be used even when only a single stereo pair is required. Experimental results included in this chapter illustrate the 3D voxel environments produced by a single omnidirectional stereo pair captured at the automatically selected baseline and the multibaseline stereovision technique based on voxel voting. A discussion of the experimental results is included at the end of this chapter.

Chapter 4: Visual Odometry

This chapter covers the development of a real-time 3DOF visual odometry system. The appearance-based panoramic visual compass, used for the tracking of the mobile robot's heading, and the ground plane optical flow tracking system, used for estimating the distance travelled by the mobile robot, are thoroughly described in this chapter. Rigorous performance evaluation of the proposed system, conducted in an indoor lab environment covered with carpet and a semi-outdoor environment with concrete slab paving, and comparing the resulting trajectory estimated by visual odometry with ground truth, is provided in this chapter. An independent evaluation of the panoramic visual compass in an outdoor setting is also available in this chapter.

Chapter 5: Mobile Robot Localization and Mapping

This chapter is divided into 3 sections. The first section discusses and illustrates the use of Haar wavelets in a place recognition system. The second section presents the two possible frameworks (rank-based and probabilistic) that can be employed by the place recognition system. In addition, it also details a method for recovering the metric information between two nodes using the data from the omnidirectional catadioptric stereovision system when the place recognition system believes the mobile robot to be revisiting a previously seen location. Since the mobile robot builds a topological map, this is referred to as topological SLAM. Experimental results in this chapter illustrate the performance of Haar wavelets independently as an image retrieval system on a database of images taken in semi-outdoor and outdoor environments, provide additional insight to the proposed probabilistic framework for the place recognition system and show how the global consistency of the topological map is maintained using an existing global relaxation technique.

Chapter 6: Autonomous Vision-based Topological SLAM

This chapter details a pure vision-based autonomous mobile robot performing topological SLAM. It describes its path planner, the active loop closure detection and validation system which balances the objectives of exploration and loop closing, and a reactive obstacle avoidance system based on the disparity maps returned by a Bumblebee stereovision system. In addition, a semi-autonomous navigation system (with *a priori* information in the form of a map provided to the system) which performs goal-seeking is also illustrated. Experimental results of the mobile robot in the autonomous, semi-autonomous and manually driven modes are illustrated and described. This chapter concludes with the comparison of the rank-based and probabilistic frameworks for the place recognition system and a discussion of experimental results, with suggestions on future work.

Chapter 7: Online Map Merging

This chapter illustrates an application of the probabilistic place recognition system based on the Haar Wavelets to solving the map merging problem. The mobile robot platform used in this chapter performs metric SLAM using laser scan matching and performs online

map merging by fusing it with an appearance-based probabilistic place recognition system (exploiting the advantages of both structural and visual information). 3 experiments with varying difficulty levels conducted in an indoor lab and office environment are illustrated in this chapter. Lastly, this chapter concludes with a discussion of experimental results and outlines possible future improvements.

Chapter 8: Conclusion and Future Work

This chapter discusses how the research challenges highlighted in the introduction have been addressed by this research. It summarizes the research findings and provides a thorough discussion of possible future work.

Appendix A: Multimedia DVD Contents

Details the contents of the accompanying multimedia DVD.

Appendix B: Calibrating a Non-central Equiangular Catadioptric System

Details the camera calibration technique for a non-central equiangular catadioptric system, including an analysis on the impact to the system if the assumptions made for the calibration process were violated.

Appendix C: Camera Motion Estimation using Ground Plane Optical Flow

Details the derivation of converting ground plane optical flow to motion estimates.

Appendix D: Modeling the State Transition Probabilities and Likelihood Voting Scheme

Provides additional information on the derivation of the new state transition probabilities and likelihood voting scheme for the refined probabilistic framework used by the place recognition system.

Appendix E: Supplementary Experimental Results

Provides additional experiments conducted in the fully autonomous and semi-autonomous modes.

Appendix F: Ground Plane Detection using Stereovision

Details the ground plane detection algorithm and results using the Bumblebee stereovision system.

Appendix G: Global Positioning System (GPS)

Provides a brief review on the development of Global Navigation Satellite Systems (GNSS) around the world and illustrates preliminary results from an inexpensive GPS receiver for mobile robot localization.

2

Omnidirectional Catadioptric Stereovision

2.1 Introduction

Visual information can be regarded as the most important sensory mode for humans. Visuals are useful for learning, recognizing objects, navigating and everyday activities. In fact, most of our actions are being guided by vision. As such, researchers attempt to replicate these complex vision systems by means of computers and cameras. Similar to human eyes, conventional cameras have a limited/narrow field of view which makes the matching and tracking of salient visual features in the scene a more difficult problem. In contrast, omnidirectional vision systems sidestep the commonly encountered *windowing* problem by capturing a much wider and continuous field of view. Furthermore, it is an established fact that omnidirectional images are well suited for both appearance-based and landmark-based localization approaches because most features are still visible after some small arbitrary amount of motion is taken by the robot. Thus, omnidirectional vision systems, which started off initially merely as an additional tool for roboticists, have become the main field of study for many researchers as it gains popularity.

In the literature, there is a rich collection of techniques that can be used to create an omnidirectional vision system. Common techniques are: (1) combination of camera(s) with mirror(s), (2) using a single camera scanning the environment while it rotates in the horizontal plane around the axis perpendicular to its optical axis or (3) a fixed multiple camera system. The combination of camera(s) with mirror(s) is also known as catadioptric vision systems (or omnidirectional catadioptric vision system if the system provides 360^o field of view) and a popular configuration is to combine a single camera with a curved (dome-like) mirror. As defined by Hecht and Zajac (1987), *dioptrics* is the science of refracting elements (e.g. lens) and *catoptrics* is the science of reflecting surfaces (e.g. mirrors) and this combination is therefore referred to as catadioptrics. The second technique is capable of producing high resolution omnidirectional images but unfortunately, scanning and stitching could not be achieved in real time. On the other hand, the third

option resolves the scanning and stitching issue outright but such systems normally come with a heftier price tag. These systems are still required to stitch the images together but the fixed camera positions will enable more efficient algorithms to take advantage of the fixed geometry between cameras. Since the proposed system in this thesis utilizes the popular single camera with a curved mirror configuration, the remainder of this introduction will be dedicated to this class of catadioptric vision systems. Readers interested in learning more about other classes of omnidirectional vision systems should refer to the book compiled by Benosman and Kang (2001).

This class of single camera with a curved mirror catadioptric vision system can be further categorized into central or non-central catadioptrics. Central catadioptric systems are desirable since they only have a single effective viewpoint which allows the derivation of the epipolar geometry of two omnidirectional images, whereas non-central catadioptric systems have multiple effective viewpoints. In addition, having a single effective viewpoint is more desirable since it permits the generation of geometrically-correct perspective images from the original image captured by the system. Central catadioptric systems have been thoroughly studied by Baker and Nayar (1999) and Svoboda and Pajdla (2002). In (Baker and Nayar, 1999), a complete set of lens and mirror (quadric surfaces) combinations which have this attractive single viewpoint property were listed. Furthermore, the epipolar geometry for such lens and mirror combinations had been fully derived by Svoboda and Pajdla (2002). With the epipolar geometry derived, the epipolar constraint can be applied to make the search for correspondences a less computationally expensive process. Subsequently, by using these correspondences together with the parameters of the calibrated catadioptric vision system, the surrounding environment can be represented in 3D by means of triangulation. The ability to perceive the environment in 3D is particularly useful for mobile robots, being highly relevant for navigation and path planning purposes.

Unfortunately, most of the catadioptric systems are non-central. This can be due to the pairing of a non-telecentric lens with a parabolic mirror, the centre of the perspective camera is not located in one of the focal length points of the hyperbolic or elliptic mirror or a mirror profile which does not possess the single effective viewpoint property (i.e. equiangular, equiresolution, spherical mirrors) is used instead of those listed in (Baker and Nayar, 1999). When these systems are mounted on a mobile robot, there are other external factors such as vibration and human errors during the alignment of camera and mirror which transforms a central catadioptric to a non-central system unintentionally. In order to understand the properties of these non-central systems, Swaminathan et al. (2001) had conducted an in-depth analysis of the locus of the multiple viewpoints (known as caustics) for non-central catadioptric systems with conic reflectors. As a result of this study, a calibration technique for conic non-central catadioptric systems with known motion was proposed. As for the purpose of extracting 3D information from two images taken from the same scene, Gonçalves and Araújo (2004) proposed an estimation technique based on the geometrical relationships of the mapping of 3D scene points to image points on quadric surfaces. On the other hand, Mičušík and Pajdla (2004) proposed a suitable non-central model, which is approximated from a central model, to obtain a projection

model for spherical mirrors. In summary, this directly implies that a completely different model might be required for a particular non-central camera-mirror combination since no single model can be general enough to cater for all possible combinations.

In the context of general stereovision using catadioptric systems, there have been a few notable works in recent years. For instance, Arican and Frossard (2007) developed a global energy minimization algorithm based on the well known graph-cut technique in order to produce dense estimation of disparities between omnidirectional images using a spherical framework. The proposed method maps omnidirectional images captured using a catadioptric system with a parabolic mirror onto a 2D sphere (central catadioptric) via inverse stereographic projection (Geyer and Danilidis, 2001) and performs dense disparity estimation using graph-cut techniques directly on the 2D sphere. In another work (He et al., 2007), an omnidirectional stereo sensor was developed by using a single perspective camera with two hyperbolic mirrors where dense disparity estimation was achieved by using a combination of dynamic programming and graph cut techniques. The proposed disparity estimation technique and calibration procedure yielded accurate depth measurements but the proposed camera-mirror configuration would occlude part of the top mirror. Additionally, the use of a single camera would introduce a defocusing effect due to the different depth of each mirror from the camera (the amount of defocusing and occlusion would depend on the separation between the top and bottom mirror). Lastly, Ragot et al. (2008) proposed an innovative calibration methodology to establish the discrete relationship between 3D points and the corresponding 2D locations of the pixel using an external calibration pattern made of luminous markers and a voxel based 3D reconstruction technique.

2.2 Hardware Configuration

As depicted in Fig. 2.1, the ActivMedia P3-AT was used to support the research and development of the omnidirectional catadioptric stereovision system. This is a highly versatile, all-terrain platform used in many laboratories across the world. The two catadioptric systems, each consisting of a Canon Powershot S3 IS and an equiangular mirror, were stacked in a vertical configuration. The Canon Powershot S3 IS can capture still images with a maximum resolution of 6 megapixels and this is highly desirable for vision-based outdoor mobile robots since distant natural features will become undetectable in low resolution images. However, there is a fair trade-off between computational workload and efficiency, and striking this balance is an important task for the system developer. This issue will be discussed more thoroughly in Section 2.4.

The robotic platform shown in Fig. 2.1 was equipped with a novel camera elevation device, which we refer to as the *Eye-Full Tower* (pun intended). This device was primarily used to translate the upper catadioptric system vertically upwards/downwards in order to vary the effective baseline of the system through the control of a DC motor via a parallel port. This DC motor would drive the gearing system (enclosed at the bottom of the device), effectively rotating the threaded rods connected to it. These threaded rods are connected to the brass bits attached on the top of the upper catadioptric system.

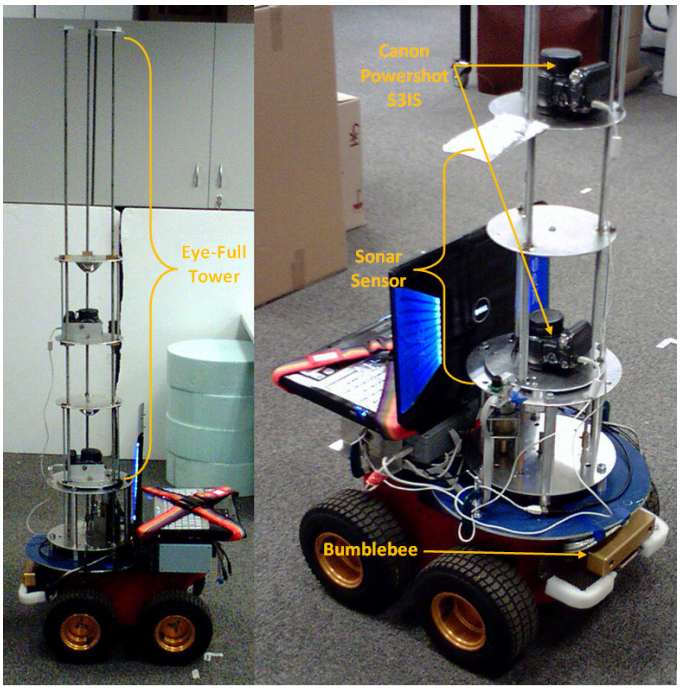


Figure 2.1: The ActivMedia P3-AT Equipped with the Eye-Full Tower

Thus, when these threaded rods rotate, the upper catadioptric system would translate upwards/downwards depending on its direction of rotation. The *Eye-Full Tower* has an effective baseline of 30cm to 90cm (measured from the lower camera). The effective baseline would be constantly monitored by a sonar sensor which was found to produce an average error of 0.5cm. A Dell XPS M1730, equipped with dual SLI Nvidia 8800M GTX graphics cards, was used to control this hardware. For your information, the ActivMedia P3-AT mobile robot platform was replaced by another mobile platform powered by a differential drive wheelchair motor/gear set and will be described in a later chapter.

2.3 3D Reconstruction with Non-Central Catadioptric Cameras

3D information is recovered from a pair of vertically stacked non-central catadioptric cameras by using our proposed camera calibration technique. Since equiangular mirrors were employed in this work, we will firstly proceed with a brief description of the properties of these mirrors. Then, we will describe the camera calibration process, which facilitates the recovery of 3D information from a pair of non-central catadioptric cameras by exploiting the properties of these mirrors.

2.3.1 Equiangular Mirrors

Equiangular mirrors are a special class of curved mirrors which possess a linear relationship between the ratio of change in radial angle with respect to change in angle of elevation as shown in Fig. 2.2. The mirror used in this project was designed by Chahl and Srinivasan (1997). One of the main advantages of using this mirror is the relatively simple

algorithm (Polar to Cartesian coordinates) used to unwarp the original image to a more intuitively understandable form. In addition, this unwarping process was found useful as it sidesteps the scaling problem of image neighbourhoods during the establishment of stereo correspondences via an area-based method.

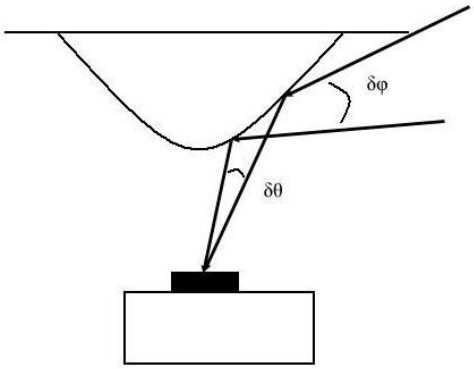


Figure 2.2: The ratio of change in radial angle ($\delta\theta$) with respect to change in angle of elevation ($\delta\varphi$) is described by the elevation angular magnification, α, which is a constant (linear relationship)

As shown in Fig. 2.1, the two catadioptric systems were stacked in a vertical configuration. Ollis et al. (1999) analyzed possible physical configurations for a catadioptric vision system using dome-like mirrors and found that the vertical configuration employed in this work exhibited many desirable qualities. Firstly, the resolution of both catadioptric vision systems was found to be falling off at an approximately equal rate. Secondly, the angular field of view of both systems was found to match. Most importantly, this vertical configuration eased the search for the corresponding epipolar lines since corresponding points in two images taken in this vertical configuration would lie on the radial line which propagates from the centre to the rim of the mirror in the image in the same direction. These radial lines are actually epipolar lines. Additionally, it was found that the employed camera and mirror combination exhibited a unique property which maintains an approximately linear relationship between the radial distances (measured in pixels from the centre of the mirror in the image) and angles of elevation. This property was exploited by the camera calibration technique and will be described in the following sub-section.

2.3.2 Camera Calibration

Most calibration techniques for central (Mei and Rives, 2007; Scaramuzza et al., 2006) and non-central (Caglioti et al., 2007; Mičušík and Pajdla, 2004) catadioptric systems aim to extract the intrinsic and sometimes the extrinsic parameters of the catadioptric system. Common intrinsic parameters are such as the principal point, which describes the origin of coordinates in the image plane, and the focal length, which describes the distance between the camera centre and principal point along the principal axis. These two parameters are normally sufficient if a pin-hole camera model is assumed. Of course, there are more specific models such as the CCD camera model described in (Hartley and Zisserman, 2000), which takes into account the possibility of having non-square pixels and also the skewing

of the pixel elements in the CCD array. On the other hand, the extrinsic parameters of the camera describe the transformation of points in the scene expressed with respect to the world coordinate frame to the camera coordinate frame with its origin at the camera centre. Using this information, the camera projection matrix, which describes the transformation of 3D Euclidean points in space into image plane coordinates, can be derived. Camera calibration is vital in many robotics applications since it can be used to derive the 3D location of a given pair of corresponding pixel locations in two images in a monocular or binocular configuration. In addition, given a number of accurate correspondences, the fundamental matrix can be derived and be used as a constraint to filter out inaccurate corresponding pixel locations. For the case of central or non-central catadioptric systems, it is not as direct as compared to the case of calibrating just a standard perspective camera due to the inclusion of a dome-like mirror. In this case, the ray of light originating from a point in the scene within the field of view of the catadioptric vision system is not directly projected onto the camera's image plane but firstly reflects off the mirror before it reaches the image plane. As such, it is also vital to ensure the precision of the camera calibration process since it directly affects the accuracy of range measurements.

In contrast to these techniques, the calibration technique proposed here computes the discrete relationship between the angles of elevation of the mirror with respect to the radial distances measured from the centre of the mirror in the image. The proposed technique is dependent on the geometrical properties of the catadioptric system. This process requires an additional tool, which is a hollow metal cylinder containing a calibration grid with black and white checkerboard patterns covering its entire inner surface, referred to as the calibration bin (Fig. 2.3).

Figure 2.3: Calibration Bin

The calibration process begins with the placement of the catadioptric system inside the calibration bin, adhering to some assumptions. An image is taken from the catadioptric system in the current configuration that produces an image as shown in Fig. 2.4. Then, the user is required to select the points of all vertices of the checkerboard pattern in the image

manually. The final location of these points will be automatically refined by analyzing a small image neighbourhood based on the user defined pixel locations of the vertices. Then, the location of the selected points will be described in terms of an orientation and radial distance from the centre of the mirror in the image. Using the geometrical relationship between the calibration bin and the catadioptric system and the geometrical properties of the catadioptric system with respect to the mirror, it is then possible to derive the angles of elevation of the respective points. As such, the discrete relationship between the angles of elevation of the mirror with respect to radial distance can be computed since the respective points are measured in terms of radial distance from the mirror. This information will be conveniently stored into a lookup table.

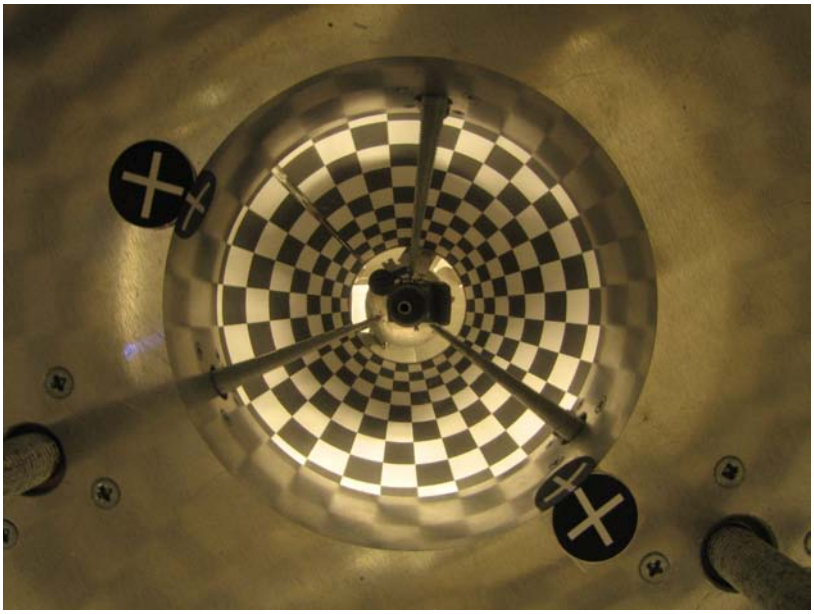


Figure 2.4: Calibration Image

The structure of the lookup table is clearly illustrated in Fig. 2.5, where each column of the radial distance array corresponds to the radial distance of a user selected point along the same orientation from the centre of the mirror and arranged in ascending order. Each element of the radial distance array is also associated with an angle of elevation. As such, whenever a pair of corresponding points from the stereo rig is provided to the system, its orientation and radial distance from the centre of the mirror are computed and the lookup table is used to retrieve the angles of elevation which correspond to the respective radial distance interval. Linear interpolation/extrapolation is then performed to compute the exact angles of elevation and is deemed suitable as it was found that the relationship between the angle of elevation and radial distance of the point in the image is approximately linear. This has been validated in Fig. 2.6. Subsequently, the 3D position of the stereo pair can be calculated by performing triangulation. Please refer to Appendix B for more details. This includes a detailed explanation on how the geometrical properties of the catadioptric system were used, assumptions made for the calibration process and analysis of the impact to the accuracy of the system due to the violation of these assumptions.

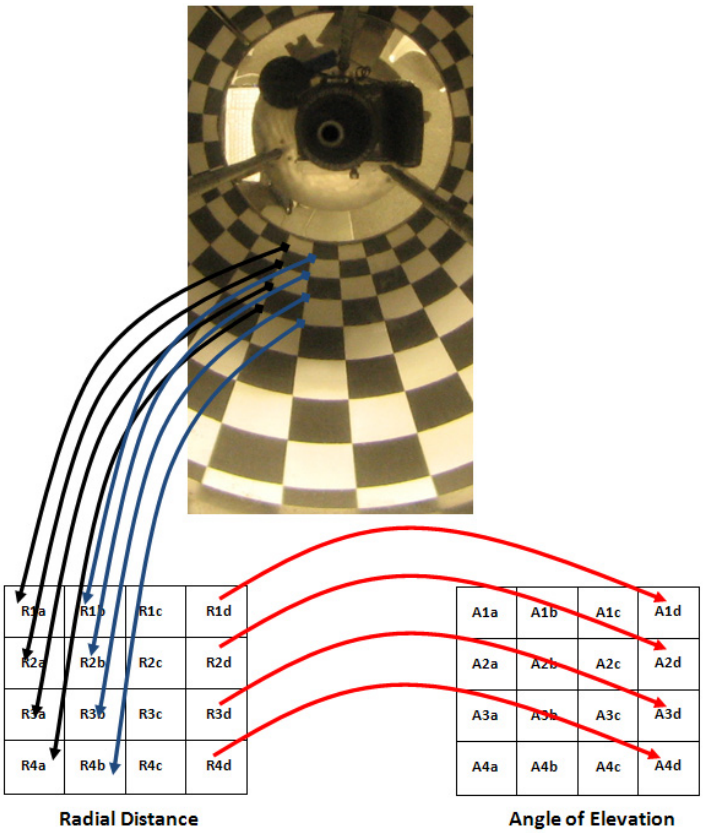


Figure 2.5: Lookup Table

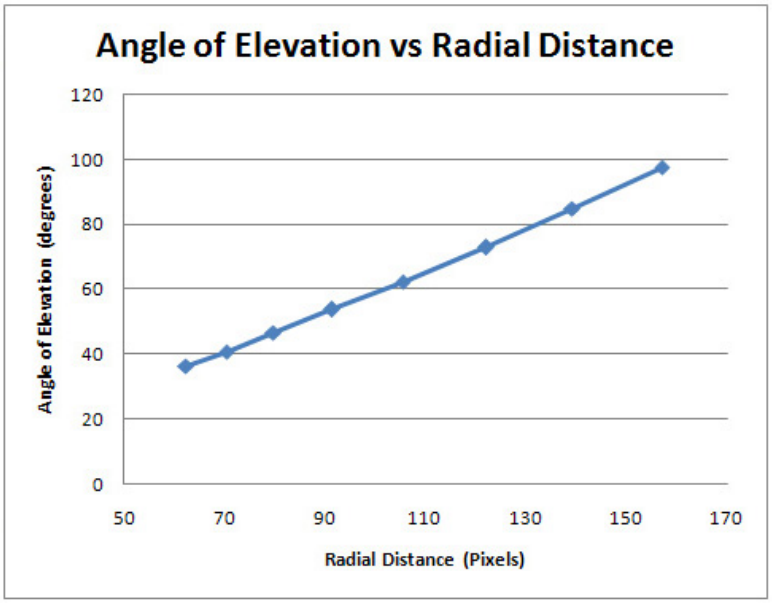


Figure 2.6: Equiangular Mirror - Angle of Elevation vs Radial Distance (using calibration data)

2.4 Stereovision

Stereovision has been an active research field for many years. This phenomenon, which describes the ability of humans to perceive depth through binocular vision, was first discovered and described by Wheatstone (1838). Stereovision techniques can be broadly categorized into feature-based or area-based approaches. In feature-based approaches, unique features extracted from the image such as distinctive edges or corners (Agrawal and Konolige, 2006), SIFT features (Lowe, 2004), etc, are matched against features extracted from the corresponding image. On the other hand, area-based or correlation-based approaches match pixel intensity patches. It is also one of the least computationally expensive techniques that produces a denser depth map (with respect to feature-based approaches), which is highly desirable for vision-based robot navigation and path planning purposes.

Area-based stereovision algorithms can be further classified into local and global approaches. As defined in (Scharstein et al., 2002), local algorithms only depend on intensity values within a finite window by making implicit smoothness assumptions whereas global algorithms solve an optimization problem by making explicit smoothness assumptions in the computation of disparity. A local, area-based stereovision algorithm was implemented in this project since local approaches were found to be less computationally expensive and were highly parallelizable. Of course, the common trade-off for lower computational workload is accuracy and in this case, less accurate disparity maps.

The most popular local, area-based approaches are based on either the computation of the sum of absolute differences (SAD), sum of squared differences (SSD), normalized cross correlation (NCC), rank or census. SAD is the most preferable approach when the stereovision algorithm runs purely on the processor of a laptop since it is relatively less computationally expensive. However, the total time difference between the computation of SAD and SSD scores was found to be negligible on a GPU-based implementation. As such, it seems to be more advantageous to compute SSD scores on a GPU-based implementation since global minimas are much more distinctive. However, although SSD is more discriminative than SAD, it is also more susceptible to changes in illumination and noise. This is also the reason why in the literature that SAD normally performs as well as SSD due to this tradeoff. It is also common to find that a mixture of constraints such as the epipolar constraint, continuity constraint, ordering constraint, left-right consistency constraint (Fua, 1991) or Single Matching Phase (Stefano et al., 2004) are applied in CPU based algorithms in order to reduce stereo ambiguities and produce higher quality depth maps. Unfortunately, it is difficult to comply with some of these constraints when the stereovision algorithm is parallelized. The implementation of this GPU-based stereovision algorithm only applied the epipolar constraint, continuity constraint, ordering constraint and a SSD score threshold.

In contrast to omnidirectional catadioptric vision systems, stereo correspondences were established after the captured image pair was rectified on conventional stereovision systems. However, on omnidirectional catadioptric vision systems, establishing stereo correspondences directly on the original image would require complicated image neighbourhoods for area-based methods. As described in Section 2.3.1, this problem could be avoided by

unwarping the original image into a rectangular image. The epipolar lines would then transform into columns of the rectangular image and correspondences could be searched for along the same column of the corresponding line.

The local, area-based stereovision algorithm was implemented on the GPU by utilizing the Nvidia CUDA libraries which are freely downloadable from the Nvidia CUDA website (Nvidia CUDA Zone, 2008). There are certainly different ways of parallelizing and optimizing the implementation of a stereovision algorithm on the GPU using Nvidia CUDA, but the implementation developed by Stam et al. (2008), which is also freely downloadable from the OpenVIDIA website, is highly optimized. However, the version of the Nvidia CUDA libraries used in this work was found to conflict with the ARIA libraries (library provided by MobileRobots to control the ActivMedia Pioneer Robot P3-AT) when used in a .Net environment. To resolve this issue, the wrapper class, CUDA.NET (GASS CUDA.NET, 2008), was used and necessary modifications were made to the original codes provided by Stam et al. (2008) to ensure the compatibility with the CUDA.NET wrapper and also with the unwarped omnidirectional image.

Once stereo correspondences were established, the information stored in the lookup table (as described in Section 2.3.2) and the effective baseline could be used to perform triangulation. The resulting 3D position of a scene point would be described in the form of (x_p, y_p, θ_e), where θ_e describes the direction of the epipolar line in the original image and, x_p and y_p is the coordinate on the plane (see Fig. 2.7) defined by the actual location of the stereo pair on the mirror. (x_p, y_p, θ_e) could then be converted into Cartesian coordinates (x, y, z) by applying simple trigonometric relationships. As a sidenote, the linear interpolation/extrapolation of values from the lookup table and the triangulation process have also been parallelized using the Nvidia CUDA libraries. For completeness, the GPU-based stereovision algorithm is described in the following sub-section.

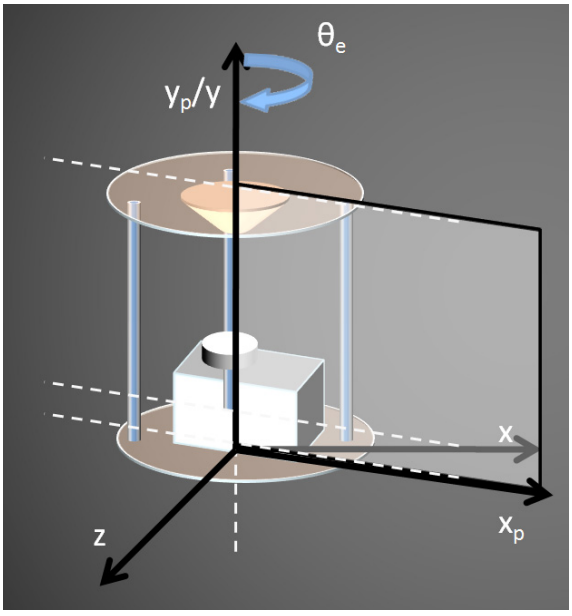


Figure 2.7: Epipolar Plane

2.4.1 The GPU-based Omnidirectional Catadioptric Stereovision Algorithm

As discussed previously, a local area-based technique based on SSD was used to search for the best matching local window in the corresponding image along the epipolar line. Since we employed Stam et al. (2008)'s technique to establish stereo correspondences, we will briefly summarize this algorithm and then proceed with the description of our proposed algorithm, which also utilizes the GPU for the computation of the 3D positions from the respective stereo correspondences via triangulation.

a Establishing Stereo Correspondences

The stereo pair is read using OpenCV functions. Then, CUDA.NET functions are called to initialize the GPU. These images are unwarped before it is transferred from host (CPU) memory into the texture memory of the GPU. As described in (Stam et al., 2008), storing the images in texture memory allows fast, cached access, boundary clamping, bilinear interpolation for subpixel accuracy and the flexibility to change the image source data type without the need for reoptimization. However, texture memory has to be initialized beforehand and does not provide write access to the kernels. In short, this means that during the search for stereo correspondences, the threads can only read from but cannot write to texture memory. Intuitively, we should also try to minimize global memory access as much as possible since it is slow. However, global memory access is inevitable at the end of the algorithm's execution since results have to be stored into global memory before it can be transferred back to host memory. Therefore, the algorithm has to be carefully designed and planned such that global memory access is minimized and the limited but fast local shared memory is properly utilized.

Fig. 2.8 illustrates the search for stereo correspondences using a 5x5 correlation mask in a stereo pair at a resolution of 20x10. Let's assume in this example that *image1* is the reference image and *image2* is the corresponding image. In order to find the corresponding location of pixel 0, we will utilize the correlation mask bounded by the blue rectangle and compute SSD values using the pixel intensities of this correlation mask with the correlation mask in the corresponding image at the same location of pixel 0 with an offset of d (disparity). Depending on the disparity step size and maximum disparity search range, this process will be executed (maximum disparity search range)/(disparity search step size) number of times. The disparity which yields the minimum SSD value will then be denoted as d_{min}. The same process is performed for pixels 1,2.. and subsequently to pixels on the next row until all pixels in the image are processed.

As illustrated, the standard local area-based technique is very inefficient since it involves a large number of redundant calculations. In order to improve the efficiency of these algorithms, redundant calculations can be significantly reduced by using a rolling window scheme such as the one described in (Stefano et al., 2004) on a CPU implementation. Nevertheless, it is still quite difficult to achieve real time speeds on a laptop with higher resolution images. This is where the GPU comes into the picture. The chosen approach minimizes redundant calculations and computes SSD values for a local window

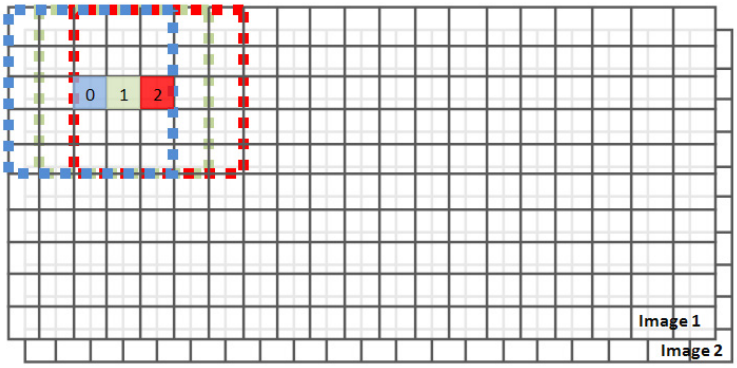


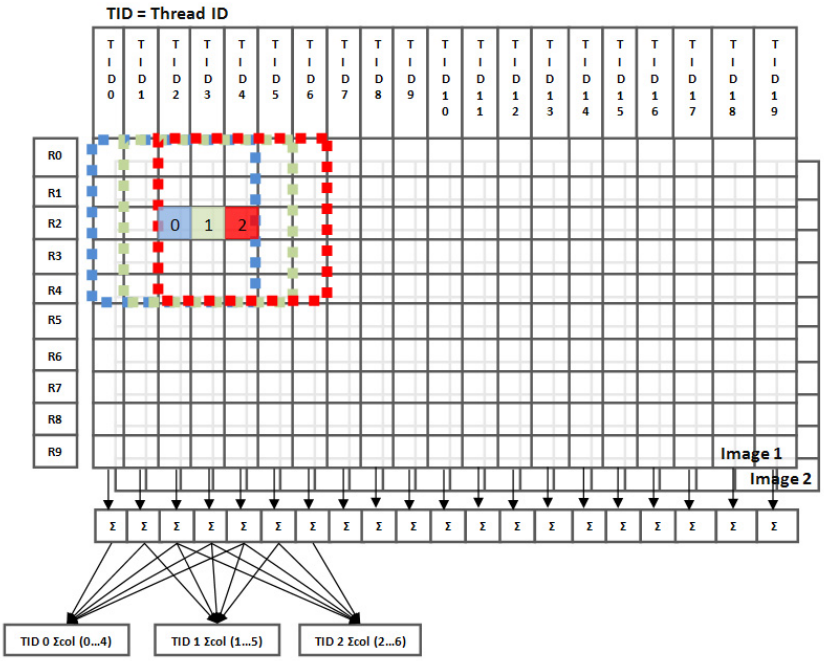
Figure 2.8: Search for stereo correspondences using 5x5 correlation mask

in parallel. The overall scheme is illustrated in Fig. 2.9. Similarly, a 5x5 correlation mask and 20x10 stereo pair are used in this example. Thread blocks are initialized to a size of 16x1 (denoted by the thread IDs 0,1,...,15). For the first row as illustrated in Fig 2.9(a), each thread computes and accumulates the squared differences of a column of pixels the height of the local window with respect to the pixel in the corresponding image with a disparity offset of d. The sum of squared differences for each column is stored in local shared memory. Once all computations are completed for the first row, each thread will then sum up the accumulated column SSD values from the neighbouring columns within the width of the local window in order to determine the SSD value for the local window of the respective pixel. This value is then tested with the current minimum value stored in global memory (which is initialized to a large number) in order to determine whether the current disparity value is the best correspondence match for the current pixel.

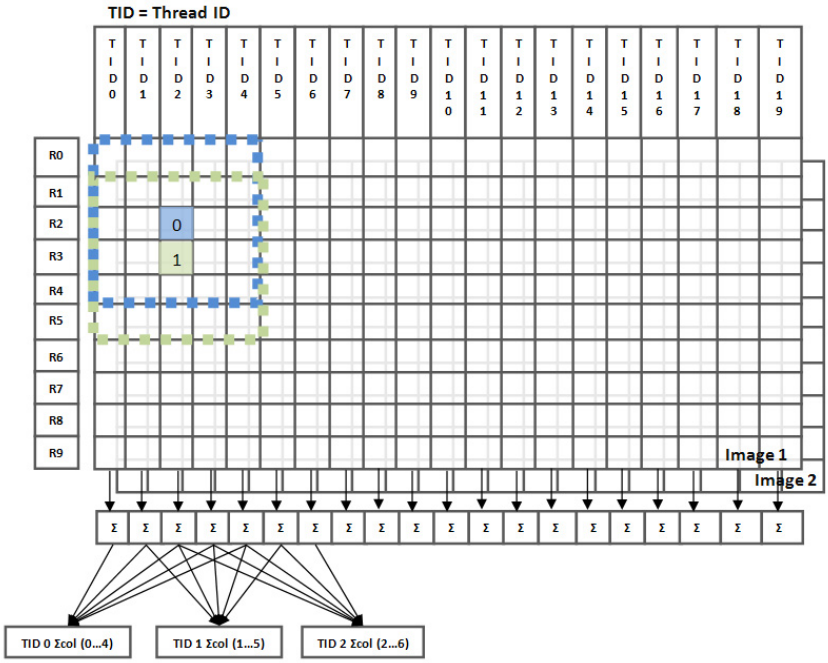
After the first row is processed, the rolling window scheme is applied. As illustrated in Fig. 2.9(b), there is significant overlap between the correlation mask of pixel 1 in row *R3* with the correlation mask of pixel 0 in row *R2*. Therefore, each thread can compute the new column SSD value by subtracting the corresponding squared difference value of the corresponding pixel in row *R0* and subsequently adding the new squared difference value of the corresponding pixel in row *R4*. This rolling window computation continues until SSD is computed for all rows allocated to a thread. The entire process is then repeated (maximum disparity search range)/(disparity search step size) number of times with the disparity value, d, incremented by the value of the disparity search step size at the end of each iteration. The final disparity values which reside in global memory are subsequently transferred back to host (CPU) memory.

b Triangulation

Once stereo correspondences are established, the disparity value associated with each pixel location of the reference image can be used to compute its respective 3D position by utilizing the lookup table created by the camera calibration process described in Section 2.3.2. With this lookup table, it is then possible to translate a 2D location in the omnidirectional image (measured in terms of radial distance from the centre of the mirror in the image) to the angle of elevation on the surface of the equiangular mirror. Thus, once the pixel



(a) Processing the first row



(b) Processing subsequent rows

Figure 2.9: Establishing stereo correspondences using the GPU

locations of the stereo correspondences are converted into radial distances from the centre of the mirror in the image, triangulation is performed in order to compute its respective 3D position.

This operation turns out highly suitable for the GPU since the same operation is required for each disparity value in the disparity image returned by the GPU-based stereo correspondence algorithm. In fact, as compared to the stereo correspondence algorithm,

this algorithm is relatively straightforward. Similarly, once the GPU is initialized by CUDA.NET, the disparity image is transferred to the GPU and stored into its texture memory. Since the information in the lookup table is used very frequently, it is also stored into texture memory. Depending on the image size, we initialize n number of thread blocks. As illustrated in Fig. 2.10, which is a continuation of the previous example in Fig. 2.9 (16x6 disparity image computed), 16x1 thread blocks are initialized and each thread block converts stereo correspondences of those pixels in the non-overlapping regions in the disparity image into radial distances, uses the lookup table in order to perform triangulation and stores the results into global memory. Please be aware that this is just a simple example which can be easily extended to a larger image size and works optimally when the thread block size is 64 or 128 x 1.

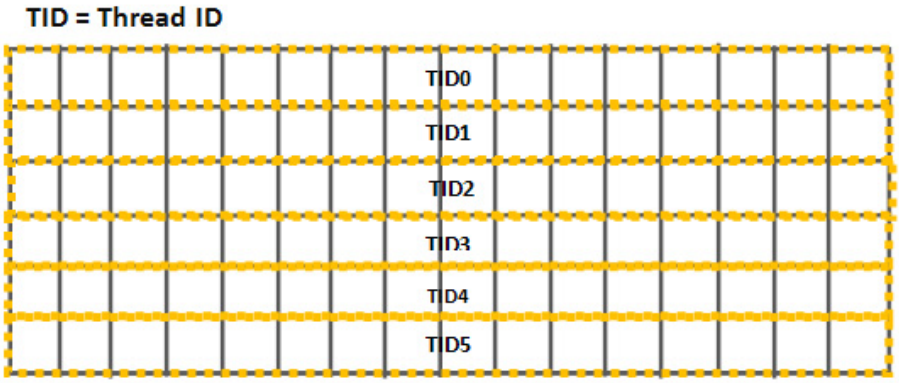


Figure 2.10: Triangulation via GPU

2.5 Results

In this section, results illustrating the performance of the proposed GPU-based stereovision algorithm, evaluated with different stereo parameters such as image size, correlation mask size, maximum disparity search range and disparity search step size, will be provided. This is then followed by the illustration of the resulting disparity maps produced by a single stereo pair taken from different semi-outdoor and outdoor environments. By using these stereo correspondences, triangulation is performed by utilizing the information in the lookup table. The 3D positions of these stereo correspondences will be illustrated in a 3D voxelized environment using OpenGL and the accuracy of the proposed omnidirectional stereovision system will be provided.

Referring to Fig. 2.11, the changes in correlation mask size from 15x15 to 41x41 had negligible effect on the total time required to compute the disparity map. However, the maximum disparity search range, disparity step size and image size did have a significant impact on the overall performance. On a 640 x 480 image, real time performance of approximately 50fps could be achieved using a maximum disparity search range of 32 pixels and disparity step size of 1 pixel. With this parallelized implementation, higher resolution images (1600 x 1200) could be used for stereovision. Although these higher

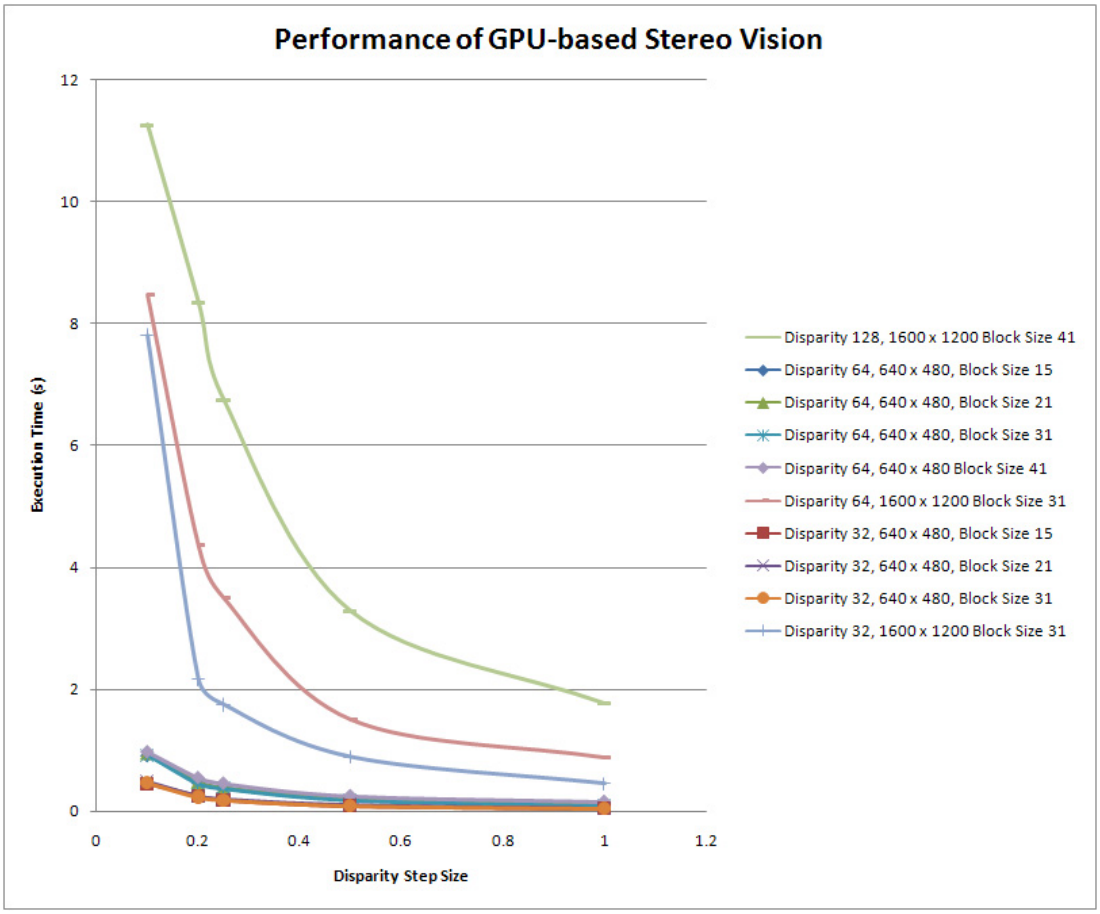


Figure 2.11: Stereo Performance

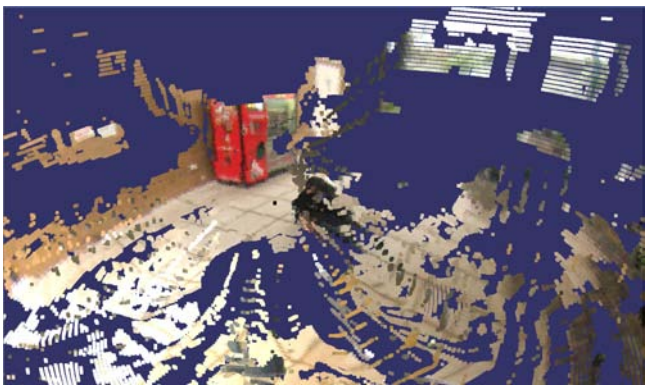
resolution images could not be processed in real time (≈1.2fps with a maximum disparity search range of 64 pixels and a disparity step size of 1 pixel or ≈0.305fps with a maximum disparity search range of 128 pixels and a disparity step size of 0.5 pixel) but it might be worthwhile spending this extra time in some cases (i.e. in certain natural environments when distinct features are otherwise too small). Subsequently, the system could adjust itself accordingly to use an image size of 640 x 480 when high resolution images were not required.

Experimental images were collected from various semi-outdoor and outdoor environments and the voxelized environments generated by the system are illustrated in Fig. 2.12-2.17. For each environment, the stereo pair (images captured by the top and bottom catadioptric vision system) and the corresponding disparity map are illustrated and this is followed by the rendered 3D voxelized environment captured from certain viewpoints which contain features that are easily recognizable from the stereo images. Results illustrated in Fig. 2.12-2.15 were generated by using unwarped images at a resolution of 1269 x 202 (original image size: 640 x 480) whereas Fig. 2.16-2.17 illustrate the resulting disparity maps and rendered 3D voxelized environments generated by using unwarped images at resolutions of 3185 x 507 (original image size: 1600 x 1200). The respective stereo parameters used are highlighted in the captions of each figure and the time required to process each stereo pair can be found using the stereo performance graph in Fig. 2.11.

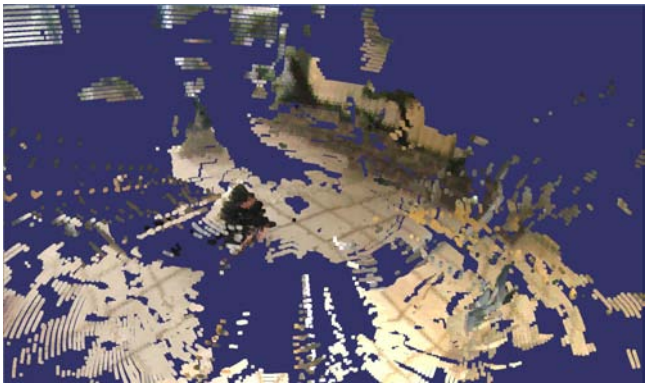
(a) Bottom Image

(b) Top Image

(c) Disparity Image



(d) Voxelized Environment 1



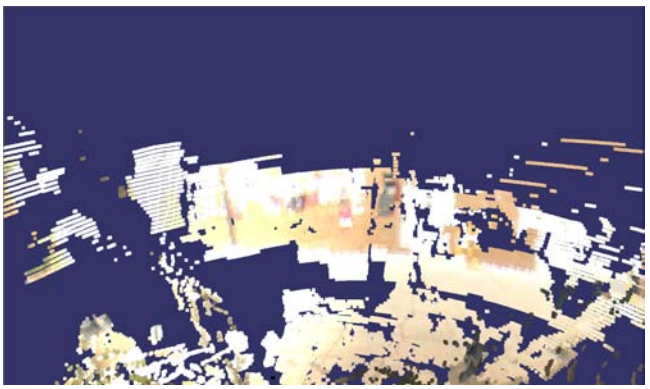
(e) Voxelized Environment 2

Figure 2.12: Stereovision Results in Semi-Outdoor Environment 1 - 30cm baseline, 15x15 correlation mask, max. disparity search range 64, step size 0.5

(a) Bottom Image

(b) Top Image

(c) Disparity Image



(d) Voxelized Environment 1

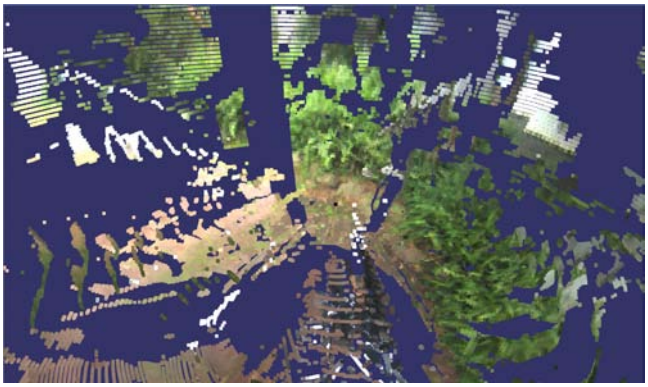
(e) Voxelized Environment 2

Figure 2.13: Stereovision Results in Semi-Outdoor Environment 2 - 30cm baseline, 15x15 correlation mask, max. disparity search range 64

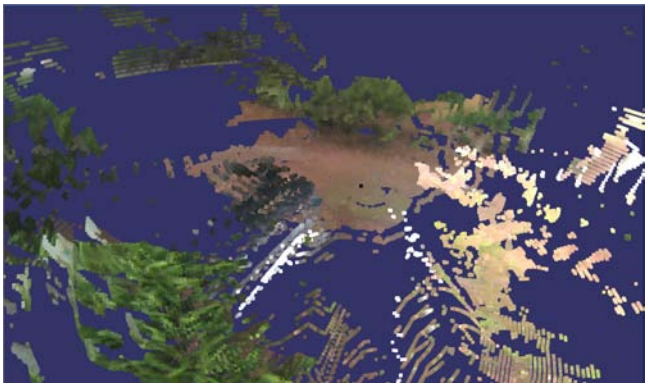
(a) Bottom Image

(b) Top Image

(c) Disparity Image



(d) Voxelized Environment 1

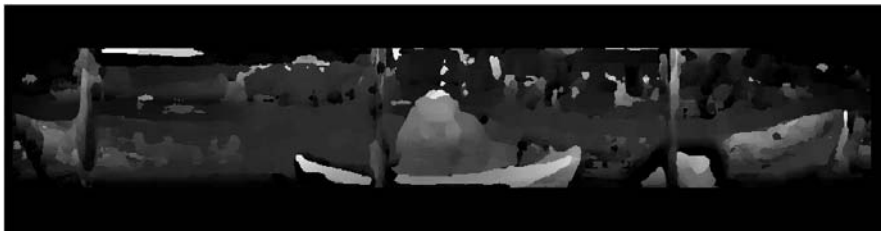


(e) Voxelized Environment 2

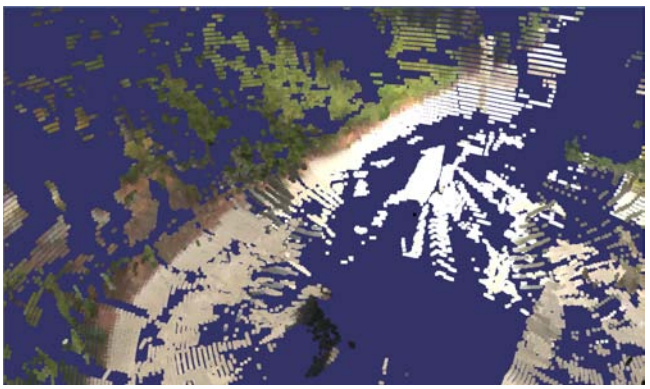
Figure 2.14: Stereovision Results in Outdoor Environment 1 - 20cm baseline, 15x15 correlation mask, max. disparity search range 64, step size 0.5

(a) Bottom Image

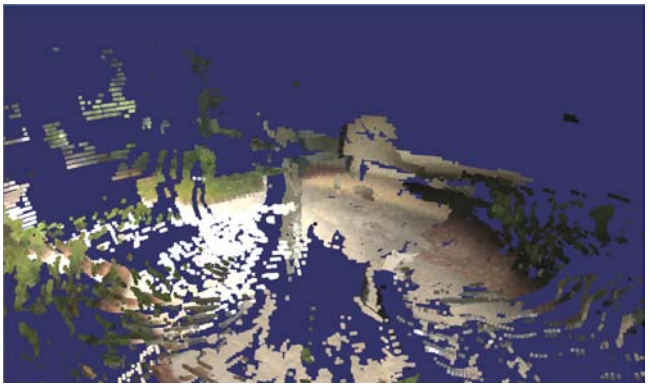
(b) Top Image



(c) Disparity Image



(d) Voxelized Environment 1



(e) Voxelized Environment 2

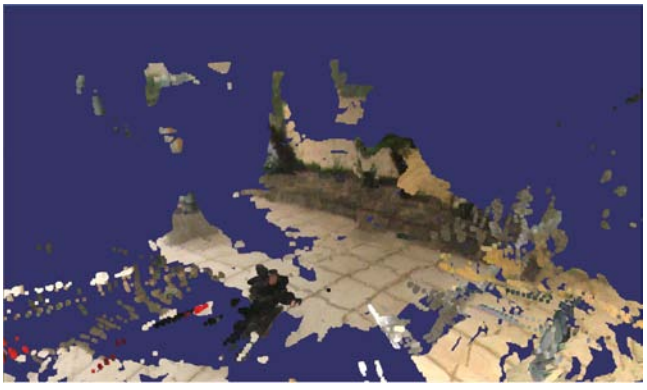
Figure 2.15: Stereovision Results in Outdoor Environment 2 - 30cm baseline, 15x15 correlation mask, max. disparity search range 64, step size 0.5

(a) Bottom Image

(b) Top Image

(c) Disparity Image

(d) Voxelized Environment 1

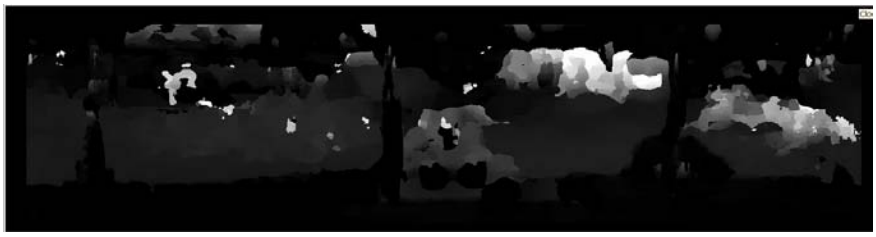


(e) Voxelized Environment 2

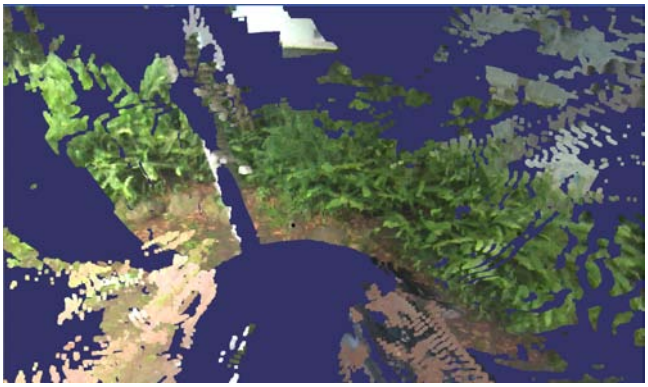
Figure 2.16: Stereovision Results in Semi-Outdoor Environment (High Res.) - 30cm baseline, 41x41 correlation mask, max. disparity search range 128, step size 0.5, 1600 x 1200 image size

(a) Bottom Image

(b) Top Image



(c) Disparity Image



(d) Voxelized Environment 1

(e) Voxelized Environment 2

Figure 2.17: Stereovision Results in Outdoor Environment (High Res.) - 20cm baseline, 41x41 correlation mask, max. disparity search range 128, step size 0.5, 1600 x 1200 image size

2.6 Discussion

As illustrated in the experimental results, a fixed disparity search range of 64 pixels was used for the lower resolution unwarped images (1269 x 202), whereas this disparity search range was increased to 128 pixels for the higher resolution unwarped images (3185 x 507). In these experiments, the stereo system was separated by a vertical baseline and the specified disparity search range of 64 pixels for the lower resolution image and 128 pixels for the higher resolution image would cover 32% of the entire possible search range of 202 pixels and 25% of the entire possible search range of 507 pixels respectively. Under such circumstances, the system would probably fail in correlating pixels belonging to objects lying too close to the platform. To avoid colliding into these obstacles, the mobile robot will be equipped with a short range reactive obstacle avoidance system that utilizes the forward facing Bumblebee stereovision system. A 0.5 disparity step size was chosen because, as reported in Gallup et al. (2008), it was found that a 0.5 disparity step size is adequate since the image is band-limited by resolution.

As mentioned in Section 2.4, a GPU version of a local area-based stereovision algorithm using SSD scores was implemented. No doubt that this is by far one of the most cost efficient methods to establish stereo correspondences but there is still room for improvement. One of the possible improvements is to increase the quality of the dense depth map without amplifying the computational time required to an unacceptable level. In comparison to recent works related to general stereovision using catadioptric systems, Arican and Frossard (2007) and He et al. (2007) made use of the global energy minimization algorithm based on graph-cut techniques. Notably in (Arican and Frossard, 2007), the dense disparity maps produced using the proposed graph cut technique are more accurate as compared to the disparity maps produced by our system (based on local area-based methods). As discussed previously, the main cost for the improved accuracy using global algorithms is higher computational workload. A list of the current state-of-the-art local and global stereovision algorithms is available in (Scharstein and Szeliski, 2008). One of the previous best performing algorithms on the Middlebury dataset, based on segment-based methods and belief propagation, was proposed by Klaus et al. (2006). Belief propagation is currently a very popular technique but, unfortunately, many of its variants are too costly except for the GPU-based hierarchical belief propagation algorithm proposed and implemented by Yang et al. (2006), which was reported to run at 16fps for a 320 x 240 image with 16 disparity levels. There are also less complex and advance local area-based algorithms which can provide competitive results. The tricky part is to strike a balance between computational workload and accuracy. Since it is now relatively easy to tap into the power of the GPUs, researchers should ensure that their proposed algorithms are highly parallelizable such that even complex algorithms can be executed in real time. With better quality depth maps, it will then be possible to further reduce artifacts in the voxelized environment.

The error of the estimated 3D position of scene points can be found in Fig. 2.18. Stereo correspondences were manually selected in the image and their estimated depths were compared against ground truth. This process was repeated three times such that an

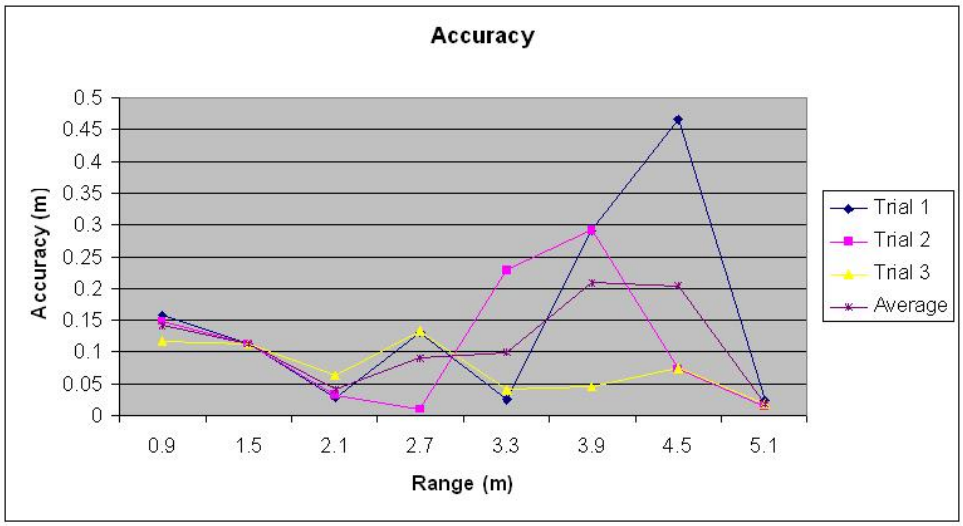


Figure 2.18: Accuracy of estimated 3D positions of scene points for stereo pairs with an effective baseline of 30 cm.

average error could be computed. The accuracy of the estimated 3D positions is affected by both the accuracy of the manually selected stereo correspondences and the information provided by the lookup table. This graph is merely a rough approximation of the accuracy of the omnidirectional system at an effective baseline of 30cm and can be further improved by providing more accurate geometrical values of the system and points on the calibration grid during the calibration process, and manually selected stereo correspondences with subpixel accuracy.

2.7 Chapter Summary

This chapter introduced a GPU-based omnidirectional stereovision system. By exploiting the characteristics and properties of the equiangular mirror and the geometry of the catadioptric system, a new camera calibration technique was proposed. This camera calibration technique computes the relationship between the radial distance from the centre of the mirror with respect to the angle of elevation on the mirror and stores this information into a lookup table. Subsequently, stereo correspondences are established via a local area-based method on the GPU. The entire GPU algorithm, which includes the establishment of stereo correspondences to the computation of 3D locations of these correspondences via triangulation using the information in the lookup table, have been thoroughly described. Additionally, the performance and accuracy of the proposed omnidirectional stereovision system have been clearly illustrated and discussed.

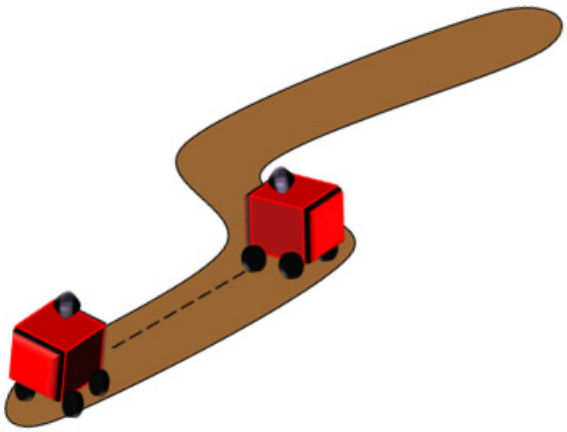
3

Multibaseline Omnidirectional Stereovision

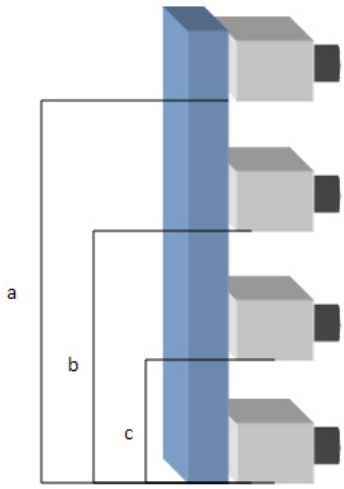
3.1 Introduction

As described in the previous chapter, stereovision enables depth perception of the environment through binocular vision. However, it must be noted that ambiguities in stereo correspondences are difficult to resolve in environments where repetitive patterns such as a picket fence, low textured areas such as the facade of man-made buildings and pavements are present, or when objects in the scene are occluded. There are many models which try to implicitly or explicitly address these problems and results so far are of mixed success. A multibaseline stereovision system attempts to reduce stereo ambiguities implicitly by combining results obtained from stereo pairs produced at different baselines either in the stage of the computation of disparity or in the post disparity computation stage where 3D positions of the scene points are obtained from triangulation.

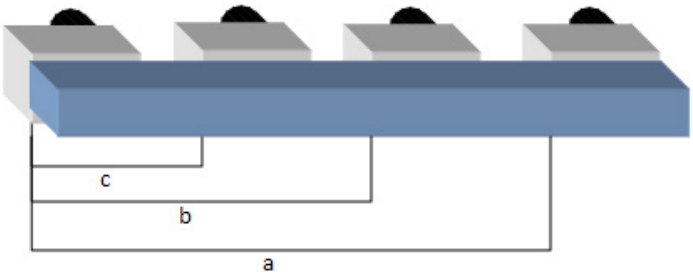
Multibaseline stereovision systems can be configured in several ways. Generally, existing systems in the literature closely resemble one of those illustrated in Fig. 3.1. Fig. 3.1(a) shows a single camera mounted on a mobile platform. Its effective baseline is highly flexible and is only constrained by the environment. However, using such a configuration requires robust and accurate tracking of the camera's location and orientation in order to derive the effective baseline which is vital to the accuracy of 3D positions calculated based on corresponding 2D image feature locations. In Fig. 3.1(b) and 3.1(c), a single/multi camera setup with a variable baseline constrained within a 1D vertical/horizontal line motion, where the values a, b and c is proportional (i.e. $a = k$, $b = 2k$ and $c = 3k$) or just some arbitrary values which satisfies the condition, $a > b > c$, are illustrated. Lastly, Fig. 3.1(d) illustrates a fixed multi-camera system which also facilitates the combination of multiple stereo pairs by exploiting the fixed geometry between the multiple cameras. The 2D arrangement of the fixed multi-camera system has the benefit of being able to detect both horizontal and vertical features despite it having a shorter baseline as compared to those illustrated in Fig. 3.1(b) and 3.1(c).



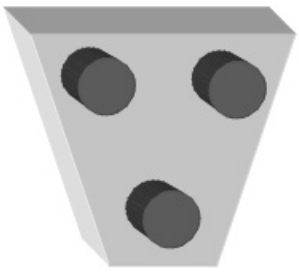
(a) Single Camera Variable Baseline



(b) Single / Multi Camera with a Variable Vertical Baseline (1D)



(c) Single / Multi Camera with a Variable Horizontal Baseline (1D)



(d) Multicamera with Fixed Baselines

Figure 3.1: Multibaseline Stereovision Setup

Ideally, depth estimates calculated via triangulation are most accurate with a wide/long baseline. However, if the separation between the two cameras is too wide, features in one camera might be occluded in the other camera and in some extreme cases, there might be no overlap between the viewpoints of both cameras (known as the missing parts problem). On the other hand, a narrow/short baseline makes the search for correspondences easier but may not produce reliable depth estimates. Consequently, even if the single camera configuration illustrated in 3.1(c) can theoretically have a very wide/long baseline since it is only constrained by the environment, it has to ensure that there is sufficient overlap between the viewpoints of the camera such that decent correspondences can be established to produce useful depth estimates. As illustrated in Chapter 2, our system features a variable vertical baseline with the mirror pointing downwards (covering a great deal of ground area). This configuration maintains the overlapping of crucial regions surrounding the mobile robot even when a wide/long baseline is used. The missing parts due to this wider/longer separation between the cameras are taller objects such as tree tops and higher levels of buildings, which does not concern most ground mobile robots.

3.2 Combining Multiple Stereo Pairs

A popular approach to combine multiple stereo pairs is the SSSD-in-inverse-distance proposed by Okutomi and Kanade (1993). In standard local area-based stereovision algorithms, the SSD scores are represented with respect to disparity. However, the Okutomi-Kanade system represents SSD scores with respect to the inverse distance/depth. By using this representation, SSD scores computed for different stereo pairs based on the same reference image can be easily integrated by summation. This approach was further extended by Sato et al. (2002) to develop detailed 3D models of outdoor scenes by utilizing the SSD's median value to avoid occlusion and a multiscale approach to improve depth estimation. The downside for the creation of such detailed 3D models is the requirement of hundreds of dense depth maps. Eventually, this translates to high computational requirements. Of course, in the context of mobile robot navigation, such detailed models are impressive but not a necessity. In addition, a more efficient implementation of the original SSSD-in-inverse-distance is available in (Kang et al., 1995) where active illumination was used to improve depth estimation on its multi-camera rig.

On the contrary, Collins (1996) defined that a true multi-image matching technique should satisfy the following 3 conditions; (a) The method generalizes to any number of images greater than 2, (b) the algorithmic complexity is *O(n)* in the number of images, and (c) all images are treated equally. According to the definition provided by Collins (1996), the Okutomi-Kanade system has violated the last condition since it requires the same reference image / base image to be used for all stereo pairs. As a result, salient features present in the other images but occluded in the reference image would not be detected and accounted for using this approach. Based on these 3 conditions, Collins proposed a multibaseline stereovision algorithm based on the space-sweep approach. The idea is to back project features from each image onto successive positions of a plane sweeping through space and concurrently establishing 2D image correspondences and triangulating the 3D positions of the feature points in the scene. This approach was further extended by Yang and Pollefeys (2003) such that dense depth estimations could be achieved in real-time on a standard stereo camera configuration as opposed to the original idea where only the depth of sparse features were estimated. To resolve issues with slanted surfaces in the original algorithm, Gallup et al. (2007) proposed the use of multiple plane sweeping directions.

The approach adopted in our system is based on voxel-voting. As defined by Dyer (2001), a voxel is a volume in Euclidean 3D space to represent a regular tessellation of cubes. This volumetric model can be subsequently used to construct a polygonal surface representation (mesh model) by applying the Marching Cubes algorithm (Lorensen and Cline, 1987). This enables textures to be mapped on top of the mesh to create a photo-realistic representation of the environment (Se and Jasiobedzki, 2006). Nevertheless, we will not discuss this in detail since it is not our intention to reconstruct the environment with such fine details. Our system only represents the surrounding environment using these volumetric models via a voting process. This can be realized in two ways; (a) take an image as the reference image and (b) treat all images equally. For the former approach,

a possible implementation is to firstly compute the disparity maps for the stereo pairs taken at different baselines which maintain the same reference image. This is followed by the triangulation of the resulting stereo correspondences for the estimation of its 3D positions. In this case, the disparity of the same pixel for stereo pairs at different baselines should translate to the same 3D position. By enforcing this condition, ambiguous/false stereo matches can be reduced. The confidence level computed for each voxel can then be used to represent obstacles probabilistically similar to the Occupancy Grids (Elfes, 1989) framework. As for the latter approach which treats all images equally, a naive implementation is to take each image as the reference image in turns or to apply the space-sweep approach, which is only suitable for central catadioptric systems. However, the space sweep approach, which requires the back projection of point features from each image onto the sweeping plane, is not as direct when omnidirectional images were used. Moreover, there will be issues with computational efficiency when a denser depth map is required.

The first approach was chosen even though it was clear that other images would not be treated equally and occluded features in the reference image would not be taken into consideration. This decision was justified through the observation that occlusion would only become an issue for objects located very close to the robot when comparing images taken from the minimum and maximum baselines of the system described in Chapter 2. These objects would be detected and avoided by using the Bumblebee stereovision system (Point Grey, 2010). The other drawback of the voxel-voting approach based on a fixed reference image would be its intensive use of memory. The amount of memory required would depend on the resolution of the voxel. Therefore, to reduce the amount of memory required, this voxel-voting approach was implemented using hash tables. The intention of this variable multibaseline omnidirectional stereovision system is not to create highly accurate and detailed 3D models of the environment but to represent the surrounding environment reasonably accurate such that the mobile robot can plan its path and refines it accordingly during navigation. Before the results are presented, the following section will describe the proposed automatic baseline selection technique.

3.3 Automatic Baseline Selection

It is well understood that a lower disparity value in the disparity map indicates that the scene point is located relatively further away from the system as compared to those of a higher disparity value. As the distance/depth of the scene point increases, the baseline between the two cameras should also increase accordingly to provide more accurate depth estimates. It was also due to this reason that Nakabo et al. (2005) proposed to use a high speed linear slider to vary the effective baseline of the stereo cameras depending on the depth of the target object to be tracked in the scene. However, a wide/long baseline will make the search for correspondences a more difficult task when the feature is not tracked from frame to frame as the baseline increases. As a result, this produces many false matches. This is also one of the main reasons for using a multibaseline stereovision system since it helps in reducing ambiguity in stereo correspondences when a wide/long baseline

is used. However, the system has to decide which stereo pairs (all of different baselines) to utilize. Eventually, this motivated the development of the proposed automatic baseline selection technique. The proposed automatic baseline selection technique was developed based on the characteristics of the employed equiangular mirror and the histogram of the disparity distribution for a stereo pair separated by baseline b. We will firstly present findings of the equiangular mirror's characteristics and subsequently describing how this information can be used to analyze the histogram of the disparity distribution.

Assume that the point of interest in the omnidirectional image taken from the lower catadioptric system is located 200 pixels from the centre of the mirror in the image on the epipolar line at angle θ_e. Since this is a stereovision system separated by a vertical baseline, the only possible location of the corresponding point in the omnidirectional image taken from the upper catadioptric system must be less than or equivalent to 200 pixels measured from the centre of the mirror in the image on the same epipolar line at angle θ_e. Based on the lookup table computed in Chapter 2 and the coordinate systems illustrated in Fig. 2.7, the relationship between x_p and y_p with respect to possible locations of the corresponding point in the omnidirectional image taken from the upper catadioptric system with a baseline of 30cm and θ_e at 45^o, 135^o, 225^o and 315^o can be approximated. These plots are illustrated in Fig. 3.2 and 3.3.

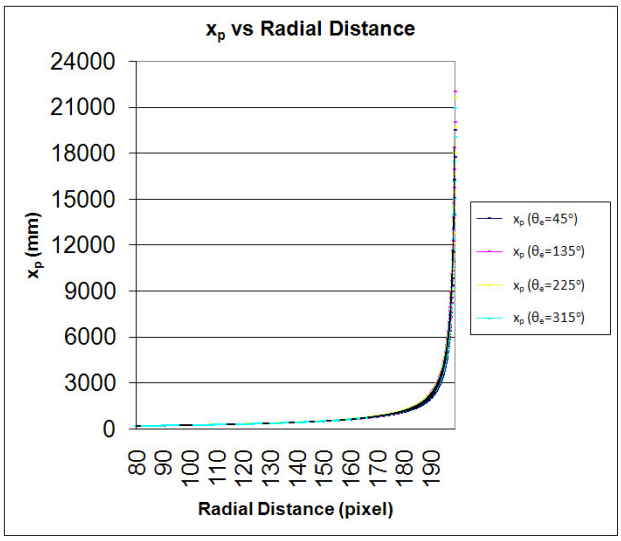


Figure 3.2: Plot of x_p vs Radial Distance (30cm baseline with interest point located 200 pixels from the centre of the mirror in the image)

Referring to the graphs in Fig. 3.2 and 3.3, it is clearly shown that as the radial distance (measured from the centre of mirror in the image) of the corresponding point gets closer to the location of the point of interest (in this case, a radial distance of 200 pixels), the value of x_p increases drastically. By using this information, minimum disparity thresholds could be determined and enforced such that a small error in the disparity would not translate to large errors in depth estimates. Subsequently, the graphs in Fig. 3.4 and 3.5 show the relationship between x_p and y_p with respect to possible locations of the corresponding point in the omnidirectional image taken from the upper catadioptric

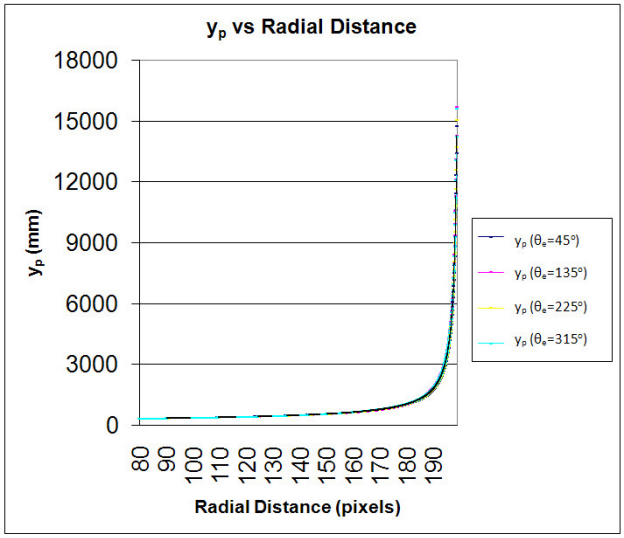


Figure 3.3: Plot of y_p vs Radial Distance (30cm baseline with interest point located 200 pixels from the centre of the mirror in the image)

system with baselines of 30cm, 50cm and 100cm, θ_e at 45^o and with locations of the point of interest in the omnidirectional image of the lower catadioptric system at 200, 175, 150, 125 and 100 pixels from the centre of the mirror in the image. In Fig. 3.4 and 3.5, all curves were derived from the angles of elevation on the epipolar line at $\theta_e = 45^o$ since it was found that the plots produced by θ_e at different angles were not significantly different (illustrated in Fig. 3.2 and 3.3). In addition, the graphs in Fig. 3.4 and 3.5 also show that a higher minimum disparity threshold could be selected for wider/longer baselines.

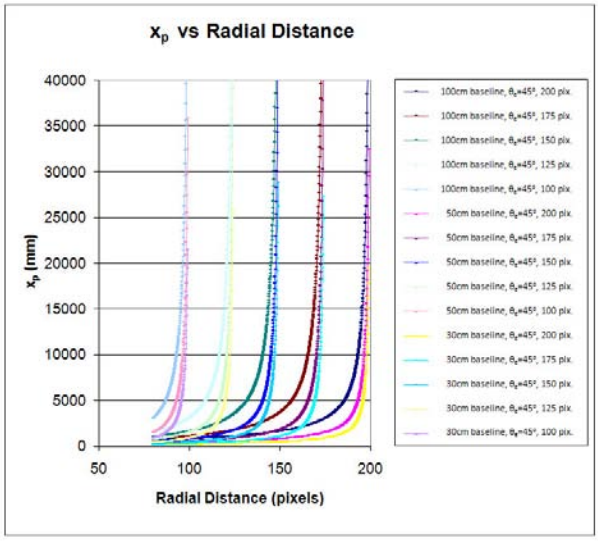


Figure 3.4: Plot of x_p vs Radial Distance at different baselines and locations of point of interest

To determine the minimum disparity thresholds for different baselines, the maximum change in x_p or y_p with respect to the change in radial distance has to be specified. For this system, the maximum change of x_p was specified to fall less than 250mm with respect to a

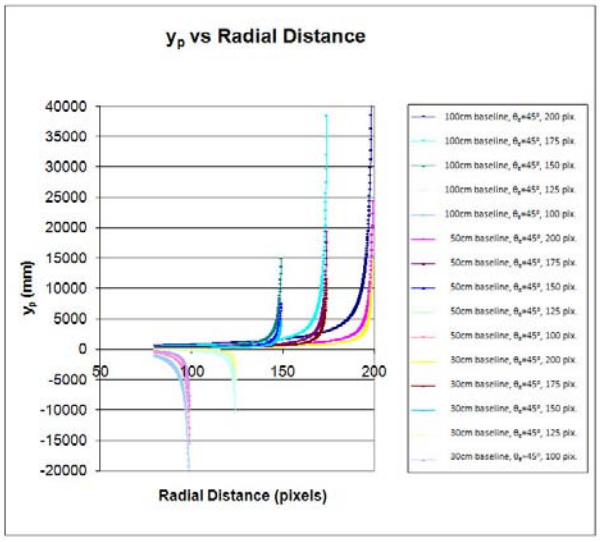


Figure 3.5: Plot of y_p vs Radial Distance at different baselines and locations of point of interest

change of 0.5 pixels in radial distance. This selection was convenient because the disparity step size for the GPU-based stereovision algorithm was also set at 0.5 pixels. Assuming no false matches, the maximum error due to the subpixel accuracy resolution would be contained within 250mm (this upper error bound is normally applicable to the far range whereas the near range error bound is much lower than specified). A 0.5 disparity step size was chosen because in (Szeliski and Scharstein, 2004), it was found that a 0.5 disparity step size is adequate since the image is band-limited by resolution. The relationship between the minimum disparity thresholds for different baselines and the resulting change of x_p and y_p with respect to a change of 0.5 pixels in radial distance at this minimum disparity threshold are illustrated in Fig. 3.6 and 3.7.

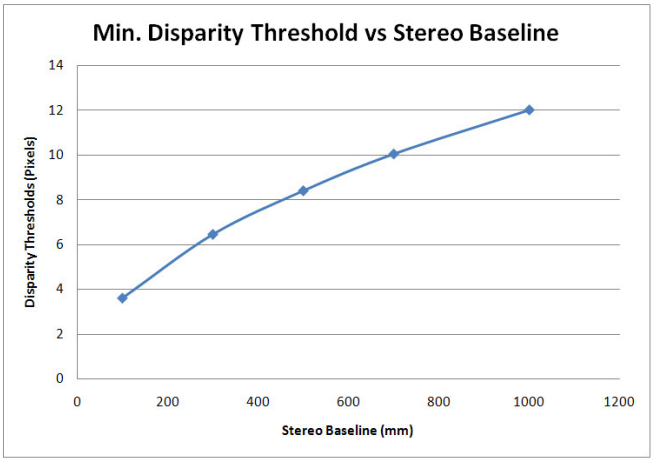


Figure 3.6: Plot of Minimum Disparity Threshold vs Stereo Baseline

As described previously, the histogram of the disparity distribution would be utilized for the baseline selection process. For a maximum disparity search range of 64 pixels, the

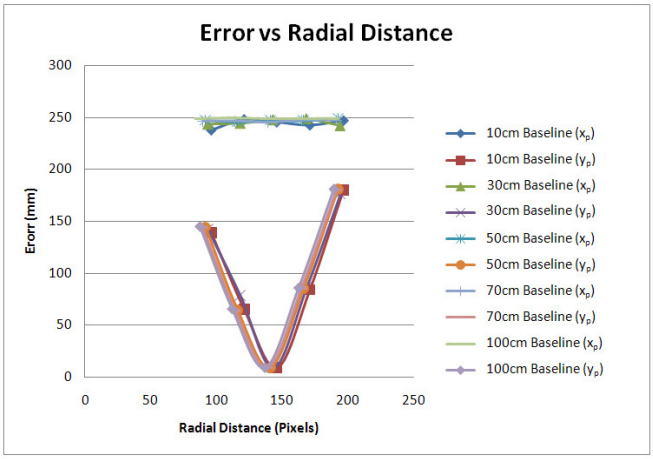


Figure 3.7: Plot of Errors vs Radial Distance

histograms of disparity distribution for two different environments, (a cluttered environment and the other, a relatively less cluttered environment) at different baselines shown in Fig. 3.8 and 3.9, were compared. It was found that both distributions gradually spread out to the right as the baseline increases. By utilizing the minimum disparity threshold shown in Fig. 3.6, the number of features in those disparity bins smaller than the specified threshold for that baseline would be summed up. At the same time, the total number of features in the last three disparity bins at different baselines would be monitored in order to provide an indication of the number of features perceived to be lying close to the system in the current baseline. All these data are illustrated in Fig. 3.8 and 3.9. From these graphs, we discovered a trend. It was found that the total number of features, with a disparity value less than the specified threshold for that baseline, would begin from a very high value and reduce to a point where it would start to stabilize, whereas the total number of features in the last 3 disparity bins would start off in the lows and experience a drastic increase at some point. Since we would like to minimize the total number of features below the minimum disparity threshold without sacrificing too much of the near range in order to produce a disparity map with the right balance of features, this would imply that a "good" baseline should be one lying in the stable region in Fig. 3.8 and those close to the lows in Fig. 3.9.

In the search for a "good" baseline for a given environment, we would compute the first and second order derivatives of the functions shown in Fig. 3.10 and 3.11. As a result, 3 or more stereo pairs would be required by the system to determine a "good" baseline for the given environment. Based on an acceptable range of values for the first and second order derivatives, which were determined empirically, the baseline would be incremented (assuming that it always starts from the minimum baseline) until an appropriate baseline was found. (For the record, the first data point Fig. 3.10 is an outlier because the summation of the total number of features in a disparity bin excludes the '0' disparity bin and it was found that at this short baseline, the majority of the disparities were found to be in the range of 0-1.) The following equations define a set of values which was found to yield good results for this system,

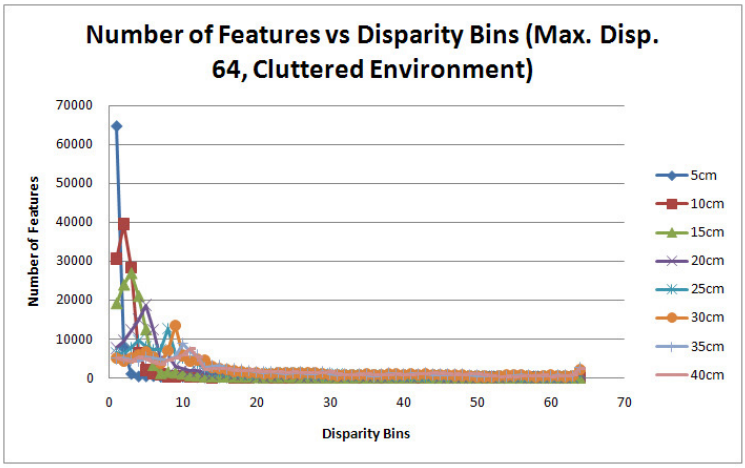


Figure 3.8: Histograms of disparity distributions at different baselines in a cluttered environment

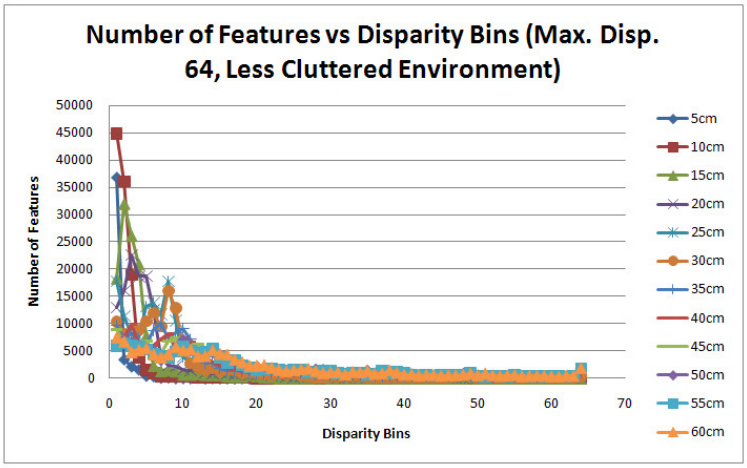


Figure 3.9: Histograms of disparity distributions at different baselines in a less cluttered environment

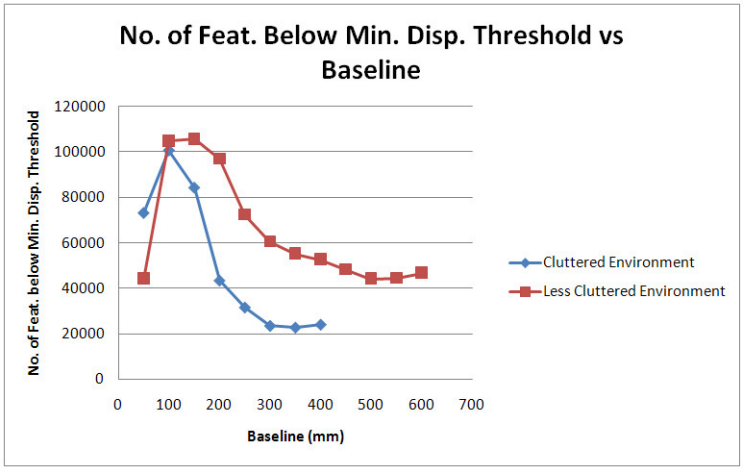


Figure 3.10: Plot of total features below min. disparity threshold vs baseline

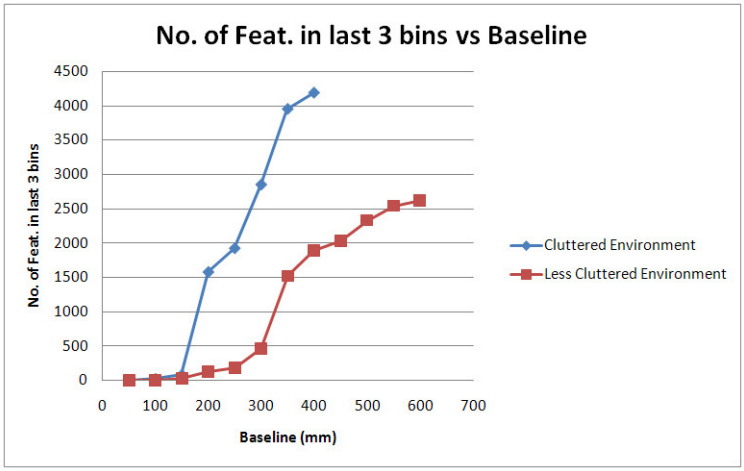


Figure 3.11: Plot of total features in the last 3 disparity bins vs baseline

$$-300 \leq \frac{dF_b}{db} \leq -100 \tag{3.1}$$

$$-10 \leq \frac{ddF_b}{ddb} \leq 0 \tag{3.2}$$

$$\frac{dF_l}{db} \geq 10 \tag{3.3}$$

$$\frac{ddF_l}{ddb} \approx 0 \tag{3.4}$$

where F_b measures the number of features below the minimum disparity threshold, F_l measures the number of features in the last 3 disparity bins and b refers to the baseline. To ensure that the selected baseline lies in the stable region shown in Fig. 3.10, the selected baseline would be required to satisfy the conditions described by Equations (3.1) and (3.2). At the same time, the selected baseline should lie close to the lows as shown in Fig. 3.11. To satisfy both conditions, the system initially has to wait until the condition described by Equation (3.4) to occur and then selecting the baseline that satisfies the conditions described by Equations (3.1) and (3.2).

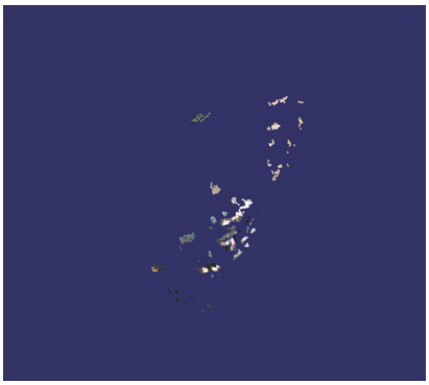
By examining the shape of the histograms of disparity distribution in Fig. 3.8 and 3.9, it was found that the histograms for both the cluttered and relatively less cluttered environments have some sort of resemblance. However, in the previous section that describes the baseline selection process, the values of the bins between the minimum disparity threshold and the last 3 bins were not taken into consideration. This is because the shape of the histograms is highly dependent on the environment. Although for these two examples, the shape of the histogram might be skewed to the low disparity values, but it is always possible to have a histogram skewed to the mid or high disparity values given the right environment and combination of system parameters. The subsequent question will then be whether this will affect the selection of a "good" baseline if given such a situation. The answer is yes and no. The variable multibaseline system has its physical limitations

in terms of the minimum and maximum baseline that it can vary between the two catadioptric systems. As such, it is not always possible to have the best baseline for a given environment. On the other hand, the automatic baseline selection process was designed such that errors as a result of subpixel accuracy resolution could be minimized by selecting baseline(s), within the physical limitation of the variable multibaseline system, that have relatively low number of features in the bins below the minimum disparity threshold and at the same time taking into account of the number of features in the last 3 disparity bins. Consequently, this implies that the system will always attempt to search for the best baseline within its physical limitations but this selected baseline may not necessary be the best for the given environment.

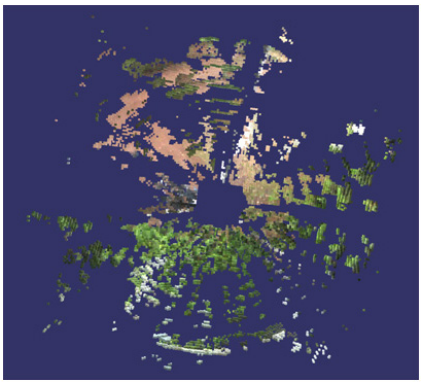
3.4 Results

The current system captures colour images of 640 x 480 in resolution but these images are converted to grayscale images before stereo correspondences are established. Although the Canon Powershot S3 IS can produce images up to 6 megapixels in resolution, the maximum resolution that can be used in this system without severely slowing it down is 1600 x 1200. However, for the multibaseline stereovision algorithm, 640 x 480 images are more desirable since accuracy can be met without significantly increasing computational time. As illustrated in Fig. 2.1, the system has usable effective baselines between the bottom and top catadioptric ranging from 30-90cm. On the other hand, if the first image captured by the top catadioptric system is treated as the reference image, then, subsequent images captured using the top catadioptric system can have effective baselines ranging from 5-60cm. In order to illustrate the effect of using a "bad" effective baseline and to compare it with the results from a "good" effective baseline consistently, the following results were generated by using the top catadioptric system only starting from the shortest achievable baseline on the system.

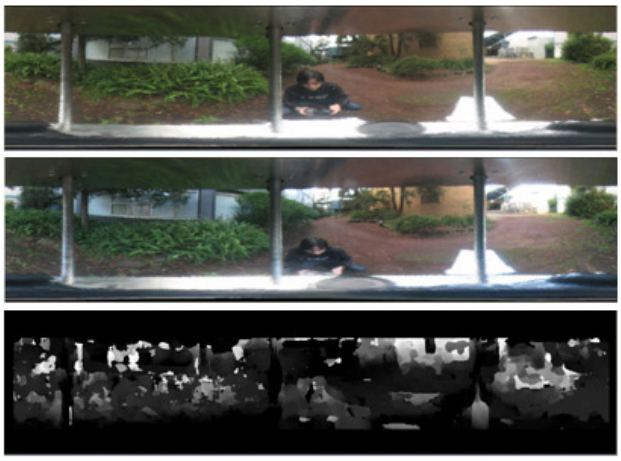
Experimental images were collected from various environments and the voxelized environments generated by the system are illustrated in Fig. 3.12-3.16. For each environment (excluding Fig. 3.16), the plan view of the voxelized environment, stereo pair and disparity map of the selected baseline will be presented and placed next to the voxelized environment created from a baseline which is not in the stable region and the voxelized environment perceived by means of the multibaseline omnidirectional stereovision algorithm. For the results in Fig. 3.16, the environment is much larger and open as compared to the other environments. As such, providing the plan views of the voxelized environment will not allow the reader to make any useful comparisons. Instead, we have selected only a part of the voxelized environment to make the results more meaningful. A 15 x 15 neighbourhood with a maximum disparity search range of 64 pixels with 0.5 disparity step size were used for establishing stereo correspondences and the subsequent parameters used for the multibaseline stereovision algorithm for each environment are provided in the captions of Fig. 3.12-3.16.



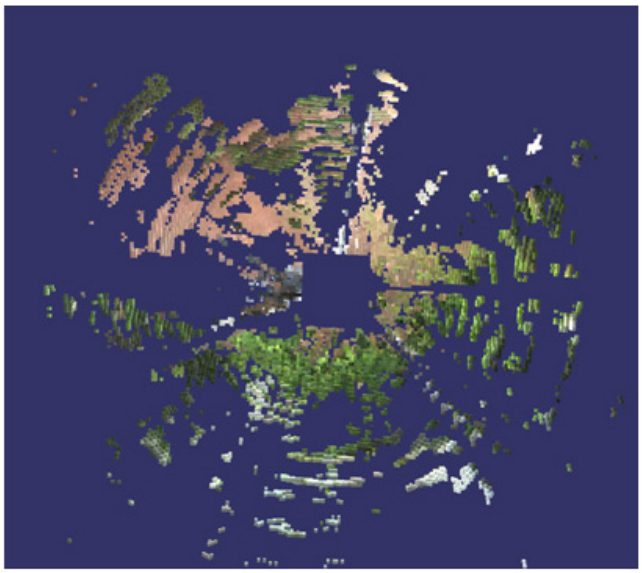
(a) Plan view of voxels created using stereo pair with 10cm baseline (not in the stable region)



(b) Plan view of voxels created using stereo pair with 25cm baseline (automatically selected baseline)

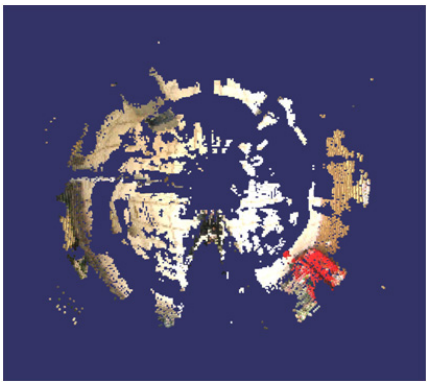


(c) Stereo pair and disparity map of the automatically selected baseline

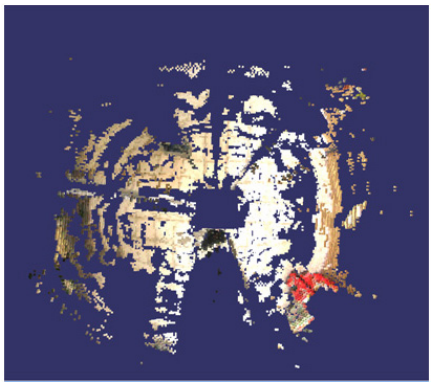


(d) Plan view of voxels created using 5 stereo pairs, voxels with uniform lengths of 5cm for each dimension with a minimum voxel threshold of 2 features

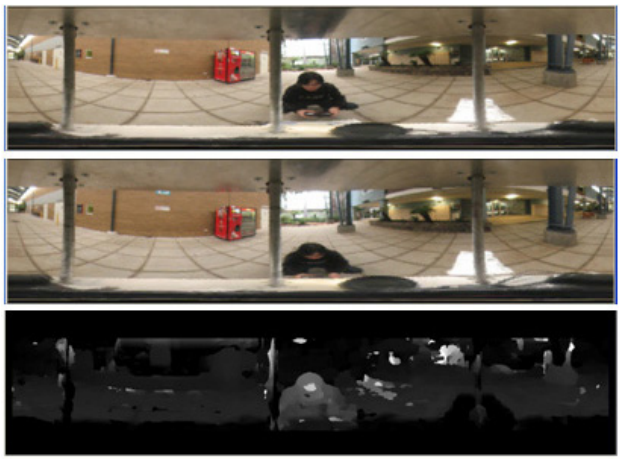
Figure 3.12: Multibaseline Stereovision Results in Outdoor Environment with Numerous Natural Features



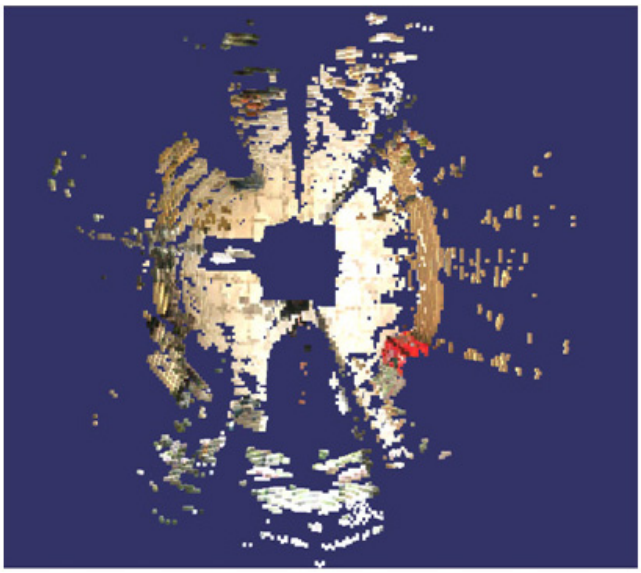
(a) Plan view of voxels created using stereo pair with 10cm baseline (not in the stable region)



(b) Plan view of voxels created using stereo pair with 30cm baseline (automatically selected baseline)

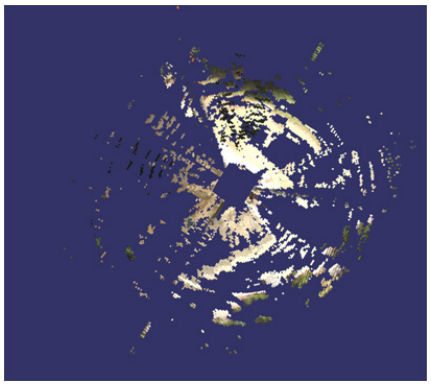


(c) Stereo pair and disparity map of the automatically selected baseline

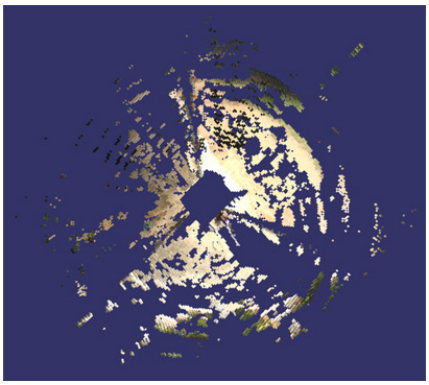


(d) Plan view of voxels created using 7 stereo pairs, voxels with uniform lengths of 5cm for each dimension with a minimum voxel threshold of 4 features

Figure 3.13: Multibaseline Stereovision Results in Semi-Outdoor Environment with Numerous Man-Made Features

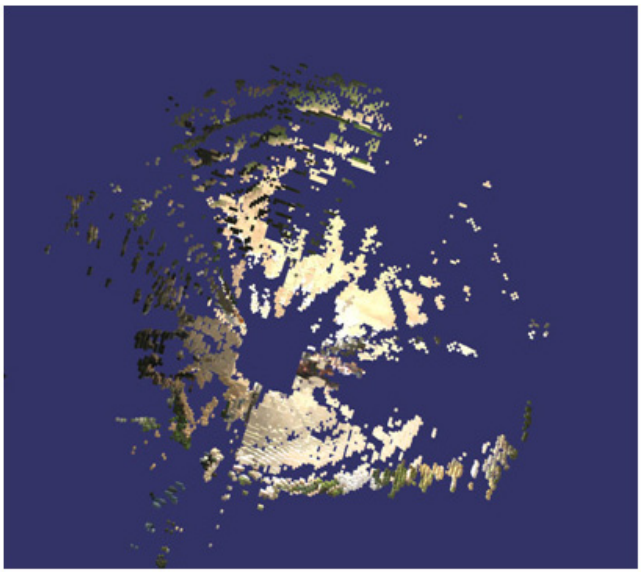


(a) Plan view of voxels created using stereo pair with 30cm baseline (not in the stable region)



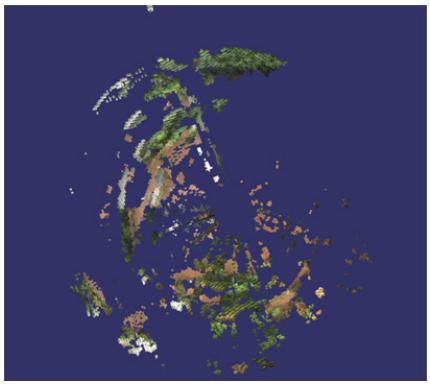
(b) Plan view of voxels created using stereo pair with 35cm baseline (automatically selected baseline)

(c) Stereo pair and disparity map of the automatically selected baseline

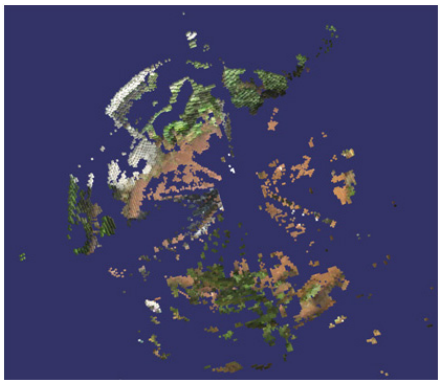


(d) Plan view of voxels created using 7 stereo pairs, voxels with uniform lengths of 5cm for each dimension with a minimum voxel threshold of 4 features

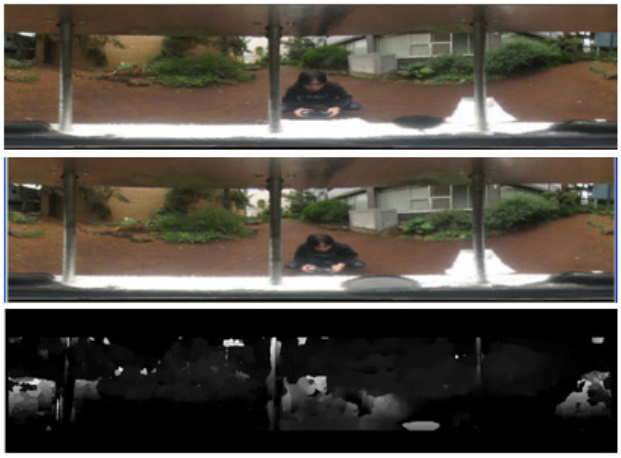
Figure 3.14: Multibaseline Stereovision Results in Outdoor Environment with both Natural and Man-Made Features



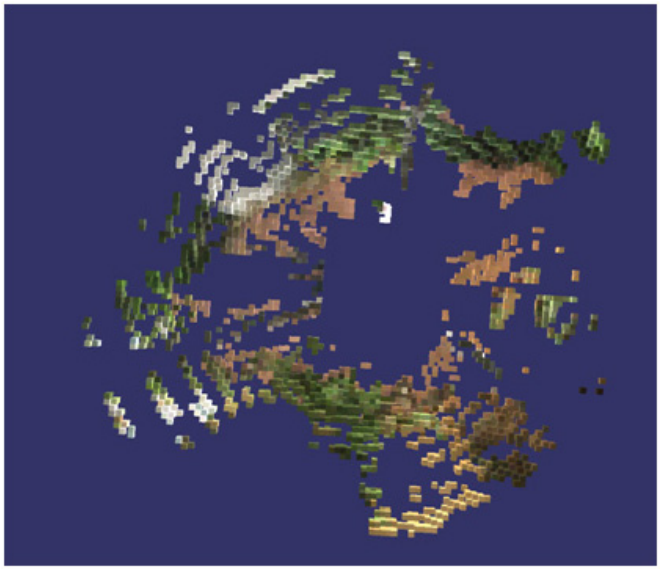
(a) Plan view of voxels created using stereo pair with 10cm baseline (not in the stable region)



(b) Plan view of voxels created using stereo pair with 30cm baseline (automatically selected baseline)

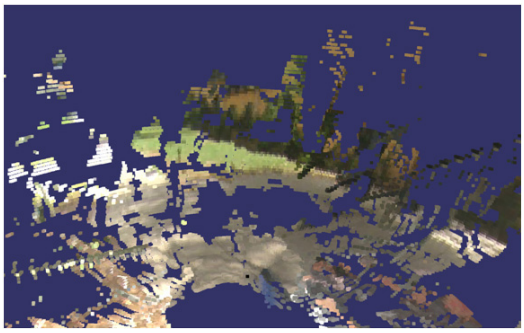


(c) Stereo pair and disparity map of the automatically selected baseline

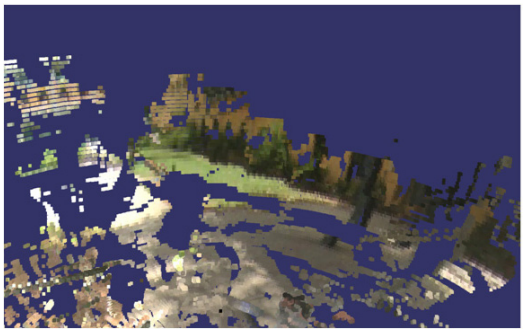


(d) Plan view of voxels created using 6 stereo pairs, voxels with uniform lengths of 5cm for each dimension with a minimum voxel threshold of 4 features

Figure 3.15: Multibaseline Stereovision Results in Semi-Outdoor Environment with Numerous Man-Made Features

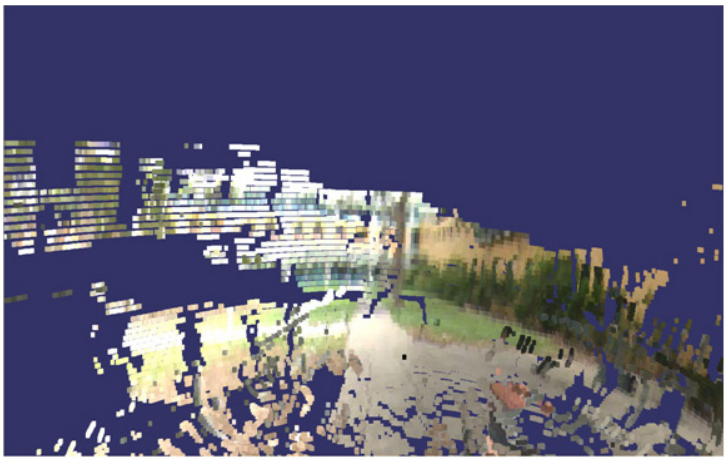


(a) Plan view of voxels created using stereo pair with 30cm baseline (not in the stable region)



(b) Plan view of voxels created using stereo pair with 45cm baseline (automatically selected baseline)

(c) Stereo pair and disparity map of the automatically selected baseline



(d) Plan view of voxels created using 4 stereo pairs, voxels with uniform lengths of 5cm for each dimension with a minimum voxel threshold of 2 features

Figure 3.16: Multibaseline Stereovision Results in An Open Outdoor Environment with Both Natural and Man-Made Features

3.5 Discussion

For most of the time, the stereo pair at the automatically selected baseline produced a satisfactory representation of the environment surrounding the mobile robot. On the other hand, stereo pairs captured with baselines that were not in the stable region might sometimes fail to reliably perceive the environment, such as the one illustrated in Fig. 3.12. This failure mode was found attributed to the short baseline used (10cm), where the majority of the disparities produced for this stereo pair lie in the region of 0-1 (which were rejected) in this environment. For other cases, using a baseline outside the stable region might produce a less accurate version of the surrounding environment as compared to the one created by the automatically selected baseline. This could be clearly observed on the right hand side of the voxelized environment in Fig. 3.13. The location and shape of the brown wall and vending machine were found to be not as accurate as compared to the environment perceived by using the automatically selected baseline. In addition, when the voxelized environment produced by using the automatically selected baseline was compared with the voxelized environment produced by the multibaseline stereovision algorithm, the latter approach produced was found to produce a denser representation of the environment and is more suitable for long range depth perception (see Fig. 3.15). In order to truly evaluate the dynamic range of the automatic baseline selection system, it was further tested in an open outdoor environment as shown in Fig. 3.16. The selected baseline for this environment was 45cm (15cm from the maximum usable baseline on the system). It was found that even for such typical open environments, the automatic baseline selection system might not select the maximum usable baseline of 60cm because in such scenarios, a significant portion of the scene would still be composed of wide and open ground surfaces that is close to the vicinity of the robot, although overall, the number of far range features have increased.

Of course, under such circumstances, it might be worthwhile knowing whether it is necessary to use two vertically stacked catadioptrics instead of one since it was illustrated that the selected effective baselines range from 20-45cm in the experiments. As for efficiency reasons, if the selected baseline is larger than 30cm, the mobile robot can make use of both catadioptrics. Once the desired baseline is found, it can remain fixed at this baseline until the robot decides again that it should probably perform another test to find the best baseline at its current position. However, if the selected baseline is smaller than 30cm, the only way to achieve this on the system is to use the top catadioptric system only. Unfortunately, in this case, the robot will be required to vary its baseline every time it attempts to obtain range measurements. For static or less dynamic environments, this is still practical for this robot because range measurements are obtained from this system after some arbitrary amount of distance is travelled by the robot. For very dynamic environments, it is better off selecting a baseline within the physical limits of the system. Additionally, if the selected baseline is larger than 30cm, using the two vertically stacked catadioptrics can produce more stereo pairs if range measurements from the multibaseline stereovision system are required.

The following describes a simple strategy to determine whether the required baseline is smaller than the physical limitation of the two catadioptrics without slowing down the system. Assume that the top catadioptric system is currently at the minimum usable baseline (measured from the bottom catadioptric). The first stereo pair utilizing the two catadioptric systems will have a baseline of 30cm. As the baseline is incremented at fixed intervals, a new image is captured at the new baseline producing an additional stereo pair by using the image captured by bottom catadioptric system as its reference image and another stereo pair by using the initial image captured by the top catadioptric system, at a distance of 30cm measured from the bottom system, as its reference image. Since the laptop is equipped with two GPUs, the depth maps and triangulated positions of the correspondences can be computed concurrently for both stereo pairs. After 3 or more stereo pairs, if dF_b/db was found to be consistently low for the stereo pairs using the image captured by the bottom catadioptric system as its reference image, it confirms that the "good" baseline is below the physical limits of the two catadioptrics. The system will not be required to perform any additional computations since the depth maps and triangulated 3D points, for all stereo pairs using the image captured by the top catadioptric system as its reference image, are readily accessible. Subsequently, the system proceeds to find the best baseline. For static and less dynamic environments, the system can utilize this selected baseline and translate the top catadioptric system vertically whenever range measurements are required whereas for dynamic environments, the system will ignore this selected baseline and use the smallest baseline achievable between the two catadioptric systems. At this point in time, the information regarding whether it is a static, less dynamic or very dynamic environment is provided to the system.

Most multibaseline systems, such as the ones proposed by Meguro et al. (2007) and Sato et al. (2002) , combine stereo pairs using structure-from-motion techniques. For these systems to work reliably, it normally requires an accurate GPS together with an IMU or some complex algorithms to track the effective baseline and orientation of the camera between relative frames in image sequences. These systems offer several advantages over vertically configured systems, such as the one described in this thesis, because the selection of the effective baseline is not constrained by hardware limitations but the physical structure of the environment. In addition, it is possible to use hundreds of images in order to create a very dense and accurate representation of the environment as compared to a maximum of 8 stereo pairs for 5cm increments in baseline or 16 stereo pairs for 2cm increments in baseline for the proposed system. However, it is not the intention of this system to create detailed models of the environment but rather an internal representation which is sufficient for the mobile robot to plan a rough path and refine it as it moves. Moreover, creating such detailed models are too costly (up to hours for some cases depending on the number of images used). Additionally, a vertically configured multibaseline system does not require an accurate GPS or some other complex solution to determine the amount of camera movement. Although highly accurate GPS receivers are available, however, it is not surprising that GPS signals are weak in highly cluttered environments, in narrow alleys surrounded by tall high rise buildings or in natural environments with a

thick canopy layer. In such situations, the estimated baseline will not be good enough for depth estimation. On the other hand, complex solutions, which involve the estimation of extrinsic camera parameters by continuously tracking and updating natural features in the environment for each frame in the image sequence, are an expensive process if accuracy is required. Similar to any visual odometry approaches, errors accumulate over time and the relative positions of the camera computed become increasingly unreliable.

With this vision system, the mobile robot is able to produce a dense representation of the environment by combining multiple stereo pairs when required. In other cases, it can be used to find a suitable baseline for the given environment when only a single stereo pair is required. This eventually opens up many possibilities for further extensions and applications of the stereo system in order to realize the ultimate goal of this work to develop a fully autonomous mobile robot.

3.6 Chapter Summary

Perceiving depth through binocular vision has always been an effortless task for humans. However, the establishment of accurate and dense stereo correspondences between a pair of cameras has always been an issue with the problem becoming more difficult as the baseline between the two cameras increases. This chapter describes how depth estimates from stereo correspondences can be improved by combining information from stereo pairs taken at different baselines.. The baseline is automatically adjusted according to the histogram of the disparity distribution in the omnidirectional image pair. This is aimed at creating multibaseline stereo images grabbed at baselines which are adapted to the environment surrounding the robot when a dense representation of the environment is required, or finding a suitable baseline for the stereo system when only a single stereo pair is required.

4

Visual Odometry

4.1 Introduction

The word odometry is composed of the Greek words *hodos* meaning "way" or "journey" and *metron* meaning "measure". In short, it defines the estimation of one's position relative to a starting position (position in this context implies both the location and orientation) after some arbitrary amount of motion is taken. The ability to measure how far one has travelled is not only valuable to humans but practically anything that is required to navigate in free space, including mobile robots. In fact, this is a crucial piece of information which partially provides an answer to the question, "Where am I?". This is also the fundamental question which many research in mobile robotics have been trying to answer for the past decade and is referred to as the mobile robot localization problem.

Let's illustrate this through a simple example. Assume that a human subject is required to complete a task, which requires the subject to travel from point A to point B in free space, given that he/she is aware of his/her initial position but with eyes blindfolded and ears isolated from sound with earplugs. In reality, it is impossible for the subject to navigate without knowing where he/she is. In this example, since the subject has neither visual nor audio information, the most intuitive approach to estimate how far he/she has travelled is to count how many steps were taken and roughly in which direction they were taken relative to the starting position. Similarly, for a differential drive mobile robot, if the base and radius of the wheel is provided, one can derive the position of the robot by taking into account the direction and number of revolutions made by each wheel. It is also common to observe that the error associated with the estimated position of the mobile robot increases over time as it navigates itself in the environment, since each new position is calculated based on the previous estimated position. In order to impede the accumulation of errors over time, it is a common practice to design and apply sensor and robot dependent models to account for the uncertainty associated with the mobile robot's estimated position. Continuing from the previous example, the direction and total number

of revolutions (including fractions) made by each wheel can be recorded using encoders which are tightly coupled to the rotating shaft. Subsequently, factors that play a role in the accuracy of the estimated position such as the resolution of the sensor output, changes in wheel radius, etc, can serve as inputs to the model. Nevertheless, no model is perfect and issues associated with wheel slippage or the exact change in wheel radius (due to mechanical imperfections and load variations) are hard to account for precisely in reality.

In order to deal with the associated errors in sensor measurements, it is normally compensated for by fusing multiple sensors that complement one another with the aim to produce a less erroneous position estimation via this multi-sensor framework. Nevertheless, it is still a matter of time before this drift becomes too large for the mobile robot to make a well-informed decision for the next course of action. It has long been recognized within the robotics community that mapping is an integral part of the localization process. In fact, it is posed as a chicken-and-egg problem, whereby accurate localization is required to perform mapping and vice-versa. When an internal representation of the environment is available, these positional drifts can be kept within some tolerable error bounds whenever the robot closes the loop/revisits an explored location. Nevertheless, in this chapter, the localization process is isolated from the mapping process until later chapters.

Wheeled locomotion (Mobile Robots, 2010; Segway Robotics, 2010; K-Team Mobile Robotics, 2010; Rusu et al., 2009) is one of the most widely studied locomotion modes for mobile robots, with legged locomotion (Sakagami et al., 2002; Tsagarakis et al., 2007; Gouaillier et al., 2009) on the rise due to the growing interest in humanoid robots and various other biologically inspired systems (Altendorfer et al., 2001; Raibert et al., 2008; Maladen et al., 2009; Fjerdingen et al., 2009). Driven by factors such as simplicity in its kinematic model and capability to turn on the spot, it is common to find that many wheeled mobile robots operate using the drive mechanism known as differential drive. As such, in the literature, there has been abundance of work related to the error modelling of these systems. For instance, Chong and Kleeman (1997) proposed an algorithm for the calibration and error modelling of the mobile robot by taking into account systematic errors such as wheel base and wheel radius differences and non systematic errors which have a strong physical basis closely related to the design of the mobile robot. This work was further extended to include the effects of wheel separation by Kleeman (2003). In another work by Tur (2007), error modelling was performed by analyzing the kinematic model of the mobile platform. The output of both works is an error covariance matrix which can be directly used by many state-of-the-art probabilistic frameworks for mobile robot localization and mapping. However, such approaches lack platform interoperability and the changeover to a different mobile platform might require the recalibration and/or the remodelling of errors for this new platform. Partially due to this reason, we were motivated to perform localization by using visual information and this is commonly referred to as visual odometry.

The main reason for choosing visual information to estimate odometry was partly inspired by the capability of human beings to perform such tasks at ease and the availability of high quality and inexpensive cameras. Generally, visual odometry approaches can be

classified based on whether they are (1) monocular or stereo camera configurations, (2) perform feature tracking (e.g. Harris corners, SIFT, etc) or apply image-based techniques (e.g. optical flow, appearance-based, etc) through successive frames or feature matching between two frames, and (3) real-time incremental or offline batch processing techniques. More recently, the development in wide-angle and omnidirectional vision systems can further refine the first(1) classification into whether the cameras used are standard perspective, wide-angle or omnidirectional vision systems.

Generally, stereo configurations are arguably more advantageous than monocular configurations. This is mainly due to the fact that the scale of the overall motion can be easily recovered on stereo configurations given its known baseline. In addition, it is easier to deal with the situation when the mobile robot is stationary or moving rather slowly with a stereo configuration. Once the type of vision system has been decided (i.e. stereo or mono), then it boils down to which kind of technique can be used to recover motion estimates. The first step generally involves the tracking or matching of distinctive features or optical flow through successive frames. Nistér et al. (2007) proposed the detection and tracking of Harris corners where features were detected in all frames but only those matching between two successive frames were retained, while those which had lost track were replenished by new ones. This was tested on both stereo and monocular configurations with a slight difference between the algorithms since the scale has to be estimated on the monocular version and range data was not available initially. In another work, Cheng et al. (2006) proposed the tracking of sparse corner features on the image returned by the left stereo camera, associated with a disparity value that could be used for the recovery of its 3D position, for the visual odometry system onboard the Mars rovers. Both of these methods then resorted to a RANSAC procedure for outlier rejection based on some model which selects the motion estimate that fits the largest number of features within some error bounds. On the other hand, Howard (2008) proposed a visual odometry algorithm using stereovision which performs standard corner feature matching, each associated with a 3D position, and an inlier detection algorithm based on some simple rigidity constraints on the world coordinates.

Of course, not all monocular configurations are required to estimate its scale. The mobile robot in (Fernandez and Price, 2004) was equipped with a camera facing downwards performing ground plane optical flow tracking. Scale recovery was not necessary since this information was provided to the system through a calibration process. Nevertheless, the proposed system was constrained to travel on fairly flat terrain. In another work (Campbell et al., 2005), a forward facing camera was mounted on a mobile platform and used the renown LKT optical flow tracker (Lucas and Kanade, 1981). The resulting optical flow vectors were classified into the ground, sky and horizon regions whereby the vectors on the ground region were used to estimate translation parameters and those in the sky region were used to estimate orientation. However, this system suffers from the unknown scale factor. A more recent work, which can also be loosely classified as an image-based technique, can be found in (Comport et al., 2010). This system was equipped with a stereo camera and avoided the error-prone feature extraction process by

exploiting the quadrifocal geometry between four views (two stereo pairs), its respective stereo correspondences and finally a robust estimation technique based on a second order approximation technique which minimizes a non least squares cost function in order to provide the motion and pose estimates of the stereo rig in 6DoF. Furthermore, this system could deal with large inter-frame displacements with a multi-resolution image approach.

All the previously discussed approaches were developed for real-time applications without the use of the computationally expensive batch processing techniques such as the non-linear bundle adjustment algorithm. Bundle adjustment is normally used for the refinement of the 3D coordinates of features that define the scene geometry and also the transformation from one viewpoint to another viewpoint via some least squares minimization technique such as the Levenberg-Marquadt algorithm. Due to the benefits of bundle adjustment and the need to satisfy the real-time constraint in many applications, a real-time compatible version of bundle adjustment was proposed (Engels et al., 2006). In addition, all previously discussed approaches use standard perspective cameras. However, in recent years, the availability of omnidirectional catadioptric vision systems and cameras with wide angle lenses have led to the development of visual odometry systems with a wide field-of-view (FOV). These systems are advantageous for visual odometry since feature matching, feature tracking or direct image-based techniques benefit from the wide FOV as most features remain in the scene after some small arbitrary amount of motion is taken by the robot. Furthermore, omnidirectional vision systems completely eliminate the *windowing* problem suffered on standard perspective cameras. For instance, Gluckman and Nayar (1998) proposed an algorithm where motion estimates were derived using the differential form of the epipolar constraint for central catadioptrics. On the other hand, Mičušík and Pajdla (2006) proposed an offline method for non-central camera models, including cameras with wide-angle fish-eye lenses and non-central catadioptrics, by an approximation of the non-central model using central models. This provided the initial camera models and its essential matrices which could be used for 3D reconstruction. Then, the 3D reconstruction of the scene was improved by using a bundle adjustment technique that assumed a non-central model. As the initial 3D reconstruction of an image sequence was being performed, the motion of the camera was estimated and refined through bundle adjustment. In another work, Scaramuzza and Siegwart (2008) developed an omnidirectional visual odometry system by tracking SIFT features (Lowe, 2004) on the ground plane to estimate translational motion and employed a direct image-based/appearance-based technique referred to as the panoramic visual compass (Labrosse, 2007) to estimate orientation. Finally, a recent work by Hansen et al. (2010) reported a complete system using a camera with fish-eye lens to detect and match SIFT features directly on the warped image sequence with applications to visual odometry.

As discussed earlier, it is not uncommon to find odometry systems to be made up of multiple different sensors in order to compensate for the errors associated with any one particular sensing mode. For example, Agrawal and Konolige (2007) combined visual odometry with an inexpensive GPS and inertial measurement unit (IMU). Consumer-grade GPS receivers provide global localization up to $\pm$15m but, unfortunately, it is unusable

in covered areas and global positioning error is huge in typical urban environments with dense and fairly tall buildings. Similarly, Zhu et al. (2007) proposed a visual odometry system based on the tracking and matching of landmarks (Harris corners) and attempted to reduce drift in pose estimates by incorporating an inexpensive IMU. In addition, Ng (2003) studied the performance of the optic flow sensor found on a standard optical mouse, functioning as a two axis displacement sensor, on different surfaces targeted to visual odometry applications. A similar work which employed such sensors in a differential configuration integrated with an IMU was reported in (Kim and Brambley, 2007).

4.2 Visual Odometry

The visual odometry system developed for our mobile robot can be classified as a direct image-based approach which combines ground plane optical flow tracking and an appearance-based panoramic visual compass technique. It assumes that the mobile robot is restricted to pure rotation or straight line motion at any point in time and provides a 3DoF motion estimate of the robot. The real-time constraint is satisfied and motion is estimated by combining distance estimates calculated by using a pseudo optical flow algorithm, with bearing estimates provided by the panoramic visual compass system, which will be described in the following sub-sections.

4.2.1 Appearance-based Panoramic Visual Compass

The panoramic visual compass, an appearance-based technique proposed by Labrosse (2007), measures the yaw angle of any mobile robot travelling on fairly flat terrain. Although the ground plane optical flow tracking algorithm, which will be described more thoroughly in Section 4.2.2, could also provide an estimation of this angle, unfortunately, it was found to be less reliable as compared to this technique. This angle basically defines the direction of heading of the mobile robot and unreliable estimates of the heading can quickly translate to huge errors between the estimated and actual locations of the mobile robot in the environment. Thus, the bearing of the mobile robot would be estimated using this technique instead.

The panoramic visual compass algorithm was found to be suitable for both central and non-central catadioptric vision systems. To compute the relative heading difference between the current omnidirectional image with the reference image, some standard image similarity metric such as Euclidean distance, SSD, SAD, etc, was used. Both images were unwarped into its rectangular form before they were compared. By iteratively comparing the unwarped input image with the reference image and column-wise shifting the unwarped input image after each iteration for a number of times equal to the pixel count width of the unwarped input image, an array of scores was computed using the chosen metric. Subsequently, the relative heading difference between the unwarped input image with respect to the reference image could be recovered by searching for the corresponding element in the score array that minimizes the difference between the two images. The index of this element in the array represents the total shifting of the unwarped input

image. By taking into account the shift direction, the relative heading difference could be computed. This column-wise shifting operation is referred to as the *sliding window* operation and this is illustrated in Fig. 4.1 and 4.2. In this example, the relative heading difference is 0^o since the mobile platform was stationary when the input and reference image was captured.

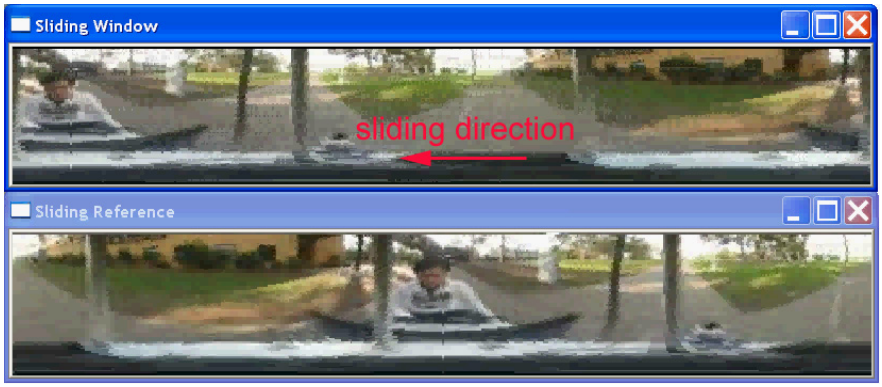


Figure 4.1: Sliding the unwarped input image (top) and reference image (bottom)

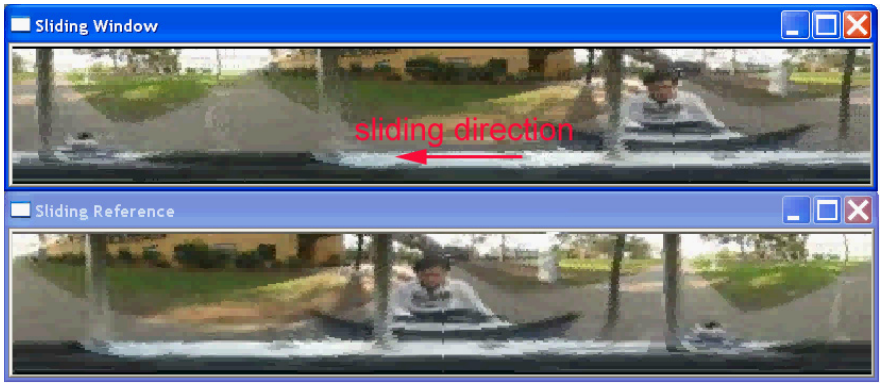


Figure 4.2: Sliding the unwarped input image (top) and reference image (bottom)

As thoroughly discussed in the original work (Labrosse, 2007), the precision of the estimated heading was limited by the resolution of the image and increasing the resolution of the image to improve its precision would proportionally increase the total processing time. In order to resolve this issue, a mixture of linear extrapolation and parabolic interpolation was used depending on the sharpness of the score function to provide subpixel accuracy. Linear extrapolation was found to perform better when the two images in comparison were found to be very similar, producing a sharp minimum, whereas parabolic interpolation was found more suitable for smooth minimums.

To detect the total change in heading direction of the mobile robot, the current unwarped input image would be compared to a reference image. This reference image, also an unwarped panoramic image, would be updated with the current unwarped input image when the difference between the two images was found to be large. This difference was measured by computing the normalized amplitude of the score function between the two images and the score function of the reference image with itself as described by the following equation,

$$A_n = \frac{A(I_t, I_{ref})}{A(I_{ref}, I_{ref})} \tag{4.1}$$

where I_t is the current image at time t and I_{ref} is the reference image and $A(I_i, I_j)$ for any image pair i and j can be defined as,

$$A(I_i, I_j) = d(I_i, I_j, \alpha_m + w/2) - d(I_i, I_j, \alpha_m) \tag{4.2}$$

where $d(I_i, I_j, \alpha)$ defines the chosen image similarity metric used to compute the score of the image pair with I_i column-wise shifted by α amount of pixels, α_m representing the amount of shifting required to produce the best matching image with respect to the reference image and w defines the width of the unwarped images. Referring to Equation 4.2, the amplitude of the score function was computed by finding the difference between the worst matching score (large difference) with the best matching score (least difference). The worst matching score was assumed to be the case when I_i was column-wise shifted to an amount such that its direction is directly opposite to the best matching pair. Although this might not be global minimum, Labrosse (2007) found that the score computed when I_i was shifted by $\alpha_m + w/2$ pixels would generally be stable and not too distant from the actual worst score in value.

To further improve the performance of the proposed heading estimation technique, Labrosse (2007) proposed to consider only certain parts of the omnidirectional images since the contribution to the optic flow by the motion of the robot is not homogeneous in omnidirectional images. For non-holonomic mobile robots, he proposed to compare only regions in the unwarped images which correspond to the front and back of the mobile robot with the width of each region approximately equivalent to 60 degrees in pixels. For holonomic mobile robots, extra processing would be required to provide an estimate of the direction of translation. As illustrated in Fig. 4.3, the region overlaid with orange colour is the region behind the mobile robot whereas the region in red highlights the region in front of the mobile robot. For completeness, the original panoramic visual compass algorithm is summarized in Algorithm 4.1.

Figure 4.3: (Red Overlay) Front Region and (Orange Overlay) Back Region

We adapted this technique for use in the proposed visual odometry system with several modifications. Firstly, the Sum of Absolute Differences (SAD) was chosen as the image similarity metric instead of Euclidean distance due to it being less computationally expensive. Next, the front 180 degrees FOV of the mobile robot in the unwarped images was used instead of the combined 60 degrees FOV of the front and back of the mobile robot.

Algorithm 4.1 Original Panoramic Visual Compass Algorithm

Parameters: $T_A = 0.6055$ - Normalized Amplitude Threshold

Require: Reference image I_{ref} (reference image) = $I_{t=0}$ (image at time = 0), $\omega_i = \omega_r = 0$ (current and reference orientations), $r_{rc} = 0$ (rotation between current and reference image) and r_{rp} (rotation between previous and reference image)

for t=1 to ∞ **do**
  $I_p = I_t$ //previous image is updated with current image
  I_t = newImage() // current image is updated with new image
  $r_{rp} = r_{rc}$
  Compute α_m between I_t and I_{ref}
  $r_{rc} = \alpha_m$/w*360 //need to take into account of shift direction
  if $A_n(I_t, I_{ref}) < T_A$ **then**
    $\omega_r = \omega_r + r_{rp}$
    Compute α_m between I_p and I_{ref}
    $r_{rc} = \alpha_m$/w*360 //need to take into account of shift direction
    $I_{ref} = I_p$
  end if
  $\omega_c = \omega_r + r_rc$
end for

This modification was made due to prolonged periods of the experimenter's presence in that region while monitoring the robot. Finally, two markers were placed beside the mirror of the catadioptric vision system such that its position could be tracked as the mobile robot moves in order to compensate for the vibration induced by the motion of the robot which affects the coordinates of the centre and rim of the mirror in the image required to unwarped the image (Fig. 4.4). The centres of these artificial markers, ($C1_x$,$C1_y$) and ($C2_x$,$C2_y$), would be tracked using the LKT tracker (Lucas and Kanade, 1981). Subsequently, the average change detected in these coordinates would be used to compensate for the change in the mirror centre (mc_x,mc_y) and mirror rim (mr_x,mr_y) coordinates respectively.

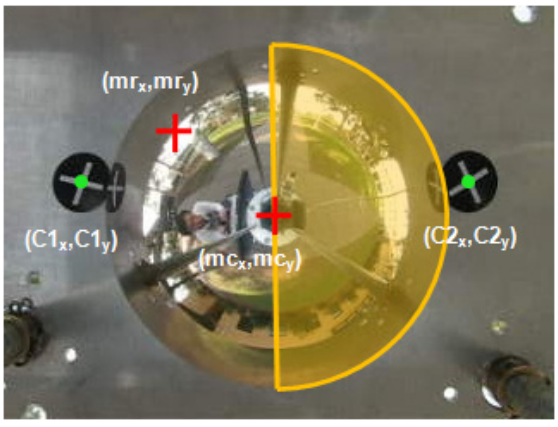
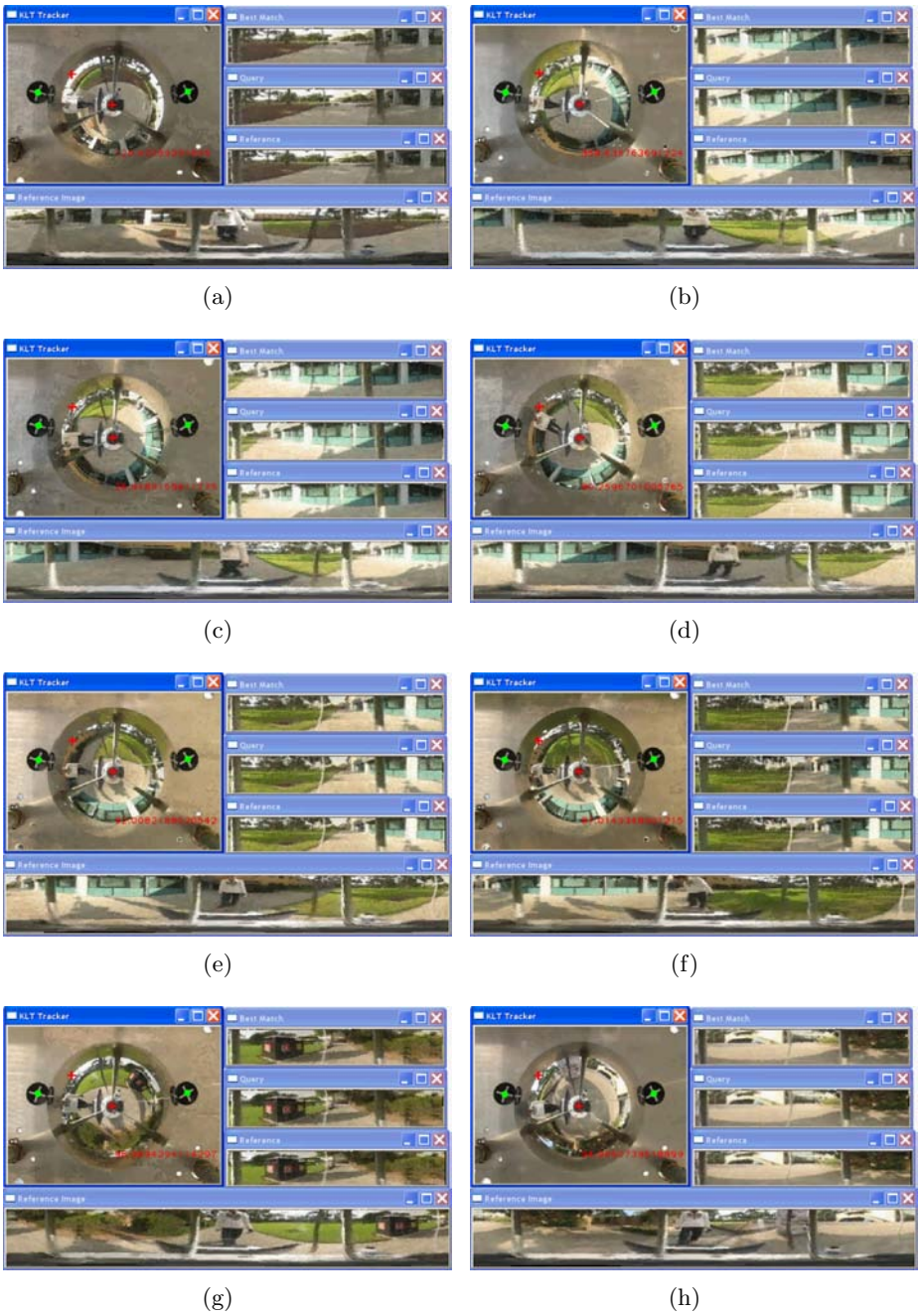


Figure 4.4: Selected Field of View and Tracking of Artificial Markers

Finally, the panoramic visual compass was put to test in an urban outdoor environment. Some randomly selected frames from the image sequence are shown in Fig. 4.5. Each subfigure includes the original catadioptric image with the location of the tracked

markers highlighted in green and the mirror centre and mirror rim coordinates highlighted in red. Images to the right of the original catadioptric image show the region in the unwarped image equivalent to the front 180 degrees FOV for the best matching, input query and reference images. The fully unwarped reference image can also be found at the bottom of each frame. Finally, by combining the distance travelled estimated by the wheel odometry system on the ActivMedia P3-AT mobile robot, the resulting trajectory was compared to ground truth as illustrated in Fig. 4.6 (plotted using Google Earth). The final algorithm is summarized in Algorithm 4.2.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.5: Selected Frames of Panoramic Visual Compass Experiment in Urban Outdoor Environment

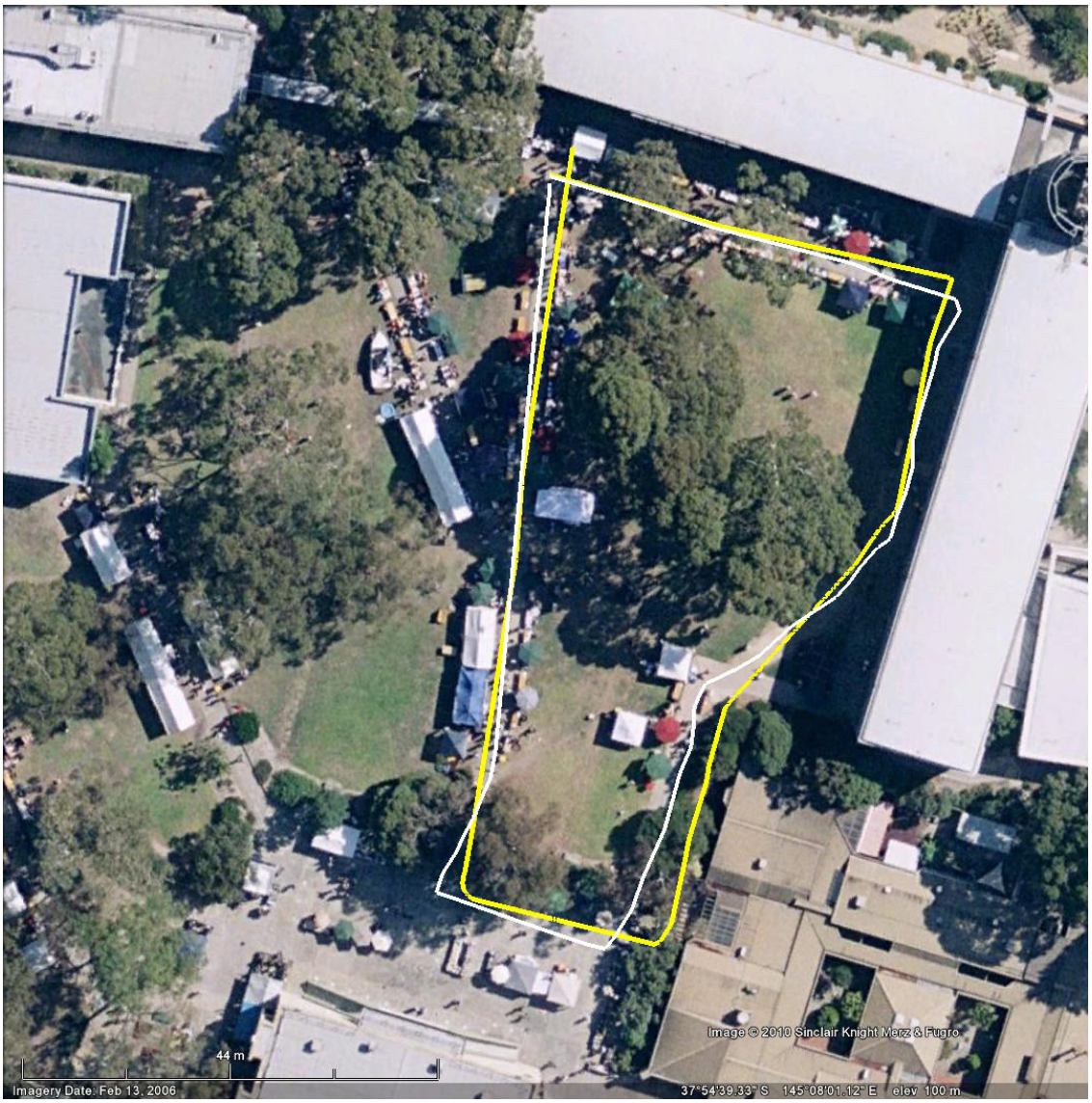


Figure 4.6: Comparing Estimated Trajectory against Ground Truth - Wheel Odometry and Bearing Estimate (Yellow) and Ground Truth (White)

4.2.2 Ground Plane Optical Flow Tracking

The ground plane optical flow tracking algorithm employed by the proposed visual odometry system to estimate the distance travelled by the mobile robot is similar to the approach described by Fernandez and Price (2004), where a camera was mounted on the mobile robot observing the surface, with the assumption that the camera image plane is parallel with the ground plane. In addition to the estimation of the two-axis displacement (dx and dy) of the mobile robot, the original technique described in (Fernandez and Price, 2004) could also estimate the change in heading (θ) of the mobile robot on flat terrain provided that the axis of rotation of the mobile robot is along the camera's optical axis. The displacement and rotation of the mobile robot, between two neighbouring frames from the image sequence returned by the camera, was estimated by measuring the change in displacement, dx and dy, in both the x and y directions (ground plane) estimated by

two corresponding points between the two frames. To further improve its robustness, this process was executed a number of times, taking two corresponding points at a time and averaging the final result. The full derivation of dx, dy and θ can be found in Appendix C.

Due to the growing payload, we replaced the ActivMedia Pioneer P3-AT mobile platform with a larger and stronger mobile platform powered by a differential drive wheelchair motor/gear set as shown in Fig. 4.7. An off-the-shelf Logitech web camera was used for the acquisition of ground plane images and a once off calibration technique was performed to find out the actual displacement in the x and y directions on the ground plane with respect to the width and height of a pixel in the image, since dx and dy values were measured in terms of pixels. This process has to be repeated only if there is a change in the height of the camera measured from the ground. The camera on the new mobile platform and the calibration image are clearly shown in Fig. 4.8. The horizontal and vertical blue lines were used to align the checkerboard pattern on the ground before the selection of any 3 vertices from the same square with known width and height. Once selected, the image coordinates of these 3 vertices would be used to derive the actual width and height of a pixel in the image on the ground.

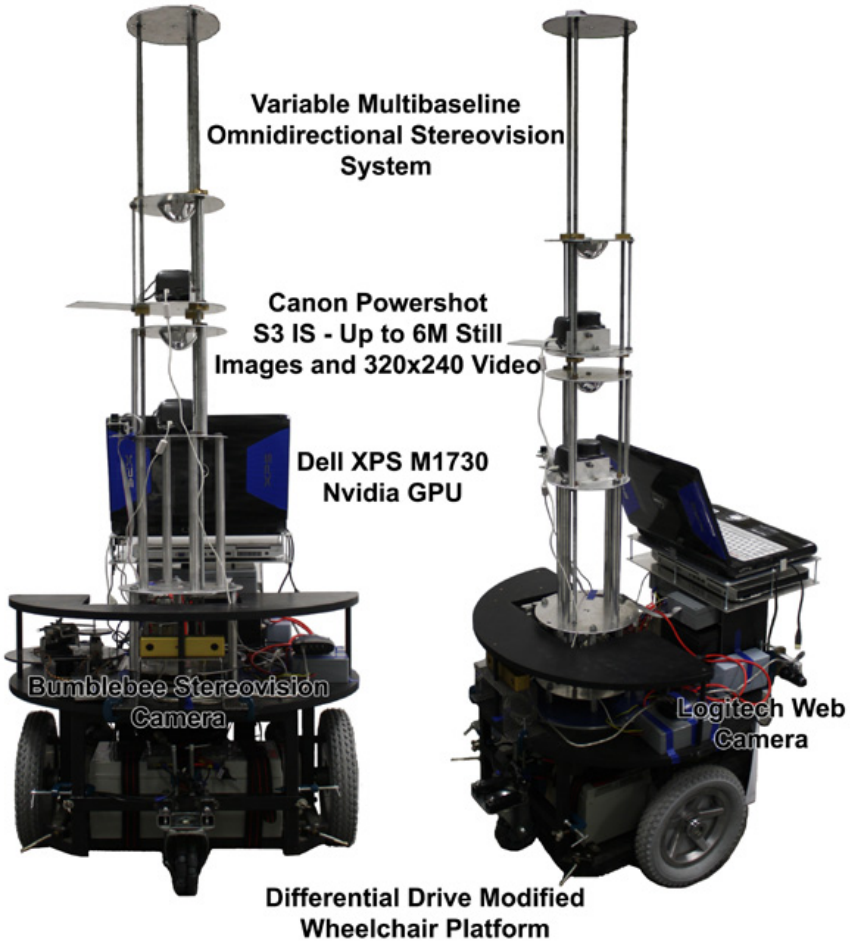
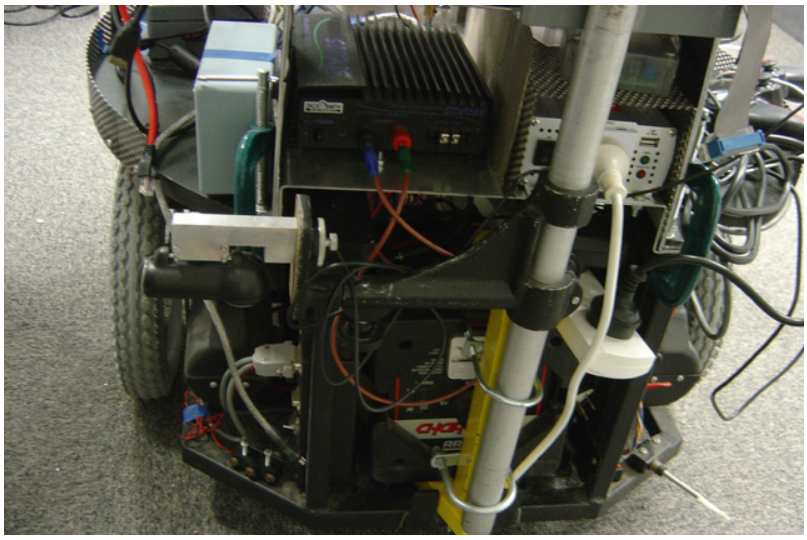
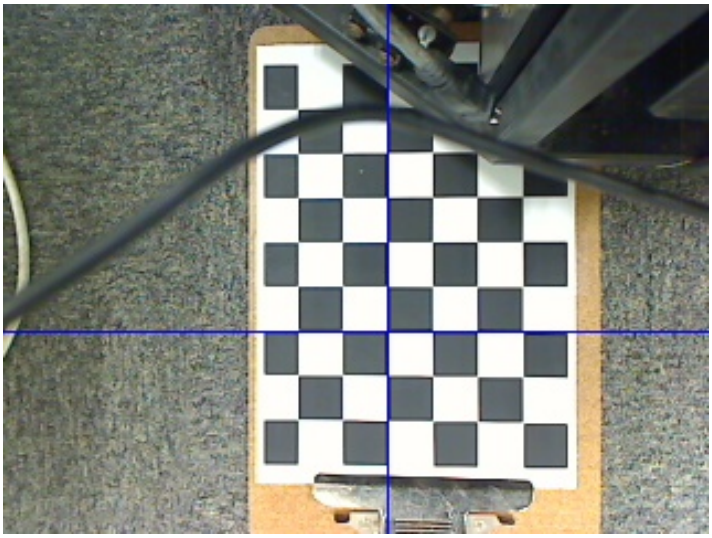


Figure 4.7: New Mobile Robot Platform



(a) Logitech Web Camera



(b) Calibration

Figure 4.8: Logitech Camera Calibration

To establish the image correspondences between two consecutive frames from the camera, the original method proposed to minimize the normalized sum of absolute differences between a local patch in the first image with the local patch in the subsequent image within some specified search window. For our proposed visual odometry system, image correspondences between two consecutive frames were established by initializing good features to track (Shi and Tomasi, 1994) and then tracking their corresponding positions in the subsequent frame using the LKT tracker (Lucas and Kanade, 1981). Fig. 4.9 and Fig. 4.10 illustrate the tracking of ground plane optical flow in a sequence of images using our method when the mobile robot was instructed to move in the forward direction. Take note that the magnitude of the optical flow has been magnified in these figures. In addition, all input images with a resolution of 320 x 240 would be cropped to 320 x 160 in order to exclude parts of the mobile robot's chassis in view.

Of course, there were times when this visual odometry system failed, such as in the short image sequences shown in Fig. 4.11 and 4.12. This might be due to various reasons such as textureless surface, motion blur or drastic variation in lighting intensity within a short time frame. The actual distance travelled by the mobile robot would not be recovered in these frames but at least, by filtering these outliers based on some simple constraints such as the standard speed the mobile robot would normally be travelling in, we could avoid having to accumulate these huge errors into the final estimated distance travelled by the mobile robot. The final visual odometry algorithm which combines bearing estimates from the appearance-based panoramic visual compass and distance travelled estimates from ground plane optical flow tracking is summarized in Algorithm 4.2.

Algorithm 4.2 Visual Odometry - Fusion of Optical Flow and Appearance based Techniques

Distance Travelled Estimates

Input: Features initialized and tracked by Kanade-Lucas-Tomasi (KLT) feature tracker available in OpenCV.

for every two image point pairs returned by KLT **do**
  Calculate the translation motion vector using dx and dy derived in Appendix C
  Filter and exclude vectors that are not achievable by the robot (i.e. amount of translation per frame)
end for
Average the resultant translation motion vectors

Bearing Estimates

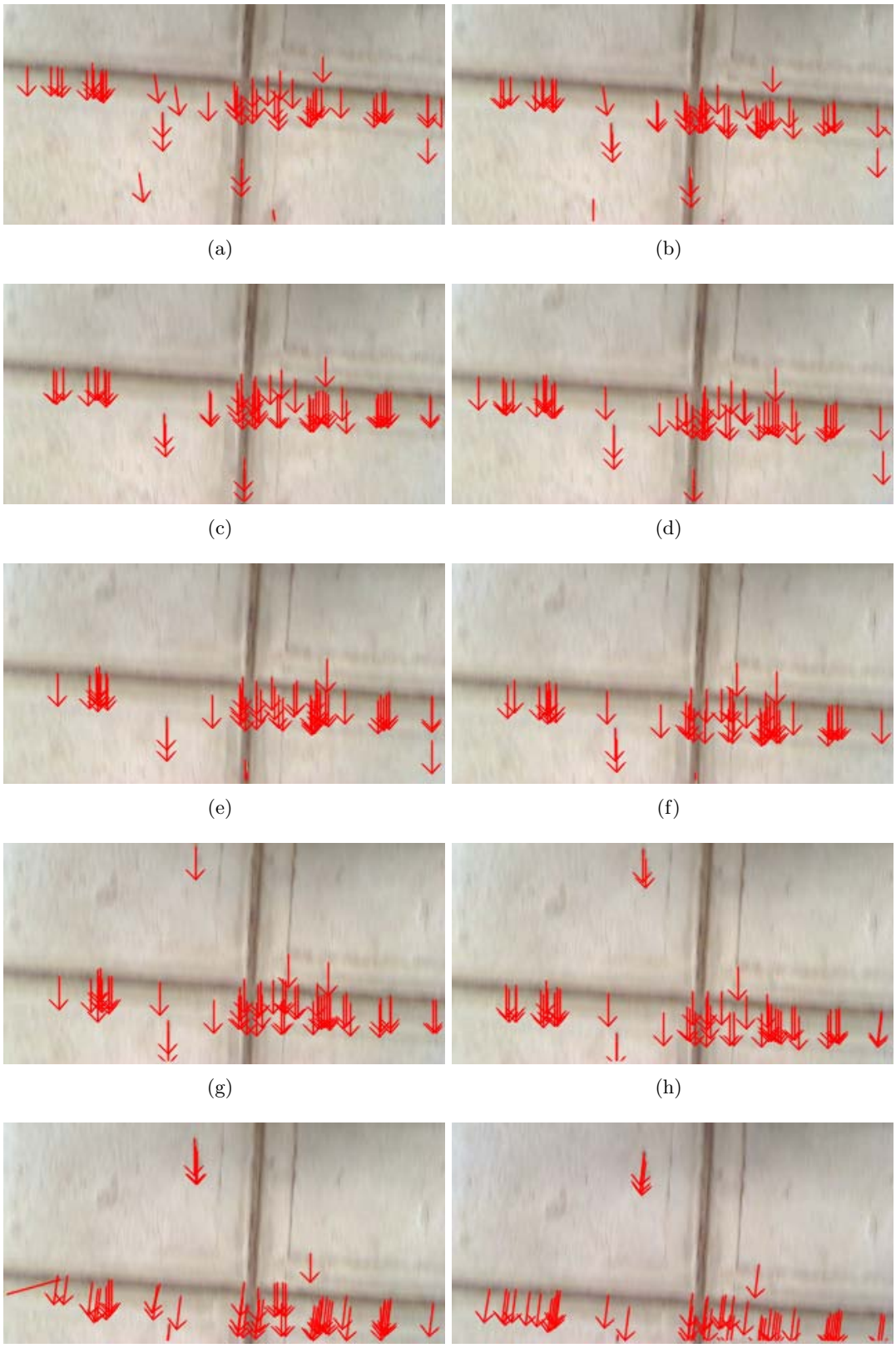
Input: Coordinates of mirror centre (mc_x,mc_y), mirror rim (mr_x,mr_y) and centre of the two crosses ($c1_x$,$c1_y$) and ($c2_x$,$c2_y$) in the omnidirectional image (Fig. 4.4)

Parameters: T_A - Normalized amplitude threshold

Require: Reference image $I_{ref} = I_{t=0}$ (image at time = 0)

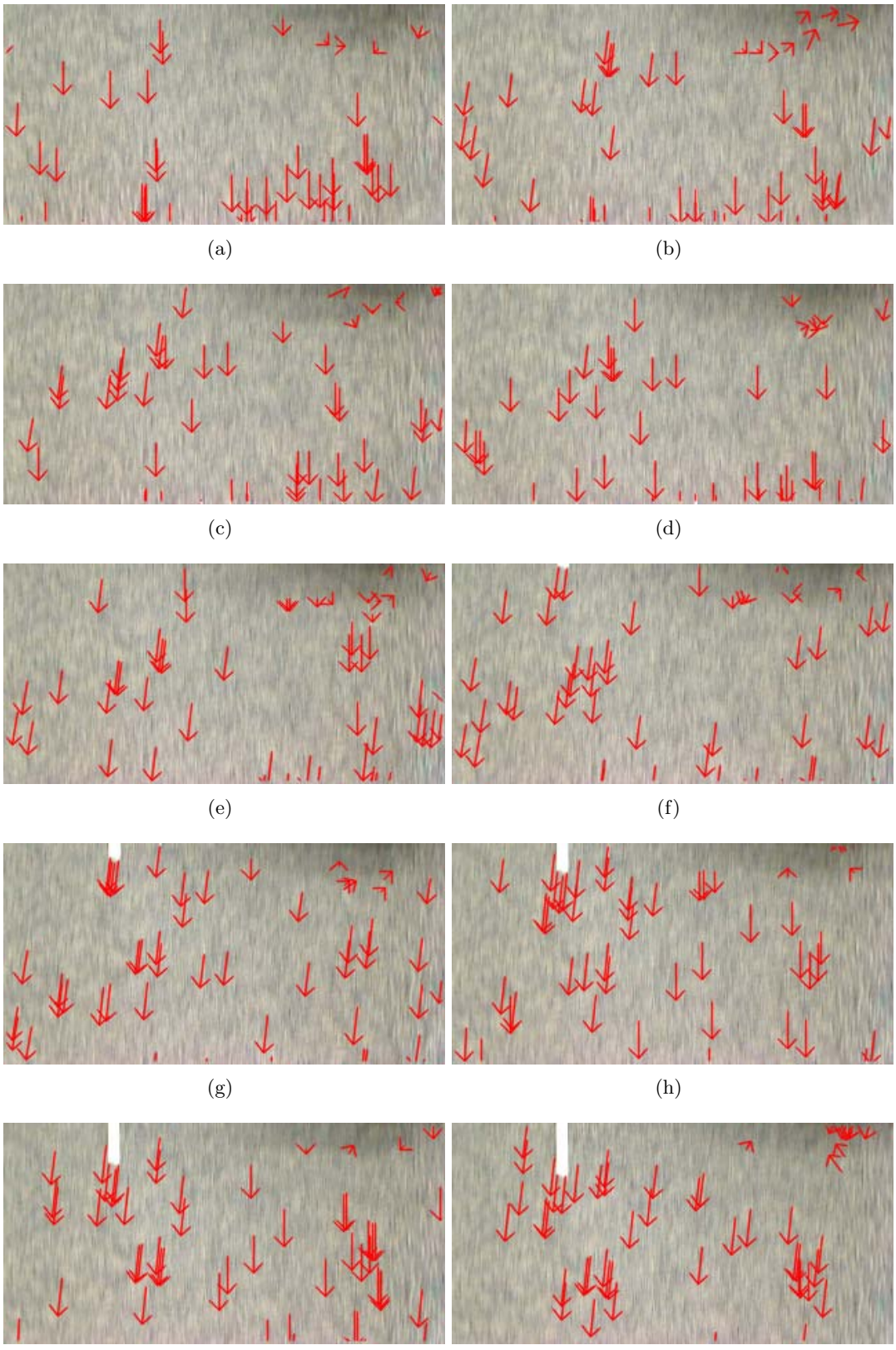
for t=1 to ∞ **do**
  Track and update($c1_x$,$c1_y$) and ($c2_x$,$c2_y$) using KLT and average the differences between the current and previous coordinates in the x and y directions
  Use average differences to update (mc_x,mc_y) and (mr_x,mr_y)
  Unwarp I_{ref} and I_t using the coordinates (mc_x,mc_y) and (mr_x,mr_y)
  for i=0 to width of I_t **do**
    Column-wise shift the unwarped image of I_t
    Compute SAD (for the front 180^o FOV of the robot) between unwarped I_t with unwarped I_{ref} and store score into array
  end for
  Find minimum score using interpolation/extrapolation
  Calculate normalized amplitude, A_n, described by equation 4.1
  if $A_n < T_A$ **then**
    $I_{ref} = I_t$
  end if
end for

Finally combine distance and bearing estimates to track the position of the robot



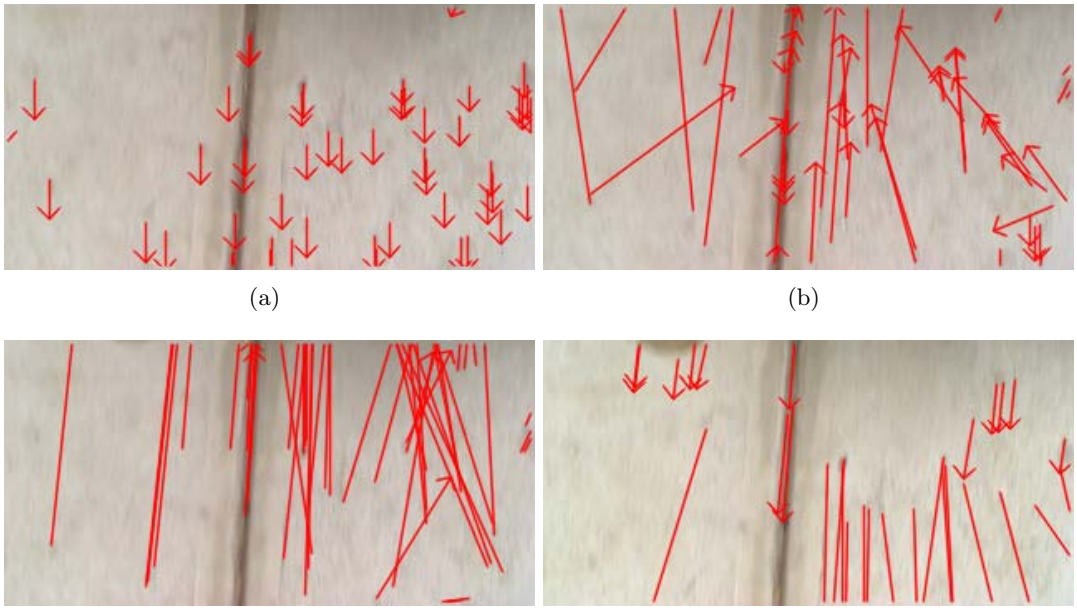
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 4.9: Ground Plane Optical Flow Tracking on Concrete



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

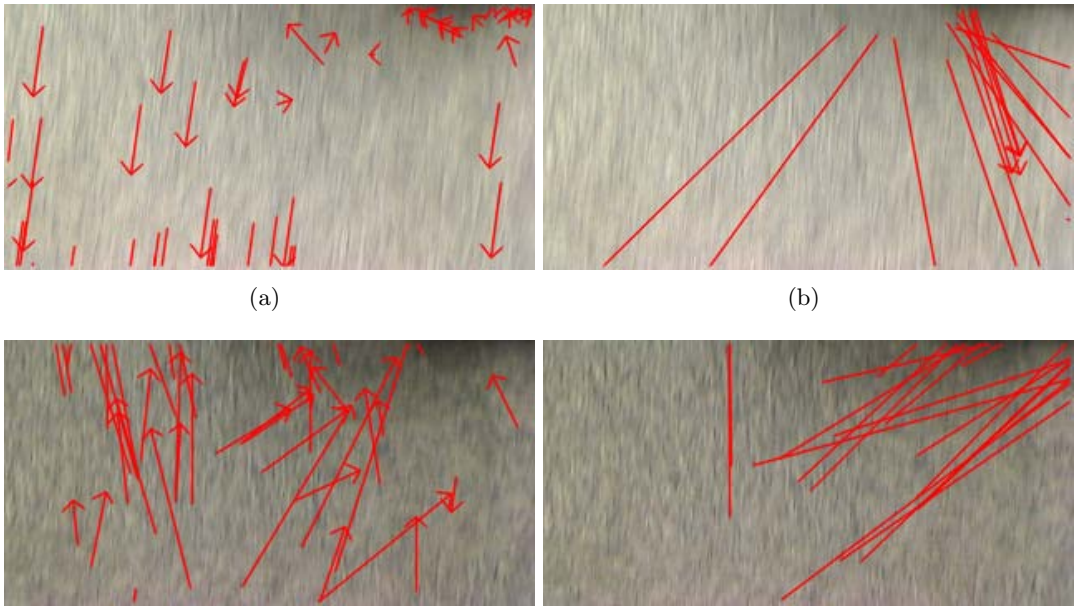
Figure 4.10: Ground Plane Optical Flow Tracking on Carpet



(a) (b)

(c) (d)

Figure 4.11: Ground Plane Optical Flow Tracking on Concrete Failed



(a) (b)

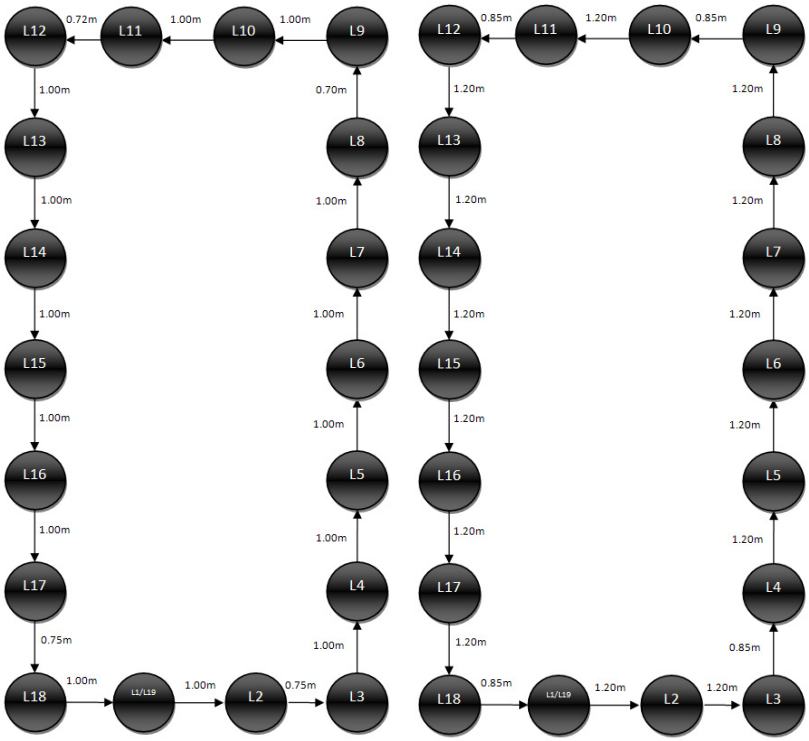
(c) (d)

Figure 4.12: Ground Plane Optical Flow Tracking on Carpet Failed

4.3 Results

The visual odometry system was tested extensively in an indoor lab environment covered with carpet and a semi-outdoor environment with concrete slab paving. The mobile robot was manually driven in a loop fashion starting from node L1 and ending at L19 where node L19 shares the same location with node L1 as illustrated in Fig. 4.13.

Since the location of the mobile robot may deviate slightly from the actual position of each node in the designated trajectory during each experiment, this deviation has to be

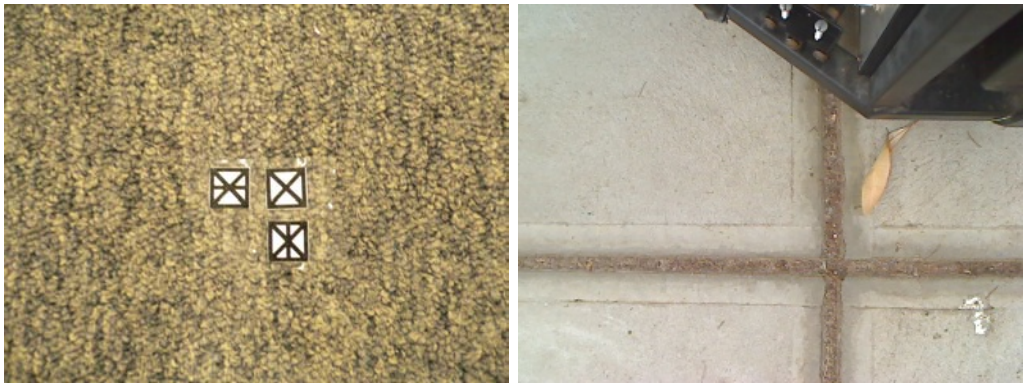


(a) Indoor Environment with Carpet (b) Semi-Outdoor Environment with Concrete Slab Paving

Figure 4.13: The Planned Trajectory of the Mobile Robot for the Visual Odometry Experiments

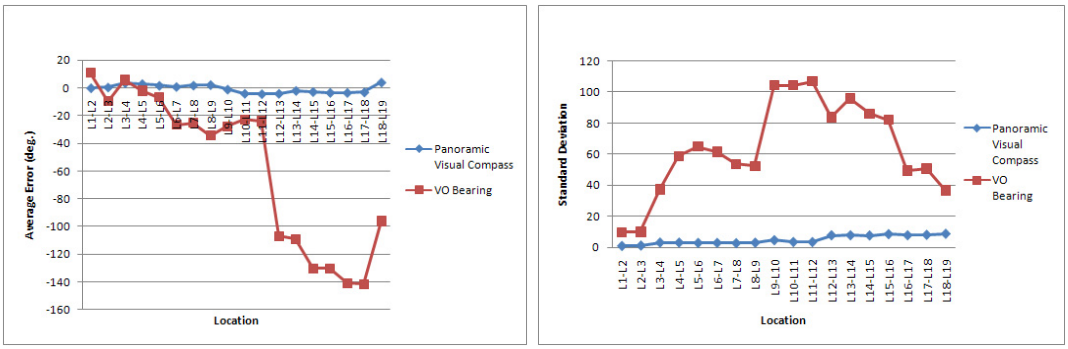
accounted for in the generation of the ground truth locations which the estimated trajectory of the mobile robot would be compared to. In order to account for these differences, artificial markers were strategically placed at each node in the indoor environment (Fig. 4.14(a)) and man-made markers in the semi-outdoor environment (intersection between the concrete slab paving in Fig. 4.14(b)) were utilized. A set of reference locations for these markers in the image, which corresponds to the actual locations of the nodes in the designated trajectory, was recorded before the experiments commenced. Subsequently, a new set of ground truth locations for each node was calculated based on the offset between the current marker's location in the image with the marker's location in the reference image.

The mobile robot was manually driven around in the same direction for a total of 59 times for the indoor experiment and a total of 22 times for the semi-outdoor experiment. For both cases, it started from L1 to L18 and then closing the loop at L19. In this chapter, we will omit the results and discussion in the situation where loop closing occurred until future chapters. The average drift (average distance travelled was 16.95m) for the indoor experiments before loop closing was 5.58%, with an average distance estimate error of 0.044 $\pm$ 0.06m and an average heading error of 0.713 $\pm$ 4^o. For the semi-outdoor experiments, the average drift (average distance travelled was 20.17m) before loop closing was 5.64%, with an average distance estimate error of 0.0147 $\pm$ 0.097m and an average heading error of 0.913 $\pm$ 4.9^o. For more details, please refer to Fig. 4.15-4.17.



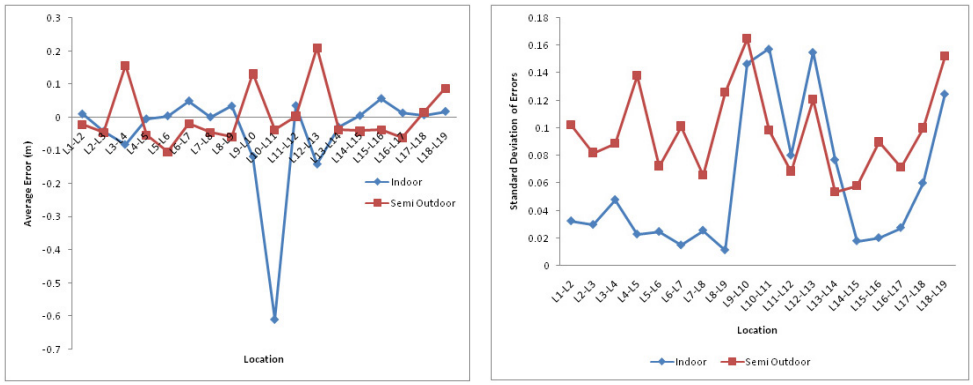
(a) Artificial Markers for Indoor Experiments (b) Existing Man-made Markers for Outdoor Experiments

Figure 4.14: The Planned Trajectory of the Mobile Robot for the Visual Odometry Experiments



(a) Heading estimate error (b) Average standard deviation of errors

Figure 4.15: Performance of the Bearing Estimation



(a) Distance travelled estimate error (b) Distance travelled standard deviation of errors

Figure 4.16: Performance of the Distance Travelled Estimation

Fig. 4.18(a) and 4.18(b) show the ground truth (white) and the estimated trajectory(red and green) of the mobile robot in one of the experimental runs conducted in the indoor and semi-outdoor environment.

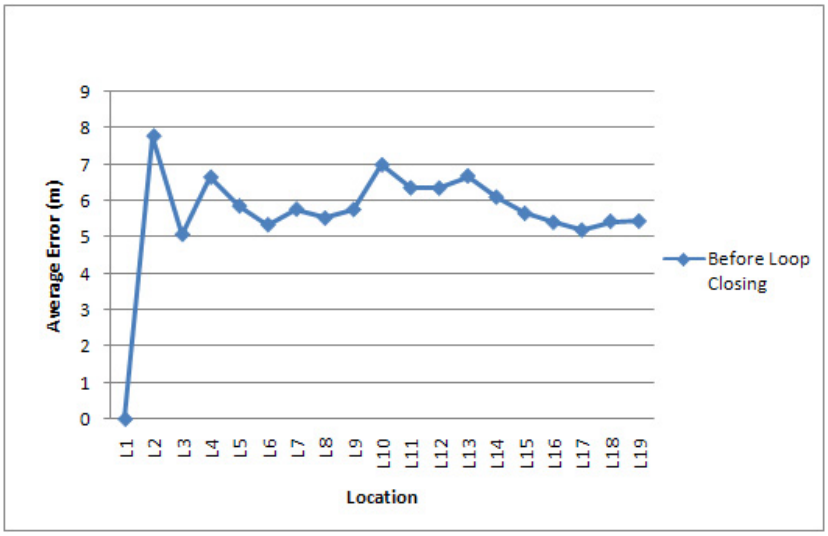
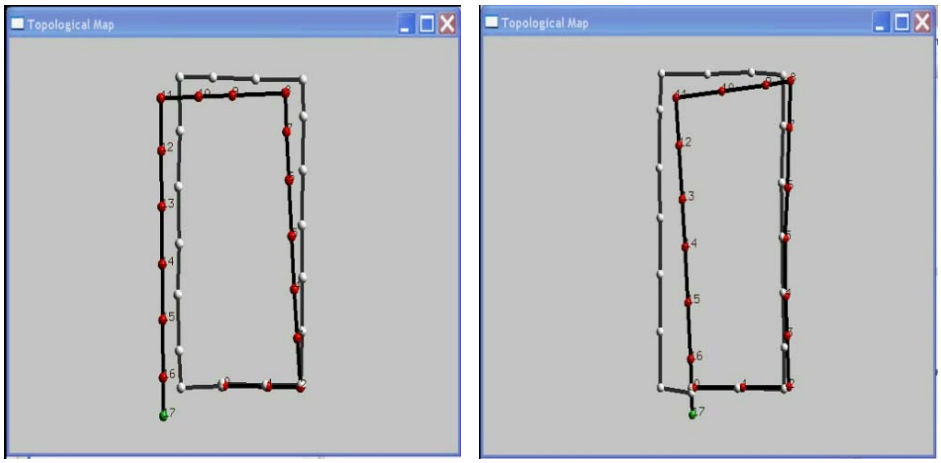


Figure 4.17: Average Drift



(a) Indoor Trajectory (b) Semi Outdoor Trajectory

Figure 4.18: Selected Experimental Runs Comparing Against Ground Truth

4.4 Discussion

It was clearly illustrated in the experimental results that the proposed visual odometry system has yielded satisfactory results and was shown to work reliably in indoor and semi-outdoor environments. Referring to Fig. 4.15, it was experimentally demonstrated that the heading estimate of the mobile robot provided by the ground plane optical flow tracking accumulated a substantial amount of error by the end of each experiment. Take note that the derivation in Appendix C could not be directly applied to the proposed system for heading estimation (θ) since it was assumed that the mobile robot's axis of rotation was aligned to the camera's optical axis. Differential drive mobile robots would normally take pure translational or pure rotational motions since this is the easiest way to control the mobile robot. The axis of a pure rotation would be at the centre of the mobile robot, implying that the only place to mount the camera is the space under the vehicle. This configuration also limits the height of the camera from the ground which generally makes

it difficult for standard web cameras to focus and also issues related to insufficient lighting under the vehicle which causes severe motion blur and creates underexposed images. As a result, it is normal to make sure that the offset between the camera's optical axis and the axis of rotation of the mobile robot is taken into account in the perceived change in rotation due to the issues in aligning these axes. The error associated to this specified offset was identified as one of the main contributors to the drastic accumulation of errors in the heading estimate. Other factors such as the amount of motion blurring due to the the mobile robot's rotational speed, textureless surfaces, severe lighting variation within a short timeframe, subpixel inaccuracies in rotational motion estimation and the sensitivity of the algorithm towards heading estimation using a single camera instead of a dual differential optical flow system contributed to this drastic accumulation of errors in heading estimates. Nevertheless, the modified panoramic visual compass was shown in the experiments to provide reliable heading estimates and justified the fusion of the heading estimated by the panoramic visual compass with the distance travelled estimated by the ground plane optical flow tracking algorithm.

The ground plane optical flow tracking algorithm was found to be more reliable in measuring the two axis displacement of the mobile robot on the ground plane. Nevertheless, its performance generally degrades when the lighting variation between two consecutive frames is too great, presence of moving shadows induced by humans, trees, etc, in the FOV of the camera, severe motion blur and/or when the image is over or under exposed. The effect of lighting variation is illustrated in Fig. 4.12 across four consecutive frames. This was also found to be the reason for the estimated positions of L10 and L11 to be more erroneous than others in Fig. 4.16(a) (due to the drastic change in lighting when the mobile robot travels in and out of an area which was saturated by the fluorescent light located directly above the mobile robot). Although not tested in a fully outdoor environment, it is expected that this system will still work as long as the optical flows are not corrupted by the previously identified issues. This is also assuming that perceptual aliasing is not too severe to affect the appearance-based heading estimate technique and it is a fair assumption to make in general that, outdoor environments will have less problems with aliasing and more features and texture will be present relative to indoor and semi-outdoor environments. Nevertheless, the proposed system is inexpensive, operates in real-time, can be easily used on different mobile platforms, is not affected by wheel slippage and is more tolerant to wheel radius changes due to mechanical imperfections and load variations. Moreover, some of these issues can be resolved by providing consistent lighting to the camera such as an array of LEDs which properly illuminates the area within the FOV of the camera. In addition, with a properly designed platform that includes the array of LEDs, the camera can be mounted in the middle of the mobile platform with its optical axis aligned with the mobile platform's axis of rotation.

4.5 Chapter Summary

Odometry has long been recognized as a key component in the domain of mobile robot localization and mapping. This ability is crucial to partially providing an answer to the

question, "Where am I?". Due to dependence of humans on visual information and the wide availability of inexpensive vision systems, it has inspired many researchers to provide a solution to perform visual odometry. However, visual information acquired from cameras is in a very crude form which does allow direct interpretation of the returned information for a specific application without further processing. Nevertheless, this is also one key advantage of vision systems since it facilitates many high-level interpretations which can be derived using this data alone. Despite so, there may be different ways to derive the robot's odometry by visual means, but for it to be applicable to mobile robots, the visual odometry system has to operate in real time. In this chapter, a visual odometry system, which combines bearing estimates using a direct image approach based on unwarped panoramic images with distance travelled estimates derived from the tracking of ground plane optical flow, was proposed, developed and validated. The proposed solution is purely vision-based and operates in real time. To test the performance of the proposed solution, rigorous experiments were conducted in indoor and semi-outdoor environments, with an additional outdoor experiment conducted for the bearing estimate technique.

5

Mobile Robot Localization and Mapping

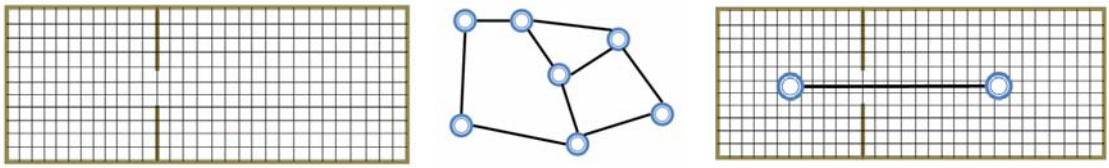
5.1 Introduction

A mobile robot without *a priori* information of the environment is required to explore, build and maintain a globally consistent map. However, during the map building process, the mobile robot is required to localize itself in the environment or simplistically, to know where it is relative to a global position. Unfortunately, localization is normally performed locally and thus, the drift associated with its estimated position increases over time. On the other hand, global localization achieved by means of artificial beacons/landmarks is not practical for many applications where the mobile robot is required to operate in an uncontrolled environment such as search and rescue missions. In addition, many of these applications only require the mobile robot to operate as a once-off process, further making the additional cost associated to the installation of such artificial beacons/landmarks unjustifiable. Similarly, it is not possible for some applications to use the Global Positioning System (GPS) since it restricts the mobile robot to large and open environments. However, if a map is built at the same time, the drift in the mobile robot's estimated position can be contained within some tolerable error bounds by associating the current location of the mobile robot relative to the map and regularly performing loop closing. It has long been recognized that localization and mapping should be performed concurrently. This problem is commonly referred to as Simultaneous Localization and Mapping (SLAM).

As indicated, map building should be performed concurrently with localization in the SLAM paradigm since the mobile robot has to know where it is in order to build a globally consistent map. In applications where the mobile robot is operated as a once-off process, the mobile robot should only be exploring and mapping the environment sufficiently to achieve its objective (e.g. reaching target destination). In this context, the map produced may not be complete and optimal in terms of accuracy, and feasible paths found to the target destination during this process may not be globally optimal. As such, it is always important to strike a balance between exploration and goal seeking behaviours of the

mobile robot. At the other extreme, where the mobile robot is continuously operating in the same environment for a prolonged period of time, other solutions which involve the use of offline maps become more justifiable due to its reliability and simplicity. This solution can be justified by considering the ratio of the total time required to build the offline map and the mobile robot's total operational time in the environment. Whilst the idea of building an offline map, normally through manual or semi-autonomous methods, may not be as impressive when compared to the autonomous SLAM approach, but there are many applications when such an approach is desirable and practical. Nevertheless, SLAM is a prerequisite for mobile robot navigation applications in unknown environments where autonomous map building is desired. Depending on the target application, the resulting map may be fully or partially complete and paths to locations based on the explored environment (map) may be globally optimal or sub-optimal.

Maps built by the mobile robot can either be in the form of a metric map which represents spatial data in fixed or dynamic resolutions, a topological map which represents the explored environment in terms of a linked collection of waypoints/nodes based on some distinctive abstract feature, or a combination of both. For example, Fig. 5.1(a) shows how the plan view of two connecting rooms can be represented using a 2D metric map with fixed resolution and Fig. 5.1(b) illustrates a topological map with several waypoints/nodes. The linkage between the waypoints/nodes may represent the connectivity between neighbouring countries, cities, connecting junctions along a pathway, abstract landmarks or high-level features, etc. As an example for the hybrid map, Fig. 5.1(c) illustrates the topological map being used as a higher level representation of the environment, which allows the environment to be represented in a more compact manner, where each node is associated with the local metric map of the respective rooms.



(a) Metric Map (b) Topological Map (c) Hybrid Map

Figure 5.1: Different Map Representations

The localization of mobile robots can be achieved in many ways. Generally, it requires the acquisition of range measurements using laser, vision or sonar sensors. Subsequently, it involves the tracking of distinctive landmarks such as edges, corners, SIFT or SURF features, etc, which are directly extracted from the acquired sensory data. Finally, this information is normally fused with wheel odometry (Kleeman, 2003; Tungadi and Kleeman, 2009; Angeli et al., 2009), visual odometry (Fernandez and Price, 2004; Campbell et al., 2005; Cheng et al., 2006) or GPS information (Agrawal and Konolige, 2006), which then serve as inputs to a probabilistic framework such as an Extended Kalman Filter (Kalman, 1960; Thrun et al., 2005; Paz et al., 2008; Tungadi and Kleeman, 2009). The fusion of this information is motivated by the uncertainty associated with the location of the

tracked landmarks and the robot's odometry. By incorporating the error models based on the characteristics of the sensors into the probabilistic framework, it is then possible to produce a better estimate of the mobile robot's position and thus, impeding the rate of accumulation of errors. The location of these landmarks are part of the local range measurements which form the local map of the surrounding area. As the landmark's locations and mobile robot's estimated position are refined, these refinements are similarly applied to the local range measurements since they are structurally dependent. As a result, a dense map is built and maintained. However, there are systems that produce sparse feature-based maps instead (Se et al., 2002; Goncalves et al., 2005). For example, in the case of a monocular vision system, distinctive features have to be extracted first from the images before its range can be estimated resulting in a set of sparse features. Of course, if a stereovision system is used instead and dense disparity maps are available, range measurements can be derived from the disparity maps before features are extracted which can be used to produce a denser representation of the environment. For the case where sonar sensors (Kleeman, 2003; Kang et al., 2010) are employed, the resulting local map is normally a set of sparse but accurate edge, corner and/or line features in an indoor environment. Nevertheless, regardless of the sensing modality, it is a matter of time before the drift in the mobile robot's estimated position becomes too large and unfit for navigation. To properly contain these errors within some specific bounds, the mobile robot has to perform loop closing such that the observed discrepancy between the location of corresponding landmarks at the loop closing point can be used to maintain the global consistency of the map. Loop closing or the revisiting of previously explored locations, is of vital importance for building a healthy map and this ability to recognize different locations have always been a key strength in vision systems.

Being able to visually distinguish one place from another is, in fact, a complicated task. Place recognition is an innate capability in human beings. Having said that, it requires years of development, learning and training of the brain in order to facilitate the extraction of salient visual cues where the type of visual cues used are still under investigation. However, once in a while during our adult life, we may still encounter situations when we find it difficult to differentiate whether we have been to this place before a few hours ago. This is especially true when we embark on an overseas trip to unfamiliar grounds. Nevertheless, the human place recognition system still outperforms any man-made system of this era by at least an order of magnitude. In the efforts to unravel the underlying mechanisms which provide us with such a robust place recognition system, neuroscientists have discovered cells in the hippocampus and parahippocampal regions of our brain that responded uniquely at specific spatial locations by increasing their firing rates. These cells were known as place cells and were first discovered by O'Keefe and Nadel (1978) on rodents. In their work, O'Keefe and Nadel (1978) hypothesized that the hippocampus of the rodents was primarily used to form a cognitive map of its environment. This hypothesis has been strongly supported by a lot of subsequent work. In a more recent work by Ekstrom et al. (2003), neural recordings of several human subjects (epilepsy patients)

showed evidence that cells in the hippocampus region responded uniquely to spatial locations whereas cells in the parahippocampal region responded uniquely to landmarks as the subjects navigated themselves in a virtual environment during the trials. This eventually inspired the development of RatSLAM by Milford and Wyeth (2010) and also became our motivation to develop a purely vision-based localization and mapping system.

The visual systems discussed previously (Se et al., 2002; Goncalves et al., 2005) fall into the category of appearance-based approaches. Nevertheless, these systems do estimate the metric locations of salient visual features and build a metric map. Normally, pure appearance-based systems consist of a set of locations, with each location being represented by an appearance model. For such systems, topological maps are highly suitable since each location can be represented as a node in the topological map with the spatial relationship between nodes described by linkages/edges. If the metric locations of salient visual features are estimated, then this information can be stored into the local metric maps associated to each node, effectively building a hybrid map. Going back to the case of a pure appearance-based system, where each location is represented by an appearance model, the system can localize itself to a previously explored area by searching for a previously visited location's appearance model that best matches the appearance model associated to the current location of the mobile robot. However, this scheme may only work provided that *a priori* information of the environment is available to the mobile robot and the mobile robot is also restricted to navigate in this known area. In cases where no *a priori* information is available, it has to be capable to distinguish between a previously seen and new locations in the environment. The advantage of using appearance-based techniques over non-visual techniques is due to them being naturally suitable for performing loop closing, multi-map merging and also solving the kidnapped robot problem (global localization).

As described previously, appearance-based techniques associate previously seen locations with their corresponding appearance models, making them equivalent to place recognition systems and closely related to image retrieval systems. The appearance model to employ depends on the target application, since different models have different strengths and weaknesses. The most convenient approach is to select an appropriate colour space and employ this as the appearance model. Images are then compared using image similarity metrics such as the Euclidean, sum of squared differences or sum of absolute differences metrics. However, such approaches are very primitive; prone to illumination changes; are scale, rotation and translation variant; and require large amounts of memory as the image database increases in size. There are other methods such as the use of image histograms and Principal Component Analysis (PCA). Image histograms (Ulrich and Nourbakhsh, 2000) are rotation and translation invariant. However, their matching accuracy is affected by illumination changes and its maximum database capacity is limited by the number of bins of the image histograms. The number of bins also determines the total memory required by each image in the database. Nevertheless, a 64 bin image histogram generally requires a small amount of memory. On the other hand, PCA, which initially gained popularity in the domain of face recognition (Turk and Pentland, 1991; Hsieh and Tung,

2009), reduces the high-dimensional data onto low-dimensional subspaces. A set of principal components (eigenvectors) is used to represent each image and also for measuring the similarity between images. Linear PCA (Kröse et al., 1999) is rotation, translation and scale invariant; however, its performance is also affected by illumination changes. As such, both appearance models are better suited for indoor environments. Nevertheless, there are other appearance models that were experimentally proven to be more robust towards illumination changes and occlusions such as the one described in (Zhang and Kleeman, 2009) that performs image cross correlation in the Fourier domain on patch-normalized omnidirectional images and the use of Haar coefficients in (Ho and Jarvis, 2008). In addition, a new paradigm known as the bag-of-visual words has recently become a popular model among researchers for appearance-based systems. This paradigm normally uses a combination of visual cues extracted from the image and builds a visual dictionary such as the ones proposed by Cummins and Newman (2008) and Angeli et al. (2008). For a complete localization and mapping system, this is normally combined with odometry and/or GPS information which serve as inputs to a probabilistic framework.

It has been acknowledged that a probabilistic framework is the appropriate solution to deal with the problem of non-ideal sensory data received by the mobile robot. It is an elegant solution because it provides a way to measure and account for the uncertainty in the location of the mobile robot and features in the map, allows the performance of the mobile robot to gracefully degrade as uncertainty becomes prevalent and provides a general platform for multisensor fusion. There are many different approaches to a probabilistic framework (Kalman filters, particle filters, etc (Thrun et al., 2005)) but in general, these frameworks are based on Bayesian reasoning. As such, it is not surprising to find Bayesian reasoning being applied to the derivation of an appearance-based localization system on mobile robots. An initial attempt to develop a probabilistic appearance-based localization system was reported in (Kröse et al., 1999). *A priori* information of the environment, in the form of PCA features extracted from grayscale panoramic images, was required by the system. These panoramic images were acquired from respective positions in the environment, where this environment could be visualized as a 2D grid map divided into resolutions of 10x10cm. Based on this information together with a nearest neighbour model, the distribution which describes the probability of the occurrence of a set of one dimensional features for every location in the environment, was derived. Subsequently, the posterior probability distribution, which describes the probability of the current mobile robot's location to be within a particular grid/cell in its metric map given a set of features extracted from the image, was derived by applying Bayes' Theorem. The mobile robot's location could then be determined, in an indoor environment, by finding the location associated to the index having the maximum posterior probability. Similarly, Ho and Jarvis (2008) proposed an appearance-based localization system, using the Haar coefficients as the appearance model, in a particle filter framework which required *a priori* information of the environment. The *a priori* information provided was in the form of a database of image signatures, created from Haar decomposed synthetic panoramic images. These synthetic panoramic images were generated from the extensive 3D virtual model of the

world built using a Riegl Z420i terrestrial laser scanner. Unfortunately, a mobile robot with no prior knowledge of the environment is required to differentiate whether a location is new or previously visited.

It was only recently that this problem in appearance-based localization and mapping systems was properly addressed using a complete probabilistic framework by two independent research groups (Cummins and Newman, 2008; Angeli et al., 2008). Both groups employed the bag-of-visual words as the appearance model to represent raw image data as a collection of attributes/words chosen from the vocabularies in the dictionary. However, there are major differences between the two systems in several aspects such as the construction and maintenance of the visual dictionary and the way that the problem was formulated and resolved. Cummins and Newman (2008) formulated this problem as a recursive Bayes estimation problem and was able to provide a measure on the probability of the mobile robot being at a previously visited location or new location. Ideally, to provide an accurate measure of the mobile robot's likelihood to being at a new location would imply the evaluation against all possible unknown places. This is not practical due to (a) it is not possible to have a database of all unknown places and (b) the computational cost associated to this process is expected to be very expensive. In order to resolve this issue and provide a reasonable estimation, a sampler, which randomly selects from a large collection of observations from street-side imagery (Google Street View or previous runs), was used to sample an observation for building the place model to represent the unmapped locations. For optimal performance, a Chow Liu tree which captures the co-occurrence statistics of visual words (SURF features) was constructed from a large number of observations. This process was reported to involve 2,800 images from 28km of urban streets and the entire process took 2 hours on a 3GHz Pentium IV. Nevertheless, the system still performed reasonably well even if standard dictionaries were used. On the other hand, Angeli et al. (2008) formulated this as a loop closure detection problem and solved it via a Bayesian framework which requires the creation of a virtual image, using a number of most frequently seen words (SIFT features and local colour histograms), to represent the appearance model of all unmapped locations. This method was reported being fast and incremental as the construction of the visual dictionary (standard tree structure) was performed as an online process as new images were acquired by the system.

Due to the appealing nature of vision-based systems for localization and mapping, specifically its suitability for performing loop closing, map merging and solving the kidnapped robot problem, we have decided to develop a mobile robot which performs visual SLAM. Our mobile robot will be equipped with the ability to recognize places based on image signatures created by applying the standard 2D Haar decomposition on omnidirectional images. It was chosen over other appearance models due to it being algorithmically simple, scalable and yet robust. As the mobile robot explores the environment, a topological map, with relative metric information between nodes provided by the visual odometry system, is built and a relaxation algorithm is employed to maintain the global consistency of the map.

5.2 Place Recognition using Haar Wavelets

Wavelets have gain tremendous success in a wide range of applications in the field of signal processing, approximation theory and computer graphics. The Haar wavelets is one of the simplest forms of wavelets and have been thoroughly described by Stollnitz et al. (1995). This simple and elegant decomposition technique has also been proven suitable for use in an image retrieval system (Jacobs et al., 1995). As described in (Stollnitz et al., 1995), both the standard and non-standard 2D Haar decomposition can be used to decompose images. The standard decomposition is easier to implement as compared to its non-standard counterpart due to it being algorithmically less complicated. However, the standard decomposition is also slightly more computationally expensive as compared to the non-standard decomposition. In addition, the standard basis functions are rectangular whereas the non-standard basis functions are square, which effectively makes the latter more suitable for features with the same aspect ratio. Jacobs et al. (1995) built a database of image signatures by keeping the average colour and top m coefficients (quantized values) of the Haar decomposed images and proposed an efficient image querying metric that could robustly match the decomposed query image to similar images in the database. The standard 2D Haar decomposition was chosen instead of the non-standard version since it was found to work better with the type of images being dealt with. In addition, the effects to matching accuracy due to the use of different colour spaces, the number of top m coefficients and different image metrics were validated by experiments. The following summarizes the algorithm proposed by Jacobs et al. (1995),

1. Build a database of image signatures. For each image, perform the following operations,
   (a) Convert to YIQ colour space
   (b) Decompose image using the standard 2D Haar decomposition described in Algorithm 5.1
   (c) Truncate the sequence to keep only the average colour and top 60 coefficients for each channel
   (d) Quantize them to just two levels (+ve and -ve)
2. Using a set of training images, determine a set of weights for the image querying metric using the logistic regression model.
3. For each query image,
   (a) Repeat steps 1(a) to 1(d)
   (b) Compute similarity score with all images in the database using the image querying metric and weights
   (c) Rank the scores in ascending order and the matching image should be ranked within the top 1% of the returned scores.

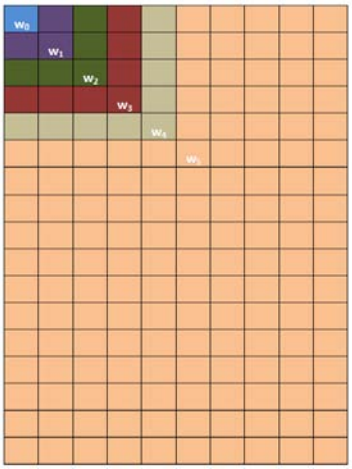


Figure 5.2: Weights and Bins

The proposed image querying metric efficiently compares the image signature of the query image Q with a target image T in the database and is expressed by the following equation,

$$w_0|Q[0,0] - T[0,0]| - \sum_{i,j:Q(i,j)\neq 0} w_{bin(i,j)}(Q[i,j] == T[i,j]) \tag{5.1}$$

where w are the weights determined using a set of independent training images via the logistic regression model, i and j represent the location of the coefficient in the decomposed image with the overall average intensity of the channel stored in the location $i = j = 0$ and $w_{bin(i,j)}$ is the weight which corresponds to the bin number that can be expressed as,

$$bin(i,j) = min(max(i,j),5) \tag{5.2}$$

The weights associated to each bin can be visually represented as in Fig. 5.2. These bins group the coefficients depending on its location in the decomposed image and its corresponding weights bias the final similarity score based on the locations of matching coefficients.

Ho and Jarvis (2008) adapted the system proposed by Jacobs et al. (1995) for panoramic images by converting the image into YIQ colour space, downsampling the original unwarped image to a size of 512 x 128 and retaining the coefficients within a bounding box of size 64 x 16 originating from the (0,0) coordinate of the decomposed image. Similarly, the magnitudes of the Haar coefficients were quantized but were conveniently stored into a bit array in order to reduce the memory footprint for each image signature (location of the coefficient in this case was not required). Since the Haar wavelets are rotation variant, each unwarped panoramic image was column-wise shifted every 10 degrees equivalent in pixels and decomposed, quantized and stored in the database. Our system employs the algorithm proposed by Ho and Jarvis (2008) since it has been adapted to work with

Algorithm 5.1 Standard 2D Haar Decomposition Algorithm for a Single Channel Image

Haar Decomposition for a Single Channel Image

Input: Query image Q with resolution of h x w

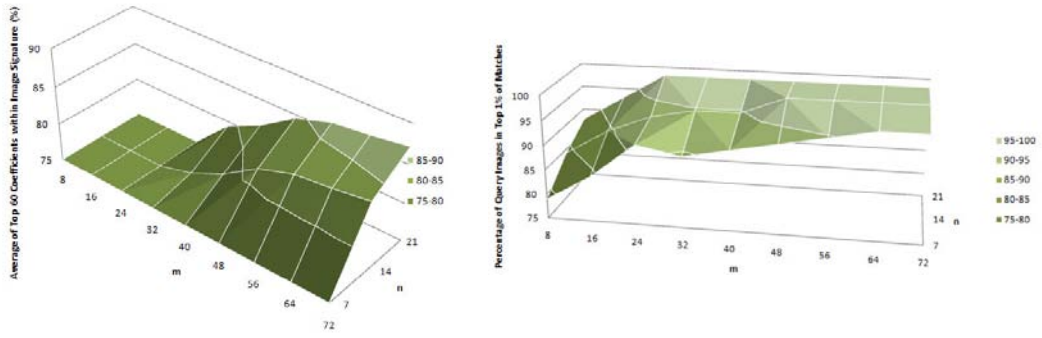
1: **for** each row $i = 1$ to h **do**
2: DecomposeArray($Q(i, 1...w)$)
3: **end for**
4: **for** each col $j = 1$ to w **do**
5: DecomposeArray($Q(i, 1...w)$)
6: **end for**

DecomposeArray Function

Input: 1D array A with e elements

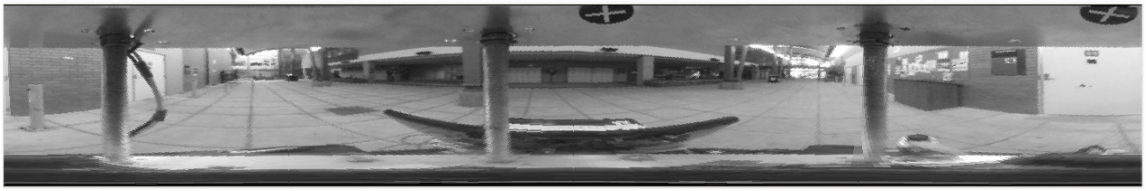
1: $A \leftarrow A/\sqrt{e}$
2: **while** $h > 1$ **do**
3: $e \leftarrow e/2$
4: **for** $i = 1$ to e **do**
5: $A'[i] \leftarrow (A[2i] + A[2i+1])/\sqrt{2}$
6: $A'[e+i]] \leftarrow (A[2i] - A[2i+1])/\sqrt{2}$
7: **end for**
8: $A \leftarrow A'$
9: **end while**

panoramic images. It creates image signatures of size 56 x 14 instead of 64 x 16. This modification was made after a total of 335 panoramic images were used to find the average number of top 60 coefficients within the bounding box of size m by n. Fig. 5.3 shows the effect of matching accuracy with different bounding box sizes using 57 query images for a database with 2052 image signatures. With these quantitative results, an image signature of size 56 x 14 was chosen since it contains an average of 82.8% of the top 60 coefficients and performing at an average of 98.2% to rank the correct image signature in the database in the top 1% of all returned matches for the 57 query images. Sample panoramic images taken from a semi-outdoor and outdoor environment, decomposed using the standard 2D Haar decomposition algorithm without downsampling, are illustrated in Fig. 5.4 and 5.5.



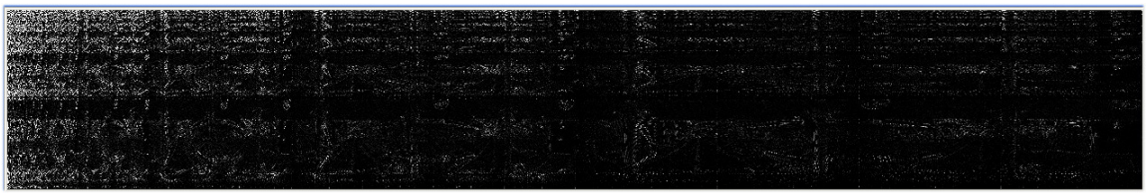
(a) (b)

Figure 5.3: (a) Average Top 60 Coefficients and (b) Average Top 1% Matches using Bounding Box of Size m by n



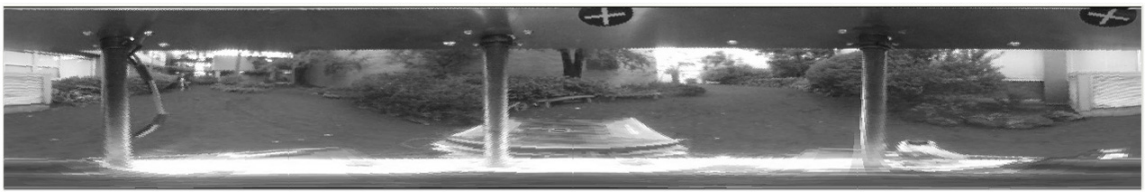
(a) Unwarped Grayscale Image

(b) After Row Decomposition



(c) After Column Decomposition

Figure 5.4: Standard Haar Decomposition on A Semi-Outdoor Unwarped Panoramic Image



(a) Unwarped Grayscale Image

(b) After Row Decomposition

(c) After Column Decomposition

Figure 5.5: Standard Haar Decomposition on An Outdoor Unwarped Panoramic Image

5.3 Maintaining the Global Consistency of Topological Maps

Metric mapping systems based on the Kalman filter update its state vector, which consists of the pose of all landmarks and the mobile robot, and its covariance matrix during the update phase of the filter. This essentially maintains the global consistency of the metric map, since the inconsistency with the observation and association of landmarks in adjacent locations of the mobile robot or during loop closure is properly distributed to all previously observed landmarks. Similarly, the global consistency of the map built by a topological mapping system, with relative spatial information between nodes, has to be maintained due to inconsistencies in sensor measurements. A common model is to conceive these links between nodes as springs (Fig. 5.6) and the objective to maintain the global consistency of the topological map is achieved by minimizing the total energy of the network of springs.

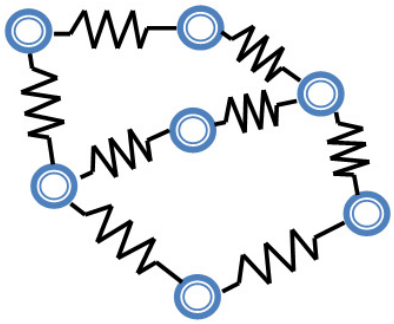


Figure 5.6: Topological Map with Links Conceived as Springs

There are several notable works in this area. Lu and Millios (1997) outlined a solution to this problem with the intention to maintain the global consistency of metric maps via a topological structure. In their approach, a metric map was constructed through the registration of overlapping local laser/sonar scans with the spatial relationship between local scans obtained by means of the mobile robot's odometry or recovered by performing scan-matching on adjacent scans. At the same time, a topological network, where each node in the network was associated to a local scan with its links representing the spatial relationship between local scans, was formed. Then, the problem of maintaining the global consistency of the map was posed as an optimization problem and was solved by minimizing the total energy of an objective function with variables defined by all the pose coordinates in the network and the link between each node conceived as a spring. Basically, the energy of a spring would be minimum when the relative pose between two nodes in the map was the same with the measurements. The pose of the nodes were then refined by minimizing the Mahalanobis distance between the measured and observed pose differences over the entire network.

Similarly, Golfarelli et al. (1998) proposed an error correction for topological maps based on the analogy of a mechanical system. The topological map was visualized as a truss with each landmark represented by a node in the topological map with links between

nodes modelled as an elastic bar. These elastic bars between nodes could be visualized as springs (combination of linear axial and rotational springs) which connected one landmark to another. Finally, the deformations induced by the inconsistency of measurements made could be corrected by finding the equilibrium position (where the total energy of all springs achieved global minimum) of the whole structure.

Unfortunately, both methods (Lu and Millios, 1997; Golfarelli et al., 1998) were required to invert a large matrix (a computationally expensive process). This eventually motivated Duckett et al. (2000) to develop an online incremental algorithm to maintaining a globally consistent topological map known as the relaxation algorithm. Similar to previous works, links in the topological map would contain relative metric information between the two connected nodes in the form of (d_{ij}, θ_{ij}), where d_{ij} is the relative distance and θ_{ij} is the relative orientation between nodes i and j. In addition, links in the topological map were constrained to be bidirectional. This implied that for every two spatially connected nodes, its respective link would contain the relative metric information from node i to j described by (d_{ij}, θ_{ij}) and the relative metric information from node j to i described by (d_{ji}, θ_{ji}) where $d_{ij} = d_{ji}$ and $\theta_{ij} = \theta_{ji} + \pi$ (given that the relative orientation was measured in radians). Finally, the uncertainty in the position of each node in the topological map was modelled according to a Gaussian distribution, effectively distributing the uncertainty in sensor measurements equally in all directions in the Cartesian space surrounding the node. This could be represented as a single variance measure, u_{ij}, which could be visualized as a circle surrounding the node. The value of this variance was set as 5% of the total distance travelled by the mobile robot in the system described by Duckett et al. (2000).

Due to its simplicity and efficiency, this algorithm is naturally suitable for mobile robot applications. Consequently, this algorithm was employed by our system for maintaining a globally consistent topological map. The following two-step procedure, originally outlined in (Duckett et al., 2000), is reproduced here for completeness.

Step 1 - For each of the neighbours j of node i, an estimate (x'_{ji}, y'_{ji}) of the coordinate of node i is obtained using,

$$x'_{ji} = x_j + d_{ji} cos\theta_{ji} \tag{5.3}$$

$$y'_{ji} = y_j + d_{ji} sin\theta_{ji} \tag{5.4}$$

where the coordinate of j is denoted by (x_j, y_j), and the variance v_{ji} in this estimate is obtained using

$$v_{ji} = v_j + u_{ji} \tag{5.5}$$

where v_j refers to the variance for the node j and u_{ji} to the variance for the link from j to i.

Step 2 - The position estimates (x'_{ji}, y'_{ji}) for all j are then combined to produce a new coordinate for node i. Firstly, the new variance v_i for node i is calculated as,

$$v_i = 1/\sum_j{}' \frac{1}{v_{ji}} \tag{5.6}$$

where $\sum_j{}'$ refers to the sum over all neighbours of node i. The new coordinate (x_i, y_i) is then obtained by calculating the mean of the estimates (x'_{ji}, y'_{ji}) weighted by $1/v_{ji}$ as,

$$x_i = \sum_j{}' \frac{x'_{ji} v_i}{v_{ji}} \tag{5.7}$$

$$y_i = \sum_j{}' \frac{y'_{ji} v_i}{v_{ji}} \tag{5.8}$$

This algorithm has been proven to converge and more details regarding the proof of convergence can be found in Duckett (2000).

5.4 Appearance-based Localization and Mapping

As indicated before, appearance-based localization and mapping systems are highly compatible with topological maps, since each node in the map represents a unique location in the spatial domain that is associated with an appearance model of the location and the links connecting the nodes specify the spatial relationship between them. The main advantage of such systems over non-visual metric mapping systems is due to it being naturally suitable for solving the multi-map merging problem, the kidnapped robot problem and, particularly, the loop closing problem. Loop closing is an integral part of autonomous map building systems since inconsistent onboard sensor measurements produce inconsistent maps over time due to accumulated errors in the robot's estimated pose. The ability to detect which previously visited location the mobile robot is currently at, which then facilitates the detection of loop closures, was achieved by comparing the Haar decomposed image signatures described in Section 5.2. However, a complete appearance-based localization and mapping system without *a priori* knowledge of the environment is also required to discriminate between whether this is a new or previously explored location. In addition, if the mobile robot finds that it is revisiting an explored location, it has to recover the relative metric transformation of the current location of the mobile robot with respect to the matching node's location in the topological map since it is highly unlikely for the mobile robot to be precisely at the exact location of the matching node. The following describes two complete appearance-based localization and mapping frameworks which is then followed by the description on how the relative transformation between the current and matching locations is recovered when the mobile robot revisits an explored location.

5.4.1 Rank-based Framework

The rank-based framework operates directly on the weighted scores computed using Equation 5.1. An image signature, I_1, in the database is considered more similar to the current input query image signature, Q, than another image signature, I_2, in the database if the returned weighted score for the I_1-Q pair is smaller than the weighted score for the I_2-Q pair (I_1-$Q < I_2$-Q). This framework basically ranks the weighted scores in ascending order, where each weighted score is the comparison of the input query image signature with an image signature in the database, and selects the top S matches that are smaller than a threshold, T_R. In addition, for a match to qualify as a probable candidate, the location of the matching node has to be within the boundary of the circle centred in the current position of the mobile robot with a radius defined by the variance measure described in Section 5.3. Each candidate, which satisfies these conditions, is also associated with a pair of omnidirectional images taken at the point in time when the node is created in the topological map. Using the unwarped bottom omnidirectional image of the candidate and the unwarped bottom omnidirectional image taken at the current location of the mobile robot, image correspondences are established by matching SURF features (Bay et al., 2006). Only candidates that contain more than T_S number of similar SURF features detected in the current unwarped bottom omnidirectional image will be retained in the pool of final candidates. The rank-based framework is summarized into a flowchart shown in Fig. 5.7.

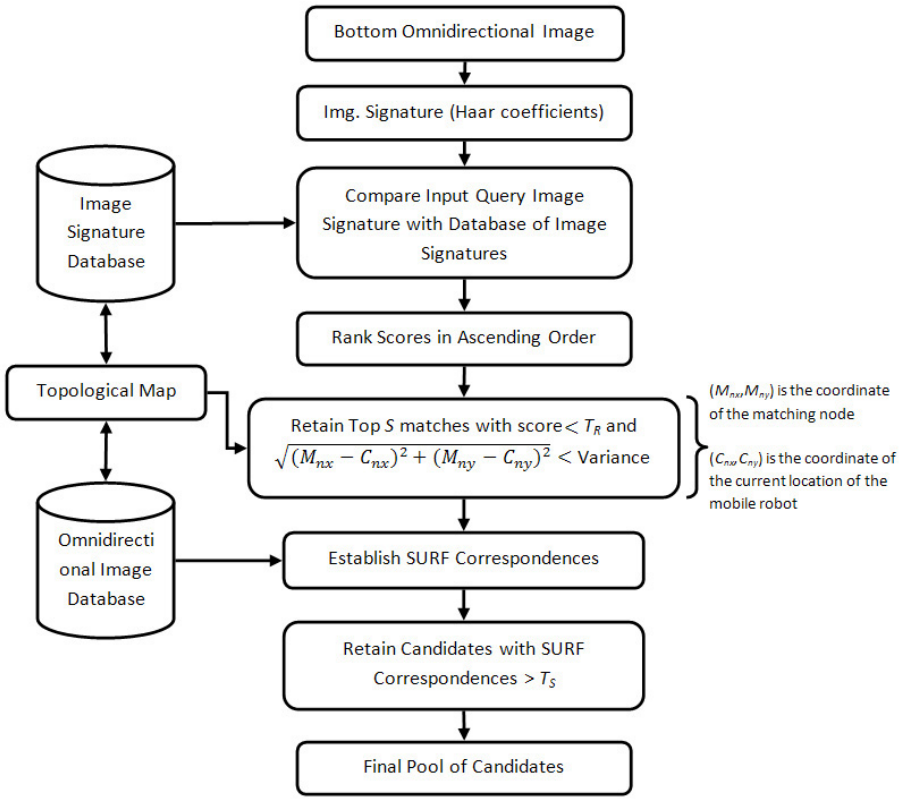


Figure 5.7: Flow Chart of Rank-based Framework

5.4.2 Probabilistic Framework

The proposed probabilistic framework was developed based on the incremental and online loop closing detection system proposed by Angeli et al. (2008), which employed the discrete Bayes filter. It was originally proposed by Angeli et al. (2008) to maintain temporal coherency and reduce transient detection errors which give rise to false positive detections. However, several modifications were made due to some significant differences in system characteristics. Firstly, Haar coefficients of the omnidirectional images were used to discriminate between the images instead of the bag-of-words approach using perspective cameras. Then, omnidirectional images were captured intermittently (i.e. after the mobile robot has travelled a considerable amount of distance from the previous node) instead of using a continuous stream of video images. Moreover, the topological relationship between these images were maintained in a bidirectional graph structure.

The problem is defined as the following. Given a set of nodes, $n_s = 0, 1, ..., t$, each associated with an omnidirectional image in the sequence $I_0, I_1, ..., I_t$, compute the probability of the mobile robot being at a previously explored node and the probability of it being an an unmapped/yet to explore location. The variable S_t, is used to describe the hypothesis that at time t, the mobile robot is at a previously visited node if $S_t = j$, where j denotes the index of an existing node in the topological map, or a new location if $j = -1$. In a Bayesian framework, this is equivalent to searching for the past image I_k where the index k, is derived by using the following expression,

$$k = \underset{j=-1,...,t-e}{argmax}\ p(S_t = j|I^t) \tag{5.9}$$

where $I^t = I_0, I_1, ...I_t$. Subsequently, by using Bayes rule and the Markov assumption, the full posterior can be decomposed into,

$$p(S_t|I^t) = \eta p(I_t|S_t)p(S_t|I^{t-1}) \tag{5.10}$$

where η is the normalization factor and $p(I_t|S_t)$ is the likelihood $\mathcal{L}(S_t|I_t)$ of S_t given image I_t. By marginalizing the belief distribution, $p(S_t|I^{t-1})$, it can then be expressed as,

$$p(S_t|I^{t-1}) = \sum_{k=-1}^{t-e} \underbrace{p(S_t|S_{t-1} = k)}_{state\ transition} \underbrace{p(S_{t-1} = k|I^{t-1})}_{prior\ posterior} \tag{5.11}$$

where e is the number of most recently visited nodes to be excluded and the prior initialized to $p(S_{t-1} = -1|I^{t-1}) = 1$. This decomposition is simply elegant because the state transition model can be used to maintain temporal coherency and reduce transient errors due to perceptual aliasing and the prior posterior is readily available from the previous time step. The following describes the new state transition model and the likelihood voting scheme. These modules were redefined, resulting in a set of general expressions. As

a result, the proposed framework can be easily adapted to various systems with different characteristics such as those mentioned previously (i.e. convergence speed, image similarity metrics). For more details on how the state transition probabilities and likelihood voting scheme were modelled, please refer to Appendix D. A flowchart of the probabilistic framework is provided in Fig. 5.8.

a State Transition Model

The state transition probabilities were modelled according to our system's characteristics which were highlighted previously. The state transition probabilities can be described as follows (note that the value of e is ignored until $t \geq e$ since it is not possible to exclude more nodes than what is available),

- $p(S_t = -1|S_{t-1} = -1) = \frac{2.0}{t-e+2.0}$ if the system believes it was at a new location in the previous time step ($SI = -1$) or $\frac{1.5}{t-e+1.5}$ otherwise: describes the probability that the mobile robot is at a new location at time t, given that it was previously at a new location at time $t-1$.

- $p(S_t = j|S_{t-1} = -1) = \frac{1-p(S_t=-1|S_{t-1}=-1)}{t-e+1.0}$, $j \in [0, t-e]$: describes the probability that the mobile robot is currently at node j, given that it was previously at a new location at time *t-1*.

- $p(S_t = -1|S_{t-1}) = k = 1 - \frac{N_k+1}{N_k+2}$ if the system believes it was at new location in the previous time step (SI = -1) or 0.2 otherwise, $k \in [0, t-e]$: describes the probability that the mobile robot is currently at a new location, given that it was previously at node k at time $t-1$.

- $p(S_t = j|S_{t-1} = k)$, where $j \in [0, t-e]$ and $k \in [0, t-e]$: describes the probability that the mobile robot is currently at node j, given that it was previously at node k at time *t-1* can be expressed as,

$$\frac{1.6}{N_k+2.0} : SI > -1, j = k \tag{5.12}$$

$$\frac{2N_k+2.0}{(N_k+2.0)^2} : SI = -1, j = k \tag{5.13}$$

$$\frac{0.8-\frac{1.6}{N_k+2.0}}{N_k} : SI > -1, j \in \text{neighbours of k} \tag{5.14}$$

$$\frac{0.8-\frac{2N_k+2.0}{(N_k+2.0)^2}}{N_k} : SI = -1, j \in \text{neighbours of k} \tag{5.15}$$

where N_k represents the total number of neighbours of node k, which can be found using the topological map, and SI is the selected index (largest probability) from the prior posterior, $p(S_{t-1} = k|I^{t-1})$, which represents the actual state of the system being at an unexplored location if $SI = -1$ or previously visited location if $SI > -1$ in the previous time step.

b Likelihood Voting Scheme

The likelihood, $\mathcal{L}(S_t|I_t)$, is computed by finding a subset $H_t \subseteq I^{t-e}$ of images whose scores, D_i ($i \in [-1, t-e]$)), are smaller than the threshold computed using the mean of all scores, μ_a, minus its standard deviation, σ (smaller Haar scores representing a better match). At the same time, the mean of all inliers, μ_{in}, which represents the average score of those larger than the threshold, is computed. Subsequently, $\mathcal{L}(S_t|I_t)$ is expressed as,

$$\mathcal{L}\left(S_t|I_t\right) = \begin{cases} \frac{(D_i-\mu_{in})^B+\mu_{in}}{\mu_{in}} & : D_i \leq \mu_a - \sigma \\ 1.00 & : \text{otherwise} \end{cases} \tag{5.16}$$

Given that if $D_i - \mu_{in} >= E$ is considered a discriminative score for a matching node and given that $E^B p(S_t = j|S_{t-1} = -1) = F$, then B can be expressed as,

$$B = \frac{log(F/p\left(S_t = j|S_{t-1} = -1\right))}{log(E)} - H \tag{5.17}$$

where H is basically an offset of the power factor B when a loop closure was detected in the previous time step. If the loop closure detected was indeed a true event, there is a higher probability for new locations in the vicinity exhibiting similar appearance. Due to the previously detected loop closure, there is also a higher probability for the system to be matching to an existing node in the map (e.g. possibly neighbour of the previously detected loop closure) given sufficient similarity in the location's visual appearance. As such, this offset ensures that false positives are reduced for such scenarios and a discriminative matching score is required for the system to remain in the state of loop closure detected from the previous to the current time step. H is currently defined as,

$$H = \begin{cases} 0.3 & \text{:j=-1 and SI>-1} \\ 0 & : \text{otherwise} \end{cases} \tag{5.18}$$

In Equation 5.17, E defines what is deemed as a discriminative score for a matching node and the combination of E and F affects the convergence speed of this framework. In (Angeli et al., 2008), a virtual image, I_{-1}, is created and maintained such that it is statistically more likely for this virtual image to match the incoming query image if the mobile robot is currently at an unexplored location. However, for our system, the score of this virtual image is calculated by subtracting the mean, μ_a, with G times the standard deviation, σ. As such, $\mathcal{L}\left(S_t = -1|I_t\right)$ is expressed as,

$$\mathcal{L}\left(S_t = -1|I_t\right) = \mu_a - G\sigma \tag{5.19}$$

Finally, the full posterior, $p\left(S_t|I^t\right)$, is computed and subsequently normalized. Similar to the rank-based framework, the probabilistic framework ensures the matching node to be

within the boundary of the circle centred in the current position of the mobile robot with a radius defined by the variance measure and the total number of SURF correspondences is above the threshold T_S if the matching node corresponds to a previously visited location. The probabilistic framework can be summarized into the flowchart illustrated in Fig. 5.8.

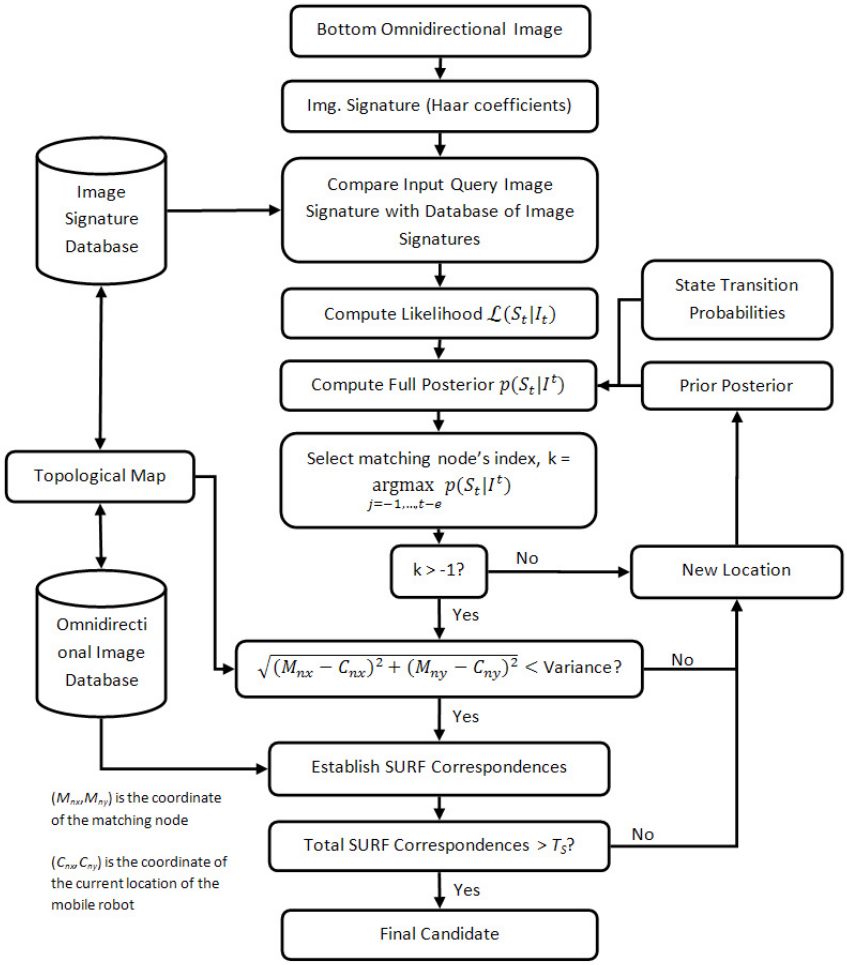


Figure 5.8: Flow Chart of Probabilistic Framework

5.4.3 The Revisiting Problem

Regardless of whether the rank-based or probabilistic framework was used, the final output should either be one or more candidates qualifying as probable matching nodes, implying that the mobile robot was revisiting a previously explored location, or the other case where there were no qualifying candidates which suggests that it was at a new location. In the case where the mobile robot was believed to be revisiting a location, the rank-based framework could produce one or more candidates whereas the probabilistic framework would only provide a single candidate. Since it is highly unlikely for the mobile robot to have exactly the same pose as the matching node, it is important to recover the relative metric information between the current position of the mobile robot with the location of the matching node.

For each qualifying candidate, corresponding to a particular node in the topological map, a pair of omnidirectional stereo images associated to the node at the time of its inclusion was retrieved. Stereo correspondences were established and its 3D location in the environment was estimated using the method described in Chapter 2. Subsequently, based on the SURF correspondences established between the unwarped bottom omnidirectional image of the current location of the mobile robot with the unwarped bottom omnidirectional image of the matching node, two 3D point patterns were generated. Using a RANSAC procedure (Fischler and Bolles, 1981) and a closed form solution to the least squares estimation of transformation parameters for two 3D point patterns derived by Umeyama (1991), the best transformation matrix was calculated. Relative translation vectors from the current position of the mobile robot with respect to the position of the matching node were recovered using this procedure and the relative orientation was robustly recovered using the average difference between the horizontal position of the SURF correspondences in the unwarped omnidirectional image.

The main reason to avoid using the returned transformation matrix to estimate the relative orientation was such that, even when stereo data was noisy, it would not affect the accuracy of the relative orientation estimate. This is important since the mobile robot adjusts its global heading orientation according to the recovered relative orientation and this ensures that it navigates in the correct direction. Since the rank-based framework could provide one or more candidates, the system would select the one having the minimum Euclidean distance to the current location of the mobile robot based on the recovered relative transformation matrix. The recovery of the relative orientation between the current and matching node using SURF correspondences is illustrated in Fig. 5.9. The recovery of both the relative translation and orientation between the current node with matching nodes is outlined in Algorithm 5.2 and is summarized into the flowchart in Fig. 5.10.

(a) Before Heading Correction, (Top) Registered Image and (Bot) Current Image

(b) After Heading Correction, (Top) Registered Image and (Bot) Current Image

(c) SURF Correspondences

Figure 5.9: Recovering Relative Orientation using SURF Correspondences

Algorithm 5.2 Recovering Relative Transformation of Matching Nodes

Input: Current stereo pair I_b and I_t
Candidates qualifying as matching nodes, $M_C[0...S]$
Define:
Stereo() - takes a stereo pair and returns 3D coords. using stereovision techniques
TMatrix() - takes two 3D point patterns and returns the transformation matrix
Parameters: maxIter - max iterations for RANSAC procedure
T_D - Euclidean distance difference threshold

```
C1 = Stereo(I_b, I_t)
for i=0 to S do
    Load the reference stereo pair R_b^{M_C[i]} and R_t^{M_C[i]}
    C2 = Stereo(R_b^{M_C[i]}, R_t^{M_C[i]})
    Establish SURF correspondences between I_b and R_b^{M_C[i]}
    Associate SURF correspondences with 3D points in C1 and C2 and store in S1 and S2
    Calculate relative rotation θ using horizontal positions of SURF correspondences
    for j=0 to maxIter do
        Randomly select two corresponding 3D point patterns from S1 and S2 and store in P1 and P2
        h = TMatrix(P1,P2)
        S1' = h × S1
        Find number of points in S1' that fits its corresponding point in S2 within defined error threshold T_D
        Keep the best transformation matrix in M[i]
    end for
end for
Find Euclidean distance using translation vectors of the best transformation matrices in M. Select the node within shortest range
return Euclidean distance from selected node and relative bearing θ
```

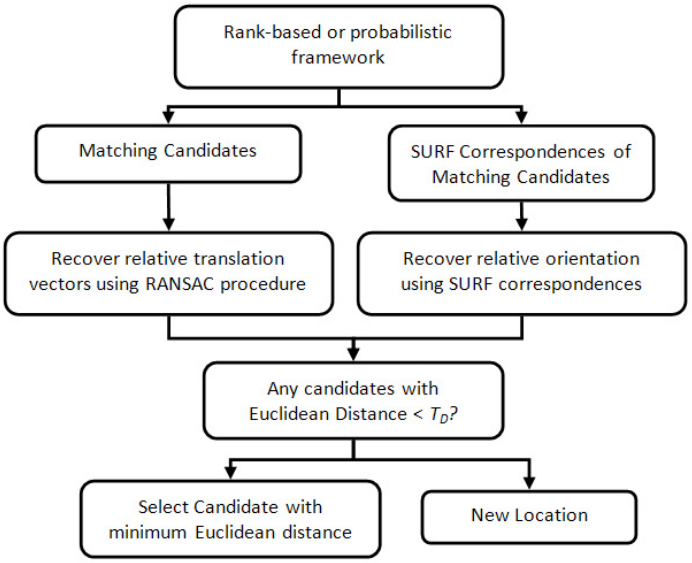


Figure 5.10: Flowchart Resolving the Revisiting Problem

5.5 Results

In order to demonstrate the reliability of using Haar wavelets as the appearance model of a place recognition system, it was subjected to an independent evaluation. This independent evaluation consists of three tests:

1. Matching a set of input query images taken at the exact locations with those in the database but at different days and times.

2. Matching a set of input query images taken at an offset of 0.6m from its corresponding image locations in the database at different days and times.

3. Repeat tests (1) and (2) with a subset of input query images. This was achieved by expanding the database to include image signatures taken from a completely different environment with respect to the subset of input query images.

Matching scores returned were sorted in ascending order for each test. Evaluation was then performed by measuring how many of the query images found their intended target in the top 1%, top 5 and top 3 of the database. As described in Section 5.2, Jacobs et al. (1995) proposed an image similarity query metric which uses a set of weights, computed using the logistic regression model, to bias the resulting similarity score between an input query image signature with an image signature in the database. The weights used in this evaluation and for subsequent experiments in this thesis were trained using a set of independent training images. A weight is associated to each bin, 3 channels with 6 bins each channel, resulting in a total of 18 values presented in Table 5.1.

	Channel Y	Channel I	Channel Q
Bin 0	-30.9657	-9.1906	-144.8866
Bin 1	0.6369	0.4206	0.8616
Bin 2	0.9599	0.8220	1.2840
Bin 3	0.7929	0.4761	0.8645
Bin 4	0.6542	0.5790	1.0526
Bin 5	0.08416	0.0991	0.1687

Table 5.1: Haar Wavelets' Weights

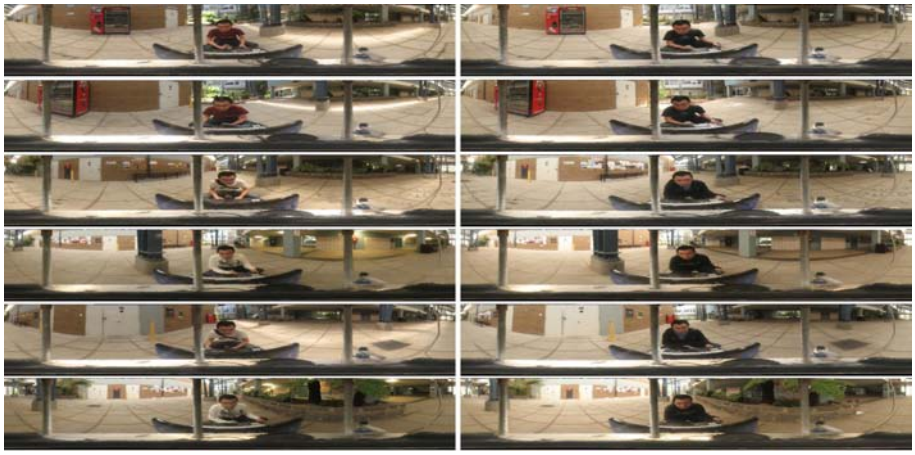
Since Haar wavelets are rotation variant, the unwarped panoramic images were column-wise shifted for every 10 degrees equivalent in pixels and a total of 36 images were used to represent any single location if it were to be included into the database. A total of 202 outdoor images and 61 semi-outdoor images at specific locations were collected for both the query and database set at different times and days. Since each location in the database would be represented by 36 images, the outdoor image database would consist of 7272 images and the semi-outdoor database with 2196 images. The first and second tests proceeded by matching the new 202 outdoor query images with the outdoor image database and the new 61 semi-outdoor query images with the semi-outdoor database. Lastly, the final test proceeded by matching the new outdoor and semi-outdoor query

images independently to a database combining both outdoor and semi-outdoor datasets. The final results were compiled into Table 5.2 with Fig. 5.11 and 5.12 showing the locations and some sample query and database images that were manually collected (approximately 1.2m apart for neighbouring locations) on different days and times.

Dataset	Query Size	DB Size	Top 1%	Top 3	Top 5
Out	202	7272	100	96.037	-
Semi	61	2196	100	100	-
Both	263	9468	100	96.958	-
Out(Off)	202	9468	100	89.552	95.532
Semi(Off)	202	9468	100	100	100

Table 5.2: Image Retrieval Matching Accuracy

(a) Semi Outdoor Environment

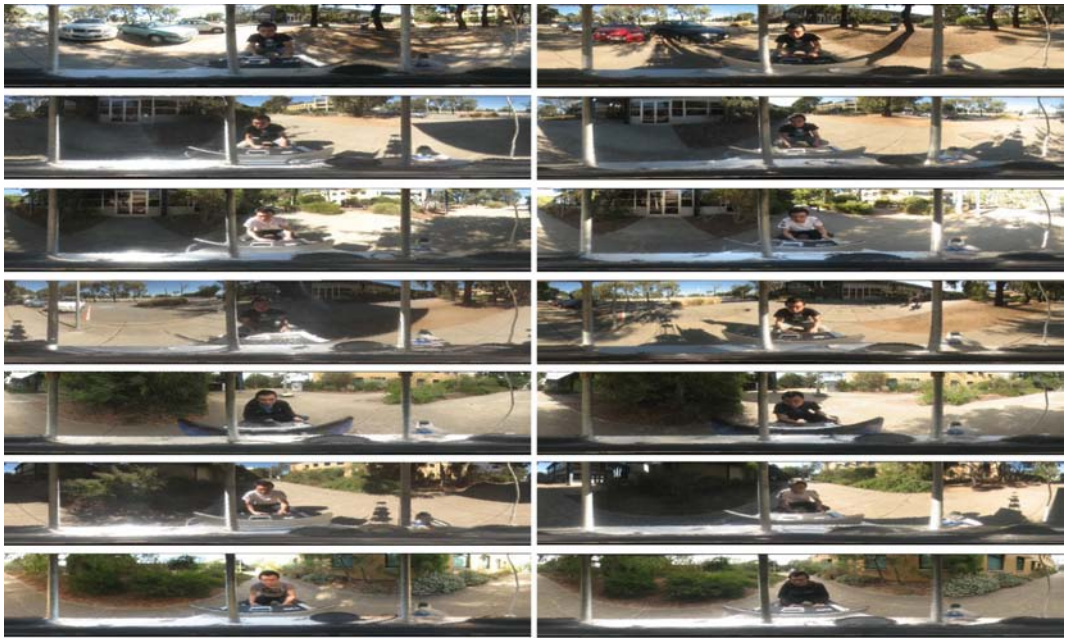


(b) Sample Database (Left Column) and Query Images (Right Column)

Figure 5.11: Sample Database and Query Images of Semi Outdoor Environment

The previous independent evaluation of the Haar wavelets clearly illustrated its matching accuracy if it were to function as an image retrieval system. This directly validates the rank-based framework described in Section 5.4.1 since it identifies the top S candidates of the returned list of scores (ranked in ascending order) that have a score less than a threshold T_R as probable candidates. These candidates will then be subjected to a number of tests in order to verify which candidate is more likely to be the better match or whether it is actually a false alarm. In addition, this also partially validates the probabilistic-based

(a) Outdoor Locations of Database and Query Images (red dots are locations where the system consistently fails)



(b) Sample Database (Left Column) and Query Images (Right Column)

Figure 5.12: Outdoor Locations of Database and Query Images

framework since the likelihood distribution is defined based on these scores. However, it does not exactly demonstrate the performance of the system. In order to provide additional insights to the performance of the probabilistic framework, the following experiment was conducted. This experiment was devised as,

1. Assume there are t number of nodes in the topological map, where node $t-2$ is connected to node $t-1$ and node $t-1$ is connected to node t in sequence.

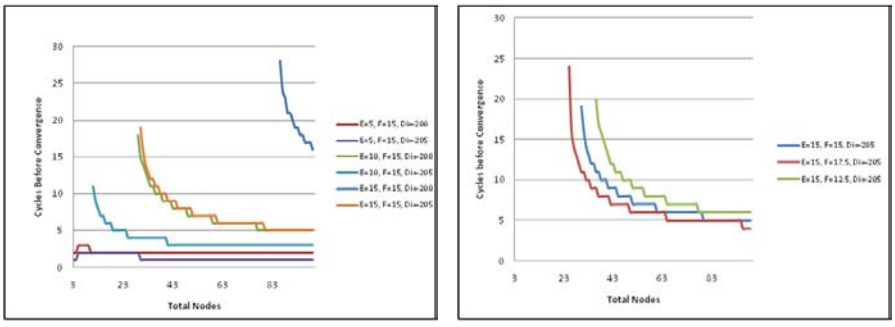
2. Initialize the prior posterior, $p(S_{t-1} = k|I^{t-1})$, such that $p(S_{t-1} = -1|I^{t-1}) = 0.75$, with the rest equally distributed.

3. Manually create a set of scores for each node (except for the virtual node) such that the score for node t is $D_{i=t}$ and the rest of the nodes with a score of -190. This can be conceived in a scenario where the mobile robot is stationary at the same location, continuously taking the same image.

4. Keep feeding this set of scores to the probabilistic framework until it converges or until some large number of iterations are performed.

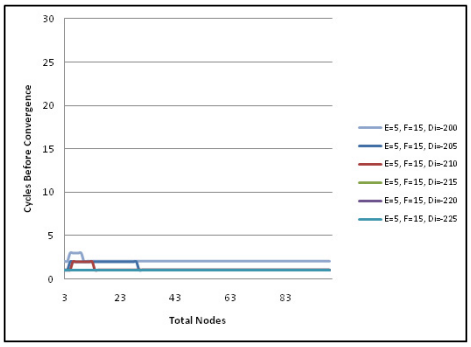
5. Repeat Steps (1) to (4) with $t = t + 1$ until t=100 with the initial value of t is 3.

This experiment was repeated with different values for the parameters E, F and G (Equations 5.16-5.19) and results are illustrated in Fig. 5.13-5.15. The graphs in Fig. 5.13(a) clearly show the effect of E as being the parameter which defines whether a score is discriminative relative to the scores of other nodes. For example, given $E = 5$, $F = 15$ and $D_{i=t} = 200$ (resulting in $D_{i=t} - \mu_{in} \approx 10$ as the total number of nodes increases), the system converged in 2-3 cycles for any number of nodes in the map whereas for the same case when $E = 15$, the system failed to converge when the total number of nodes in the map was less than 86. In reality, if $D_i - \mu_{in}$ is approximately 10.0 consistently and $E = 15$, the system will not converge even when there are more than 86 nodes in the map given the number of cycles it takes to converge and the increasing likelihood of perceptual aliasing as the database size increases. On the other hand, Fig. 5.13(b) shows how the parameter F could be used to fine tune the convergence of the system given that $E = 15$ and G was fixed at 1.5.

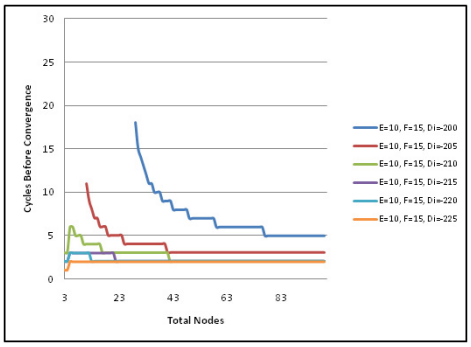


(a) E Parameter (b) F Parameter

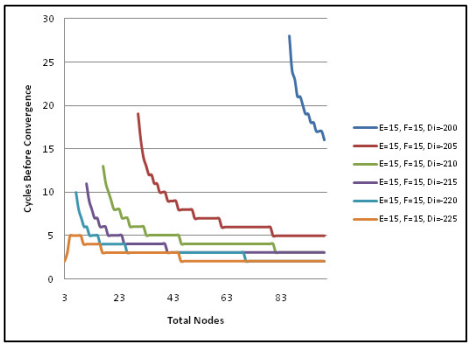
Figure 5.13: Analyzing the Effects of the E and F Parameters (G fixed at 1.5)



(a) E=5,F=15

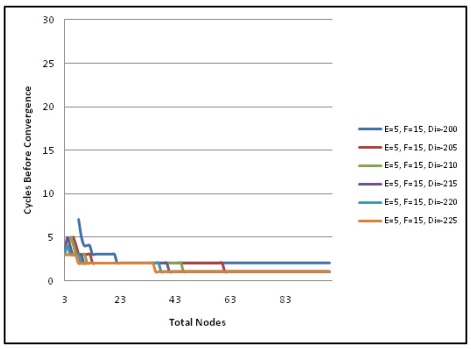


(b) E=10,F=15

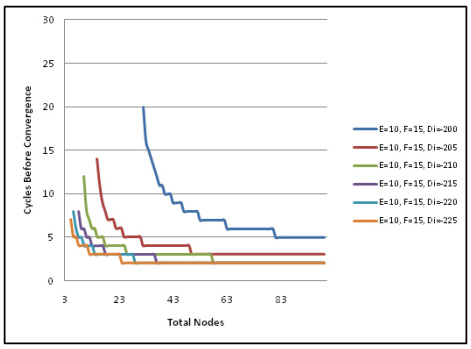


(c) E=15,F=15

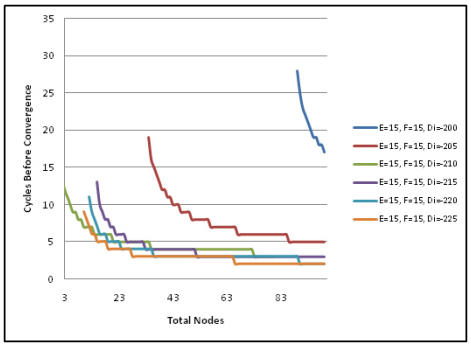
Figure 5.14: Convergence with Different Values of E when F =15 and G=1.5



(a) E=5,F=15



(b) E=10,F=15



(c) E=15,F=15

Figure 5.15: Convergence with Different Values of E when F=15 and G=2.0

Several conclusions were drawn from this experiment:

- The speed of convergence increases as the total nodes in the topological map increases. This conclusion sounded counter intuitive at first. However, after taking into account of the higher likelihood of perceptual aliasing as the map size increases, this was found to be a desirable trait.

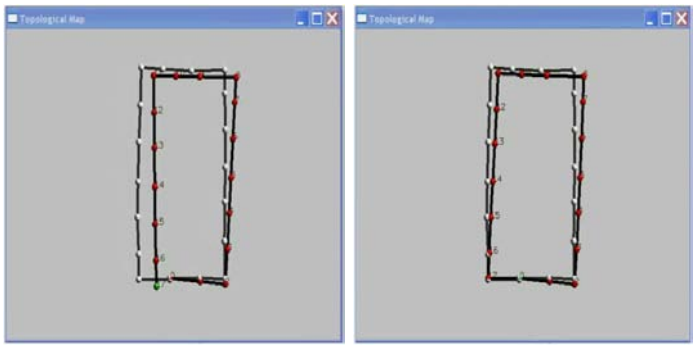
- The system would not converge even if it was continuously fed with the same set of scores if $D_{i=t} - \mu_{in}$ was deemed indiscriminative given some values for the parameters E, F and G. For example, when $E = 15$, $F = 15$, $G = 1.5$ and $D_{i=t} = 200$ (as illustrated in the experimental results), the system did not converge since this was not a discriminative score for the system to believe it was indeed revisiting a location in the map. As discussed earlier, even when if there are more than 86 nodes in the map, the system is unlikely to converge due to the higher likelihood of perceptual aliasing.

- The speed of convergence increases as the difference between $D_{i=t}$ and μ_{in} becomes larger, given that the parameters E, F and G remained constant.

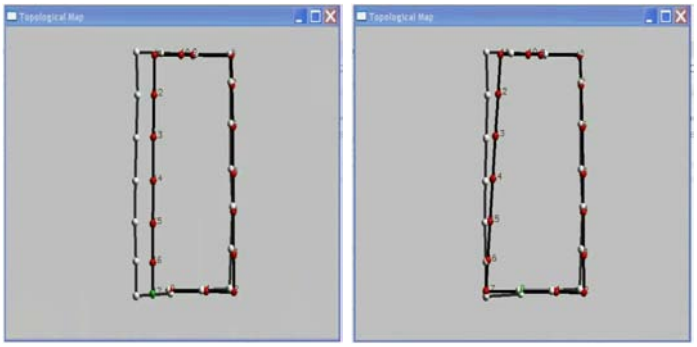
- The speed at which the framework converges might differ in real experiments depending on the path taken by the mobile robot and the severeness of perceptual aliasing.

Regardless of whether the rank-based or probabilistic framework was used by the Haar-based place recognition system, the final objective was to localize the mobile robot as it builds a globally consistent map. If the system was believed to be at a new location, a new node would be inserted into the map with the coordinates of this node computed by running the two-step relaxation algorithm described in Section 5.3 for 1 iteration. The distance travelled estimates, d_{ij}, and the mobile robot's heading estimate, θ_{ij}, were measured using the visual odometry system described in Chapter 4. On the other hand, if the system was believed to be revisiting a location in the map, links would be established connecting the previous node to the current matching node in the topological map and the two-step relaxation algorithm would be executed for 20 iterations.

To illustrate the importance of performing loop closing, the following experiment was conducted. This experiment is basically a continuation of the experiment in Chapter 4, where the mobile robot was manually driven in a loop closing fashion from L1 to L18 and then loop closing at L19 in an indoor and semi-outdoor environment. The final results are illustrated in Fig. 5.18 and selected experimental runs from both indoor and semi-outdoor environments are illustrated in Fig. 5.16 and 5.17. For the indoor experiments, the average drift (average distance travelled was 16.95m) before loop closing was 5.58%, as reported in Chapter 4. Based on the results in Fig. 5.18, the average drift was found to reduce to 3.34% after loop closing. For the semi-outdoor experiments, the average drift (average distance travelled was 20.17m) before loop closing was 5.64%, as reported in Chapter 4. Based on the results in Fig. 5.18, the average drift was found to reduce to 4.2% after loop closing.

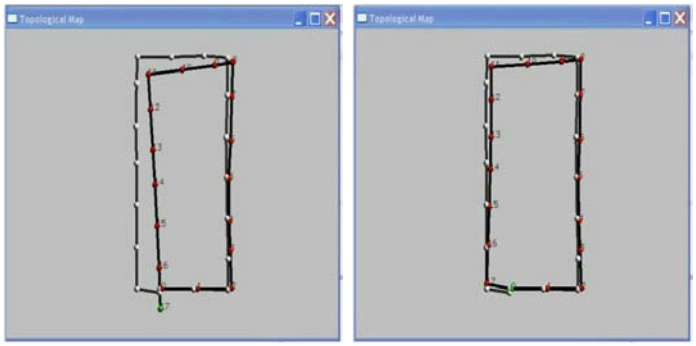


(a) (b)

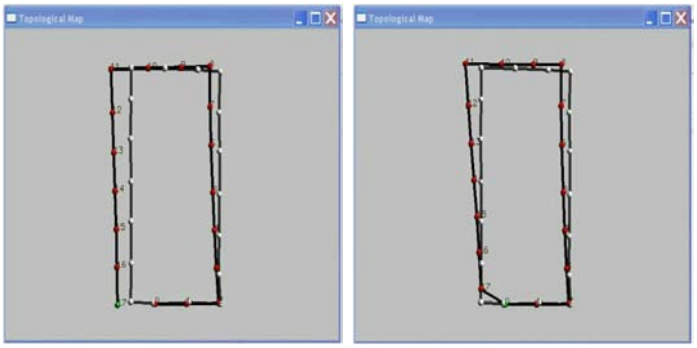


(c) (d)

Figure 5.16: Selected Experimental Runs in Indoor Environment (Ground Truth Trajectory - White Nodes)



(a) (b)



(c) (d)

Figure 5.17: Selected Experimental Runs in Semi Outdoor Environment (Ground Truth Trajectory - White Nodes)

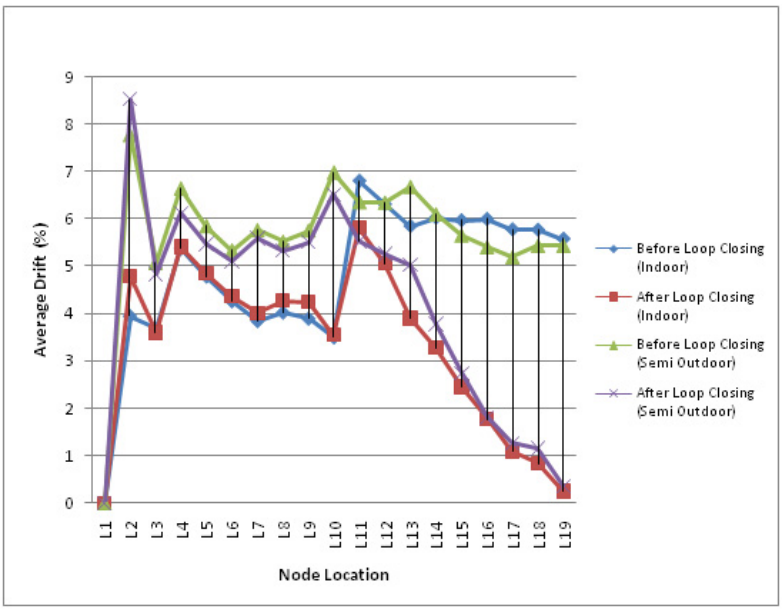


Figure 5.18: Average Drift Before and After Loop Closure

5.6 Discussion

The first experiment clearly demonstrated the robustness of the place recognition system with respect to lighting variation and occlusion. For example, the selected outdoor images in Fig. 5.12 were severely affected by lighting variation and objects such as cars parked in the scene, could be replaced by another car or might not be around anymore resulting in an empty parking space. The experiment with the offset image dataset further revealed that the system could reliably localize itself even if it was offset by a certain distance from its original location (0.6m in this case) and yet being able to accurately differentiate between two locations separated by 1.2m. The effect on matching accuracy for different image signature sizes with respect to the percentage of top 60 coefficients within this bounding box was also illustrated. In addition, although weights were trained using a totally independent dataset, it was reliably used to produce high matching accuracy for the semi-outdoor and outdoor datasets.

Then, a second experiment was devised to evaluate the performance of the refined probabilistic framework with several conclusions drawn with regards to the effect of the variation of crucial parameters introduced into the framework. These parameters facilitate the adaptation of the framework to various appearance-based place recognition systems whereby the required convergence speed might differ depending on the target application. In addition, the significance between the relative values of scores is not likely to be similar on different systems depending on the image similarity metric employed, type of appearance model, etc. By varying the parameter E of the proposed framework and fine tuning its convergence speed using the parameter F, this refined framework can be adapted specifically to the individual needs and characteristics of various systems. Comparing the probabilistic framework to the rank-based framework, the probabilistic framework can be tuned to produce similar results and converge as quickly as the rank-based framework.

The beauty of the probabilistic framework lies in the fact that it does not impose a fix threshold to determine whether a score is discriminative and converges quicker or slower depending on how discriminative the best matching score is with respect to others. Nevertheless, the selected candidate(s) is/are subjected to a number of additional evaluations before a final decision is made as to whether the current location of the mobile robot is a new or previously visited location. Finally, the last experiment showed how the global relaxation algorithm could be used to maintain the global consistency of the topological map and how loop closing could effectively keep mapping and localization errors within tolerable bounds.

5.7 Chapter Summary

A mobile robot without *a priori* knowledge of the environment is required to explore, build and maintain a globally consistent map. This was achieved by concurrently performing map building and localization. Since sensor measurements are noisy, errors accumulate over time. As such, the mobile robot has to consistently perform loop closing in order to keep errors bounded. In this chapter, an appearance-based mapping and localization system was proposed since appearance-based techniques were naturally suited for detecting loop closures. The proposed system (using the rank-based or probabilistic framework) builds a topological map, recognizes previously visited locations based on Haar coefficients and applies a relaxation algorithm for maintaining the global consistency of the topological map.

6

Autonomous Vision-based Topological SLAM

6.1 Introduction

A robot is defined as being autonomous when it can achieve a specific task in an uncontrolled environment with minimal or no human guidance. This directly implies that robots used for bomb detonation missions, the robotic submarines used in the efforts to contain the recent oil spill in the Gulf of Mexico or the American predator drones that are being used in the war against terrorism are all non-autonomous robots. This is because these robots require a human operator to continuously control, monitor its surroundings using its onboard sensors and achieve its task via teleoperation. Similarly, industrial robots such as those manufactured by KUKA (2010) or ABB Robotics (2010), which are able to perform spot welding for automobiles, palletizing, paint spraying, etc, are classified as non-autonomous due to their being restricted to operate in tightly controlled environments in order to complete tasks without human guidance. For a mobile robot to qualify as an autonomous robot, it has to, minimally, navigate itself safely from point A to point B in an uncontrolled environment. The complexity of the task is then dependent on whether the uncontrolled environment is static or dynamic, indoors or outdoors, whether *a priori* information (i.e. map of the environment) is provided to the mobile robot and type of sensors available. Of course, the complexity of any of these environmental attributes is also critical in assessing task complexity itself.

From the standpoint of autonomous robots, it is generally agreed that they exhibit some degree of intelligence. However, intelligent robots are not necessarily autonomous or, specifically, fully autonomous for the task at hand and autonomous robots are not necessarily more intelligent than non-autonomous robots. For example, consider the case of a manually driven mobile robot that is able to create a consistent and dense 3D representation of the environment. The map building process is performed autonomously and does not require human intervention, but map building requires the mobile robot to move around the environment which is achieved by a human navigator. Similarly, a domestic

humanoid robot that is designed to clean the house but is unable to react to unanticipated events such as rearrangement of furnitures in the house, also fits into this category. This category of robots is sometimes referred to as being semi-autonomous. Nevertheless, intelligence is a highly subjective descriptor and varies depending on the complexity of the task, the appearance of the robot, etc. For example, the expectations of a human-like robot are much higher as compared to a robot with an arbitrary appearance performing the same task (this issue is referred to as the *uncanny valley* and has been further studied in (Ishiguro, 2007)) and a semi-autonomous robot performing a more complicated task might be perceived as being more intelligent than an autonomous robot performing a very simple task.

What exactly makes mobile robots autonomous? Regardless of whether the mobile robot is required to bring a cup of coffee back from your favourite cafe down the street or deliver an important document to your research partner whose office is located on the other side of the university, the fundamental problem has always been related to how reliably the mobile robot can move from one point to another without supervision. Of course, to bring a cup of coffee back from the local cafe requires the mobile robot to perform other tasks such as locating and recognizing the cafe upon arrival, finding and manipulating the cup of coffee and interacting with people when ambiguity arises. However, the main focus of this work is to concentrate on the fundamental problem of autonomous robot navigation in hope that it facilitates such uses in future.

This allows the refinement of the original question to, "what exactly makes mobile robots navigate autonomously?". The main components required for mobile robot navigation are localization, mapping and path planning. In the previous chapters, it was discussed why a mobile robot without *a priori* information of the environment has to simultaneously perform map building and localization. The appearance-based localization and mapping system was also thoroughly discussed in Chapter 5. However, a mobile robot with map building and localization capabilities is useless without a human operator if it cannot decide its next course of action. This next course of action can be decided based on some objective, perhaps temporary or transient, such as enlarging and reducing the ambiguity of the map via exploration, performing loop closing in order to reduce the uncertainty in the mobile robot's estimated pose or perform goal seeking to minimize the distance to its target destination. A path planner is then required to generate viable paths based on the current position of the mobile robot. A decision is made based on the current state of the mobile robot and its map. Normally, it has to resolve conflicting objectives based on some rules specified by an expert system (i.e. incorporating human knowledge) or the adjustment of weights that help the system to place priorities on certain objectives depending on the circumstances.

6.2 Path Planning

As discussed previously, the mobile robot selects a path generated by the path planner in order to achieve some objective, perhaps temporary or transient, such as exploration, loop closing or goal seeking. As for the path planner, given the starting and goal locations, the

generated path can be optimal or suboptimal in terms of the total distance of the path, time required to travel to the specified location, path clearance (i.e. distance to obstacles), total energy required, minimal risk of collision, covertness, computational time or a combination of them depending on the type of path planning algorithm employed. This section is divided into two parts. The first part reviews the state-of-the-art path planning algorithms whereas the second part thoroughly describes our path planning algorithm.

6.2.1 Review of Path Planning Algorithms

a Dijkstra's Algorithm

This algorithm was conceived by the Dutch computer scientist Dijkstra (1959). It is guaranteed to return the shortest path from a single starting location with the assumption of non-negative edges. It starts by examining all unexplored cells closest to the starting location and places the cells adjacent to the cell currently being examined in the list of cells to be examined. Although it returns the optimal path in terms of distance, this algorithm is required to explore a relatively large area (known environment represented in grid maps) before it can produce a path to reach its goal.

b Best First Search Algorithm

The Best First Search algorithm works similar to the Dijkstra's algorithm. However, it generally requires less time and effort to plan a path since the total search area is reduced by applying an intuitive heuristic for the subsequent cell selection process. Instead of selecting cells closest to the starting location, it selects cells closest to the goal. Therefore, the path returned may not be shortest and is dependent on the complexity of the environment.

c A* Algorithm

The A* algorithm, proposed by Hart et al. (1968); Nilsson (1971), combines heuristic approaches (i.e. Best First Search algorithm) with formal approaches (i.e. Dijkstra's algorithm) for determining the minimum cost path for graph searching. This algorithm was not initially designed for path planning until Lozano-Perez and Wesley (1979) applied it to plan collision free paths among polyhedral obstacles. This work was later extended by Jarvis (1983) where the originally suggested methods for polyhedra growing were simplified, issues pertaining to the various modes of vehicle motion were explored and provided a methodology that could be extended into the third dimension. The A* algorithm is guaranteed to return the shortest path like Dijkstra's algorithm, but requires less time and effort by combining heuristics in order to reduce the total number of cells to explore. Assuming that n is the number of cells in a 2D grid map, $g(n)$ is a function of the cost of the path from the starting location to cell n and $h(n)$ is a function of the estimated cost via heuristics from cell n to the goal location, the A* algorithm always examine the cell adjacent to cell n which has the lowest cost, $f(n) = g(n) + h(n)$. Nevertheless, for this algorithm to be admissible (returning the optimal path by opening/exploring the least

amount of cells/nodes), the forward estimate, $h(n)$, should be the highest possible lower limit (i.e. not smaller than actual cost).

d D* Algorithm

The D* algorithm (Stentz, 1994) is similar to the A* algorithm except that it is dynamic in the sense that arc cost parameters can vary during the motion planning process. Optimal trajectories are generated as long as the mobile robot's motion is properly coupled to the path planner. Similar to the A* algorithm, the D* algorithm is suitable for both graph-based or grid-cell maps. Additionally, it is also suitable for path planning purposes in unknown / partially unknown and dynamic environments in an optimal and efficient manner.

e Potential Fields

As described by Huang and Ahuja (1992), the potential fields approach assumes obstacles to carry electric charges and the resulting scalar potential fields are used to represent free space. Collisions with surrounding obstacles are avoided by a repulsive force between the mobile robot and the obstacles, which is simply the negative gradient of the potential field. Additionally, they described that the potential fields approach can be used to obtain a global representation of space and facilitates the planning of a coarse path in the global level. However, this approach is intrinsically a non-global path planning methodology which suffers from a number of serious problems such as the 'entrapment' problem due to local minima caused by obstacles between the robot and the goal locations (at times resulting in a feasible goal location to be unreachable when obstacles are close to the goal) and oscillations in the presence of obstacles and in narrow passages.

f Rapidly-exploring Random Trees (RRT)

The RRT is a random path generator proposed by (LaValle, 1998; LaValle and Kuffner, 1999; Kuffner and LaValle, 2000). The basic idea of RRT in path planning is to randomly propagate in any direction from the starting and goal locations. The final path is obtained when both paths meet. Kuffner and LaValle (2000) further extended this idea by taking into consideration of the the kinematic and dynamic constraints. This algorithm does not return the shortest path but a feasible one (if such exists) and works efficiently with large maps and in high dimensional spaces (e.g. 36 DoF humanoid robot).

g Distance Transform

The Distance Transform was originally proposed by Rosenfeld and Pfaltz (1966) for describing and analyzing the shape of objects (in binary images) in computer vision applications. This algorithm was then applied to path planning for mobile robots by Jarvis (1985, 1994). As described by Jarvis (1994), the notion of propagating distances was turned inside out by considering distance contours emanating outwards from specified goal locations in unoccupied space through all free space around the obstacles. The shortest path could be

returned for any starting location, without the need to recalculate the distances again for each cell in the grid map, by tracking the steepest descent gradient path from the starting location. The Distance Transform algorithm is a global path planner (no local minima), easy to implement, straightforward, can be extended to cope with static/dynamic obstacles in a known/unknown environment and is able to cater for any number of dimensions. Also, cell translation or occupancy costs can be accommodated (e.g. rough terrain or being unobservable). The only limitation is the increasing memory requirements to store the distance transform of a large map with a fine grid resolution.

h Nodal Propagation

As described by Jarvis (1992), Distance Transform is the result of propagating adjacency from goals out through all of free space, such waves flowing around obstacles in a tessellated space. Nodal propagation is the method which takes advantage of the structure of a nodal graph (topological map) for path planning using a modified form of Distance Transform. This algorithm is general enough to accommodate different cost functions and always returns the optimal path.

i Voronoi Diagram

The Voronoi diagram is a special kind of decomposition of metric space determined by distances to a specified discrete set of objects in space. Assume a discrete set of points, p (lying on the same plane). A Voronoi diagram, which is made up of Voronoi cells, attempts to divide these cells in such a way that any point within a specific cell is closer to the point which generated the cell than any other points. Using Voronoi diagrams for path planning generally do not yield the shortest path but ensures that the path generated is amongst the safest paths since it maximizes clearance to surrounding obstacles when the mobile robot is restricted to traverse along the edges of cells in the Voronoi diagram (assuming the discrete set of points are obstacles). A recent work (Bhattacharya and Gavrilova, 2007) had reported using Voronoi diagrams for generating optimal paths in the presence of simple disjoint polygonal obstacles. In another work, Tungadi and Kleeman (2009) proposed the use of Voronoi diagrams for generating loop-paths for a fully autonomous mobile robot performing SLAM using laser rangefinders and sonar sensors.

j Probabilistic Roadmaps

The Probabilistic Roadmaps algorithm has been successfully applied to plan complex motions for articulated robots (Amato and Wu, 1996) and mobile robots (Boor et al., 1999) in configuration spaces of low and high dimensionalities. It proceeds in two phases; the learning and query phase. In the learning phase, a graph referred to as the roadmap is built by repetitively picking a random configuration of the robot and testing this configuration for collision, terminating only when it is collision-free. At the same time, a local planner is used to connect this configuration to the roadmap. In the query phase, the starting and goal locations are specified and it attempts to search for the roadmap which connects these

locations. The roadmap is probabilistically complete if, for any query, the probability of answering the query incorrectly goes to zero after a long run. In the literature, there are many variants of this algorithm targeting specific applications.

6.2.2 Path Planner

As described in Chapters 2 and 3, our mobile robot could generate a 3D representation of its surroundings by using a single or multiple pairs of omnidirectional images via stereovision techniques. The resulting 3D point cloud from the omnidirectional stereovision system would be voxelized and clipped before they were compressed into a 2D local grid map. By ray tracing from the centre of the mobile robot in all directions, the 2D local grid map was then segmented into visible, non-visible and obstacle regions as shown in Fig. 6.3(c) and 6.3(d). These rays would propagate outwards from the centre of the mobile robot in all directions and travel for a maximum radial distance, R_{dist}. They would be terminated when obstructed by obstacles in the environment or when the total radial distance travelled was found to be greater than R_{dist}. Any grid cells traversed by a ray would be labelled as visible regions to the mobile robot whereas others would be labelled as non-visible regions to the mobile robot or remain as being an obstacle if it was originally labelled as an obstacle. The examples shown in Fig. 6.3 were created using omnidirectional images taken in an indoor environment shown in Fig. 6.1.

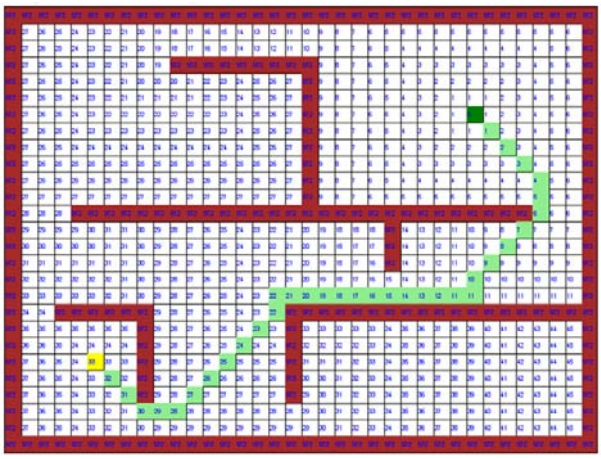
(a) View 1 (b) View 2

Figure 6.1: Indoor Environment

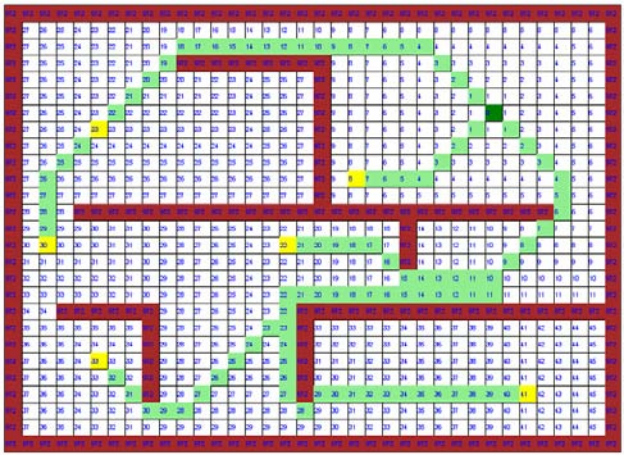
Similar to the frontier region detection algorithm proposed by Yamauchi (1997), where frontier regions were detected between explored and unexplored areas, frontier regions between visible and non-visible areas were detected in our system, with the centre of these regions calculated. Then, obstacles were dilated to create a safety margin between the perceived obstacles and the mobile robot. Paths could then be planned using 2D Distance Transform (DT) from the current location of the mobile robot to these frontier centres. The 2D Distance Transform (Manhattan distance) of a static grid map with different starting and goal locations are shown in Fig. 6.2. The cost function of the DT algorithm used in our system was purely based on distance, thus producing the shortest path, but could be extended to cater for other factors as well. Given that the goal locations were

fixed, the 2D Distance Transform of this static grid map would remain the same regardless of the mobile robot's starting location or its subsequent locations. In addition, for the case where multiple starting and goal locations were specified (Fig. 6.2(c)), the DT path planned for each starting location would always be the shortest path to the nearest goal location. Since our system has multiple goal locations (i.e. each frontier centre) and a single starting location, this implies that the path returned using DT is the shortest path from the starting position to the nearest goal location instead of producing a path to each goal location. This issue was resolved by treating these goal locations as starting locations and the original starting location as the goal location (planned paths would be reversed). This solution is simple and requires no modification to the original DT algorithm. The final 2D local grid map and DT paths for the indoor environment example are shown in Fig. 6.3(e) and 6.3(f). In addition, an example of a 2D local grid map for a semi-outdoor environment is illustrated in Fig. 6.4.

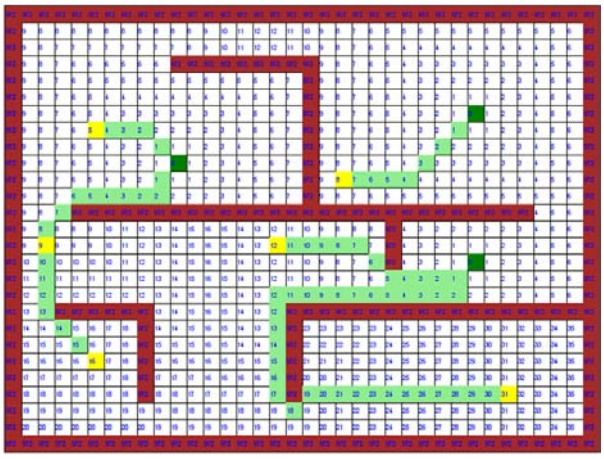
Using the previously described technique, the mobile robot was able to plan short paths to these frontier centres where the chosen frontier centre was dependent on the mobile robot's objective (i.e. exploration, loop closing, target location, etc), which will be described later in this chapter. The mobile robot was also equipped with the capability to exploit the current state of the topological map to plan a path which directs the mobile robot to an existing location in its map. For planning an optimal path to the target node in the topological map, the nodal propagation algorithm (Jarvis, 1992) was employed. A cost function was defined and the optimal path (in terms of distance) would be returned to the mobile robot. In fact, this cost function could also be extended to include other factors such as terrain traversability, likelihood to receive reliable GPS coordinates, likelihood of matching to the appearance-model of the location, etc. The mobile robot would be required to traverse in previously seen locations using the resulting path since it relies on existing information in the topological map for path planning. An example illustrating the use of the nodal propagation technique to compute the shortest path is provided in Fig. 6.5. To compute this shortest path in this example, spatial information and connectivity of each node in the topological map were required. For more details on the DT and nodal propagation algorithms, please refer to (Jarvis, 1985) or Appendix I in (Jarvis, 1994) for the basic Distance Transform algorithm and steepest descent path planning and (Jarvis, 1992) for the nodal propagation algorithm where hand worked examples were included.



(a) Single Start and Single Goal Location



(b) Single Start with Multiple Goal Locations



(c) Multiple Start and Goal Locations

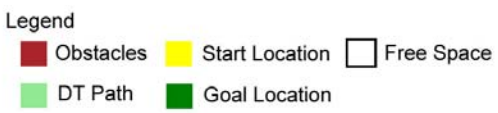
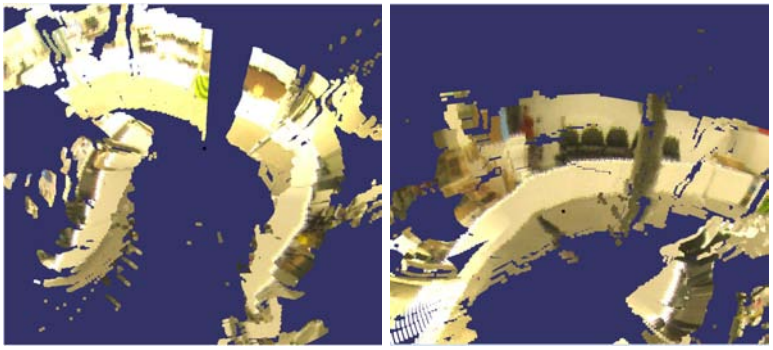
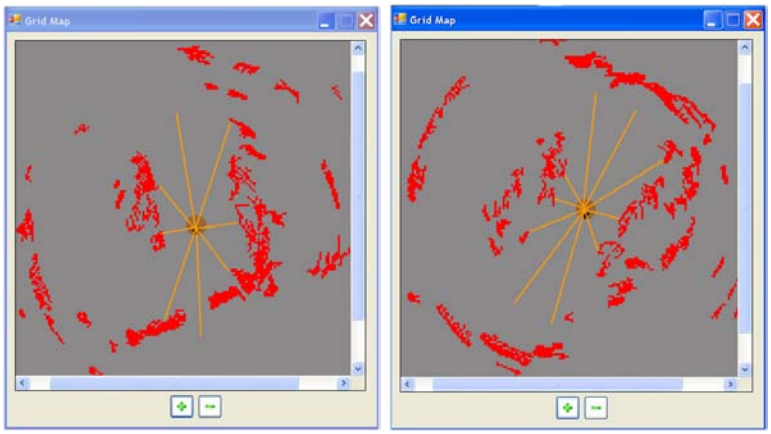


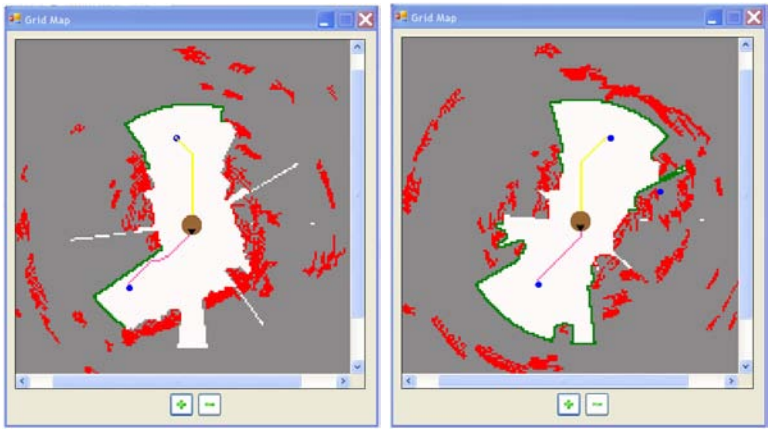
Figure 6.2: Path Planning using 2D Distance Transform (Manhattan Distance)



(a) Voxelated Environment 1 (b) Voxelated Environment 2



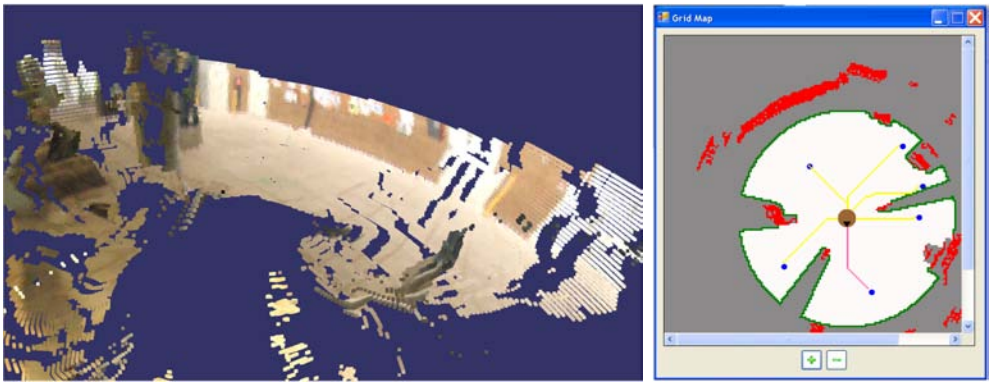
(c) Segmenting Environment 1 (d) Segmenting Environment 2



(e) Grid Map 1 (f) Grid Map 2

(g) Unwarped Image 1 (h) Unwarped Image 2

Figure 6.3: Segmenting the 2D Local Grid Map (Indoor Environment)



(a) Voxelated Environment (b) Grid Map

(c) Unwarped Image

Figure 6.4: 2D Local Grid Map (Semi Outdoor Environment)

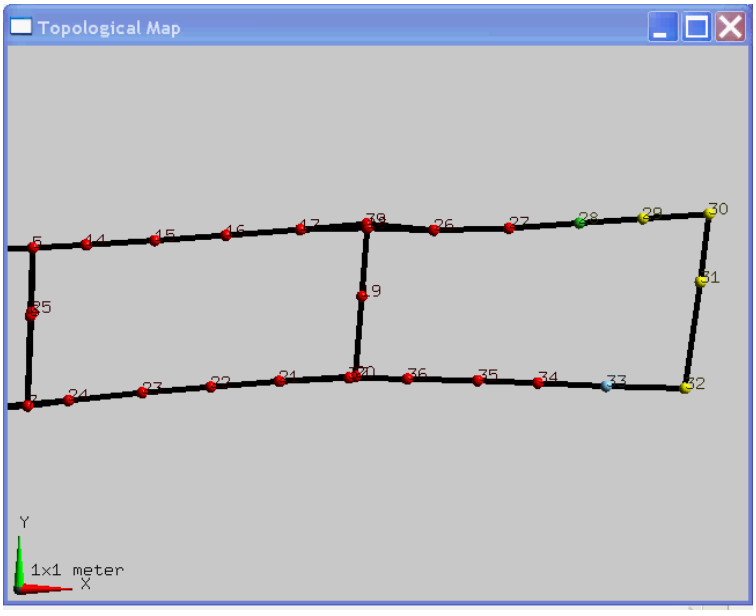


Figure 6.5: Shortest Path using Nodal Propagation

6.3 Autonomous Navigation

A mobile robot navigation system traversing in unknown environments is required to build an internal representation of the environment via map building and continuously localizing itself at the same time. Building this map requires the mobile robot to perform exploration. However, errors accumulate over time in the localization process and this requires the mobile robot to perform loop closing to break the accumulative propagation of error by readjusting the nodal locations of the topological map, treated as a network of springs, by minimizing its total energy via the global relaxation algorithm described in Section 5.3. In an earlier example, where the mobile robot was required to bring a cup of coffee from the local cafe, the mobile robot was also required to balance its effort amongst goal seeking, exploration and loop closing. However, if the objective of the mobile robot was to autonomously map the environment, then performing exploration and loop closing should suffice. In this work, the autonomous navigation system only balances its efforts between exploration and loop closing modes. Although the goal seeking mode was not considered during the map building process, the system will be demonstrated later in this chapter to be capable of reaching its target destination given a map of the environment. In this section, a brief review of how existing systems autonomously perform exploration will be provided. The motivation to perform loop closing was described in Chapter 5, but was written in the context of detecting loop closures for mapping and localization. This section aims to provide the reader with the motivation to perform loop closing in the context of autonomous navigation. Then, a detailed description of our strategy to perform loop closing autonomously will be provided.

6.3.1 Review of Exploration Strategies

Exploration can be loosely defined as the task of searching out and mapping new places. In the context of mobile robots, these new places are free space that have yet to be mapped. Exploration allows the mobile robot to enlarge its map, learn more about the environment and possibly discover other efficient or less dangerous ways to move from one point to another in the environment. The following discusses a subset of exploration strategies used in mobile robot navigation. The suitability of a particular exploration strategy depends on the type of map built by the mobile robot, sensors available and nature of the environment.

a Distance Transform

The Distance Transform (Jarvis, 1985, 1994) can also be used as an exploration algorithm by taking all unexplored cells in the grid map as goal positions. If the cost function is purely based on distance, the resulting DT path directs the mobile robot to the nearest unexplored cell and expands the map. This is a greedy algorithm in the sense that it attempts to visit all unexplored cells in the environment, closest first (unless it is obstructed). Zelinsky (1992) adapted the Distance Transform for quad-tree maps and utilized this representation to explore the environment and perform goal seeking.

b Global A-Optimal Exploration Strategy

Sim and Roy (2005) proposed a global and non-greedy path planner that provides an efficient exploration strategy which deliberately closes loops with the aim of faster convergence to the correct map. An Extended Kalman Filter was used for SLAM and the a-optimal measure was used to plan the global path. The a-optimal measure, which minimizes the mean squared error of the model, was used as the objective function of the exploration strategy such that the path which maximizes information gain for building an accurate map could be computed. This method is not greedy in the sense that it does not choose the single next best action which maximizes the expected information gain for a given point but integrates observations along the trajectory to the point. Nevertheless, it does not exhaustively search the space of all trajectories due to computational cost and may not result in an optimal exploration strategy. Due to the choice of the objective function, which aims to produce an accurate map via a formal method for generating exploration trajectories that maximizes information gain, this method also resulted in a planner that deliberately closes loops in order to improve the overall map estimate.

c Frontier-based Exploration

As described by Yamauchi (1997), frontiers were defined as regions between the boundaries of open space (explored and unoccupied space) and unexplored space. The centres of these frontier regions provided the mobile robot with possible goal locations in its map for exploration. Using these frontier centres as goal locations for a path planner, a mobile robot could autonomously navigate to unexplored spaces in the environment.

d Landmark-based Exploration

Taylor and Kriegman (1993) proposed an exploration strategy based on the association of visual landmarks and obstacles represented in a two-dimensional configuration space. The goal of this exploration strategy was to explore all landmarks seen by the mobile robot. A landmark would be labelled as fully explored only if the mobile robot had circumnavigated its associated obstacle in the two-dimensional configuration space.

e Voronoi Diagram

Besides being a useful path planner, which generates paths with the maximum clearance from obstacles, the Voronoi diagram was also reported to be useful for mobile robots performing exploration (Yershova et al., 2005; Tungadi and Kleeman, 2009). In (Tungadi and Kleeman, 2009), loop-paths were generated using Voronoi diagrams which enabled the mobile robot to explore and perform loop closing. Voronoi regions were also used in (Yershova et al., 2005) to bias the generation of paths to unexplored regions using the Rapidly-exploring Random Tree algorithm where the probability of extending a node was proportional to the size of the Voronoi region.

f Neural Networks

In (Thrun, 1993), two backpropagation artificial neural networks were used for model building where experiences were remembered and generalized via the neural networks. One of the neural networks was used for interpreting sensor measurements and the other was used for estimating the confidence level of the sensor measurements. Both networks were trained to encode the specific characteristics of the sensor and typical environments traversed by the mobile robot. By discretizing the model of the environment into grid maps, low cost paths were generated to unexplored spaces based on the confidence level (a measure of how likely that a space was occupied) of a particular cell in the grid map. A low confidence level of a particular cell in the grid map represented a higher exploratory utility. As the exploratory utility was propagated through free space, an efficient exploration strategy using dynamic programming approaches was achieved as the mobile robot traverses the environment based on the path generated using the steepest ascent.

6.3.2 Motivation to Perform Loop Closing

Literally, loop closing in the context of mobile robot navigation can be loosely defined as the act of closing the loop by empowering the mobile robot with the ability to recognize places that it is indeed revisiting in an explored area in its map after traversing the environment in a loop-like trajectory. For mobile robots without *a priori* information of the environment, traversing in a loop-like trajectory maximizes the area being explored and at the same time directing it back to a previously explored location in the map. In this manner, as the mobile robot returns and performs loop closing, the accumulated errors in the pose of the mobile robot and detected landmarks in the environment, as a result of errors associated with sensor measurements, can be partially rectified. This process is advantageous for both the localization of the mobile robot and in maintaining the global consistency of the map. However, the problem of reliably detecting loop closures is indeed difficult. Due to the vital importance of this issue and its relevance in SLAM systems, loop closure has gained a tremendous amount of attention in recent years. Lately, the term loop closing is not only used to describe the ability of the mobile robot to perceive that it is revisiting a location in its map after traversing in a loop-like trajectory but has been used to describe the ability of the mobile robot to recognize that it has returned to a previously explored location in general, irrespective of whether it is navigating in a loop-like trajectory.

In the literature, there are many ways to detect loop closures and they are largely dependent on the type of sensors available on the mobile robot. The three main sensors used for mapping and localization are laser rangefinders, sonar systems and cameras. Laser rangefinders and sonar systems normally provide range data, with those produced by laser rangefinders generally being a lot denser as compared to sonar systems. Sonar systems are capable of providing accurate locations of a sparse set of corner and edge features but they are only suitable for indoor applications (attenuation of ultrasonic signals in air is so severe that ranging much beyond 10m is unusual). Laser rangefinders, on the other hand, are less restrictive and have been successfully applied to outdoor applications. Nevertheless,

loop closure detection is difficult when only range data is available, since it relies on the physical structure of the environment in order to distinguish one place from another. This eventually becomes an issue if the mobile robot is required to traverse in man-made environments where it is likely that different places produce similar range data. Of course, the gravity of this issue reduces with respect to how much this effective measurable range can be increased. However, there is a limitation to how much this effective range can be increased due to health and safety concerns since laser rangefinders are active sensors. Despite so, current state-of-the-art laser scanners like the Riegl terrestrial scanners can range to >800m and yet still being eye safe (although it is quite costly as well). In addition, laser rangefinders emit active energy which may result in possible interference and energy detection in military situations that may sometimes be a disadvantage. Scan matching (Lu and Millios, 1997; Diosi and Kleeman, 2005) is a popular technique for comparing laser range data and in a more recent work by Gränstrom et al. (2009), rotation invariant features were extracted from the range data before scan matching was performed in an attempt to improve the accuracy of loop closure detection.

On the other hand, visual information acquired from video cameras can be further processed to provide sparse or dense range data. However, range data derived from visual information, which satisfies the real-time constraint, is not as accurate and consistent as compared to laser range data. As such, visual loop closure detection techniques are normally appearance-based, matching the appearance model of previously seen locations with the appearance model of the mobile robot's current location. As highlighted in Chapter 5, appearance-based techniques are naturally suited to solving the kidnapped robot problem, perform multi-map merging and particularly the loop closure detection problem. In fact, the majority of recent work in visual loop closure detection uses appearance-based techniques and a review of these systems was provided in Chapter 5. Unfortunately, visual loop closure detection does suffer from perceptual aliasing and severe lighting variations. This forms the motivation for utilizing both structural and visual information in the environment for reducing false positive detections such as those described by Ho and Newman (2007) and Tungadi et al. (2010) (a system that will be presented later in this thesis).

Unfortunately, a loop closure detection system without active loop closing on an autonomous mobile robot is similar to trying to perform loop closing by chance. This is due to the errors associated with sensor measurements which degrade the localization of the mobile robot over time, affecting the reliability of the system to direct itself back to possible loop closing locations. Stachniss et al. (2004) and Tungadi and Kleeman (2009) have both incorporated active closing into their hybrid map building systems. However, the active loop closing strategy proposed by Stachniss et al. (2004) performs loop closing only when the uncertainty in the mobile robot's pose becomes too large and detects loop closure candidates if the shortest length between the current and candidate node in the topological map is large whereas the shortest length between these nodes locations in the metric map is small. In this way, this is still not any better than trying to perform loop closing by chance. On the other hand, the strategy proposed by Tungadi and Kleeman (2009) is a more effective approach since they apply the Voronoi diagram for the

generation of loop-paths which facilitates loop closing (via laser scan matching) and exploration. In addition, the previously described global a-optimal path planner (Sim and Roy, 2005), which generates global paths that allow the mobile robot to perform exploration, maximizes information gain and was illustrated to deliberately plan paths that closes the loop since the aim of the exploration is to build accurate maps in an efficient manner by considering the trajectory to be taken by the mobile robot.

Irrespective of the reliability of the system to detect loop closures, it is impossible to have one which produces no false alarms without having to trade off with more false negatives or delay in convergence. This is where active loop closure validation comes into play. Active loop closure validation should not only validate the loop closure event based purely on past and present information but should attempt to proactively gather evidence in the future if necessary. If sufficient new evidence leads the system to believe that the loop closure event is indeed a false alarm, it should be able to reverse the effects of this decision. Unfortunately systems such as (Cummins and Newman, 2008; Angeli et al., 2008; Kawewong et al., 2010) validate a loop closure event only based on the past and present information. Other approaches such as those which maintain multiple hypotheses (Tomatis et al., 2002; Spero and Jarvis, 2005; Beevers and Huang, 2005; Tully et al., 2009) are able to gather more evidence before committing to accepting a loop closure event but are lacking in an active loop closure validation strategy. The problem with a multiple hypothesis framework without an active loop closure validation strategy is the exponential increase in the total number of hypotheses it has to maintain as more loop closure events are left unvalidated. In summary, an autonomous mobile robot performing map building in an unknown environment is, minimally, required to perform and balance its effort amongst exploration and loop closing, where loop closing is achieved by a combination of active loop closing and active loop closure validation when required.

6.3.3 Autonomous Loop Closing System with Exploration Strategy

We propose that a complete loop closing system should be made up of three modules: (1) loop closure detection, (2) loop closure validation and (3) system restoration. Loop closure detection can be achieved in many ways depending on the type of sensors available. However, for a fully autonomous mobile robot, loop closure detection is more advantageous in many applications if the mobile robot actively seeks it (e.g. via an active loop closing strategy). Once the mobile robot believes it is revisiting an explored location, it should then validate this loop closure event regardless of whether a single or multihypotheses framework is used. If validation fails, there should be some means of reversing the decision of loop closure to no loop closure in that particular time step, since modifications were made to the map and the mobile robot's pose when it commits to a detected loop closure. We call this process system restoration.

The ideas presented here can also be easily adapted to other systems building metric or hybrid maps. Nevertheless, it is possible for systems to retain all information or to maintain multiple hypotheses with the potential to retrospectively remove invalidated

loop closures. For such SLAM systems, some parts of the complete loop closing system might not be applicable without necessary modifications.

a Loop Closure Detection

Overview For this system, loop closure detection was achieved via an appearance-based technique described in Chapter 5, which could employ either the rank-based or probabilistic framework described in Sections 5.4.1 and 5.4.2. Haar decomposed omnidirectional images were used as the appearance model for each node in the topological map for loop closure detection. The chosen appearance model was experimentally validated to be robust for use in indoor, semi-outdoor and outdoor environments. A list of scores would be returned when a new image was presented and these scores would be sorted in ascending order. Subsequently, the top S candidates having a score less than the threshold, T_R (a lower score representing a better match) were selected by the rank-based framework. Each loop closure hypothesis would be subjected to a number of validations using information at present, such as the number of SURF correspondences established and ensuring that the Euclidean distance between the location of the matching candidates with the current location of the mobile robot to be less than its current variance value (measure of uncertainty in its position). Relative metric information between nodes would then be recovered from those qualifying candidates using the RANSAC procedure described in Section 5.4.3 and the final candidate was chosen by selecting the one with the minimum Euclidean distance. This final candidate would only be accepted as a valid loop closure event provided that it has a Euclidean distance less than T_D measured from the current location of the mobile robot.

In Section 5.4.2, a refined probabilistic framework based on the work of Angeli et al. (2008) was proposed for use with the Haar wavelets as the appearance model of a location in the environment. The state transition model and likelihood voting scheme were redefined such that it could be easily adapted to systems requiring different convergence speeds and allowing the developer to decide what is deemed as a discriminative matching score (depending on the underlying appearance model being used and its image similarity metric). If the probabilistic framework was used, a single candidate would be returned if a loop closure was detected and the same validations as described for the rank-based framework would be applied to this matching candidate.

The system would commit itself to the loop closure event if the final candidate was successfully validated (based on past and present information). Then, it would proceed into the *correction* phase where the previously recovered relative metric information between nodes would essentially be used to correct the mobile robot's pose. Lastly, the global relaxation algorithm would be iterated for 20 times in order to maintain the global consistency of the topological map

Active Loop Closing Strategy Active loop closing was achieved through a tight integration with the path planner. There are two factors to consider; (1) loop size and (2) target loop closing location. Loop size is affected by (a) the type of map built by the

system, (b) accuracy of the mobile robot's localization system, (c) the type of sensors used for loop closure detection and of course, (d) physical restrictions imposed by the structure of the environment.

For the case of a metric mapping system which provides an option to select any loop size (where the size of the loop is measured by the total distance required to complete the loop), the loop size can be determined by balancing two factors; (i) maximizing the area being explored and (ii) minimizing the uncertainty in the mobile robot's estimated pose for the navigation system to reliably direct the mobile robot to its intended target loop closing location. At times, these two factors conflict one another depending on the effective range of the employed sensor since the amount of overlap between sensor measurements in the outgoing and returning paths of the loop should be minimized (depending on the uncertainty of sensor measurements at different ranges). Assuming that the effective range of the sensor shown in Fig. 6.6 is the range where uncertainty in sensor measurements is low, the mobile robot should minimize this overlap in the outgoing and returning paths of the loop. If the resulting loop size is too large, it directly affects the ability of the mobile robot to find its target loop closing location. However, whether the area being explored is being maximized does not depend on the effective range of the sensor alone but also the structure of the environment. Fig. 6.7 depicts a typical scenario where the mobile robot would have no choice but to complete the loop and Fig. 6.8 shows an example where the mobile robot would not be required to travel in a loop-like trajectory for this part of the environment (assuming that there is more to explore in the environment than as depicted) due to the effective range of the sensor and the size of this part of the environment. However, since the mobile robot operates in unknown environments, such assumptions could not be made. For instance, the environment can be as large and open as depicted in Fig. 6.9 or as large and cluttered as depicted in Fig. 6.10. As such, given the option to select from a range of fixed loop sizes or the flexibility to pick any loop size, a higher priority is given for increasing the likelihood of the mobile robot to return to the intended loop closing location.

For the active loop closing system described by Tungadi and Kleeman (2009), where the Voronoi diagram was used, loop-paths were generated (implicitly addressing the problem of target loop closing location selection) and executed in order such that shorter loop-paths were executed first, if available, before attempting longer loop paths in order to ensure the global consistency of its map. Loop-paths that were too short were ignored and at times, when only long loop-paths were available due to the structure of the environment, the system would have no choice but to attempt these loop-paths. The definition of whether a loop-path is long or short is largely dependent on the system. On the other hand, the global a-optimal (Sim and Roy, 2005) algorithm which deliberately plan paths to close the loop achieves this via an objective function in its global path planner which performs exploration and tries to maximize the information gain based on the planned trajectory with the aim to converge to the correct map in a more efficient manner. This objective function implicitly addresses both the loop size and target loop closing location.

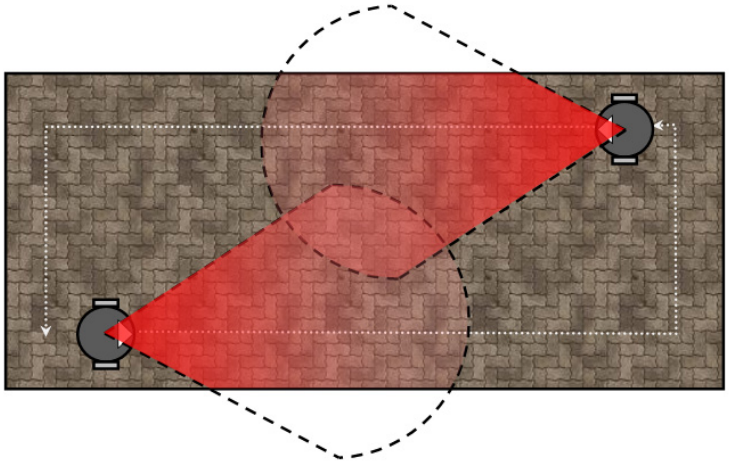


Figure 6.6: Scenario 1: Overlap in Sensor Measurements

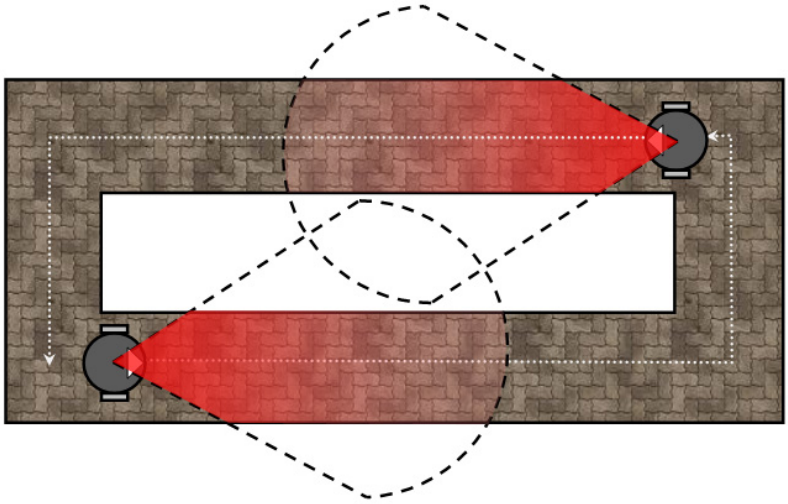


Figure 6.7: Scenario 2: Restrictions due to Physical Structure of Environment

Similarly, for topological mapping systems, loop size selection is dependent upon whether the objective is to maximize the area being explored or to reliably find the target loop closing location. For appearance-based systems, the area being explored is maximized when the loop size is chosen based on the minimum separation between images where the place recognition system employing a specific appearance model can still function reliably. Assuming that the minimum separation between images is small, the loop size can also be kept small such that it increases the likelihood of the mobile robot to return to and find its intended target loop closing location. However, navigating in small loops drastically increases the total operational time unless a dense topological map is desired. Nevertheless, this minimum separation between images changes with respect to the structure and texture available in the environment. Similar to metric mapping systems, our system selects a loop size that is biased towards increasing the likelihood of the mobile robot returning to its target loop closing location. Of course, another important factor is the length of operational time the mobile robot is expected to be in the environment to be explored. If it is to spend one year in the environment, even days of exploration can be justified. As

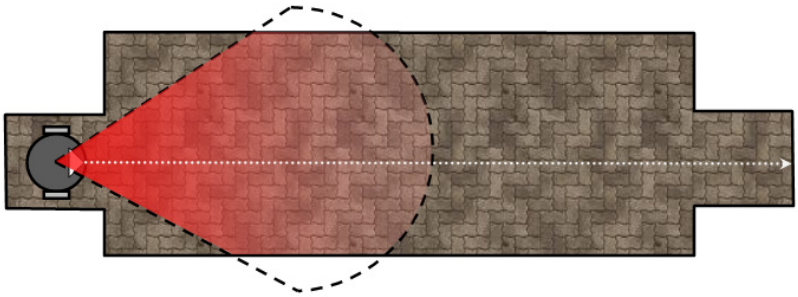


Figure 6.8: Scenario 3: Effective Range of Sensor and Size of Environment

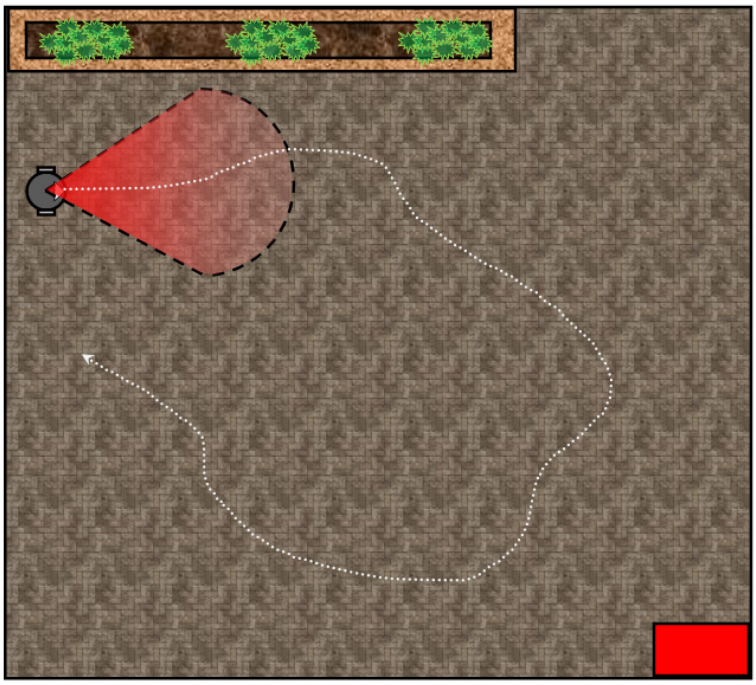


Figure 6.9: Scenario 4: Large and Open Environment

described in Section 6.2.2, our system can provide multiple goal locations (each frontier centre) for a 2D local grid map. The following describes the path selection process, target loop closing node selection process and the strategy employed to return to this selected target node.

There are three modes of operation: (1) active loop closing validation which will be described later, (2) pure exploration and lastly, (3) the active loop closing and exploration modes. The system is always initialized to start in the pure exploration mode. In this mode, it selects any one of the planned paths, each associated to a frontier heading extracted from the 2D local grid map, as long as it is unlikely to bring the mobile robot too close to an existing node in the evolving topological map. In addition, it is a lazy algorithm in the sense that it is always biased towards selecting a valid frontier heading which requires the least amount of effort (i.e. the amount of pure rotational motion required such that it minimizes the change in global heading). When the total distance travelled by the mobile robot exceeds D_I, where the total distance travelled is measured

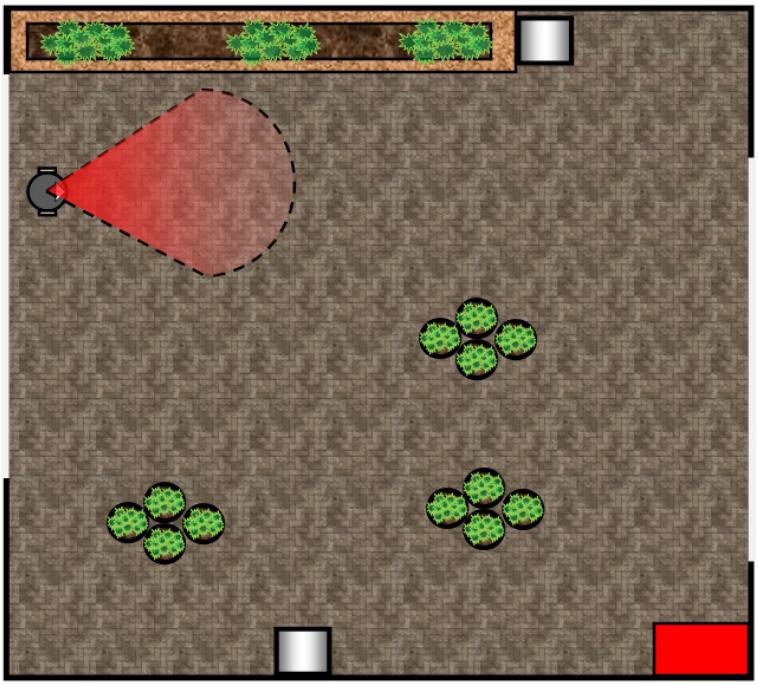


Figure 6.10: Scenario 5: Large and Cluttered Environment

by accumulating the relative distance between each node inserted into the topological map so far, the system switches into the active loop closing and exploration mode.

The objective of the active loop closing and exploration mode is to maximize the chances of exploring new areas in the map while it attempts to direct the mobile robot towards a target loop closing node. This target node is selected based on the prospect of a more accurate relocalization of the mobile robot's pose with respect to the target node's global reference position and orientation. Ideally, it tries to select the node with the least positional variance. At the same time, it takes into account whether the mobile robot is likely to reach this target node based on the distance travelled so far prior to detecting a loop closure event, D_F, and a rough estimate of how far the mobile robot is from this target node, measured based on Euclidean distance.

Specific to this work, the average positional drift of the visual odometry system was reported as 5.64% of the total distance travelled for a semi-outdoor environment in Chapter 4. Given that it is likely to detect loop closure if the mobile robot is approximately 1.5m away from the actual location of the target node in the environment, the total distance the mobile robot can travel prior to loop closing, $D_{max} = 25m$, given that the average drift is increased to 6%. D_I is then set as 7m ($0.25D_{max}$ rounded up to the nearest integer). The positional variances of all nodes are then stored into a list, L, which is sorted in ascending order and the Euclidean distance, D_E^i, from the current location of the mobile robot to each node is computed, where i is the index of the node in the list. If $D_E^0 + D_F < D_{max}$, the selected target loop closing node is the node associated with index 0 of the list, L. Else, a subset of nodes, with positional variances less than half of the mobile robot's current positional variance from list L, is stored into a new list H with its associated Euclidean distances, D_E^j, where j is the index of the node in the list H. Subsequently, if

$D_F < D_{max}$, the first node in list H (similarly sorted in ascending order according to its positional variance value) which satisfies the condition $D_F + D_E^j > D_{max}$ is chosen or else H is sorted in ascending order according to D_E^j and the first node in index 0 is selected.

Once the target loop closing node is selected, the system selects a path which is associated to a frontier heading that avoids previously explored locations in the map and at the same time directing it closer to this target node. D_F is only reinitialized to 0 if a valid loop closure event is found, matching the current location of the mobile robot to the target loop closing node or to any of its immediate neighbours (neighbours to the target loop closing node prior to committing to the detected loop closure).

b Loop Closure Validation

As described previously, a number of validations were applied to probable loop closure candidates using past and present information. However, a complete loop closing system should proactively gather more evidence in the future, to confirm or refute the earlier loop closing decision, if required. Our active loop closure validation strategy can be illustrated using a simple example based on the topological map in Fig. 6.11.

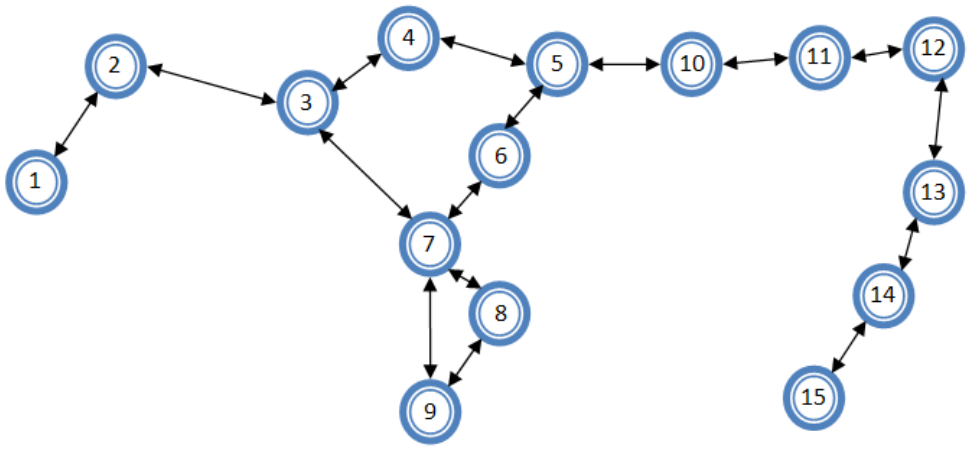


Figure 6.11: Topological Map: Example for Loop Closure Validation Illustration

Assume that at the current time, t, the mobile robot is at node 15 in the topological map. It then decides to travel in an arbitrary direction to a location which creates an image that triggers the loop closure detection module. The detected loop closure, LC_{t+1}, is matching the current location of the mobile robot at time $t+1$ to node 8 in the topological map. Assuming that the mobile robot is indeed at node 8, sequences of paths (i.e. 7-6-5, 7-3-4, 7-3-2, 9-7-6 and 9-7-3) can be generated. LC_{t+1} can thus be validated as true if any of these sequences can be executed in order with the loop closure detection module robustly matching the mobile robot to these nodes in the anticipated order. Each sequence should have the same number of nodes with the first node being an immediate neighbour of the node in the loop closure event LC_{t+1} (node 8 in this example). In addition, these immediate neighbours are limited to those which are prior to committing to the loop closure event. Increasing the number of nodes in each sequence increases the validity of the process but it also increases the total operation time. We describe the proposed

loop closure validation strategy as being based on the confirmation of its *"local context consistency"* where the context here is in reference to the loop closure event.

For our system, a simplified version of this strategy has been implemented instead (active loop closing validation operation mode). Our system stops and captures an image every time the visual odometry system finds the current location of the mobile robot to be more than x meters away from the previous node in the topological map. If the loop closure detection module matches the current location of the mobile robot with node 8 in the topological map at time $t+1$, the system directs the mobile robot to the most convenient immediate neighbour of node 8 (prior to committing to the loop closure) which requires the least change in its global heading using the nodal propagation technique described in Section 6.2.2. (1) If the matching node returned by the loop closure detection module is located within a distance of $1.25x$ meters (based on its topological relationship), this validates the loop closure event LC_{t+1}. On the other hand, (2) if a match is returned and does not satisfy the specified condition, system restoration is performed. Otherwise, (3) if no matches are found, the system directs the mobile robot to the next most convenient node. This process is iterated for n number of times and the matching node has to lie within a distance of $1.25nx$ given that $n \in (1, 3)$ for LC_{t+1} to be validated.

Referring to (1), as the matching node (a loop closure event itself) validates the loop closure event LC_{t+1} (node 8 in this example), LC_{t+1} also validates the matching node making it unnecessary to further validate it. The proposed algorithm is recursive at times if (2) occurs since the system would have to validate the matching node. To determine whether a detected loop closure requires further validation, some measures need to be in place to describe the degree of uncertainty of the detected loop closure. For the rank-based framework, this can be measured via the number of SURF correspondences whereas for the probabilistic framework, a likelihood ratio test (min. Bayes risk (Radke et al., 2005)) can be conducted in order to gauge the degree of risk involved when committing to a loop closure as compared to no loop closure. This in fact describes how the active loop closing validation operation mode is triggered active.

c System Restoration

The purpose of having the system restoration module is to enable the system to revert to a previous state of the system which has committed to a detected loop closure when the decision of loop closure is subsequently overturned. This module has strong associations with the loop closure validation module and functions analogous to system restorations available on operating systems. A restoration point is created whenever the loop closure validation module requires further evidence to validate the detected loop closure event. This restoration point contains all the information deemed necessary (i.e. state of the map, probability distributions). Assuming that a loop closure event, which requires further validation, is detected at time $t+1$, a restoration point, R_t, that contains all the necessary system information at time t is created. If the loop closure event is invalidated at time $t+3$, the system is firstly restored using R_t. This is then followed by forcing the state of the system at time $t+1$ to change from loop closure detected to no loop closure status.

Subsequent inputs to the system at time $t+2$ and $t+3$ are then reprocessed by the system again but this time having the state at time $t+1$ being no loop closure detected. As discussed previously, this process is recursive at times if the loop closure event is invalidated by another loop closure, requiring the system the perform active loop closure validation for this new loop closure event only if additional evidence is required. This example is illustrated in Fig. 6.12 and the flowchart of the complete loop closing system is illustrated in Fig. 6.13.

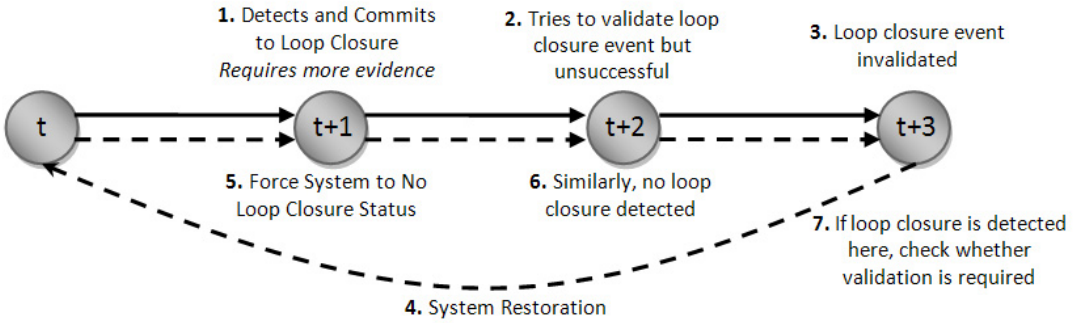


Figure 6.12: System Restoration: An Example

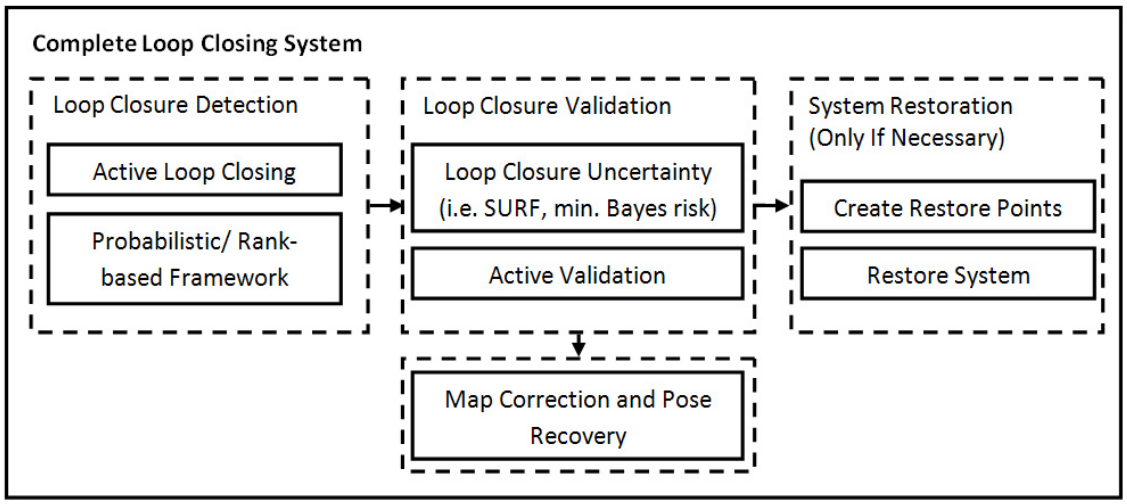


Figure 6.13: Flowchart of Complete Loop Closing System

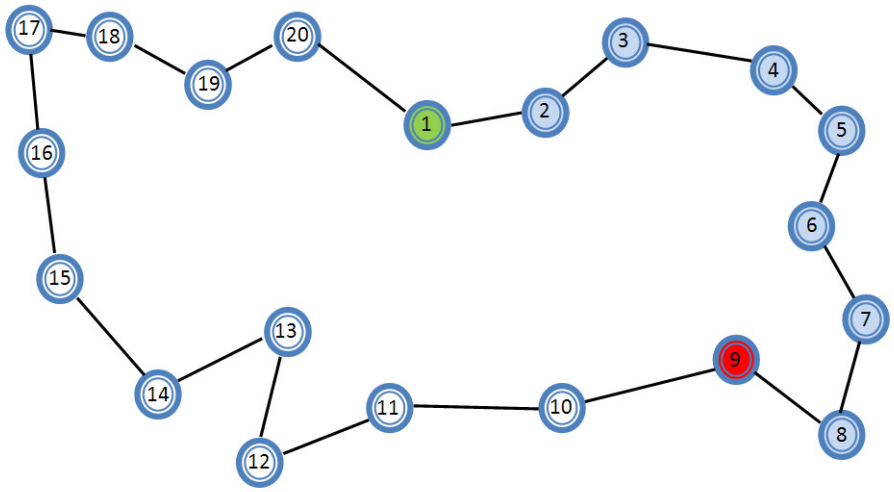
6.4 Semi-Autonomous Navigation

A semi-autonomous mobile robot is one which could not achieve its designated task completely without human guidance nor operate in an uncontrolled environment but illustrates considerable autonomy in achieving many of the sub-objectives required to complete the task. This section describes how our mobile robot performs goal seeking autonomously given a coarse map of the environment where the position of its goal is restricted to lie within the given map. It is semi-autonomous because *a priori* information in the form of a coarse map is provided to the system, where this map is built autonomously in a

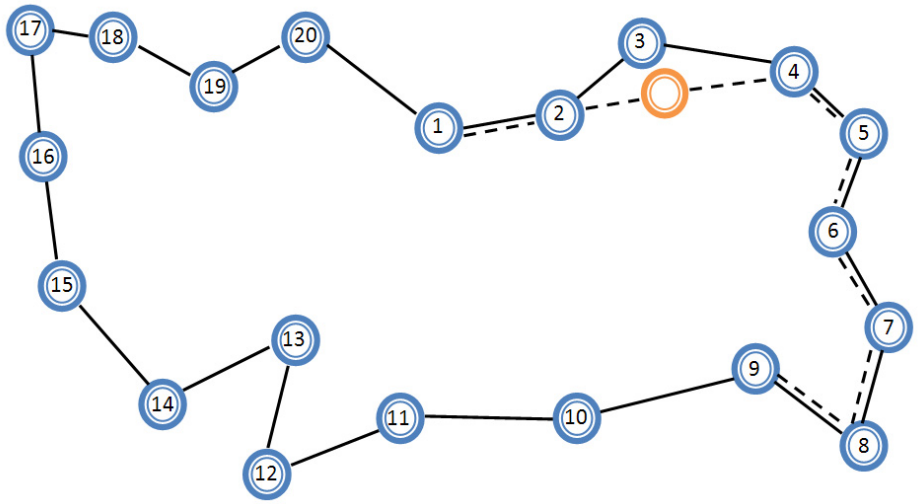
separate occasion by the system as the mobile robot was manually driven by a human operator. The support of a human operator in building a map, whilst not as impressive as autonomous map building, can be justified if the mobile robot is to operate for some time in this environment since this approach is more reliable and simpler. Once this map is loaded into the system, the human operator specifies the initial and goal locations in the map and the mobile robot autonomously finds its way to the goal.

As the mobile robot loads the map into its system (with its initial and goal locations specified), it plans a path to the goal using the nodal propagation algorithm described in Section 6.2.2. The path generated is the shortest path for our system but, as pointed out earlier, this algorithm can accommodate a more comprehensive cost function which takes relevant factors into consideration (i.e. terrain traversability). Assume that the topological map loaded into the system is as shown in Fig. 6.14(a). The information available in this topological map is identical to what has been described in Chapter 5 (i.e. global coordinates of nodes, positional variance, index of the image signature associated with this node in the database, etc). In this example, the initial location of the mobile robot is specified to be at node 1 and the goal is to reach node 9 autonomously. The system initially plans a path using the nodal propagation algorithm and yields the shortest path (based on the topological map) from node 1 to node 9 via nodes 2-3-4-5-6-7-8-9 in sequence. However, there are times when the mobile robot deviates from the planned sequence which may be due to the reactive obstacle avoidance overriding the system (will be described later in this chapter), ambiguities with the matching of image signatures, drift in estimated position as a result of severe perceptual aliasing or a combination of them. For example in Fig. 6.14(b), the mobile robot is supposed to navigate to node 3 but instead ended up at the node in orange. Nevertheless, this goal seeking mode is able to deal with deviations from the planned path and drives the mobile robot to the next best node of the current path. In this case, the current path computed using nodal propagation is 2-3-4-5-6-7-8-9 when the mobile robot is at the orange node. However, the system decides it should head to node 4 instead based on the Euclidean distance of the mobile robot's current position to each node in the current path and the 2D local grid map generated at the current location.

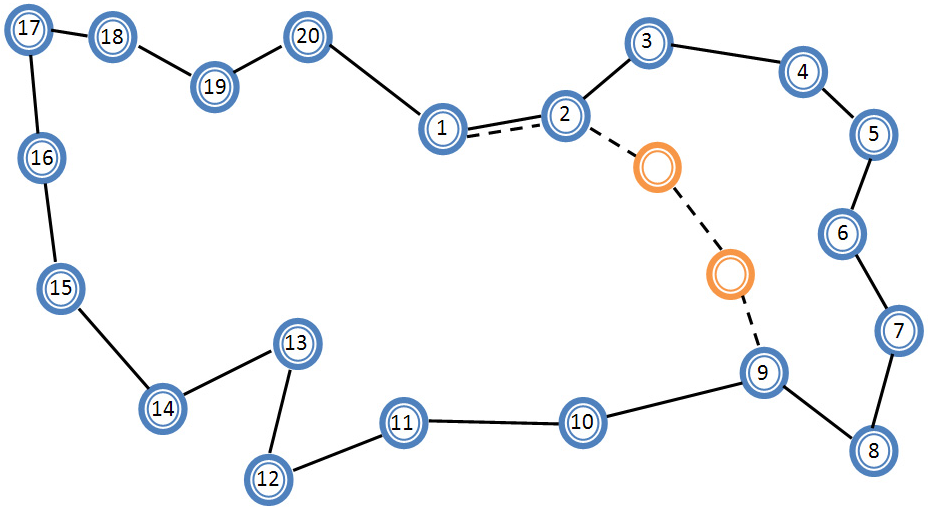
Although the path planned using the nodal propagation algorithm is guaranteed to return the shortest path, but it is restricted to the path where the human operator has driven the mobile robot in during the map building process. Normally, this map would not be a dense topological map but a rather coarse one implying that there may be better ways of reaching the target node. The system does not blindly follow the path suggested by the nodal propagation algorithm but is always on the lookout for shortcuts which may bring the mobile robot to the target node in a shorter time using the 2D local grid map. As demonstrated in Fig. 6.14, the mobile robot might find that it can possibly reach node 9 by taking this shortcut from node 2, deviating from the planned path and reaching node 9 using less time and effort. Of course, this requires the mobile robot to explore new areas and there is always an associated risk to it since this action may be leading the mobile robot to a dead end instead. Nevertheless, there are ways to prevent the mobile robot



(a) Toplogical Map with Initial Location (Green), Goal Location (Red) and Shortest Path (Light Blue)



(b) Goal Seeking Scenario 1: Node Skipping (– Actual Path Taken)



(c) Goal Seeking Scenario 2: Possible Shorter Path (– Actual Path Taken)

Figure 6.14: Example of Goal Seeking Mode

to take risky shortcuts by considering the effective range of the sensor and the Euclidean distance to the goal if the mobile robot is to head off in this direction.

6.5 Reactive Obstacle Avoidance

The disparity maps returned by the Bumblebee stereovision system are primarily used by the reactive obstacle avoidance system on the mobile robot. It determines whether the mobile robot should avoid an obstacle for the case where it is getting too close to one based on the compressed 1D disparity map, where each element of this 1D map is computed using Algorithm 6.1 and ignores any planned paths. Using this 1D disparity map, the system can decide whether it should go straight, steer left or right depending on where it perceives to have less obstruction. The Bumblebee stereovision system is mounted at the front of the mobile robot as shown in Fig. 6.15.

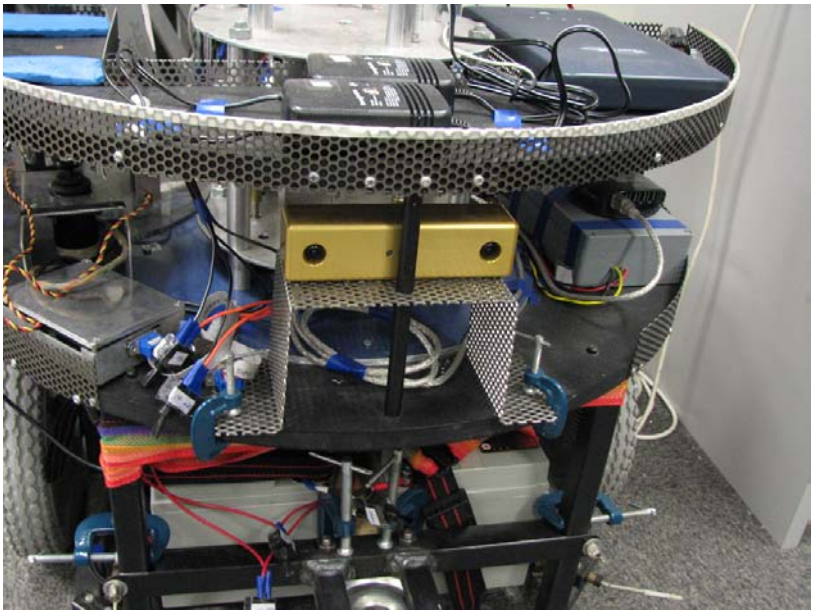


Figure 6.15: The Bumblebee Stereovision Camera

The disparity values range from 0(darkest)-255(brightest) with a lower value indicating that the location of the point in the scene is relatively further away as compared to one having a higher disparity value. To detect whether the mobile robot is too close to an obstacle, the system analyzes the compressed 1D disparity map where each element in the disparity map is basically the maximum disparity value of each column in the original 2D disparity map. If sufficient elements in the middle of the 1D disparity map are of high disparity values, the system should decide whether to steer right or left depending on where it perceives to have less obstruction. However, there are times when the Bumblebee stereovision system fails to establish corresponding points at certain pixel locations due to ambiguities. The unknown depths of these pixel elements are represented by a high disparity value by default. Since it is common to have many areas in the resulting disparity map with unknown depths due to ambiguities in stereovision, the resulting compressed 1D disparity map will not be useful if the previously described method is used since

Algorithm 6.1 Processing the Disparity Map from the Bumblebee

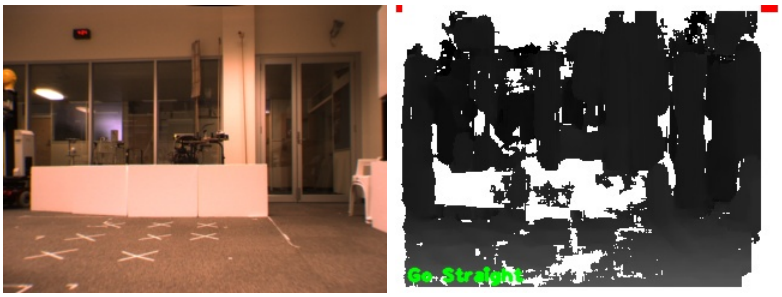
Require: 2D Disparity Map D_{map} initialized to 0

Define: 1D Compressed Disparity Map D_{comp}, accumulator $count = 0$, disparity threshold $Disp_{thresh} = 100$

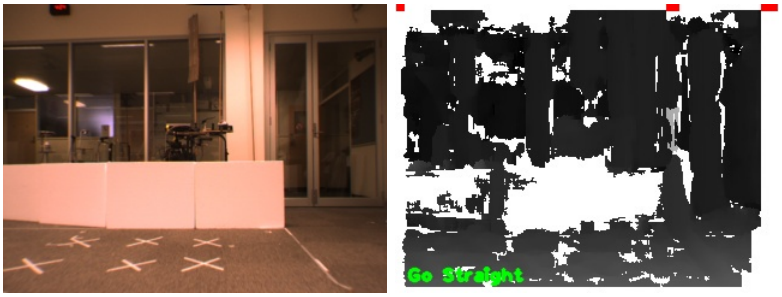
```
1:  for i = 1 to width of D_map do
2:    for j = 1 to height of D_map do
3:      if D_comp(i) < D_map(j,i) and D_map(j,i) < 200 then
4:        D_comp(i) = max(D_map(j,i))
5:      else if D_map(j,i) ≥ 200 then
6:        count++;
7:      end if
8:    end for
9:    if count ≥ 0.75*(height of D_map) then
10:     D_comp(i) = 150
11:   end if
12: end for
13: for i = 1 to width of D_comp do
14:   if D_comp(i) > Disp_thresh then
15:     D_comp(i) = 1
16:   else
17:     D_comp(i) = 0
18:   end if
19: end for
```

most of the elements in the 1D disparity map will have a high disparity value due to these ambiguities. Nevertheless, if there are sufficient high disparity values in a column (assuming that all these high disparity values are due to ambiguities), it is less risky to assume for this column to have a high disparity value (Line 9-11 in Algorithm 6.1) since there are too many elements in this column with unknown depths. These issues have been thoroughly addressed in Algorithm 6.1 and results from a discontinuous video sequence are shown in Fig. 6.16. Images on the left column in Fig. 6.16 were captured by the right camera of the Bumblebee stereovision system whereas the images on the right column in Fig. 6.16 show the corresponding disparity maps. The red pixels on the top row of each disparity map show the locations where a disparity value larger than 100 were found in the compressed 1D disparity map and the output from the reactive obstacle avoidance system is shown on the bottom left hand corner of each disparity map.

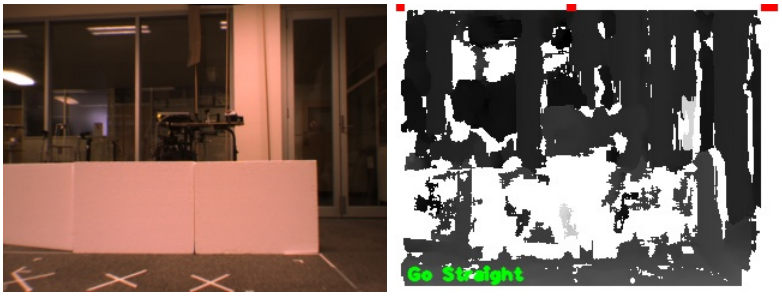
The reactive obstacle avoidance system overrides the control of the system whenever it issues a "Turn Left" or "Turn Right" direction since this means that the mobile robot is too close to a perceived obstacle in the environment which the omnidirectional stereovision system has failed to detect. The decision to turn left or right is purely based on the compressed 1D disparity map. In order to create a more intelligent reactive obstacle avoidance system; the state of the map, target loop closure location and visual odometry information are used when the mobile robot is operating in the active loop closing and exploration mode. As a result, this allows the reactive obstacle avoidance system to direct the mobile robot closer to the target node while avoiding obstacles before higher levels of the system regain control of the mobile robot.



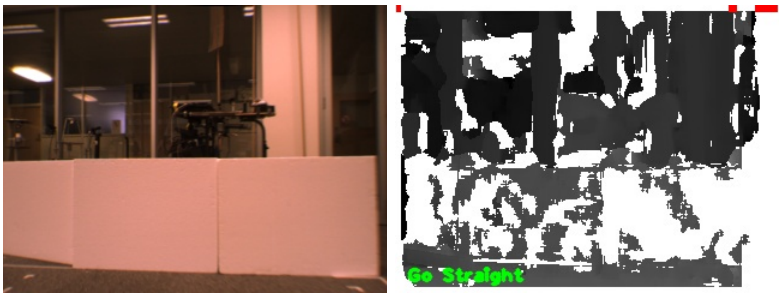
(a) (b)



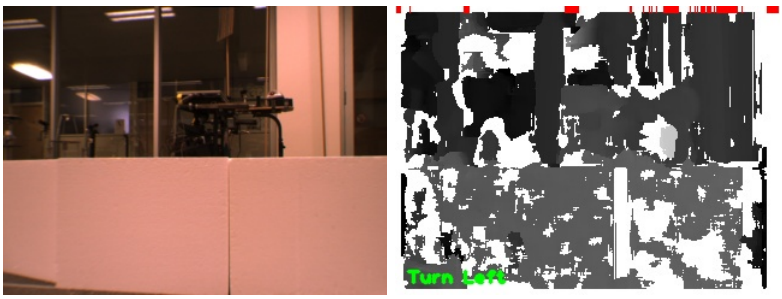
(c) (d)



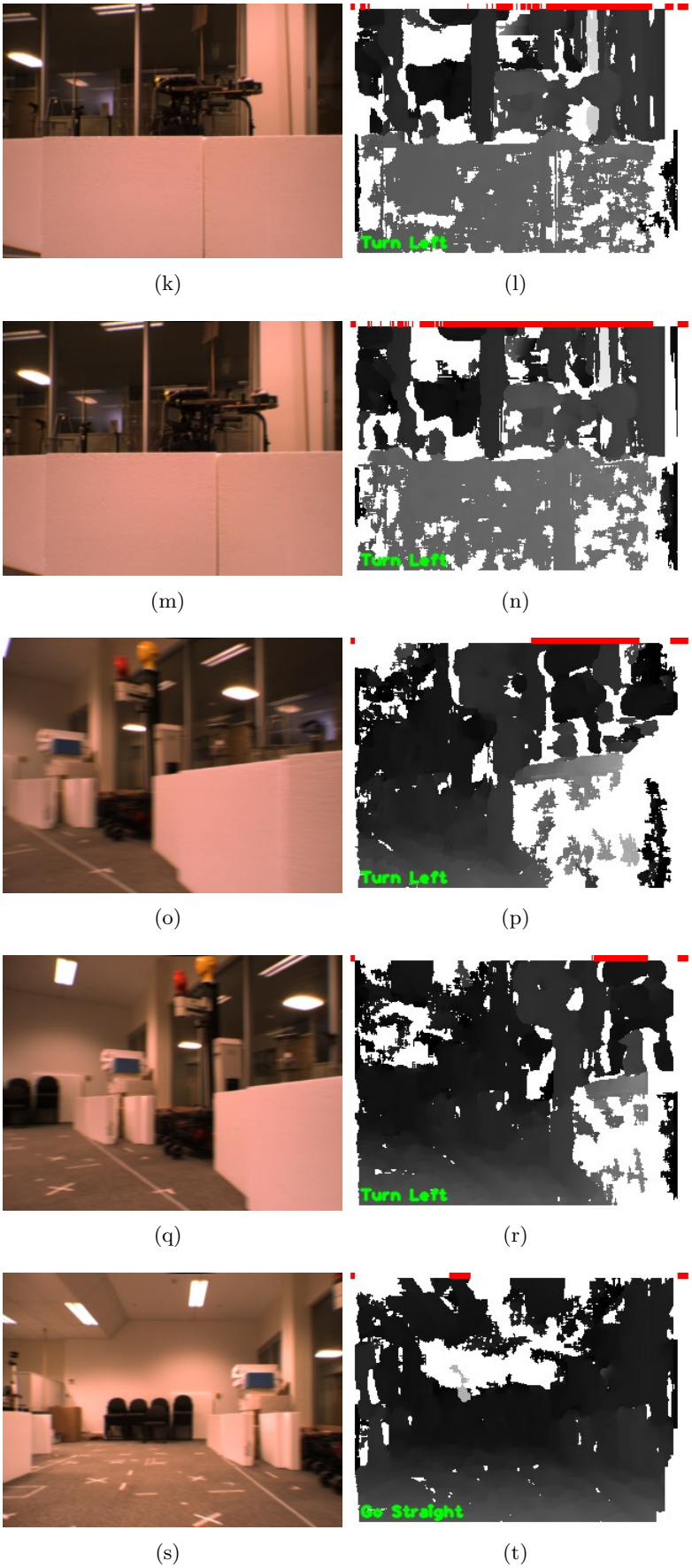
(e) (f)



(g) (h)



(i) (j)



(k) (l)

(m) (n)

(o) (p)

(q) (r)

(s) (t)

Figure 6.16: Reactive Obstacle Avoidance System

6.6 System Summary

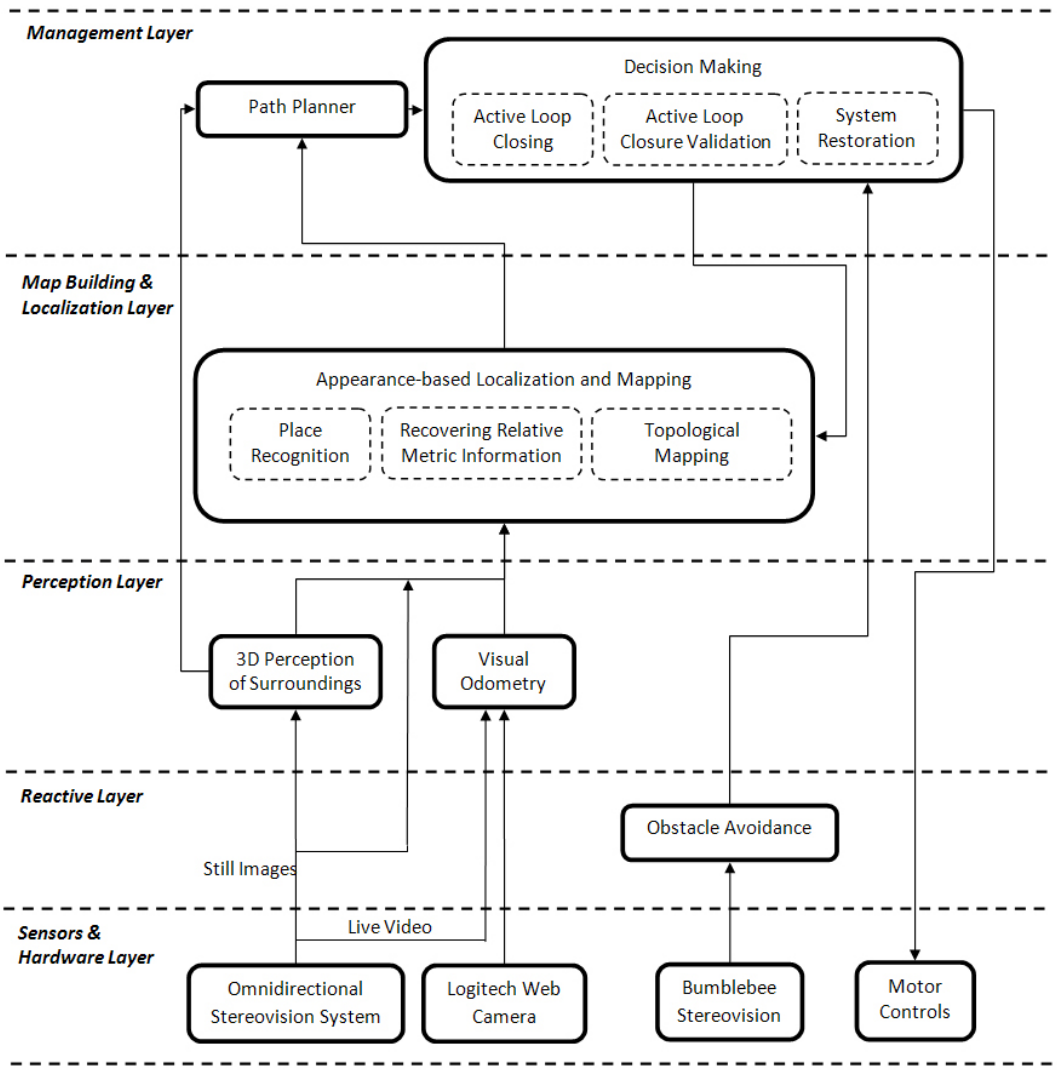


Figure 6.17: The Multilayer System Architecture

The entire system can be summarized into a multilayer architecture shown in Fig. 6.17. Following a bottom-up approach, the *sensors and hardware layer* consists of the omnidirectional stereovision system, Logitech web camera, Bumblebee stereovision system and motor controls on the mobile platform (not including the onboard computer systems). Right above the *sensors and hardware layer* is the *reactive layer*. The low-level reactive obstacle avoidance system described in Section 6.5, which steers the mobile robot away from obstacles that are not detected by the omnidirectional stereovision system, is located in this layer. The reactive obstacle avoidance system temporarily overrides the control of the mobile robot when such obstacles (not detected by the primary sensor) are detected until higher levels regain control of it.

On top of the *reactive layer* lies the *perception layer*. This layer provides a 3D perception of the mobile robot's surroundings using the omnidirectional stereovision system. Additionally, the *perception layer* also provides the mobile robot with real-time motion

estimates (visual odometry system) using the live video from the bottom catadioptric system (for orientation tracking) and the Logitech web camera (estimating total distance travelled). The omnidirectional stereovision system is thoroughly described in Chapter 2 and the visual odometry system is thoroughly described in Chapter 4.

The information from the *perception layer* is subsequently provided to the appearance-based localization and mapping system located in the *map building and localization layer*. As described in Chapter 5, the omnidirectional images are used by the place recognition system to discriminate between previously visited and new locations in the environment. 3D information from the omnidirectional stereovision system is also useful for recovering the 3D relative metric information between two locations (Section 5.4.3), required for maintaining a globally consistent topological map during loop closure. Finally, by combining the appearance-based localization and mapping system with motion estimates from the visual odometry system, the mobile robot can perform topological SLAM.

The top layer is referred to as the *management layer*. This is where the path planner and decision making module are located. As detailed in Section 6.2.2, the path planner utilizes information in the evolving topological map built by the mobile robot and 3D information from the omnidirectional stereovision system to create 2D local grid maps for path planning purposes. Finally, the decision making module will decide the next course of action for the mobile robot depending on whether it is operating in the autonomous or semi-autonomous navigation mode. In the autonomous navigation mode, the mobile robot will balance its effort between exploration and loop closing (also performing active loop closing, active loop closure validation and system restoration). In contrast, when the mobile robot is in the semi-autonomous navigation mode, the mobile robot is goal-oriented and seeks for its target destination within the given topological map. As a result, in the semi-autonomous navigation mode, the decision making module will not be equipped with the active loop closing and active loop closure validation strategies (as described in Sect. 6.4, the mobile robot will be revisiting nodes anyway on its way to the target destination unless it is taking a detour from the planned path).

6.7 Results

The complete system was tested extensively in a variety of environments ranging from indoors to outdoors. In addition, it was also tested in different modes of operations; (a) fully autonomous, (b) offline and (c) semi-autonomous modes. The following subsections thoroughly describe how the respective experiments were conducted and the results of each.

6.7.1 Fully Autonomous Experiments

The mobile robot was tested in indoor, semi-outdoor and outdoor environments in the fully autonomous mode. The underlying algorithms used are the same as described in the previous sections except for the indoor experiment. The indoor experiment was an initial attempt to demonstrate the feasibility of the proposed appearance-based localization and

mapping system which does not employ an active loop closing strategy and does not actively validate ambiguous loop closure events. In this experiment, the mobile robot selects a frontier heading (extracted from the 2D local grid map) based on the possibility of exploration and selects the path associated to the frontier heading which requires the least change in its global heading. If all frontier headings generated using the current 2D local grid map of the mobile robot have been explored, the mobile robot uses the nodal propagation algorithm described in Section 6.2.2 to return to the closest node which has been registered as having unexplored regions. Otherwise, it returns to its initial position. Please note that similar capabilities can be found on the mobile robot which performs the semi-outdoor and outdoor experiments.

General across all experiments, the mobile robot starts in an unknown environment and this starting position is the origin of the global coordinate system. The mobile robot captures a pair of stereo images from the omnidirectional stereovision system and plans its next course of action. As the mobile robot executes its planned path, it performs real-time visual odometry using the technique described in Chapter 4 and switches on the reactive obstacle avoidance system described in Section 6.5. It stops at intervals of 1m (Euclidean distance measured from the location of the previous node to the current location of the mobile robot), captures omnidirectional stereo images in order to produce the 2D local grid map for planning its next course of action and builds a topological map which is associated to a database of image signatures. The rank-based framework was used in all experiments and the resulting topological map built by the system was compared to ground truth (only for the semi-outdoor and outdoor experiments). Using the plan views of the experimental area created using stitched laser scans from a Riegl LMS Z420i terrestrial laser scanner, the ground truth location for each node in the topological map was recovered by utilizing the omnidirectional images captured by the mobile robot during the experiments and exploiting the grid pattern on the floor (for the semi-outdoor environment) or distinctive landmarks in the environment (outdoor environment) to localize the mobile robot in the environment. The plan view of the primary map created by the terrestrial laser scanner shown in Fig. 6.18 was provided as a courtesy of Ho (2010), who was previously a PhD candidate in the laboratory. Experiments were conducted in various parts of this primary map, separated into 3 different regions as shown in Fig. 6.19. The weak grid lines (from the paving pattern) in the original map created by the terrestrial laser scanner were enhanced in Fig. 6.19(a) for better clarity. Each cell of the grid map is approximately 60 x 60 $\pm$ 5 cm in each dimension. The following defines all system parameters (remain as constants for all experiments, unless stated otherwise).

- T_D = 1m - Euclidean distance threshold for accepting a matching candidate as a valid loop closure event described in Section 6.3.3.

- T_R = -200 - Haar similarity score threshold for selecting matching candidates described in Sections 6.3.3 (overview of loop closure detection) and 5.4.1.

- As described in Section 6.3.3 (loop closure validation), the risk associated to committing a loop closure event for the rank-based framework is measured by the total

number of SURF correspondences. Matching candidates which produce a number of SURF correspondences in the range of 0-19 are rejected, 20-35 requires active validation and those with more than 35 correspondences do not require further validation.

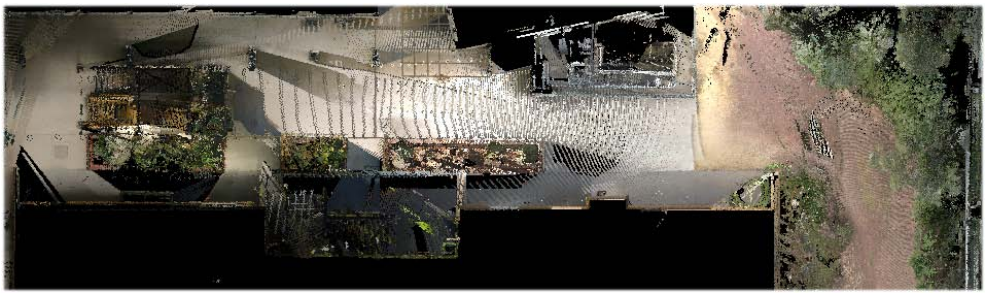
- The proposed active loop closure validation strategy is iterated for a maximum of 3 times, where n $\in$ (1,3) (defined in Section 6.3.3 (loop closure validation)).
- Size of Haar image signature: 56 x 14 (Section 5.2).
- Resolution of omnidirectional stereo images is 640 x 480 (not unwarped).
- Stereo correlation mask size: 31 x 31, Max. Disparity Search Range: 64, Disparity Step Size: 0.5 pixels with baseline fixed at 30cm unless stated otherwise.

Figure 6.18: Plan View of Stitched Riegl Laser Scans (Courtesy of Nghia Ho)

Indoor Experiment

For the indoor experiment, the mobile robot neither seeks loop closures nor validates detected loop closure events. As shown in Fig. 6.20, the indoor laboratory environment was setup in a way that its structure resembled a loop, thus, allowing the mobile robot to return to previously visited locations and detect loop closures even if it was not equipped with an active loop closing strategy. The main purpose of this experiment was to investigate the feasibility of the proposed system based on the combination of visual odometry, omnidirectional stereovision and an appearance based localization and mapping system.

(a) Semi Outdoor Environment 1 (Grid Resolution: 60 x 60 ± 5 cm in each dimension)



(b) Semi Outdoor Environment 2

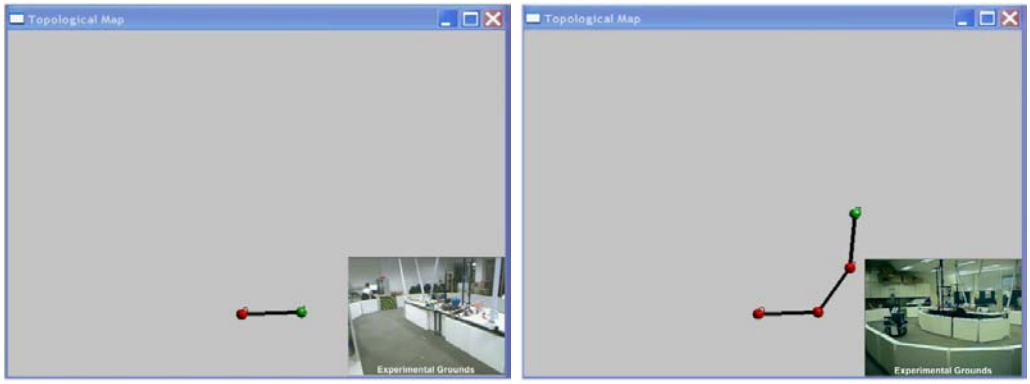
(c) Outdoor Environment

Figure 6.19: Plan Views of Different Experimental Areas

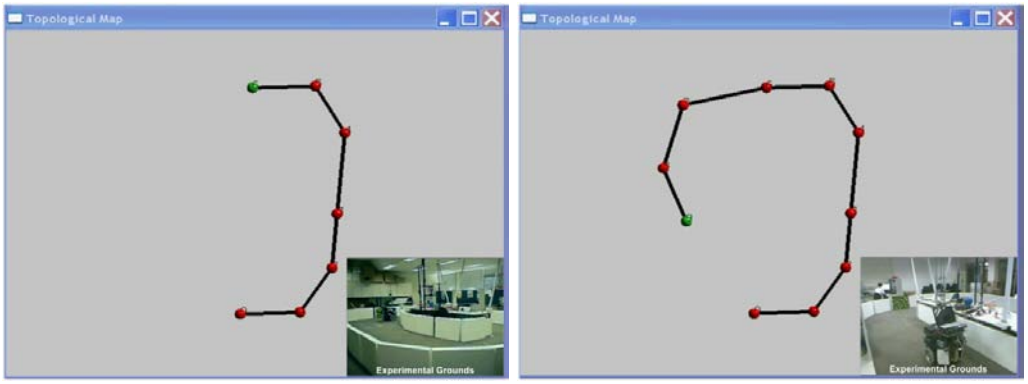
The state of the evolving topological map at different time steps, t, can be found in Fig. 6.21. The mobile robot was at its initial starting position at $t = 0$. It explored the environment and created a trajectory of nodes which resembled the structure of the environment from $t = 1, 2..., 12$. At $t = 13$, it backtracked due to noisy sensor measurements from the omnidirectional stereovision system and successfully localized itself to node 11 in the topological map. Nevertheless, it managed to return to node 0 and detected a loop closure at $t = 16$ (on the way detecting another loop closure at node 12). From $t = 17, ..., 21$, the mobile robot was able to relocalize itself to existing nodes in the topological map and Fig. 6.22 shows some of the matching omnidirectional image pairs.

(a) (b)

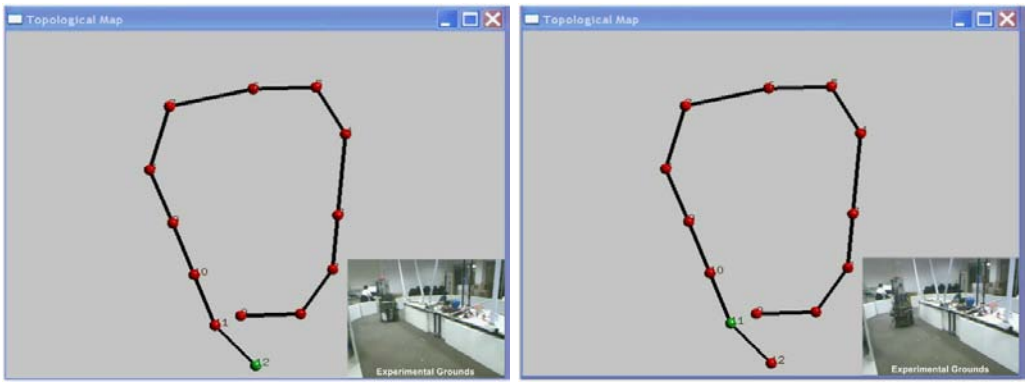
Figure 6.20: Indoor Experimental Environment



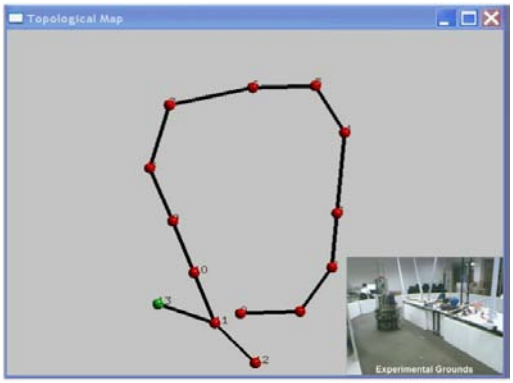
(a) t=1 (b) t=3



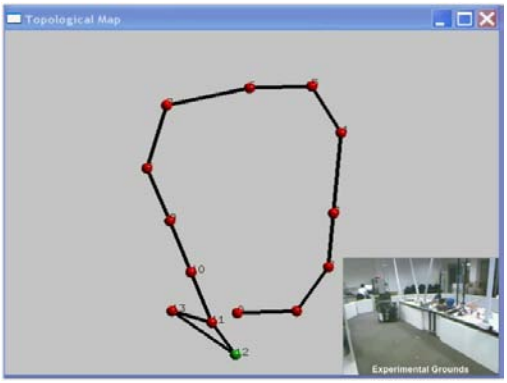
(c) t=6 (d) t=9



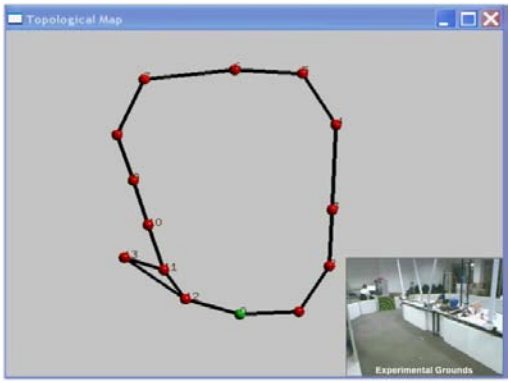
(e) t=12 (f) t=13



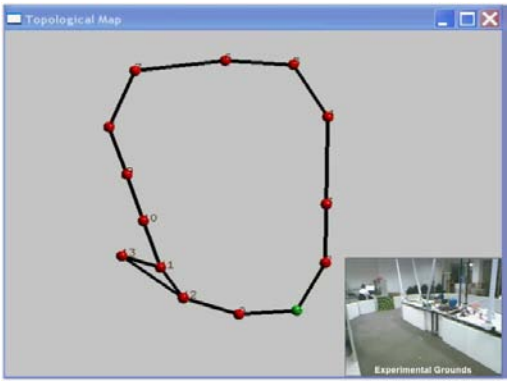
(g) t=14



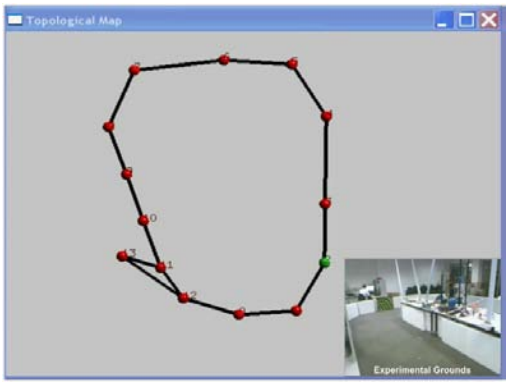
(h) t=15



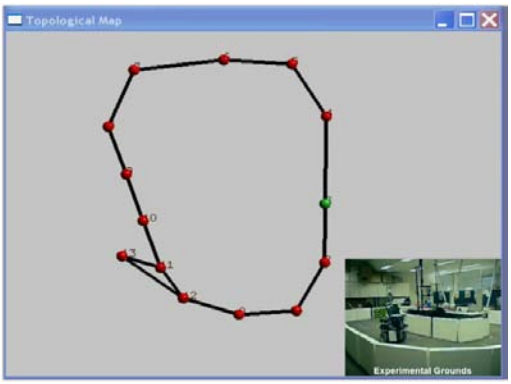
(i) t=16



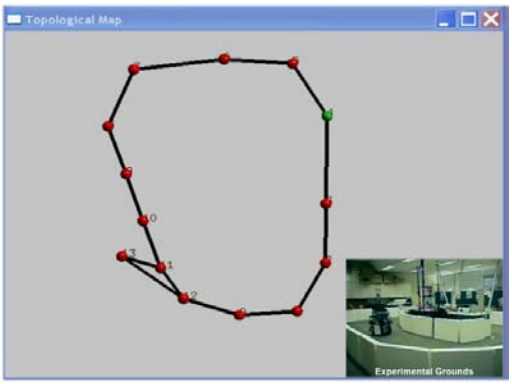
(j) t=17



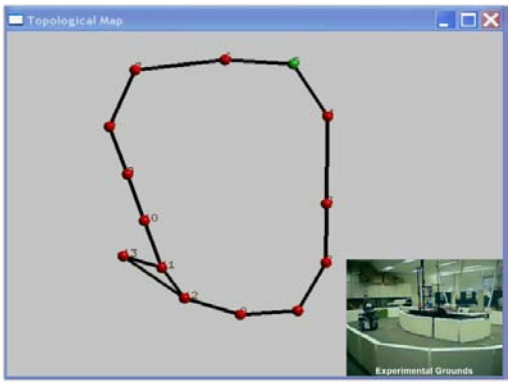
(k) t=18



(l) t=19



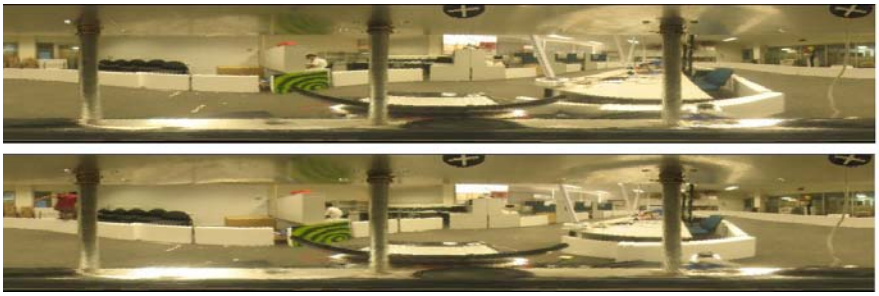
(m) t=20



(n) t=21

Figure 6.21: Indoor Experiment

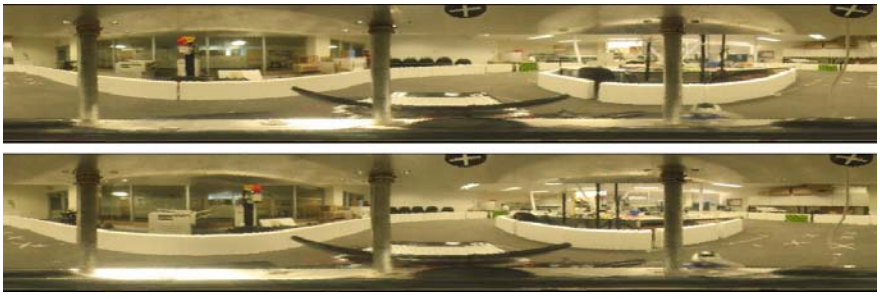
(a) t=13



(b) t=16

(c) t=17

(d) t=18



(e) t=19

Figure 6.22: Matching Omnidirectional Image Pairs for Indoor Exp. (Loop Closure Detection)

Semi-Outdoor Experiments

The semi-outdoor experiments were conducted with the mobile robot equipped with the complete loop closing system described in Section 6.3.3. Generally, it is accepted that an indoor environment is one that is unaffected by the varying light intensity due to the position of the sun at different times of the day or the occlusion of the sun on a cloudy day. An indoor environment normally has a constant light source (e.g. fluorescent lamps, incandescent light bulbs) whereas an outdoor environment is normally an area which is not covered by a roof and is directly affected by the varying light intensity of the sun. A semi-outdoor environment is the category between an indoor and outdoor environment. A total of 7 semi-outdoor experiments are illustrated in Fig. 6.25-6.43 with 3 supplementary experiments available in Appendix E. Experiments in Fig. 6.25-6.40 and the first three supplementary experiments in Appendix E were conducted in the semi-outdoor environment shown in Fig. 6.23 whereas the experiment in Fig. 6.43 was conducted in the semi-outdoor environment shown in Fig. 6.24.

Parts of the first semi-outdoor environment in Fig. 6.23 were partially covered by a roof with semi-opaque stripes and parts of it were covered by a ceiling with fluorescent lamps which automatically switches on at different times of the day or when it is too dark. The second semi-outdoor environment in Fig. 6.24 was made up of partially covered and partially open areas resulting in changing lighting illumination. All experiments were conducted in a static or partially static environment where in some cases, the mobile robot was operating while people were moving in the environment, with plants in the vicinity swaying back and forth due to strong winds.

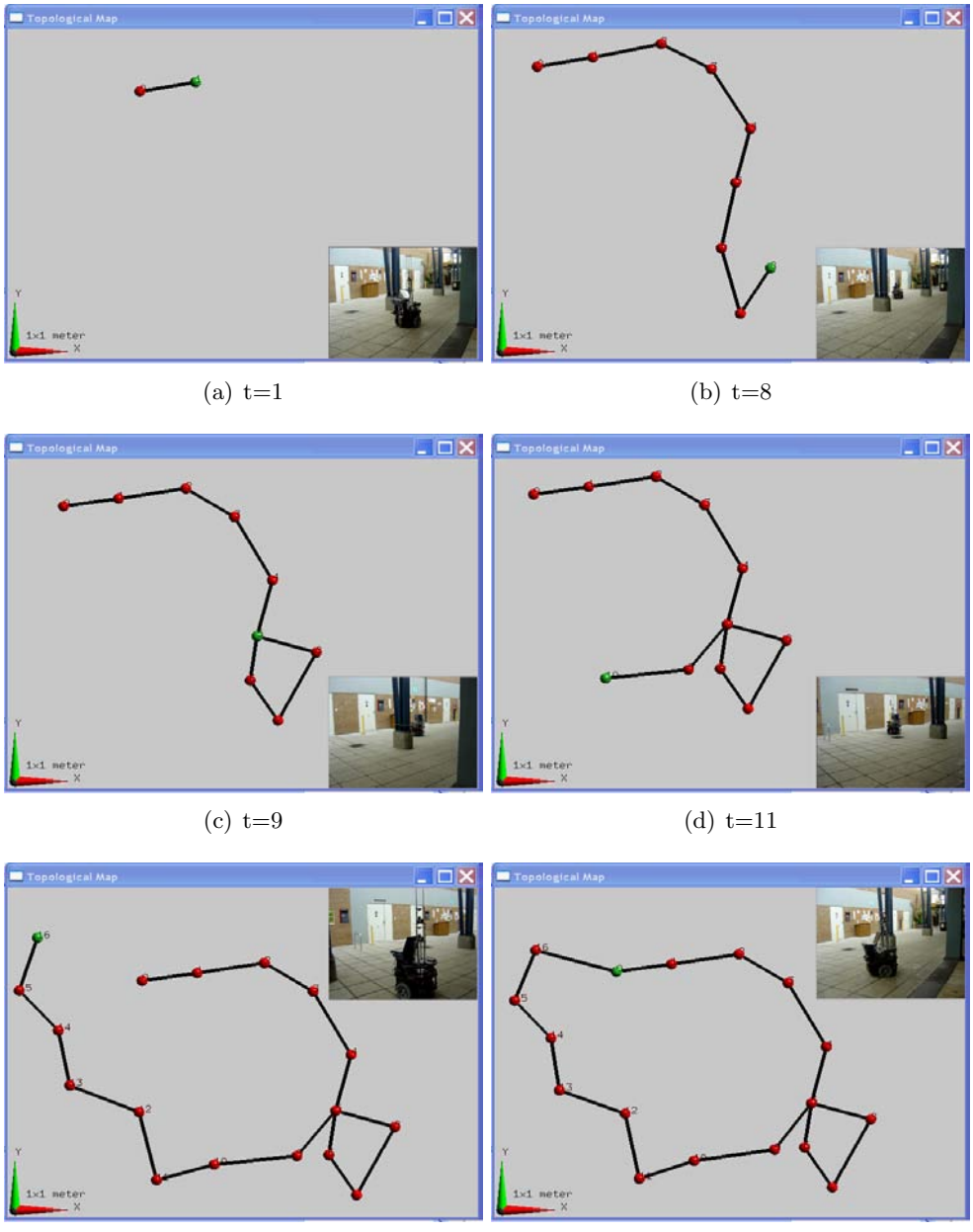
(a) (b) (c)

Figure 6.23: Semi Outdoor Experimental Environment 1

(a) (b) (c)

Figure 6.24: Semi Outdoor Experimental Environment 2

Semi-Outdoor Experiment 1 The results from this experiment were straightforward. The mobile robot started exploring the environment at $t = 0$. It seems that it was trying to perform a clockwise loop trajectory when it decided to take a turn resulting in the trajectory at $t = 8$. Nevertheless, it turned back and performed loop closing at $t = 9$ (small loop). Finally, it managed to return to its starting location and closed the main loop at $t = 18$. This simple experiment showed the effectiveness of an active loop closing strategy on the mobile robot and the importance to perform loop closing regularly for maintaining a globally consistent map. The resulting topological map was properly scaled, rotated and superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.26. The matching omnidirectional image pairs which raise loop closure events at $t = 9$ and $t = 18$ can be seen in Fig. 6.27.



(a) t=1 (b) t=8

(c) t=9 (d) t=11

(e) t=17 (f) t=18

Figure 6.25: Semi Outdoor Experiment 1

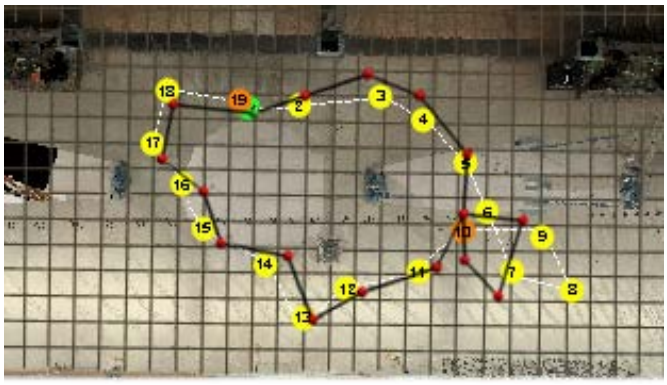


Figure 6.26: Semi Outdoor Exp.1 - Comparing Against Ground Truth

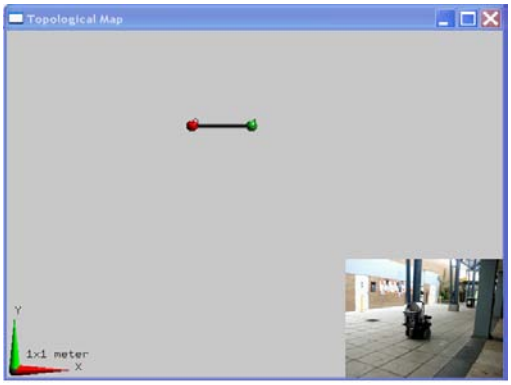
(a) t=9

(b) t=18

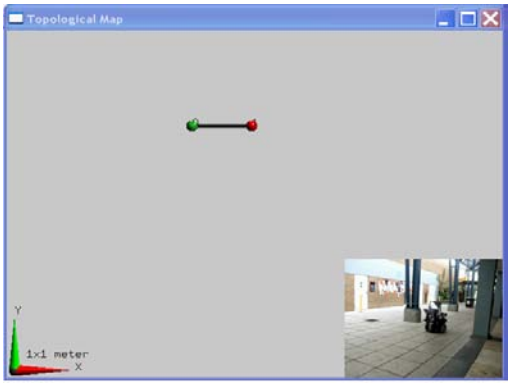
Figure 6.27: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 1 (Loop Closure Detection)

Semi-Outdoor Experiment 2 In this experiment, the mobile robot detected a loop closure event at $t = 2$, $LC_{t=2}$, which matched the current location of the mobile robot to node 0 in the topological map. This was in fact a false positive. $LC_{t=2}$ passed the initial validations but was found to require more evidence based on the total number of SURF correspondences. Using the proposed active loop closure validation strategy, it tried to validate $LC_{t=2}$. However, $LC_{t=2}$ was invalidated by a loop closure event at $t = 5$, $LC_{t=5}$, which triggered it to perform system restoration. Fig. 6.28(e) shows the state of the map being restored to the state at $t = 1$, which is the state before $LC_{t=2}$ was detected. It then forced the state of the system to change from loop closure to no loop closure detected at $t = 2$. Finally, $LC_{t=5}$, which invalidated $LC_{t=2}$, can be seen in Fig. 6.28(h). Subsequently, it detected loop closures at $t = 14$ and $t = 15$. Nevertheless, since

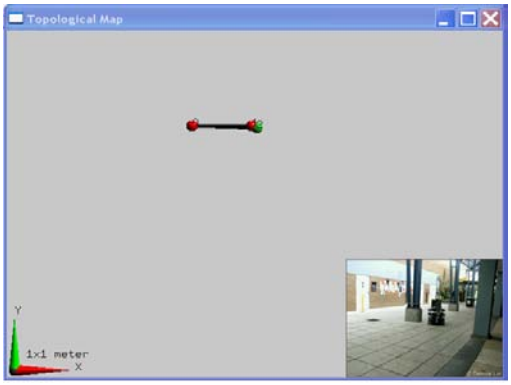
the condition $D_E^0 + D_F < D_{max}$ (Section 6.3.3(active loop closing strategy)) remained valid, its target loop closing location remained as node 0 and it successfully reached its target destination at $t = 16$. The resulting topological map was properly scaled, rotated and superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.29. Randomly selected matching omnidirectional image pairs which raise loop closure events are illustrated in Fig. 6.30. Observe how similar in terms of scene composition the image pair at $t = 2$ is, which produces the false positive. The following are node pairs (node numbers based on the ground truth trajectory in Fig. 6.29) which raise loop closure events in this experiment: 5-3, 14-10, 15-11 and 16-1. Please take note that this does not include false loop closure events that had been restored and rectified.



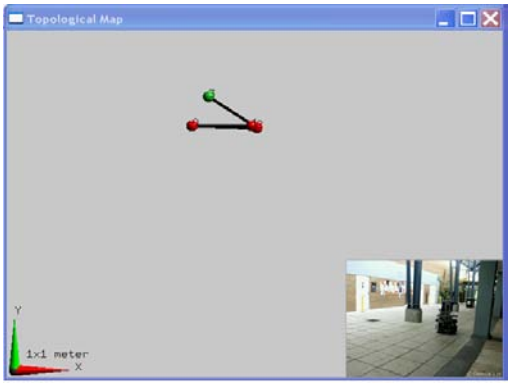
(a) t=1



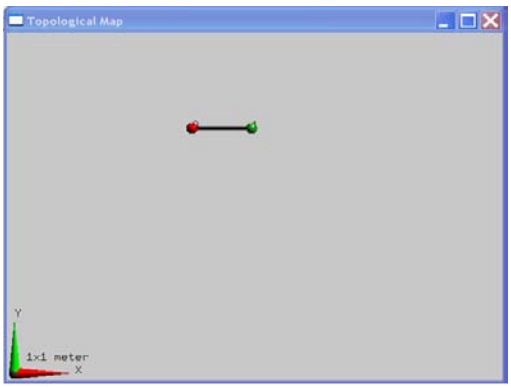
(b) t=2



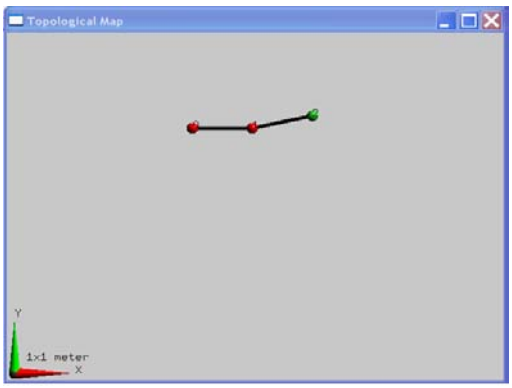
(c) t=3



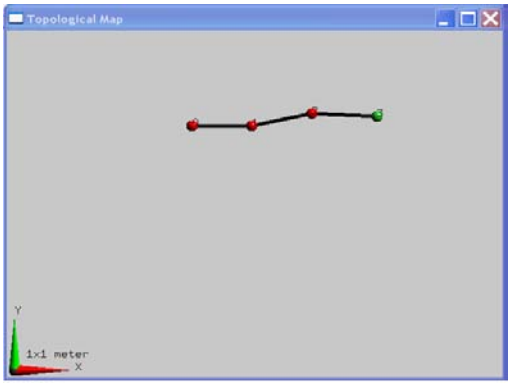
(d) t=4



(e) t=5 (Recovery)

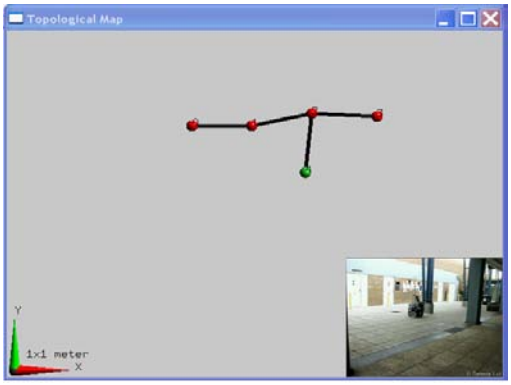


(f) Recovery



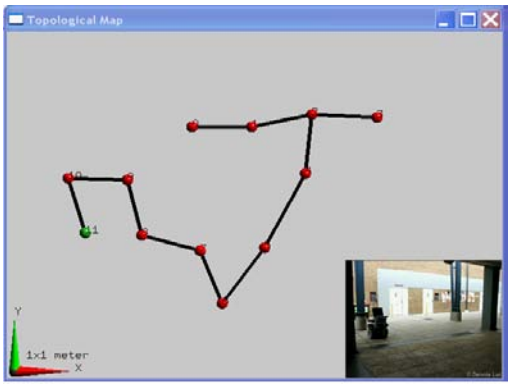
(g) Recovery

(h) Recovery

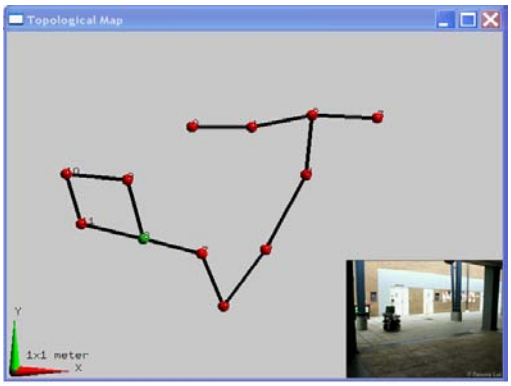


(i) t=6

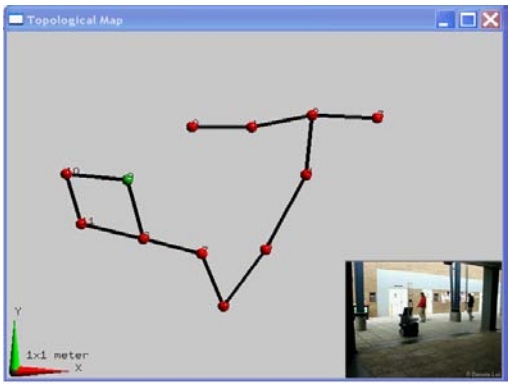
(j) t=9



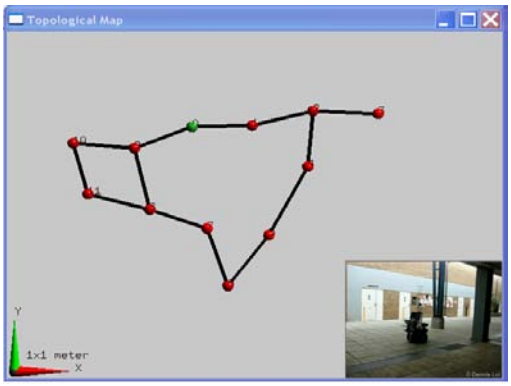
(k) t=13



(l) t=14



(m) t=15



(n) t=16

Figure 6.28: Semi Outdoor Experiment 2

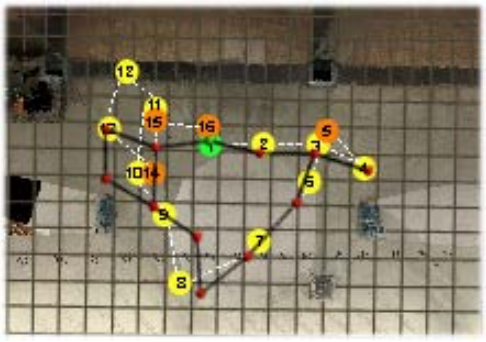
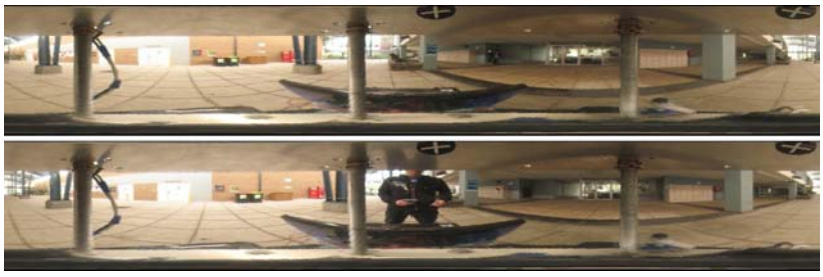


Figure 6.29: Semi Outdoor Exp.2 - Comparing Against Ground Truth

(a) t=2

(b) t=5

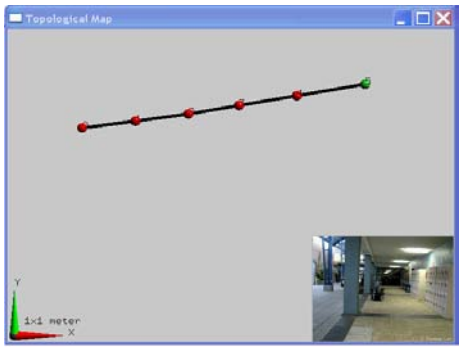
(c) t=14



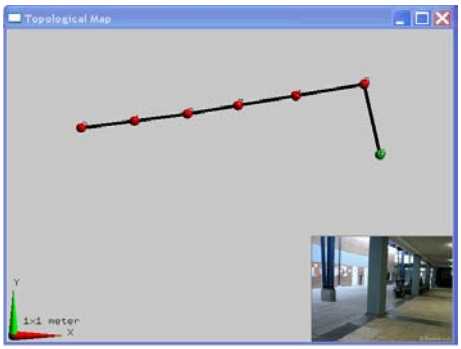
(d) t=16

Figure 6.30: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 2 (Loop Closure Detection)

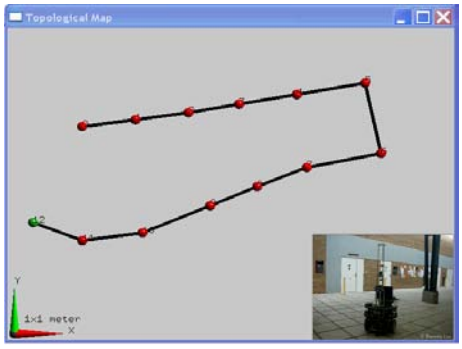
Semi-Outdoor Experiment 3 In this experiment, the mobile robot was clearly illustrated to be switching from the pure exploration mode to the loop closing and exploration mode at $t = 5 - 6$ and detected its first loop closure at $t = 13$ in Fig. 6.31. The mobile robot detected multiple loop closures throughout its entire course of navigation and a subset of these loop closures were detected at $t = 14$ and $t = 25$. All loop closures detected either did not require further validation or had been validated. As such, system restoration was not performed in this experiment. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.32. Some of the matching omnidirectional image pairs which raise loop closure events in this experiment are also provided in Fig. 6.33. The following are node pairs (node numbers based on the ground truth trajectory) which raise loop closure events in this experiment: 14-1, 15-2, 16-11, 19-11, 26-21, 27-18, 29-17, 30-11 and 32-12.



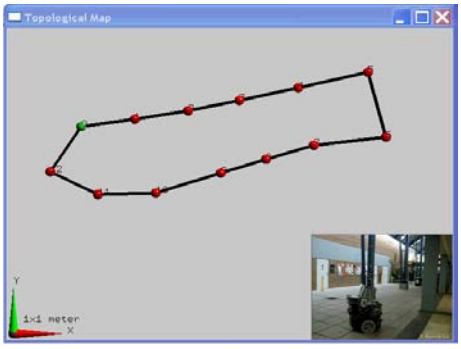
(a) t=5



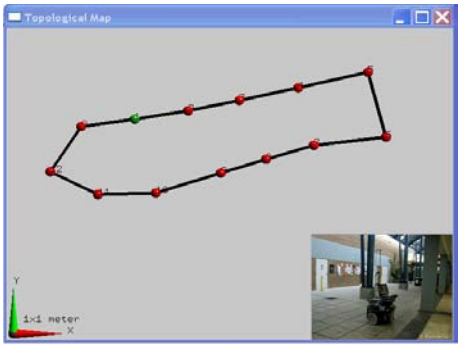
(b) t=6



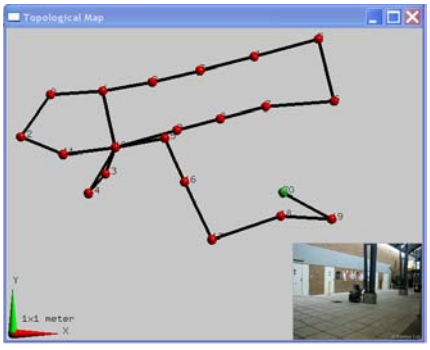
(c) t=12



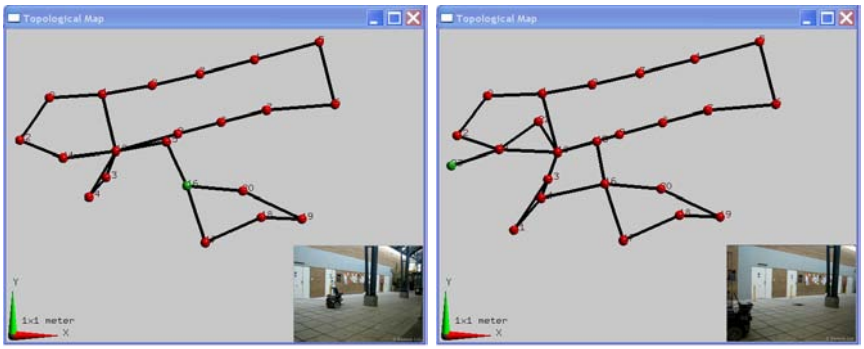
(d) t=13



(e) t=14



(f) t=24



(g) t=25 (h) t=32

Figure 6.31: Semi Outdoor Experiment 3

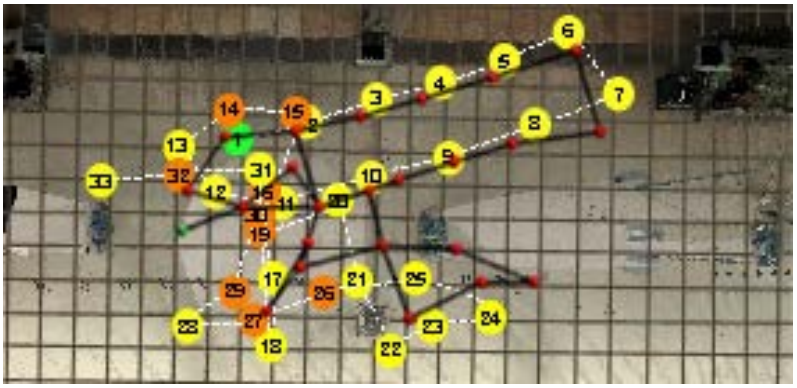
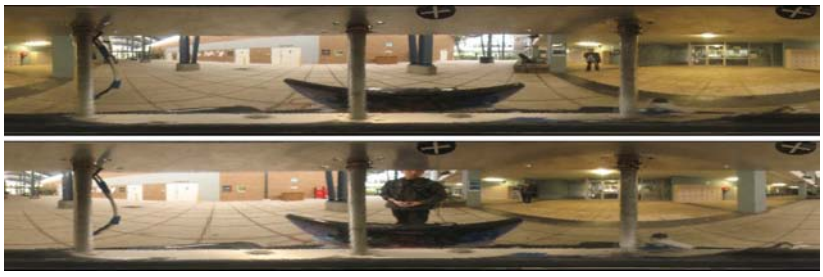
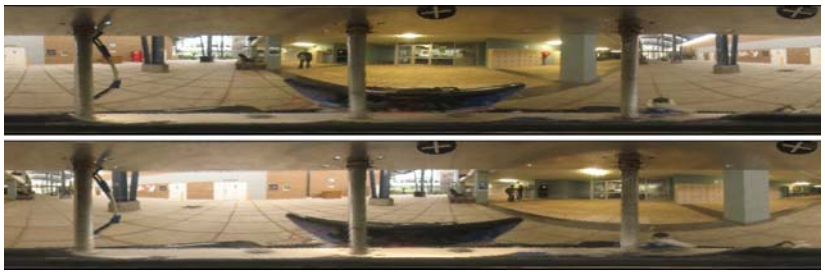


Figure 6.32: Semi Outdoor Exp.3 - Comparing Against Ground Truth



(a) t=13

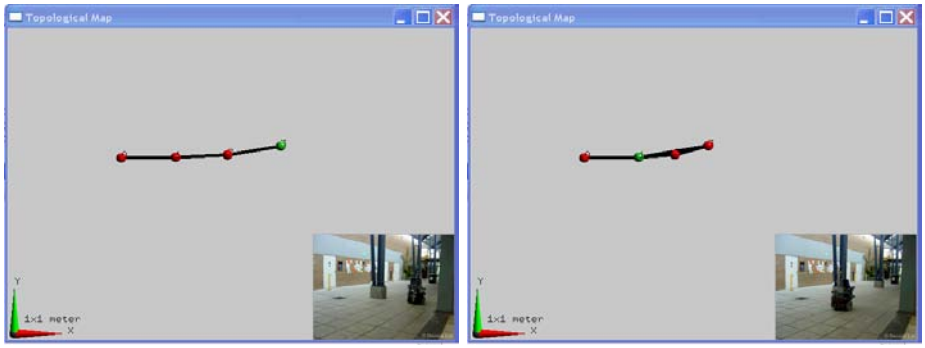


(b) t=14

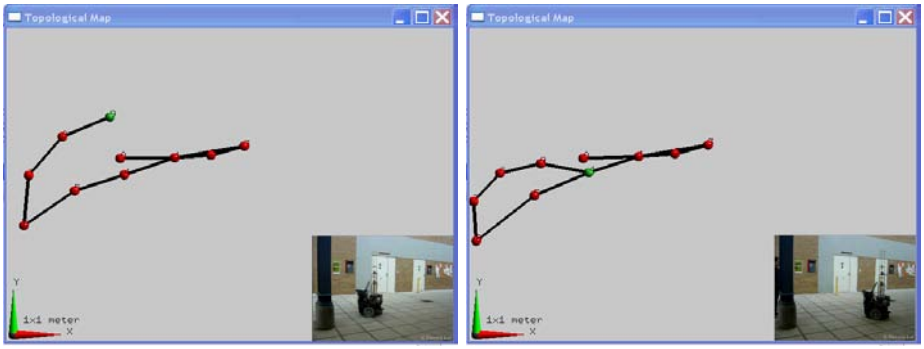
(c) t=25

Figure 6.33: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 3 (Loop Closure Detection)

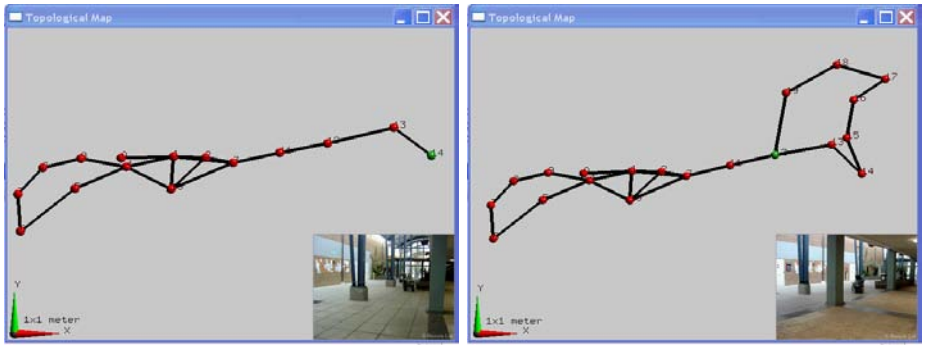
Semi-Outdoor Experiment 4 This experiment illustrates the mobile robot detecting multiple loop closures as it explored the environment. It is shown in Fig. 6.34 that the mobile robot managed to close several loop like trajectories at $t = 11$, $t = 26$ and $t = 33$. However, as being defined in Section 6.3.2, the term loop closing has been lately used to describe the revisiting of previously explored locations in general. Adhering to this definition, there were multiple loop closures detected between $t = 11, ..., 18$. The resulting topological map was scaled and rotated accordingly before it was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.35. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. 6.36. The following are node pairs (node numbers based on the ground truth trajectory) which raise loop closure events in this experiment: 5-2, 13-6, 15-3, 16-2, 17-14, 18-4, 28-20 and 35-7.



(a) t=3 (b) t=4



(c) t=10 (d) t=11



(e) t=18 (f) t=26

(g) t=32 (h) t=33

Figure 6.34: Semi Outdoor Experiment 4

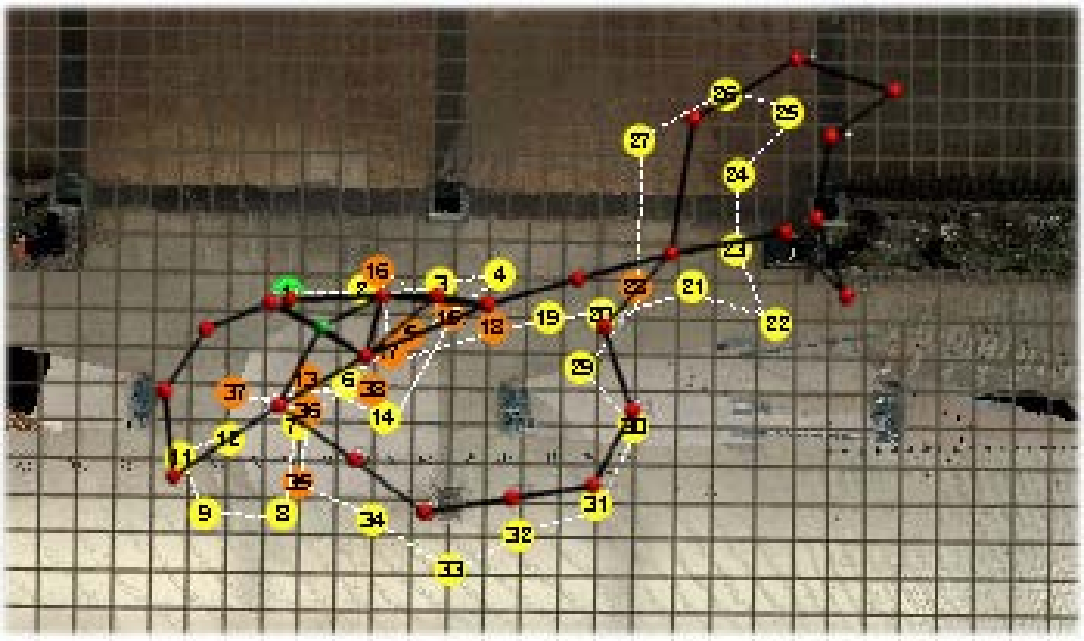


Figure 6.35: Semi Outdoor Exp.4 - Comparing Against Ground Truth

(a) t=11

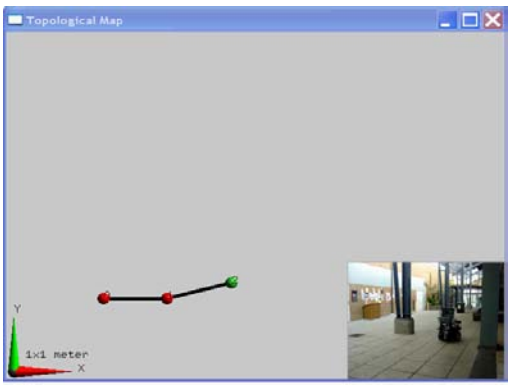
(b) t=26

(c) t=33

Figure 6.36: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 4 (Loop Closure Detection)

Semi-Outdoor Experiment 5 The following experiment is one of the more complete experiments in the sense that it illustrates the complete behaviour of the mobile robot performing active loop closing, active loop closure validation, system restoration, switching from the pure exploration to the loop closing and exploration mode and recovering from position estimate errors as loop closing was performed. At $t = 4$, it detected a loop closure at node 2 of the topological map which was verified to be valid based on ground truth (matching node 5 to node 3 according to the node numbers of the ground truth trajectory in Fig. 6.38). Subsequently at $t = 5$, it detected a loop closure event matching the current location of the mobile robot to node 3 of the topological map (matching node 6 to node 4 of the ground truth trajectory). The mobile robot tried to validate this loop closure event but was unsuccessful prompting system restoration at $t = 8$. Observe the position of node 4 in the topological map as compared to its ground truth position (node 6 of the ground truth trajectory). The mobile robot lost track of its heading for a while due to perceptual aliasing but it eventually recovered as it closed the loop at $t = 14$. It proceeded with exploration and closed another smaller loop at $t = 22$. At $t = 25$, it created node 20 in the topological map (node 26 of the ground truth trajectory). The

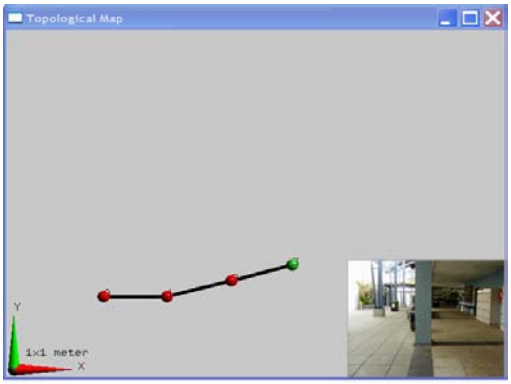
position of node 20 was found to be incorrect when compared to ground truth but the mobile robot recovered from the localization error at $t = 26$ when it revisited node 14 in the topological map. Finally, it proceeded with exploration and closed a bigger loop at $t = 32$. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.38. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. 6.39. The following are node pairs (node numbers based on the ground truth trajectory) which raise loop closure events in this experiment: 5-3, 15-3, 23-19, 24-18, 25-17, 27-17 and 33-1. Please take note that this does not include false loop closure events that had been restored and rectified.



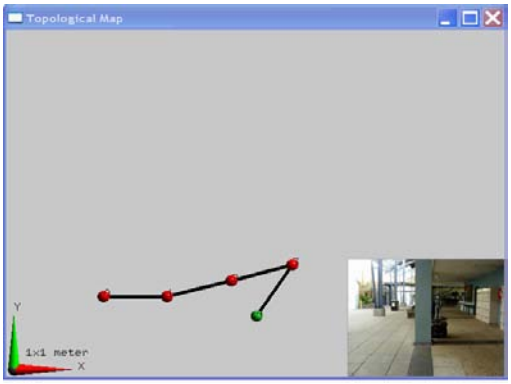
(a) t=2

(b) t=3

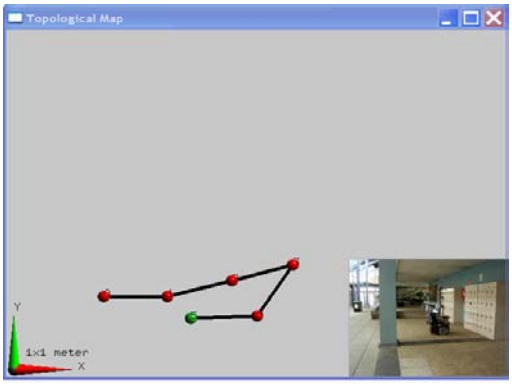
(c) t=4



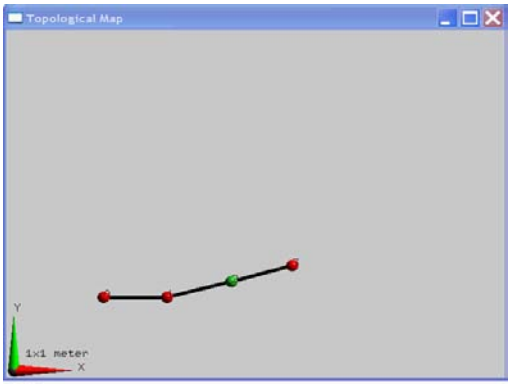
(d) t=5



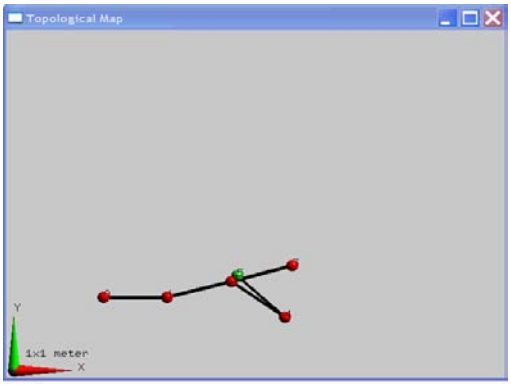
(e) t=6



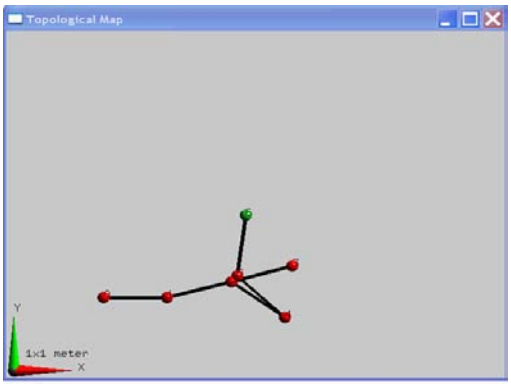
(f) t=7



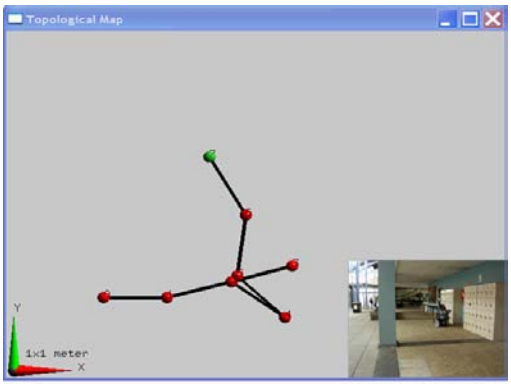
(g) t=8 (Recovery)



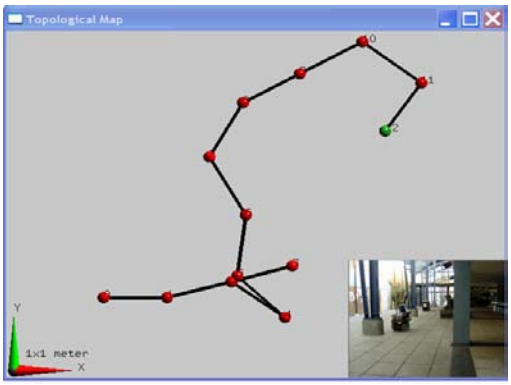
(h) Recovery



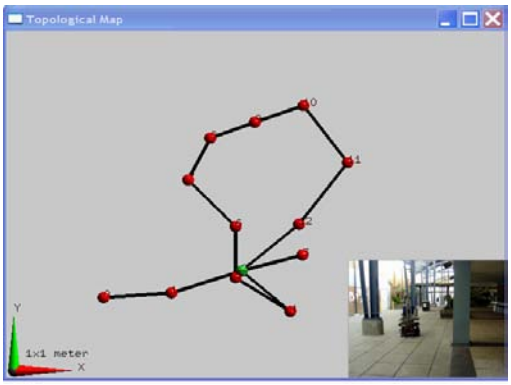
(i) Recovery



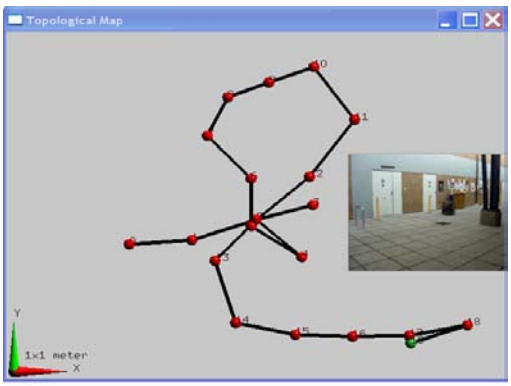
(j) Recovery



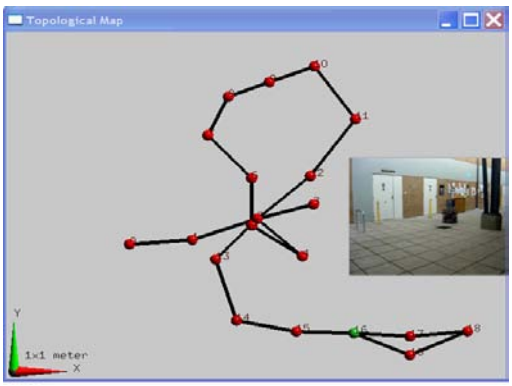
(k) t=13



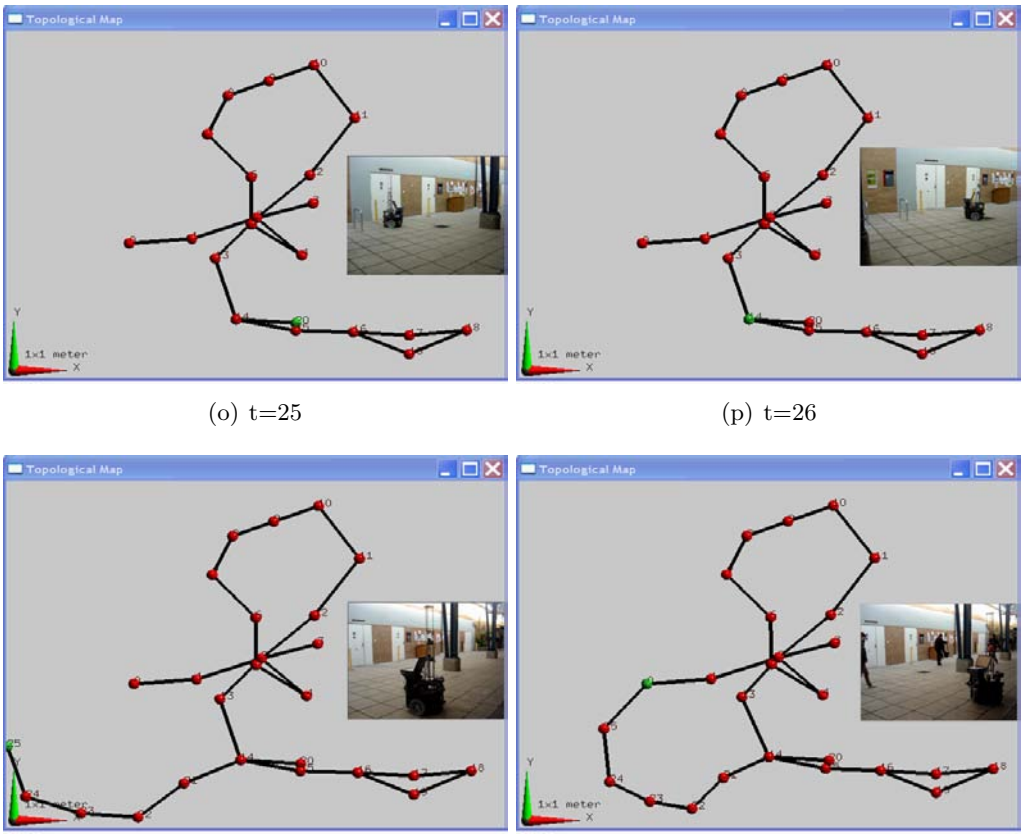
(l) t=14



(m) t=21



(n) t=22



(o) t=25 (p) t=26

(q) t=31 (r) t=32

Figure 6.37: Semi Outdoor Experiment 5

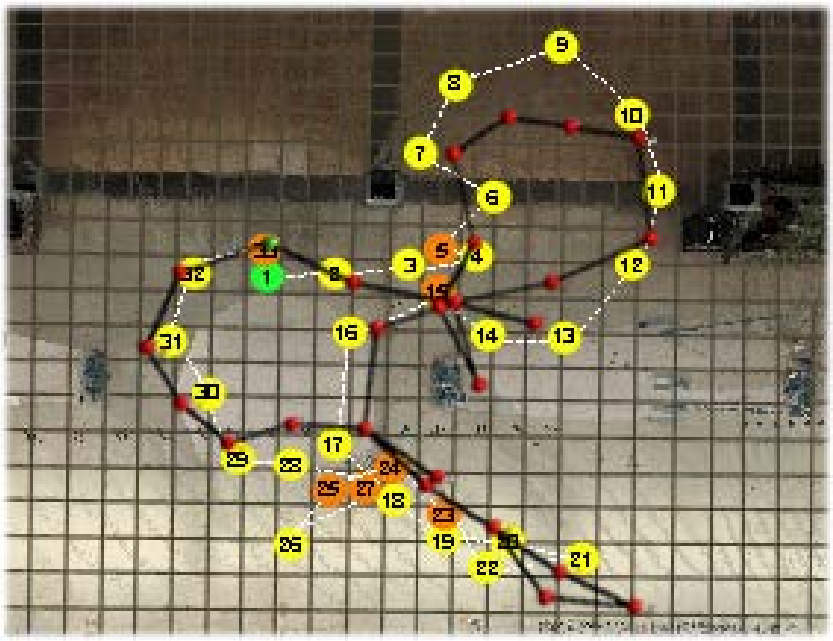
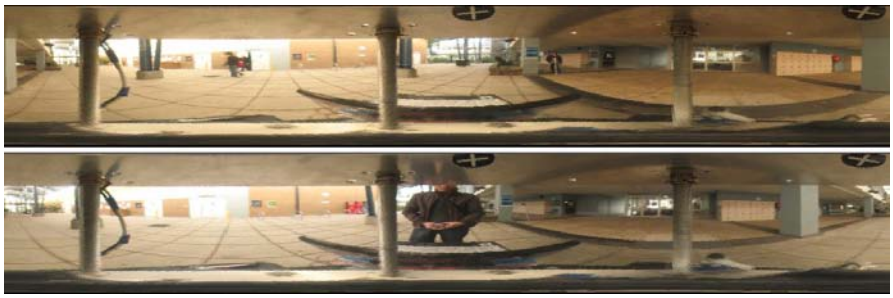


Figure 6.38: Semi Outdoor Exp.5 - Comparing Against Ground Truth

(a) t=4

(b) t=14

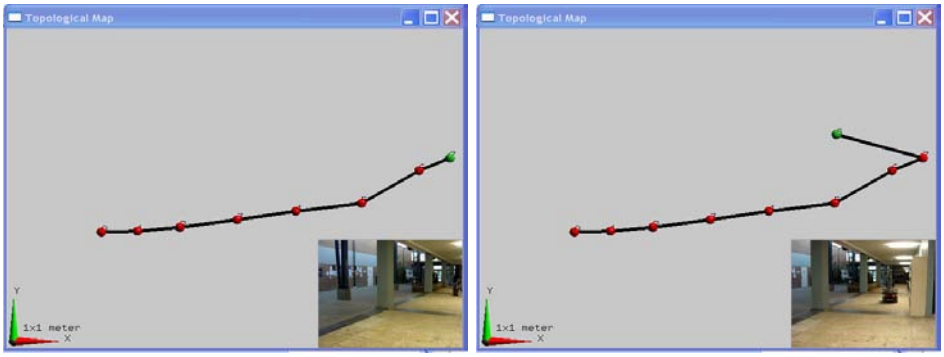


(c) t=32

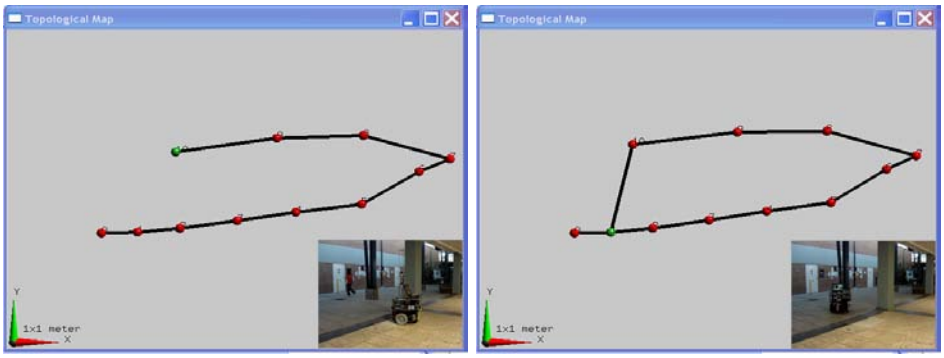
Figure 6.39: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 5 (Loop Closure Detection)

Semi-Outdoor Experiment 6 Although conducted in the same semi-outdoor environment shown in Fig. 6.23, this experiment is slightly different and interesting due to the low lighting conditions. From $t = 7$ to 8, the mobile robot switched from the pure exploration to the loop closing and exploration mode and managed to close the loop at $t = 11$. Notice that the blurring in one of the omnidirectional images in Fig. 6.42 at $t = 11$, due to the camera being out of focus, did not deter it from matching to the correct node. The mobile robot accumulated positional errors due to the low lighting conditions but managed to partially recover itself with the loop closures detected at $t = 24$ and $t = 26$. Comparing the resulting topological map with ground truth, it was clear that the low lighting conditions had caused a severe drift in orientation for the loop on the left. Nevertheless, the shape and connectivity of the estimated trajectory compared to ground truth was found to be coherent and the accuracy of the map could be improved by revisiting. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.41. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig.

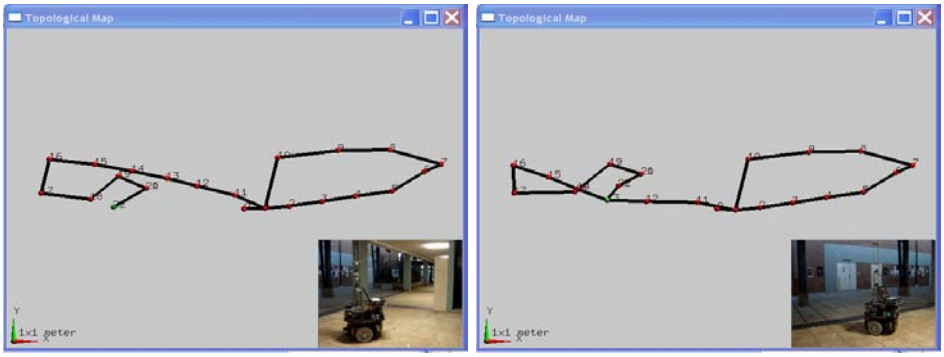
6.42. The following are node pairs (node numbers based on the ground truth trajectory) which raises loop closures events in this experiment: 12-2, 25-15 and 27-1.



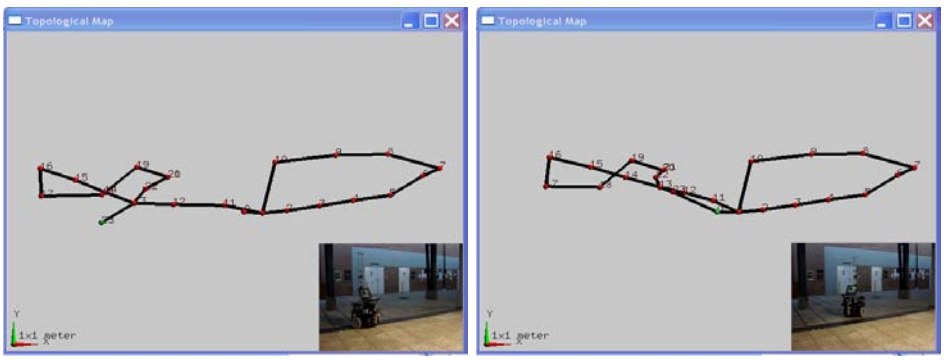
(a) t=7 (b) t=8



(c) t=10 (d) t=11



(e) t=23 (f) t=24



(g) t=25 (h) t=26

Figure 6.40: Semi Outdoor Experiment 6

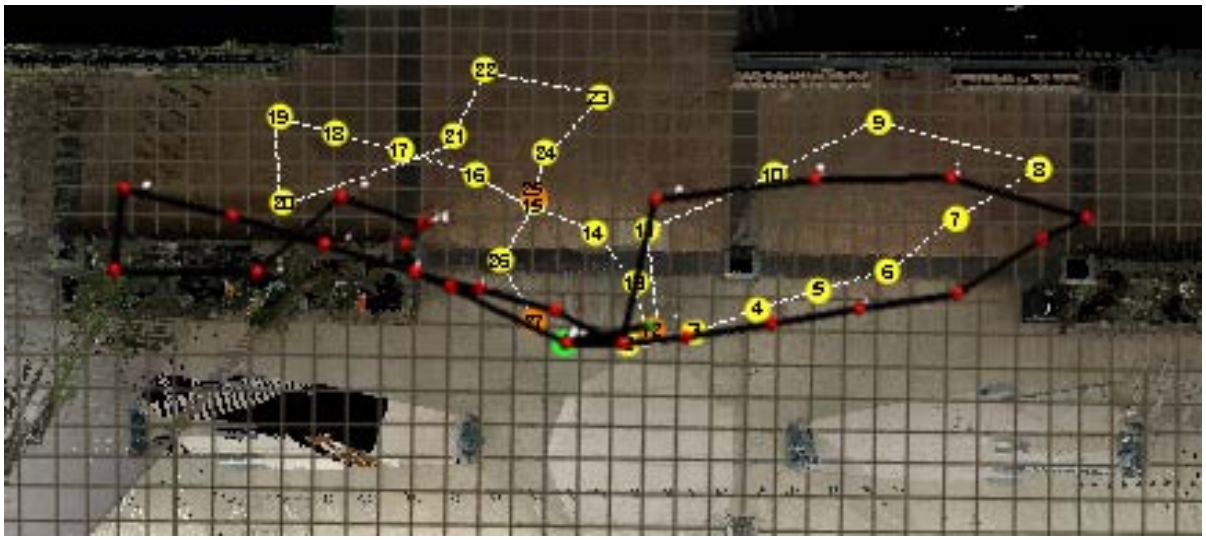
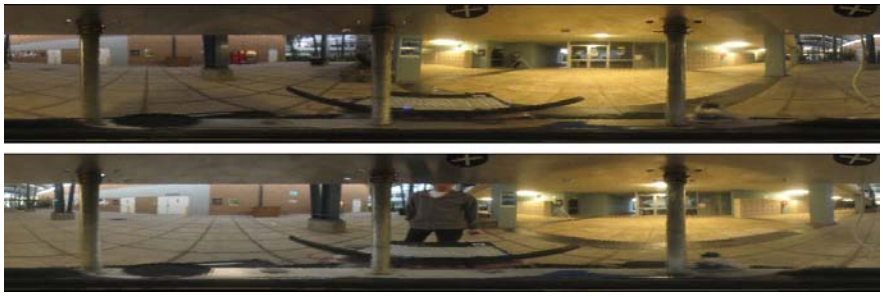


Figure 6.41: Semi Outdoor Exp.6 - Comparing Against Ground Truth

(a) t=11

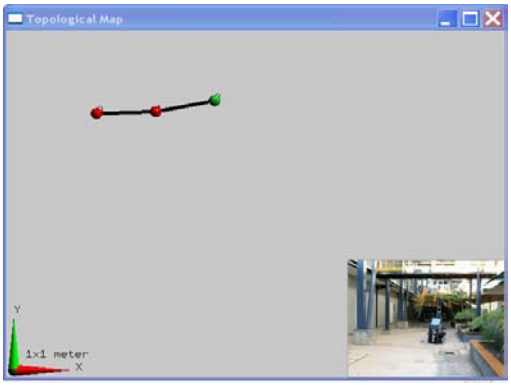
(b) t=24



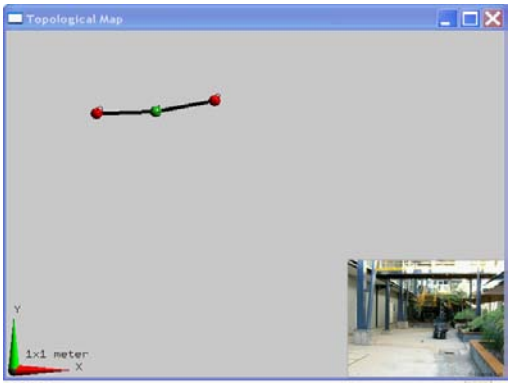
(c) t=26

Figure 6.42: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 6 (Loop Closure Detection)

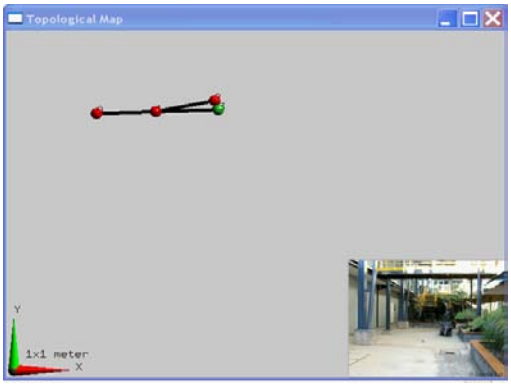
Semi-Outdoor Experiment 7 The final semi-outdoor experiment was conducted in a different semi-outdoor environment shown in Fig. 6.24. At $t = 3$, it detected a loop closure event which was further validated to be a false positive at $t = 6$, where system restoration was performed. At $t = 13$, the mobile robot had drifted away from its actual current location in the environment, deterring it from detecting loop closures even though it thought it was close to one. Nevertheless, it managed to find a valid loop closure event at $t = 21$, which matched the current location of the mobile robot with node 2 in the topological map. Several other loop closure events were detected on the way. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. 6.44. As illustrated, the map was only partially corrected with nodes 9,10 and 11 in the topological map requiring further corrections via additional observations. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. 6.45. The following are node pairs (node numbers based on the ground truth trajectory) which raises loop closure events in this experiment: 9-7, 10-8, 15-13, 16-11, 18-8, 20-7, 21-11 and 22-3. Please take note that this does not include false loop closure events that had been restored and rectified.



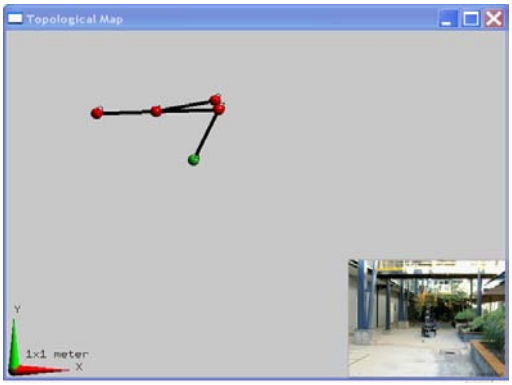
(a) t=2



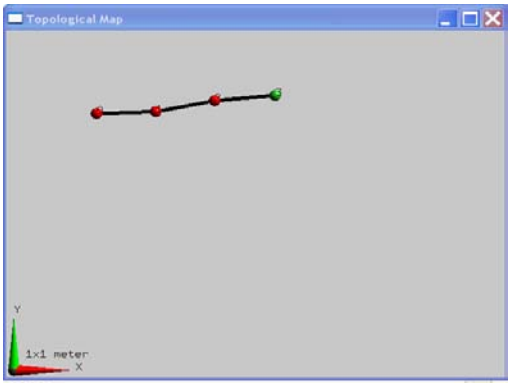
(b) t=3



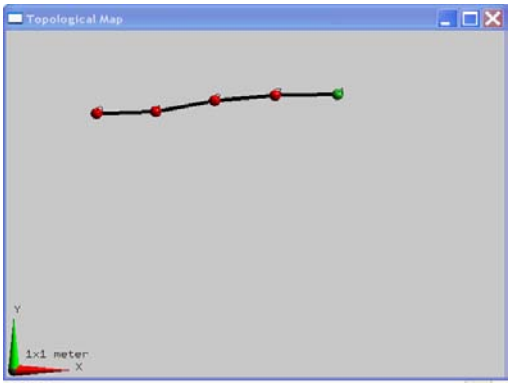
(c) t=4



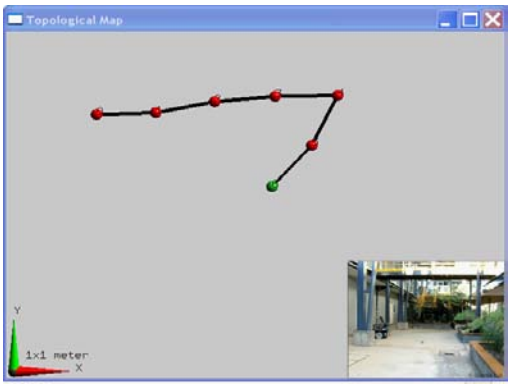
(d) t=5



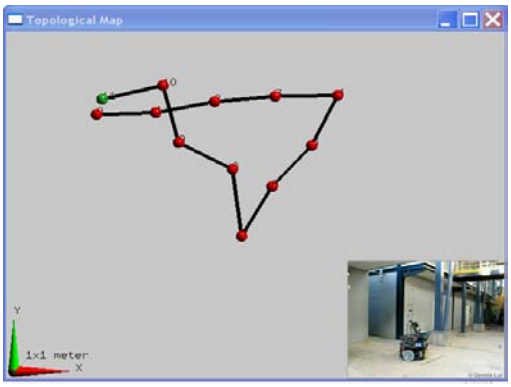
(e) t=6 (Recovery)



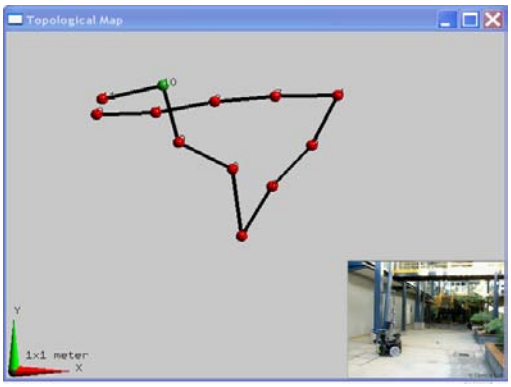
(f) Recovery



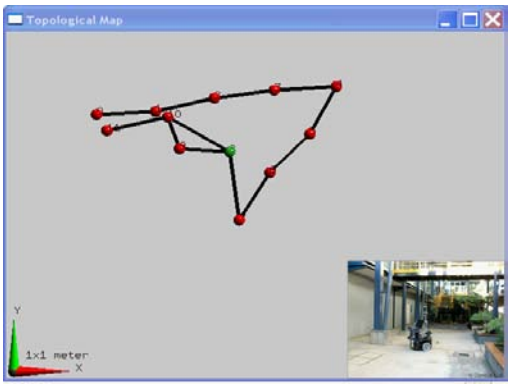
(g) Recovery



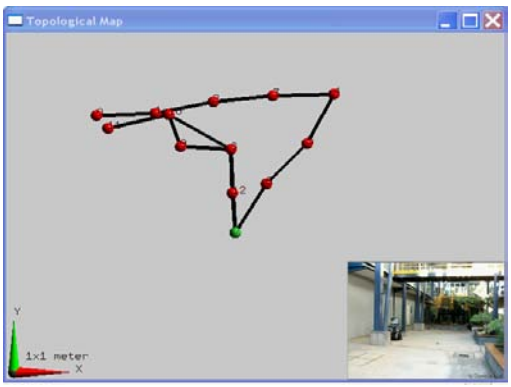
(h) t=13



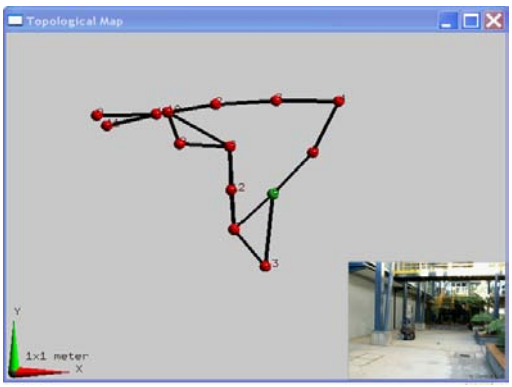
(i) t=14



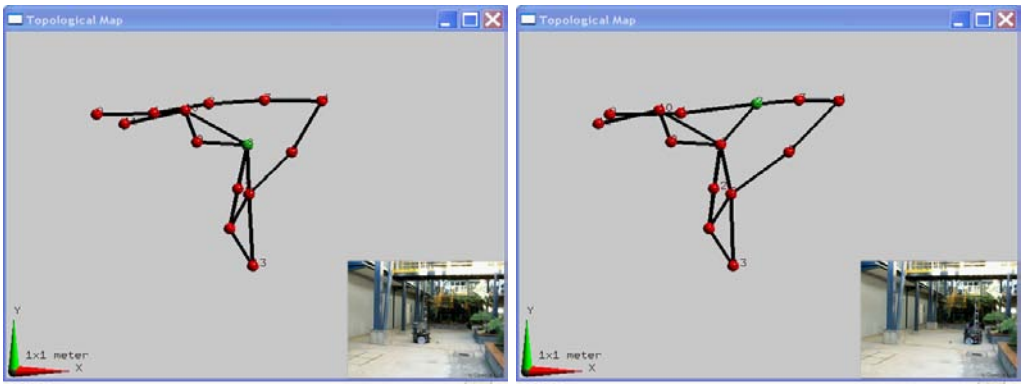
(j) t=15



(k) t=17



(l) t=19



(m) t=20 (n) t=21

Figure 6.43: Semi Outdoor Experiment 7

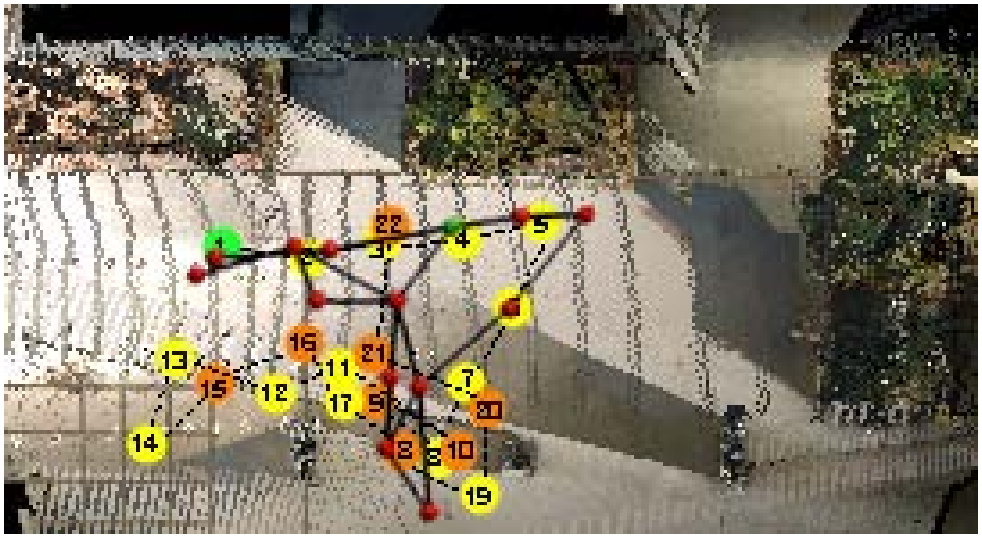
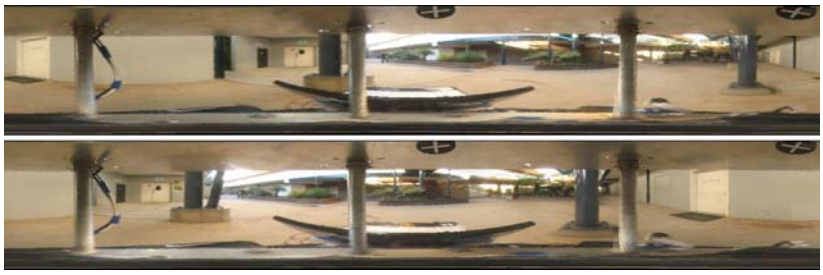


Figure 6.44: Semi Outdoor Exp.7 - Comparing Against Ground Truth

(a) t=3

(b) t=14



(c) t=17

(d) t=19

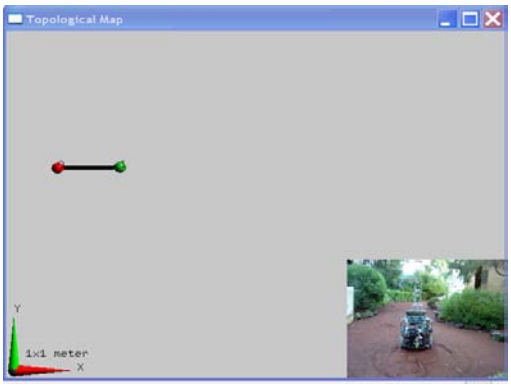
(e) t=21

Figure 6.45: Matching Omnidirectional Image Pairs for Semi-Outdoor Exp. 7 (Loop Closure Detection)

Outdoor Experiment This experiment was conducted in the outdoor environment illustrated in Fig. 6.46. This area is not covered and contains an abundance of natural features (trees, plants,etc). The terrain is an unpaved road surface which gets soft during wet seasons. The size of the pathway is relatively small as compared to previous environments but wide enough for the mobile robot to traverse in. It started exploring, turned back at $t = 7$ and detected loop closures at $t = 8$ and $t = 13$. This experiment showed that the appearance-based localization and mapping system could work in an outdoor environment. Unfortunately, the mobile robot platform was not designed to operate in such soft terrain. Therefore, there were numerous times that the mobile robot's casters were stuck in the soil, which requires some human assistance to overcome the resistance of the soil built up on the sides of the casters as the mobile robot tries to rotate. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth for comparison in Fig. 6.49. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. 6.48. The following are node pairs (node numbers based on the ground truth trajectory) which raise loop closure events in this experiment: 9-5 and 11-2.

(a) (b)

Figure 6.46: Outdoor Experimental Environment



(a) t=1

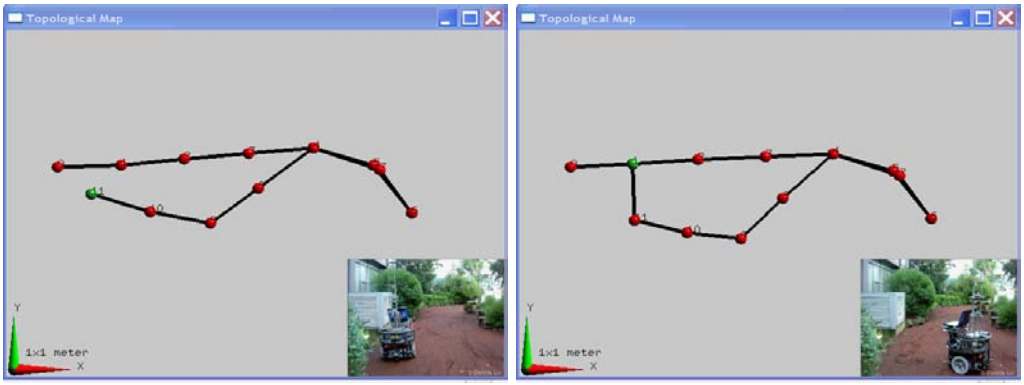
(b) t=3

(c) t=6

(d) t=7

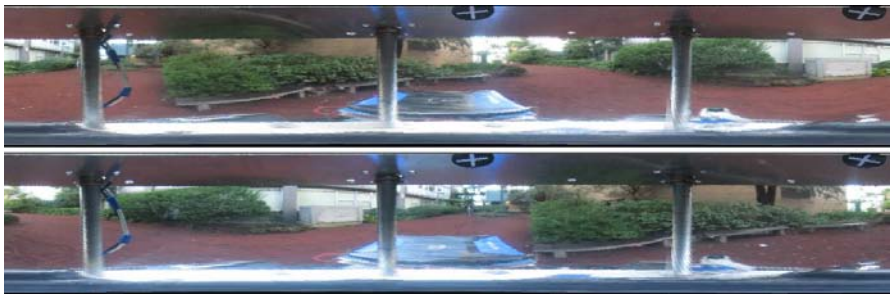
(e) t=8

(f) t=9

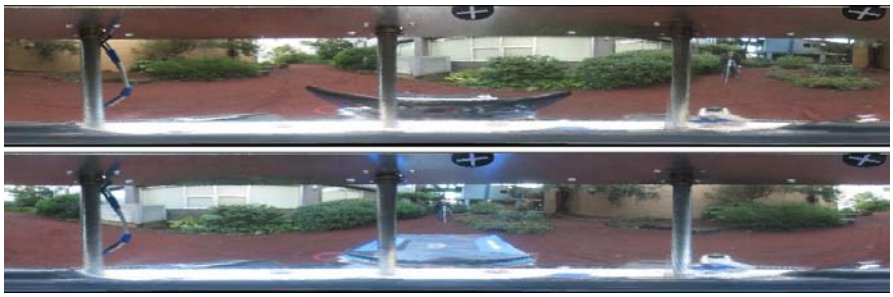


(g) t=12 (h) t=13

Figure 6.47: Outdoor Experiment



(a) t=8



(b) t=13

Figure 6.48: Matching Omnidirectional Image Pairs for Outdoor Exp.(Loop Closure Detection)

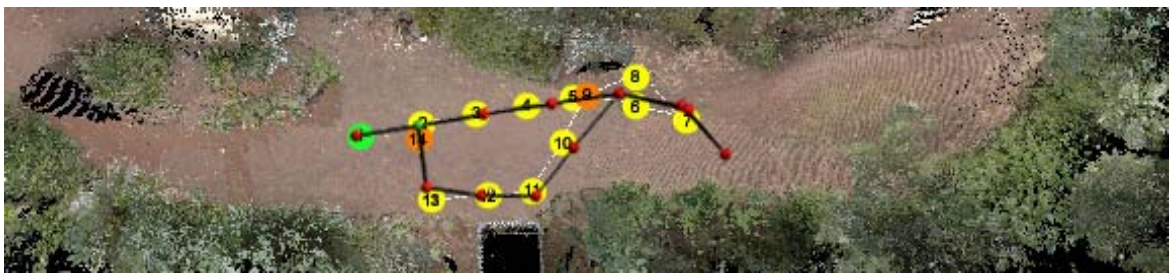


Figure 6.49: Outdoor Exp.- Comparing Against Ground Truth

6.7.2 Offline Experiments

(a) (b) (c)

Figure 6.50: Random Shots from a Manual Offline Experiment

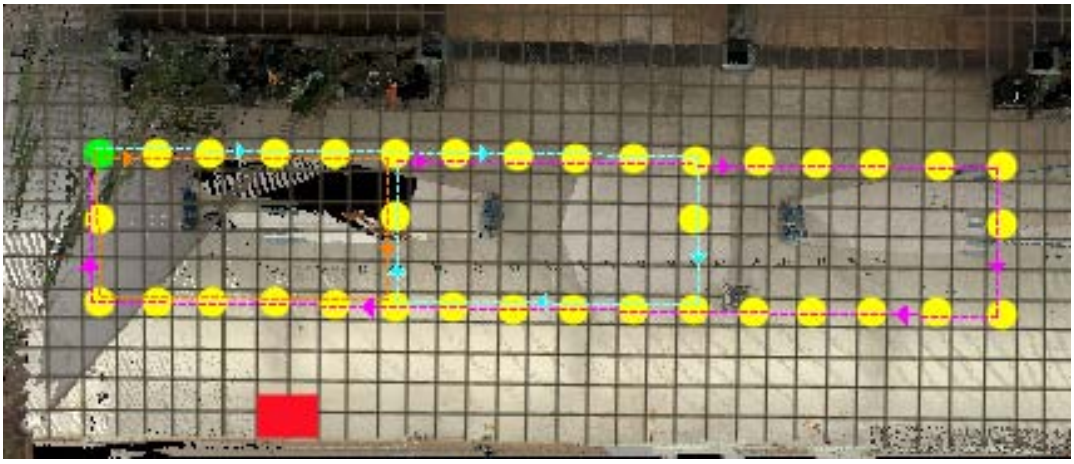
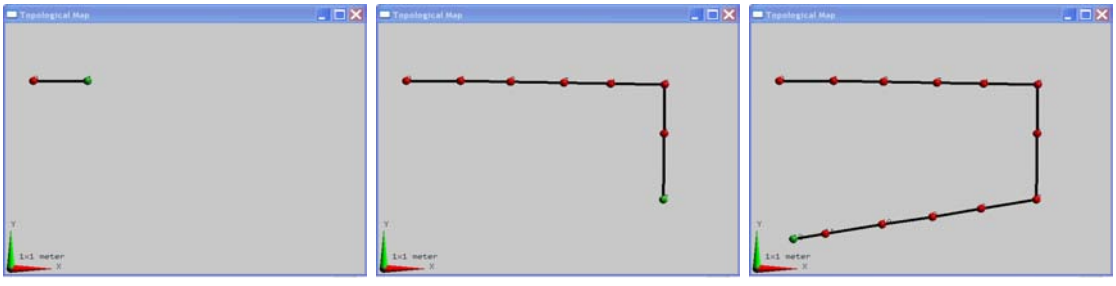


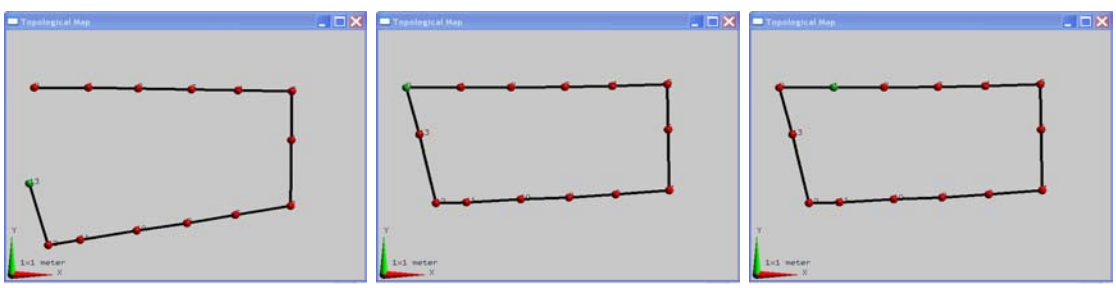
Figure 6.51: Ground Truth Trajectory

In these experiments, the mobile robot was manually driven in the semi-outdoor environment in Fig. 6.50. As it was being driven around, all data required to build the topological map and the database of appearance model for each unique location in the environment were collected. The mobile robot was driven in a fashion as illustrated in Fig. 6.51. The initial starting position is located at the green node in Fig. 6.51, where the mobile robot should proceed in the direction of the orange trajectory, completing a small loop at the end of the trajectory. The mobile robot stops at locations where the yellow nodes are located and captures an omnidirectional image, which is subsequently scaled, decomposed and quantized into an image signature (appearance model of the location). As the mobile robot completes the orange trajectory, it merges into the light blue trajectory, where the trajectory starts off by revisiting several locations of the orange trajectory before it branches off to a second small loop. As it completes the light blue trajectory, it merges with the magenta trajectory, completes the third loop, and ends at the initial starting position (the green node) of the mobile robot. The following are results for 2 offline datasets using the rank-based framework with the same parameters used in the autonomous experiments. Since the mobile robot was manually driven in a fixed trajectory, the resulting maps could be compared to the ground truth trajectory (grid resolution is approximately 60 x 60 $\pm$ 5 cm in each dimension).

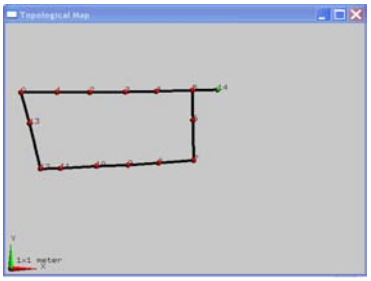
Offline Experiment 1 The state of the topological map at different times are shown in Fig. 6.52. The experiment started at $t = 0$, successfully loop closing at node 0 in the topological map at $t = 14$, as it completed the orange trajectory in Fig. 6.51. At $t = 15$, it was driven along the light blue trajectory shown in Fig. 6.51, detected loop closures at nodes 1 to 5 in the topological map before branching off to the second loop, which has yet to be explored at $t = 20$. Towards the completion of the light blue trajectory, it detected a loop closure at $t = 31$, but unfortunately, it did not detect the loop closure at $t = 32$ when it was supposed to. Nevertheless, the advantage of having two representations of the same place is debatable. Most importantly, it does not cause problems as for the case where two unique locations in the environment having similar representations. It completed the light blue trajectory at $t = 33$ and merged into the magenta trajectory. It detected loop closures at nodes 14 to 18 in the topological map before branching off to the last loop, which has yet to be explored at $t = 39$. A loop closure (a false positive) which requires further validations, $LC_{t=40}$, was detected at $t = 40$ but was later invalidated at $t = 42$. In fact, $LC_{t=40}$, was invalidated by another loop closure event, $LC_{t=42}$, detected at $t = 42$. Similarly, $LC_{t=42}$ requires further validation and was later invalidated at $t = 45$, when system restoration was performed. This experiment illustrated the recursiveness of the proposed active loop closure validation strategy described in Section 6.3.3. Of course, this is only true if a loop closure detected was invalidated by another loop closure which requires further validation. As the mobile robot was close to completing the third loop, it detected another false positive at $t = 51$ but was invalidated by another loop closure event at $t = 52$, which prompted system restoration. At $t = 64$, the mobile robot returned to its initial starting position while detecting multiple loop closures along its course. In Fig. 6.53, the topological map produced by the system was compared with the ground truth trajectory (trajectory with white nodes) and a subset of the sequence of omnidirectional images taken while traversing the orange trajectory in Fig. 6.51 is shown in Fig. 6.54.



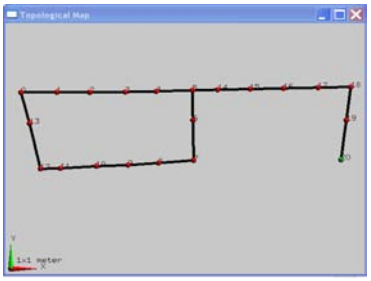
(a) t=1 (b) t=7 (c) t=12



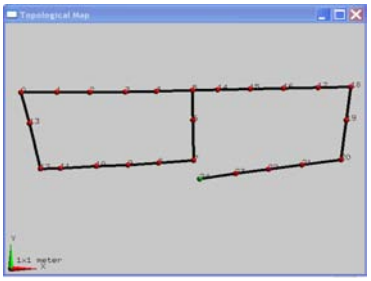
(d) t=13 (e) t=14 (f) t=15



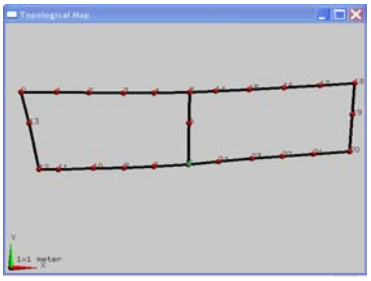
(g) t=20



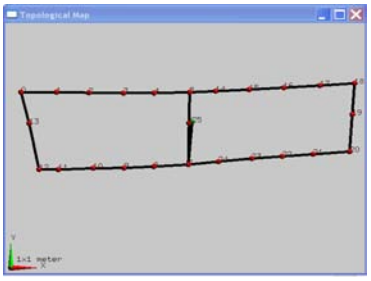
(h) t=26



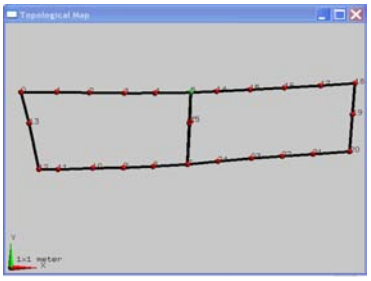
(i) t=30



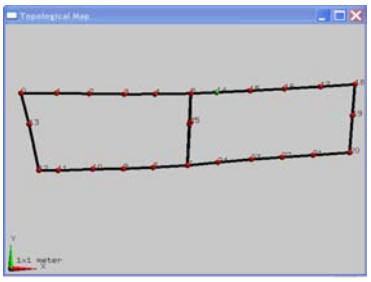
(j) t=31



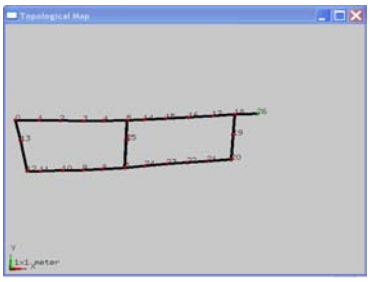
(k) t=32



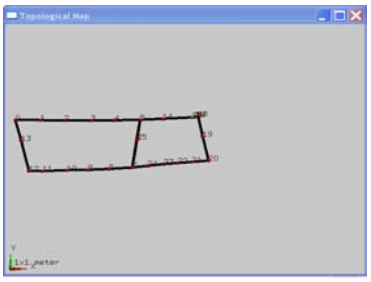
(l) t=33



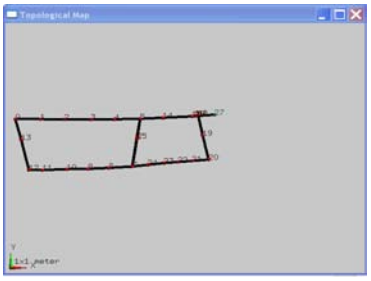
(m) t=34



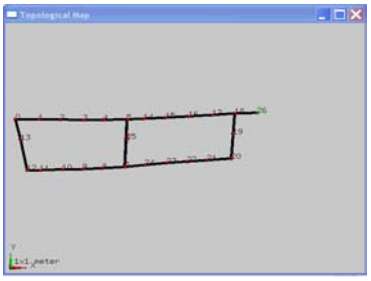
(n) t=39



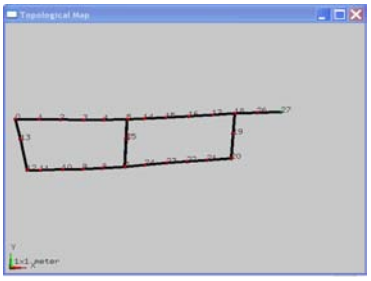
(o) t=40



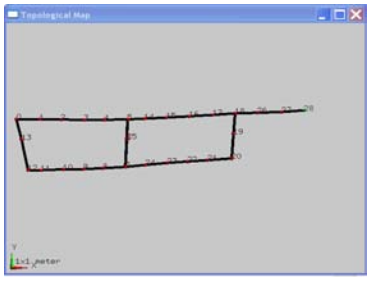
(p) t=41



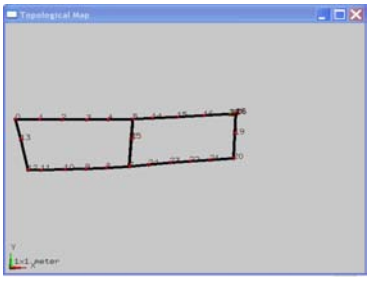
(q) t=42 (Recovery)



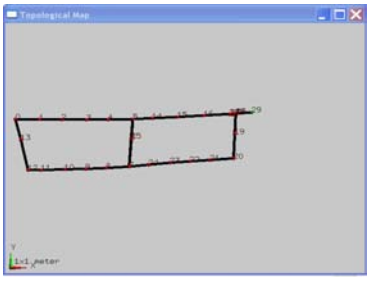
(r) Recovery



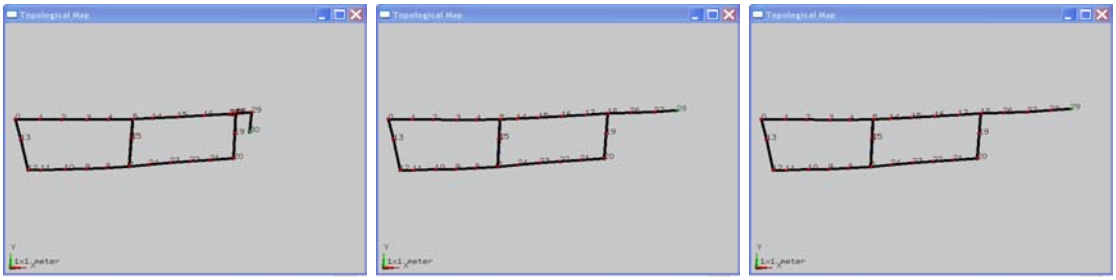
(s) Recovery



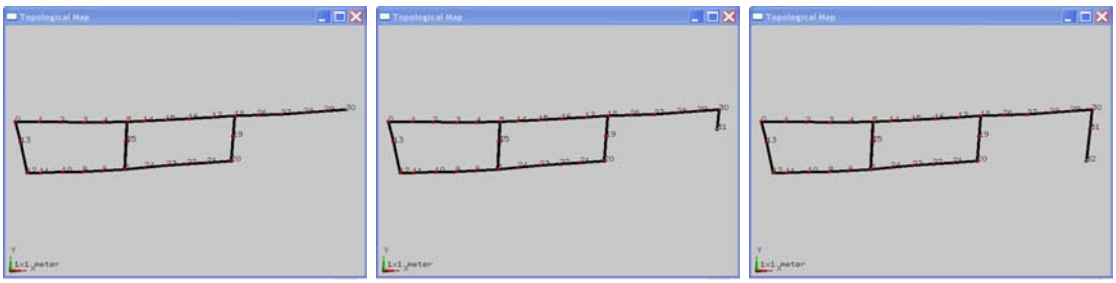
(t) Recovery



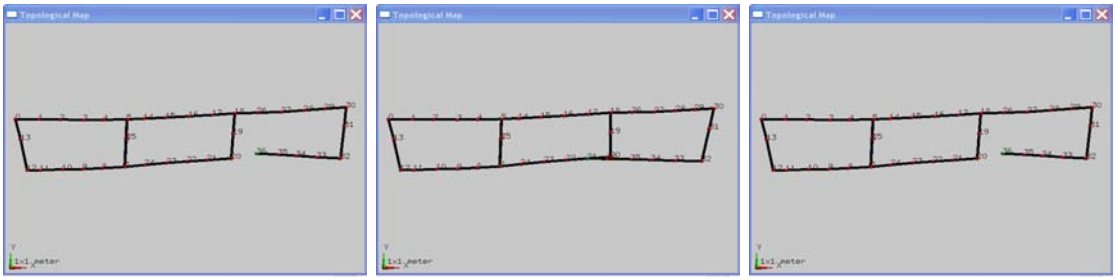
(u) t=43



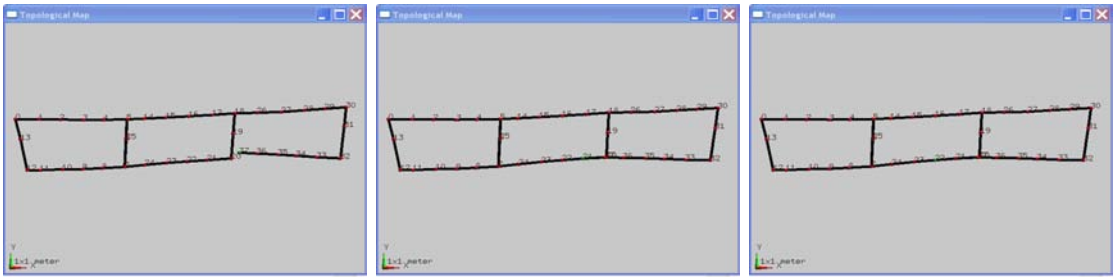
(v) t=44 (w) t=45 (Recovery) (x) Recovery



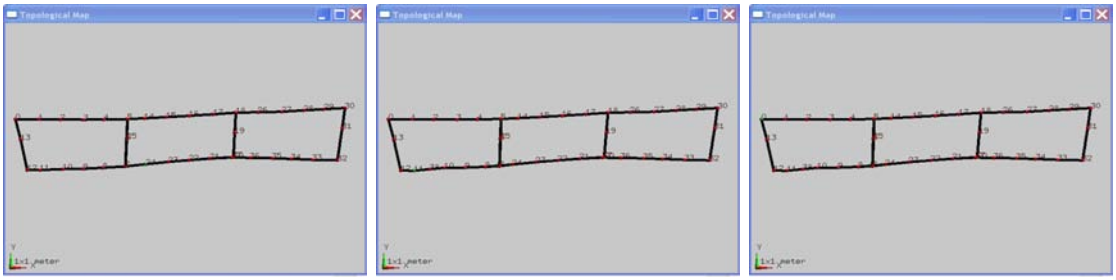
(y) Recovery (z) Recovery (aa) t=46



(ab) t=50 (ac) t=51 (ad) t=52 (Recovery)



(ae) Recovery (af) Recovery (ag) t=53



(ah) t=55 (ai) t=61 (aj) t=64

Figure 6.52: Offline Experiment 1

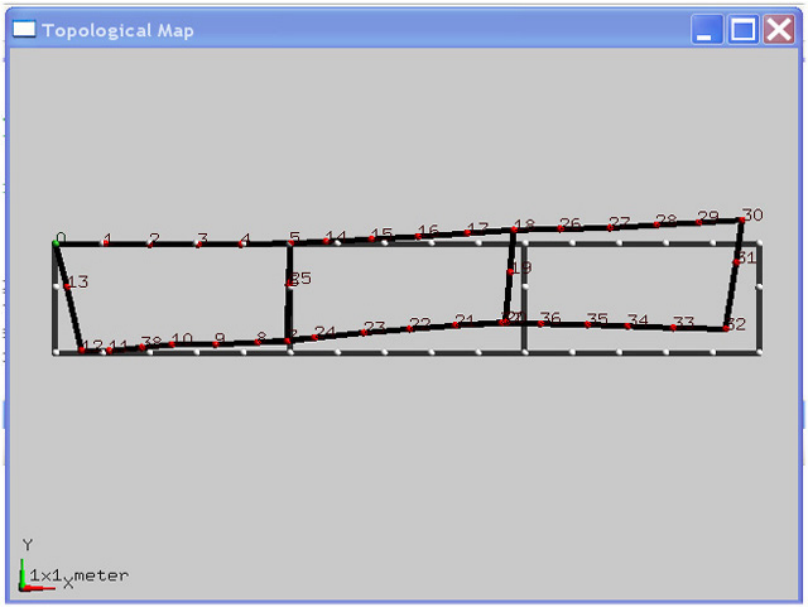
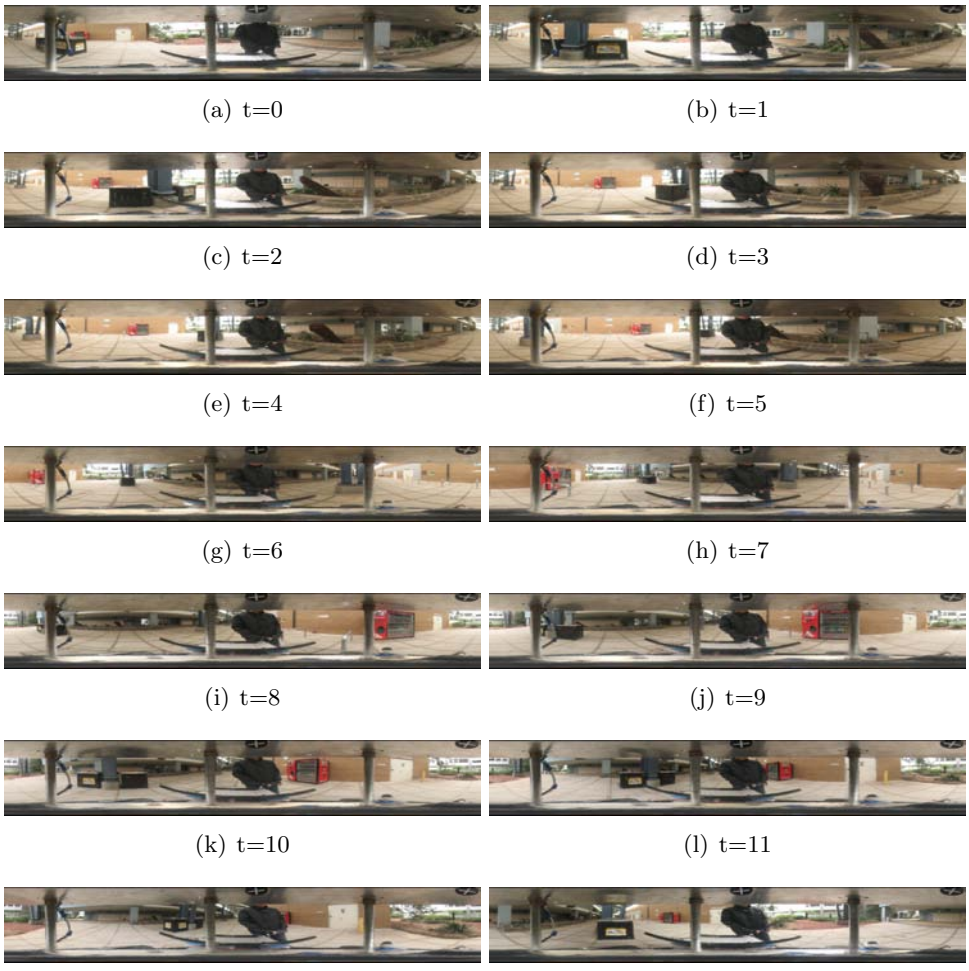


Figure 6.53: Offline Exp. 1 - Comparing Against Ground Truth



(a) t=0 (b) t=1

(c) t=2 (d) t=3

(e) t=4 (f) t=5

(g) t=6 (h) t=7

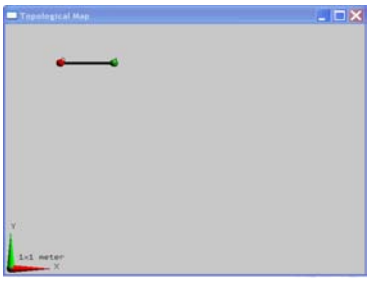
(i) t=8 (j) t=9

(k) t=10 (l) t=11

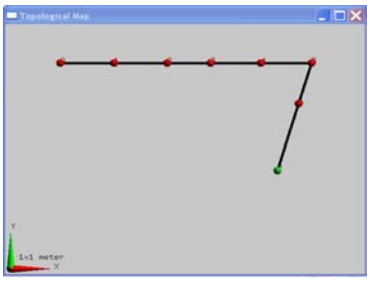
(m) t=12 (n) t=13

Figure 6.54: Subset of Omnidirectional Image Sequence for Offline Experiment 1

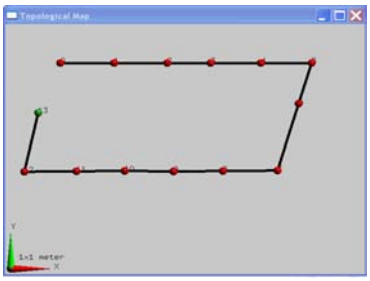
Offline Experiment 2 Similar to the first experiment, the mobile robot started off at $t = 0$ and traversed in the same trajectory. Referring to Fig. 6.55, the mobile robot made an error at $t = 31$, which incorrectly localized the mobile robot to node 8 of the topological map, where in fact it was actually at node 7. Nevertheless, the following loop closure at $t = 32$ rectified this error. Similarly, two representations were used to represent the same location at node 5/25 in the topological map at $t = 33$. Throughout the rest of its entire course, the mobile robot detected false loop closures at $t = 42$ and $t = 45$. Both loop closure events require further evidence to be gathered and was later invalidated by the system, which prompted system restoration. In Fig. 6.56, the topological map produced by the system was compared with the ground truth trajectory (trajectory with white nodes) and a subset of the sequence of omnidirectional images taken while traversing the orange trajectory in Fig. 6.51 is shown in Fig. 6.57.



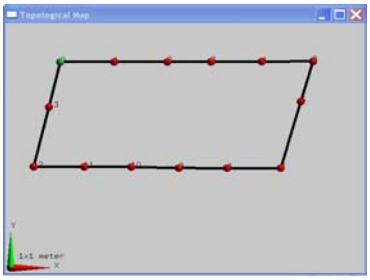
(a) t=1



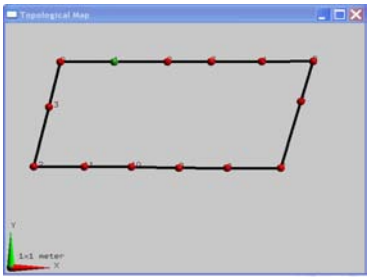
(b) t=7



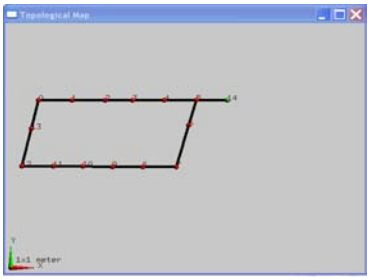
(c) t=13



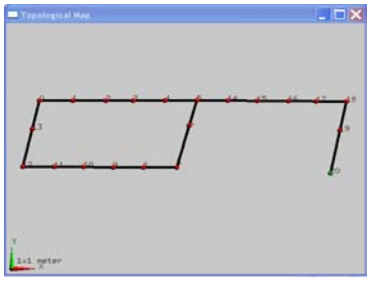
(d) t=14



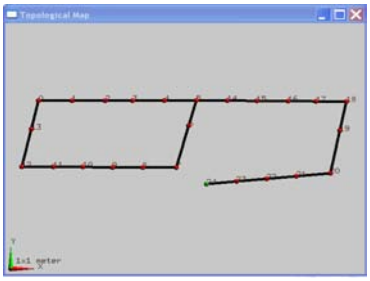
(e) t=15



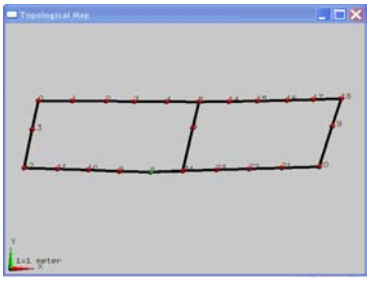
(f) t=20



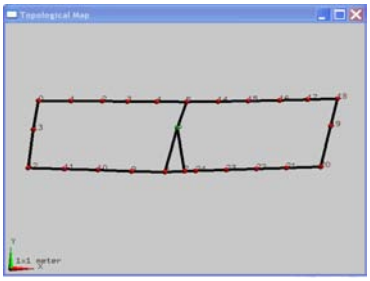
(g) t=26



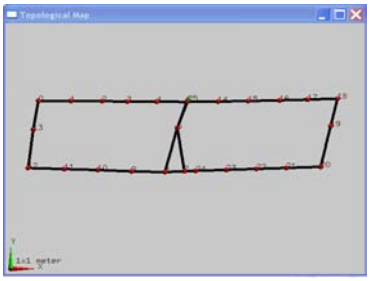
(h) t=30



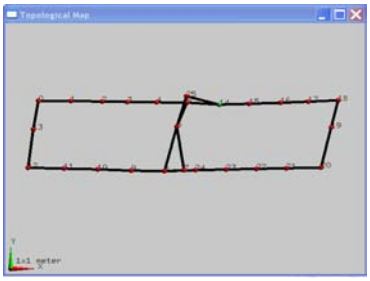
(i) t=31



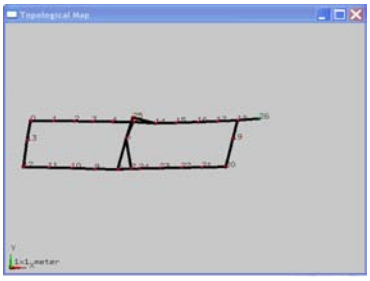
(j) t=32



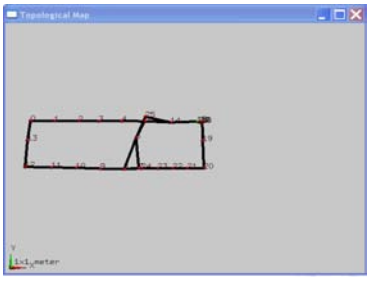
(k) t=33



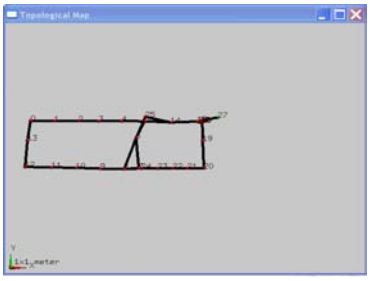
(l) t=34



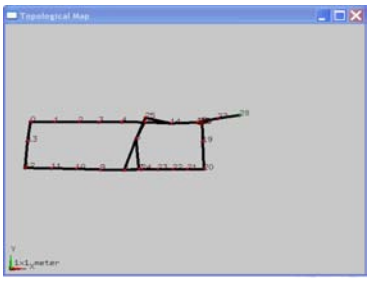
(m) t=39



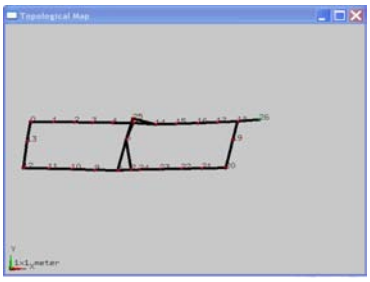
(n) t=40



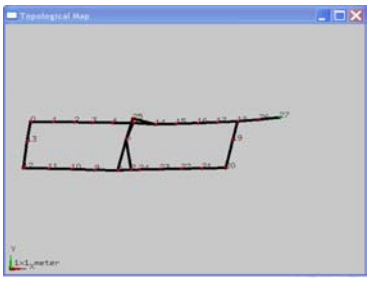
(o) t=41



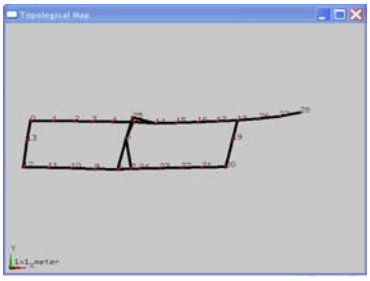
(p) t=42



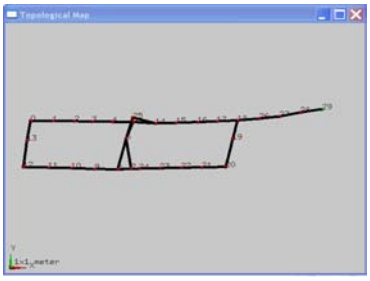
(q) t=43 (Recovery)



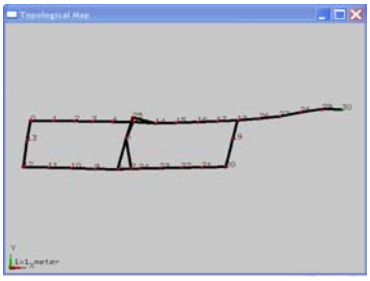
(r) Recovery



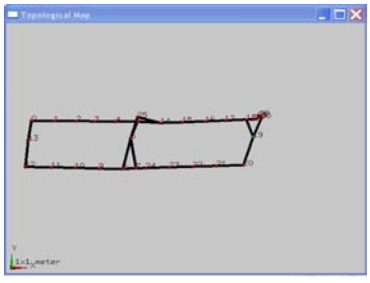
(s) Recovery



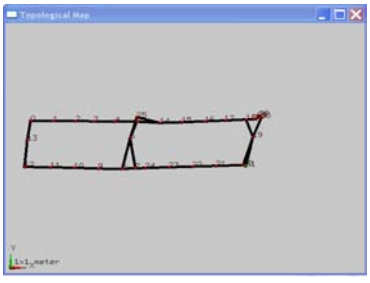
(t) Recovery



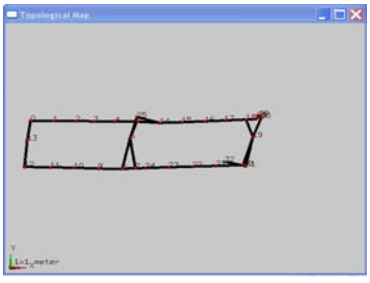
(u) t=44



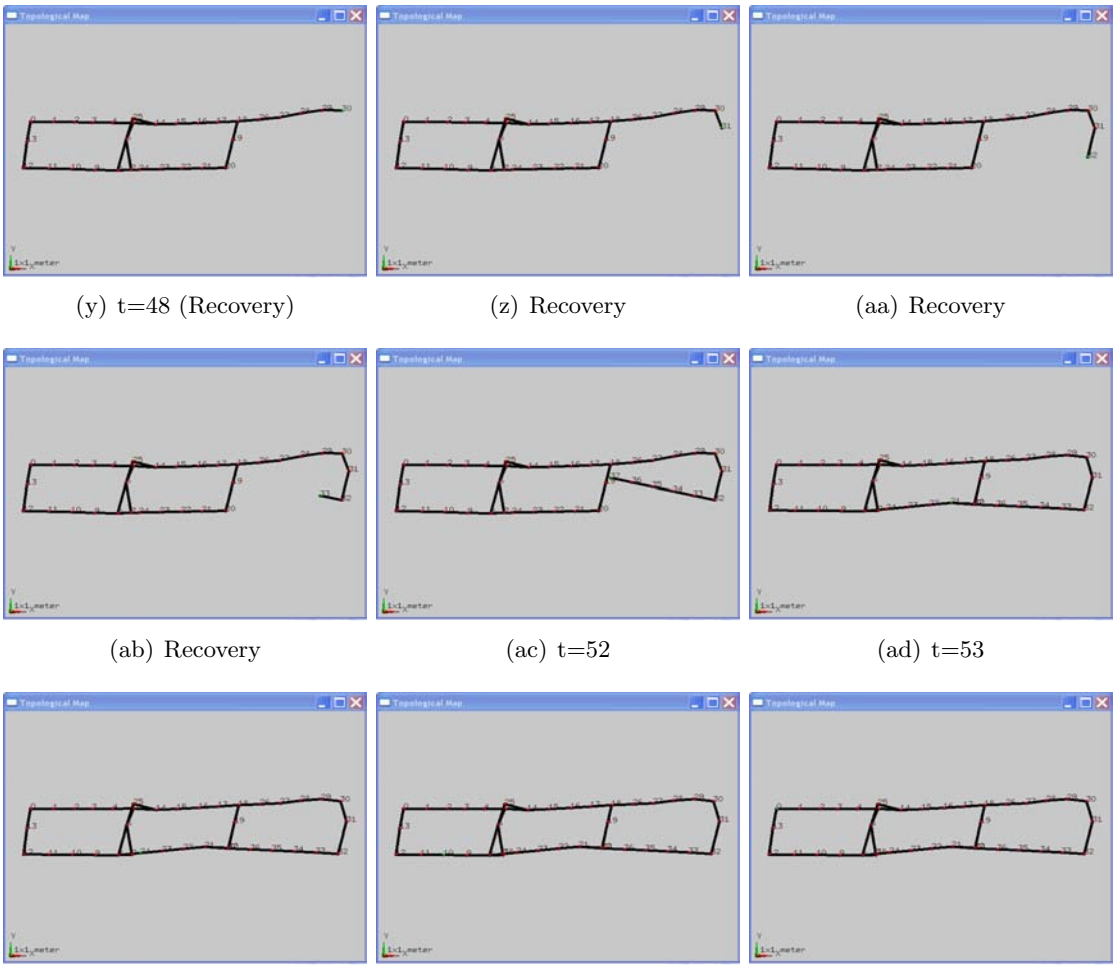
(v) t=45



(w) t=46



(x) t=47



(y) t=48 (Recovery) (z) Recovery (aa) Recovery

(ab) Recovery (ac) t=52 (ad) t=53

(ae) t=56 (af) t=60 (ag) t=64

Figure 6.55: Offline Experiment 2

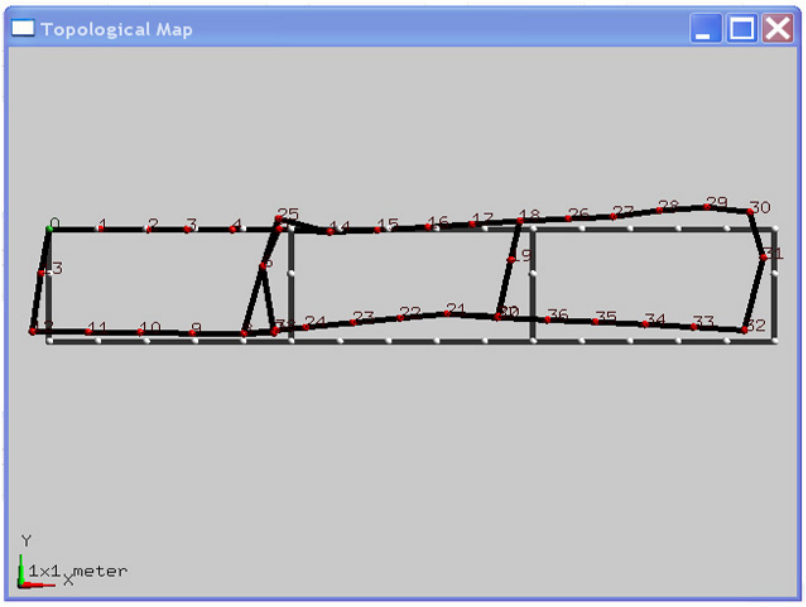
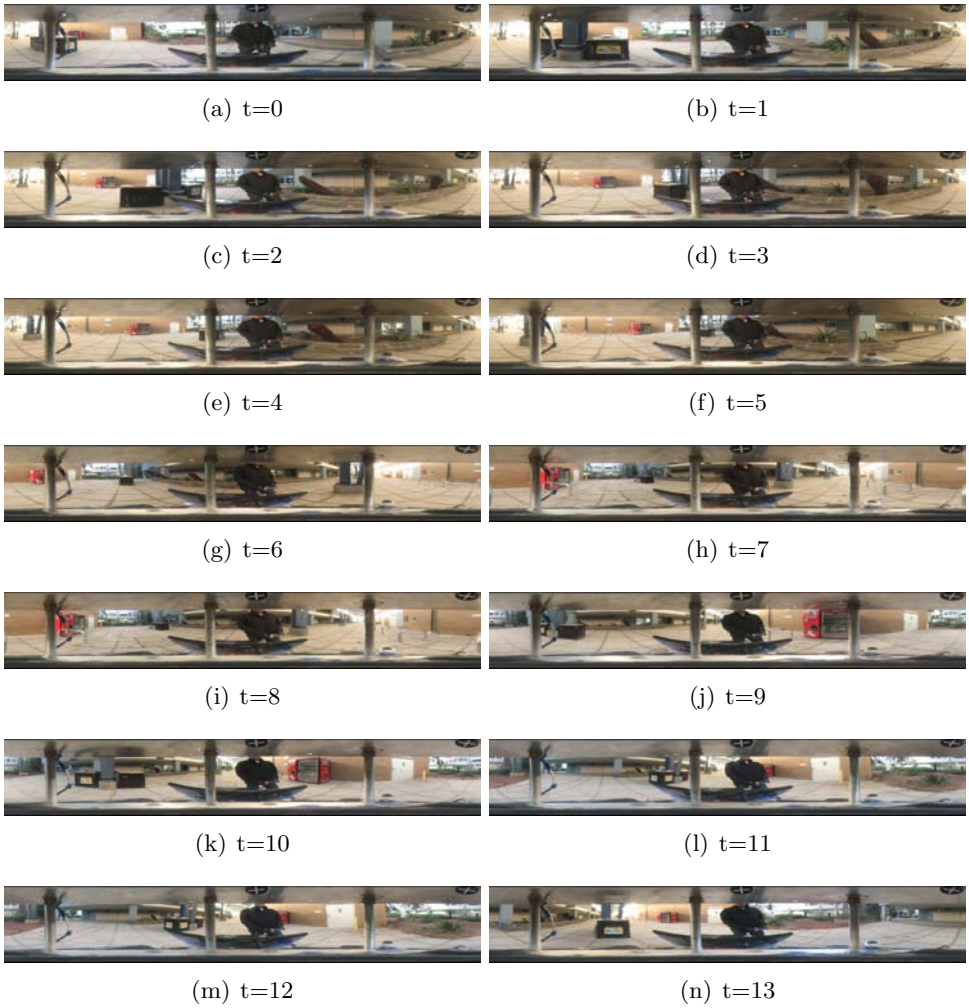


Figure 6.56: Offline Exp. 2 - Comparing Against Ground Truth



(a) t=0 (b) t=1

(c) t=2 (d) t=3

(e) t=4 (f) t=5

(g) t=6 (h) t=7

(i) t=8 (j) t=9

(k) t=10 (l) t=11

(m) t=12 (n) t=13

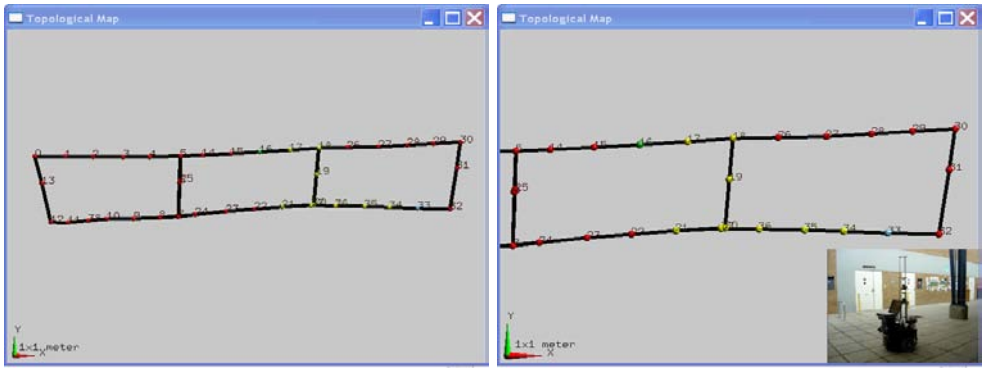
Figure 6.57: Subset of Omnidirectional Image Sequence for Offline Experiment 2

6.7.3 Semi-Autonomous Experiments

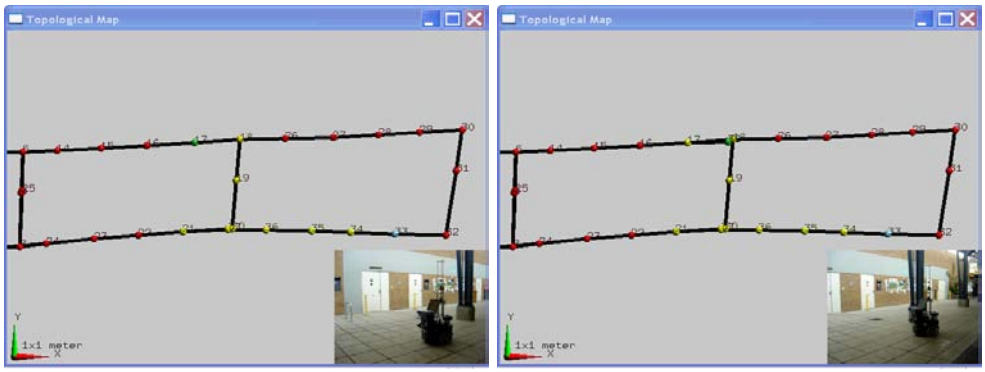
The following experimental results were conducted using the semi-autonomous navigation mode described in Section 6.4. The map created by offline experiment 1 (Section 6.7.2) was loaded into the system and used for all the following experiments. Similar to the autonomous and offline experiments, the semi-autonomous experiments use the rank-based framework with the same relevant parameters specified in the autonomous experiments.

Semi-Autonomous Experiment 1 The map loaded into the system was created by offline experiment 1. The initial position of the mobile robot was defined by the user as being at node 16 and its goal was to reach node 33. Once the map was loaded with the starting and goal locations defined, the nodal propagation algorithm (Section 6.2.2) was used to plan a path to the goal based on the topological map (yellow nodes). This planned path is basically a rough plan for the mobile robot to reach its target destination. The mobile robot can deviate from this planned path depending on whether it thinks there is a viable shortcut or whether it has been temporarily overridden by the reactive obstacle avoidance system. In this experiment, the mobile robot navigated according to

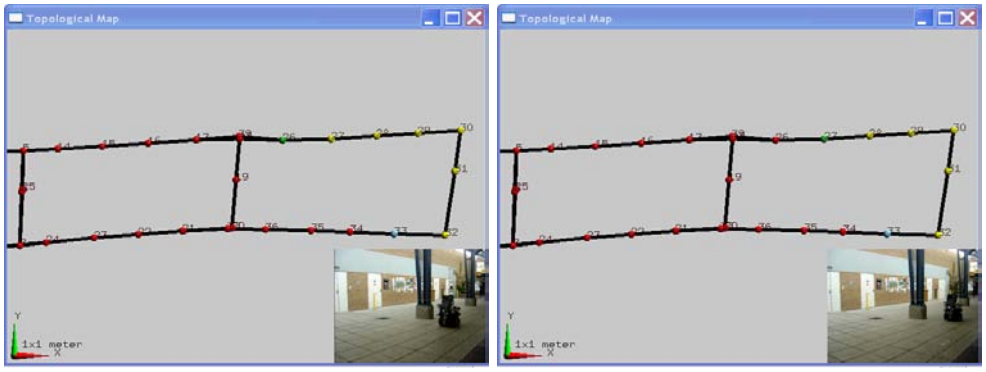
the planned path at $t = 1$ and $t = 2$. However, at $t = 2$, it failed to detect the loop closure at node 18. It deviated from the planned path and detected a loop closure at node 26. The new planned path based on the mobile robot's current position was also illustrated at $t = 2$. Subsequently, it followed its planned path until it decided to take a detour at $t = 6$ and detected a loop closure at $t = 8$, which required further validation. In the semi-autonomous mode, the mobile robot is not equipped with the active loop closing validation strategy described in Section 6.3.3 since it is regularly trying to visit previously seen nodes when it is not taking a shortcut. The loop closure detected at $t = 8$ was later invalidated at $t = 11$ and system restoration was performed. For the case when it takes a detour, it should not be detecting loop closures anyway since it is in unexplored areas. It reached its target destination at $t = 13$. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. 6.59.



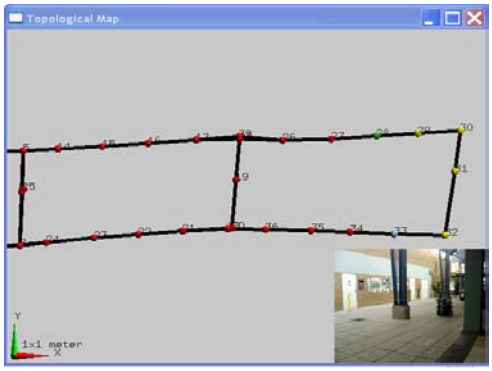
(a) Map Loaded (b) t=0



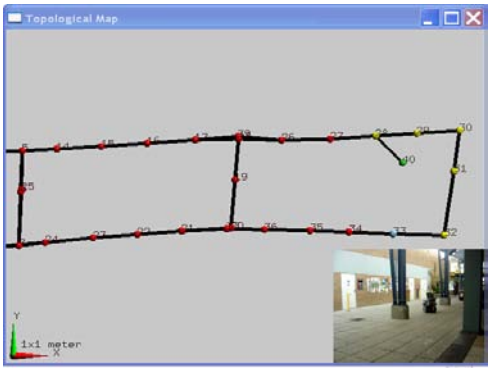
(c) t=1 (d) t=2



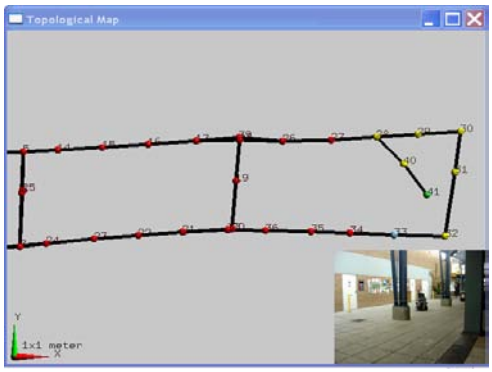
(e) t=3 (f) t=4



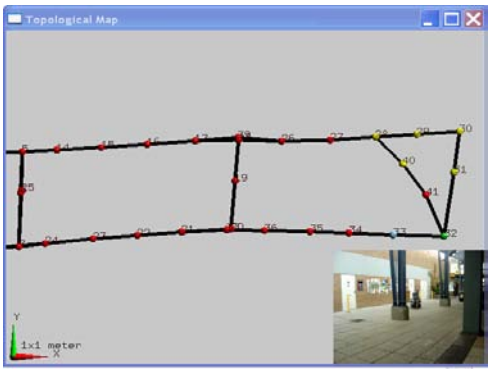
(g) t=5



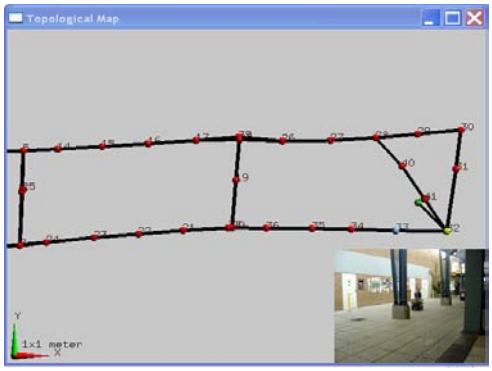
(h) t=6



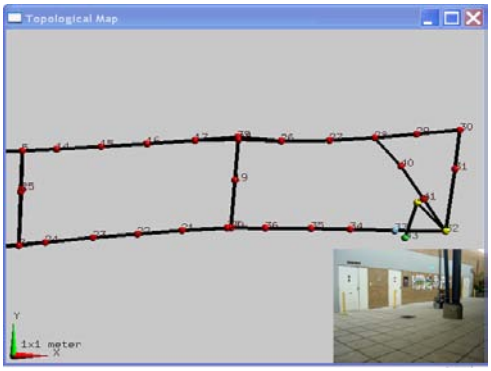
(i) t=7



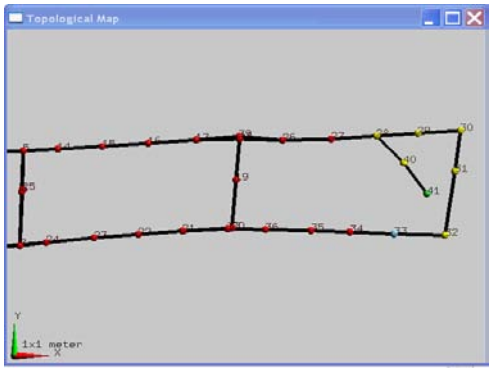
(j) t=8



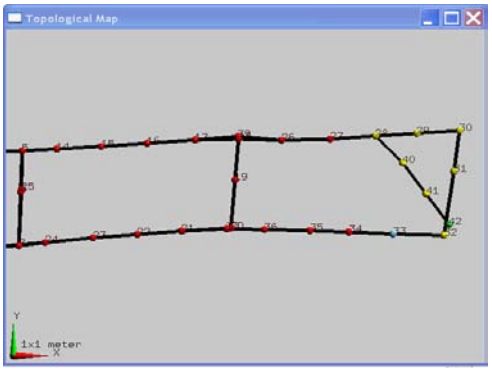
(k) t=9



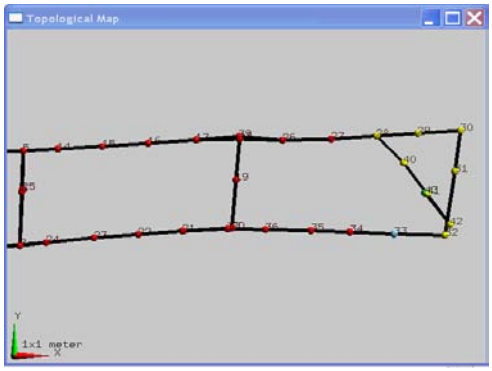
(l) t=10



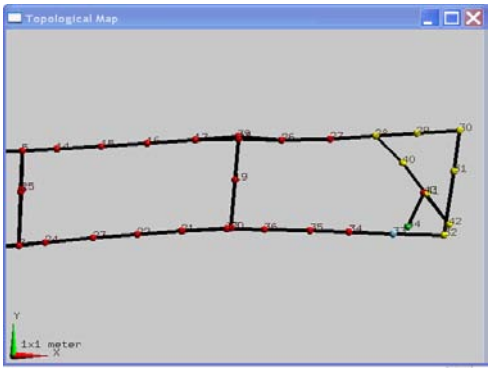
(m) t=11 (Recovery)



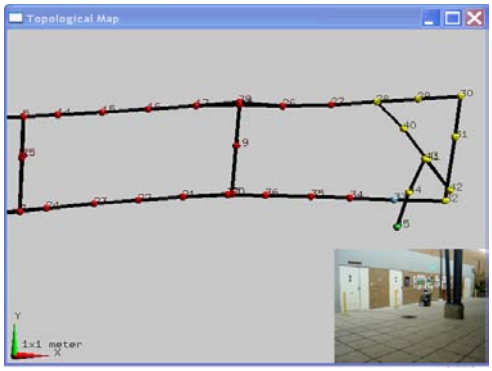
(n) Recovery



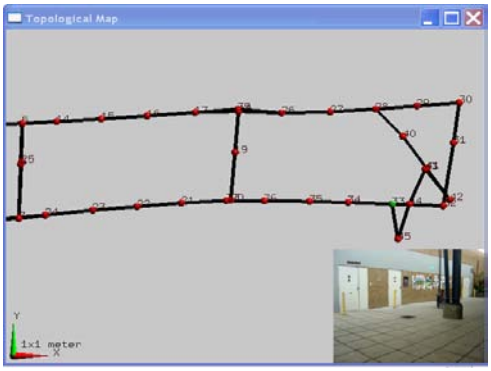
(o) Recovery



(p) Recovery



(q) t=12



(r) t=13

Figure 6.58: Semi-Autonomous Experiment 1

(a) t=1

(b) t=3

(c) t=4

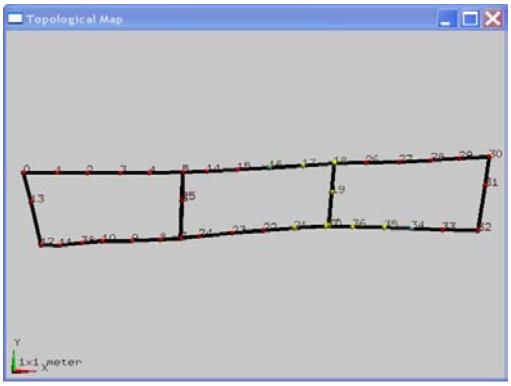
(d) t=5

(e) t=12

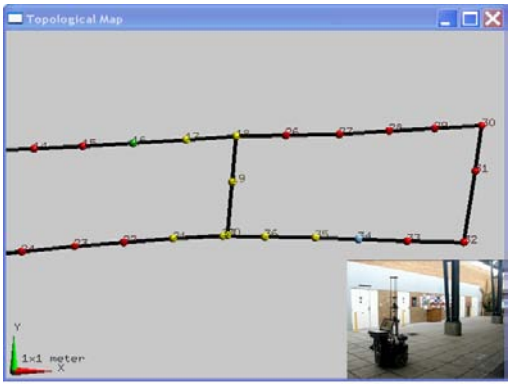
(f) t=13

Figure 6.59: Matching Omnidirectional Image Pairs for Semi-Autonomous Exp 1(Loop Closure Detection)

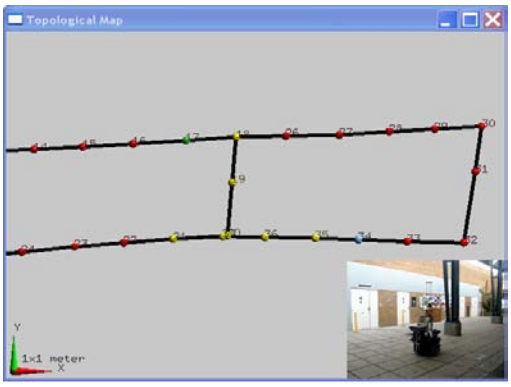
Semi-Autonomous Experiment 2 The map from offline experiment 1 was loaded into the system with the starting and goal locations defined as node 16 and node 34 respectively. The mobile robot followed its planned path (yellow nodes) from $t = 1$ to $t = 2$ and deviated from the path at $t = 3$. Nevertheless, it successfully found its target destination at $t = 8$. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. 6.61. A third semi-autonomous experiment is available in Appendix E.



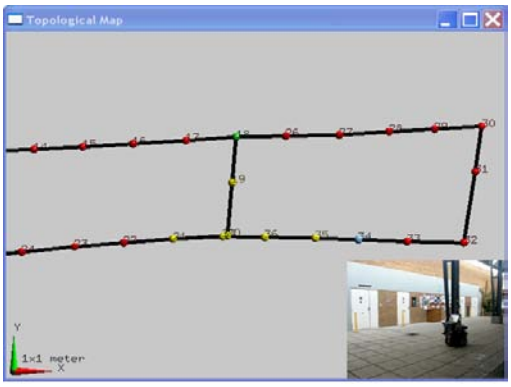
(a) Map Loaded



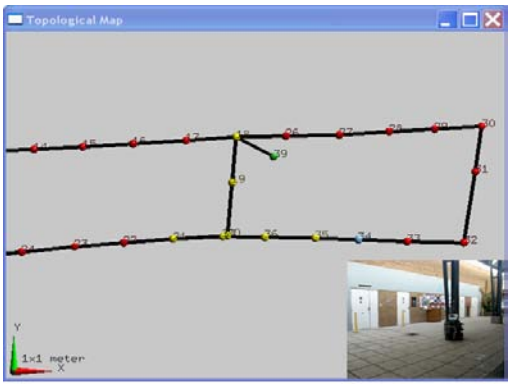
(b) t=0



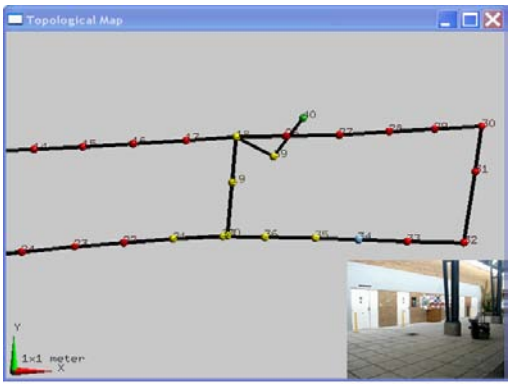
(c) t=1



(d) t=2

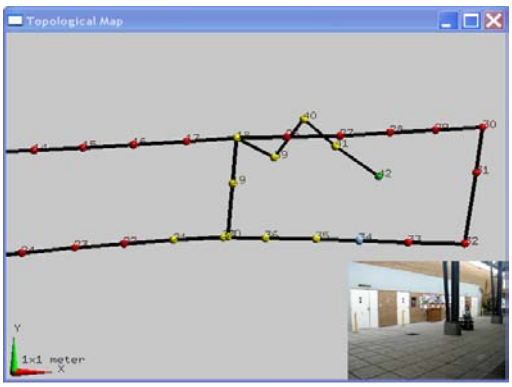


(e) t=3

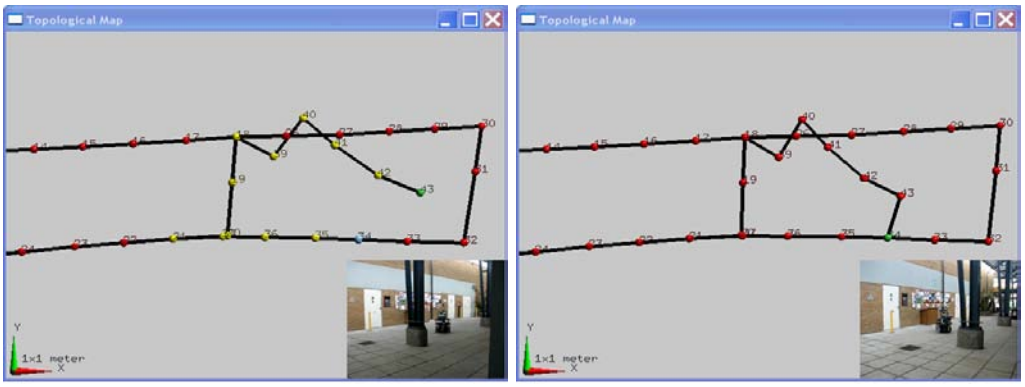


(f) t=4

(g) t=5



(h) t=6



(i) t=7 (j) t=8

Figure 6.60: Semi-Autonomous Experiment 2

(a)

(b)

(c)

Figure 6.61: Matching Omnidirectional Image Pairs for Semi-Autonomous Exp 2(Loop Closure Detection)

6.7.4 Rank-based and Probabilistic Frameworks Comparison

In the previous experiments, the mobile robot employed the rank-based framework. The reason for selecting the rank-based framework to run the experiments instead of the probabilistic framework was straightforward. Firstly, the rank-based framework was anticipated to perform worse in terms of the total number of false positive loop closure detection since a fixed threshold was used. As such, the previously presented experimental results, that showed the mobile robot successfully performing topological SLAM using the rank-based framework, illustrated the robustness of the system as a whole. Secondly, in order to compare the two frameworks, the same dataset has to be used for its evaluation. This means that the experiments have to be conducted using either one of the two frameworks. The frameworks were then evaluated based on the number of hypotheses that a loop closure has occurred is true according to ground truth and the total number of type I and II errors defined as,

- Type I(a) error - Incorrectly accepting a loop closure hypothesis, when in fact it should be rejected (false positive)
- Type I(b) error - Incorrectly accepting a loop closure hypothesis, when in fact it should be rejected after validation (false positive after validation)
- Type II(a) error - Incorrectly rejecting a loop closure hypothesis, when in fact it should be accepted (false negative) causing it to miss a loop closing opportunity.
- Type II(b) error - Incorrectly rejecting a loop closure hypothesis after validation, when in fact it should remain accepted (false negative after validation) causing it to invalidate a good loop closure.

Framework	Ground Truth	Total LC	Type I(a) Errors	Type I(b) Errors	Type II(b) Errors	Type II(a) Errors
Rank	133	134	19 (14.2%)	10 (7.5%)	2 (1.7%)	18 (13.5%)
Prob (E=5)	133	104	8 (7.7%)	4 (3.8%)	1 (1.04%)	37 (27.8%)
Prob (E=10)	133	90	7 (7.7%)	3 (3.3%)	0 (0%)	50 (37.6%)
Prob (E=15)	133	68	3 (4.4%)	1 (1.47%)	0 (0%)	68 (51.1%)

Table 6.1: Performance of Rank-based and Probabilistic Frameworks (F=15, G=2)

The results in Table 6.1 were compiled using the indoor, semi-outdoor and outdoor datasets in the autonomous experiments (Section 6.7.1); the offline experimental datasets (Section 6.7.2), the semi-autonomous datasets (Section 6.7.3) and the supplementary experimental datasets included in Appendix E. The probabilistic framework was evaluated on the same dataset using 3 different values for the parameter E (5,10 and 15) with $F = 15$ and $G = 2$, where the parameters E, F and G were thoroughly described in Section 5.4.2.

Based on the ground truth locations of each image in the dataset, the actual number of loop closures that were supposed to be detected by the system is 133. This number was derived based on the mobile robot's location with respect to previously seen locations (existing locations in the topological map). If its current location was found to be within a proximity of 0.5m from a previously seen location according to ground truth, then a loop closure should be detected, matching the current location of the mobile robot to this existing node in the map. The results in Table 6.1 show that the rank-based framework produced the least Type II(a) errors, with the value of the parameter E ranging from 5-15. The results further revealed that the rank-based framework only missed 13.5% of all loop closures that it was supposed to detect (133 for all experiments). Despite having the least number of Type II(a) errors, the rank-based framework was reported to produce the most number of Type I(a) and Type I(b) errors. Type I errors were considered worse than Type II errors since an incorrectly accepted loop closure damages the map building and localization process. In general, Type II errors were more tolerable as long as the mobile robot managed to detect loop closures before the positional drift in its localization becomes too unreliable for it to make well-informed decisions.

Without the active validation strategy, the rank-based framework produced a total of 19 Type I(a) errors, which is equivalent to 14.2% of all true loop closures to be detected according to ground truth (134-19=115 for the rank-based framework). With the complete loop closing system described in Section 6.3.3, the system managed to reduce the total number of Type I(a) errors to 10 after validation. These remaining errors were categorized as Type I(b) errors. In fact, 60% of the remaining errors were close matches since the matching node was an immediate neighbour of the true node according to ground truth. Given the system's ability to recover local metric information from its current location with the matching node, these close matches would cause minimum damage to the map building and localization process. Unfortunately, the validation process has incorrectly invalidated 1.7% of the good loop closures detected, which were classified as Type II(b) errors. As mentioned earlier, Type II errors in general were more tolerable than Type I errors.

In this comparison, the probabilistic framework was demonstrated to produce far less Type I(a) errors as compared to the rank-based framework, which implied that the active validation strategy was being used less often. With the active validation strategy, it further reduced the total number of Type I(a) by approximately 50%, with the remaining unrectified loop closures classified as Type I(b) errors. Unfortunately, the reduction in Type I errors comes at the price of higher Type II(a) errors, which implied that more effort was required for the mobile robot to detect a loop closure. The tradeoff is basically between an increase in the total time required to detect loop closures for the probabilistic framework or an increase in total operational time to conduct loop closure validation for the rank-based framework. However, the probabilistic framework is still considered to be superior to the rank-based framework since it does not impose a fixed threshold to discriminate between possible previously seen and new locations. Additionally, the probabilistic framework is arguably more general and suitable for all types of environment

(although the fixed threshold used in the rank-based framework was shown to work in indoor, semi-outdoor and outdoor environments).

6.8 Discussion

The mobile robot was tested in a wide variety of environments ranging from indoors to outdoors operating in the fully autonomous mode presented in Section 6.7.1, using the rank-based framework. These experiments clearly illustrated the capability of the mobile robot to perform autonomous navigation in unknown environments. A topological map, which maintains the spatial relationship between its nodes, was built by the mobile robot as it performs navigation. Using the active loop closing strategy, the mobile robot was able to revisit nodes in the environment in order to maintain the global consistency of the map and recover from positional drifts in the localization of the mobile robot via loop closing. The complete loop closing system, which incorporates active loop closure validation, has been shown to be effective for reducing the number of Type 1(a) errors (defined in Section 6.7.4) for both the rank-based and probabilistic frameworks. For each experiment conducted in the fully autonomous mode, the final topological map was compared to ground truth. The topological map built by the mobile robot has been shown to be coherent although not highly precise as compared to ground truth. This reinforces the fact that autonomous navigation in unknown environments does not necessarily require the utmost precision, making it suitable for use in many applications that are not interested in building the most accurate map possible. Nevertheless, these maps could be further improved given sufficient time for the mobile robot to make more observations of an ambiguous location in the topological map.

The initial system, which excluded active loop closing detection and validation, was tested in an indoor environment where the complete system was tested in both the semi-outdoor and outdoor environments. Multiple experiments conducted in the semi-outdoor environment were illustrated, starting from simple scenarios and gradually progressing to more complicated ones. The most complete experiment, which demonstrated the full capability of the mobile robot, was illustrated in semi-outdoor experiment 5. In addition, interesting results were presented in semi-outdoor experiment 6 where the mobile robot was operating in low lighting conditions (matching correctly to a blurred omnidirectional image in the database). In semi-outdoor experiment 7, the mobile robot was tested in a different semi-outdoor environment in order to demonstrate the robustness of the system. The system was further challenged to operate in an outdoor environment, which was considered a successful experiment. However, due to the soft terrain in which the mobile robot was required to travel in, there were multiple times that it required human assistance for it to overcome the resistance of the soil buildup next to its caster.

Although not presented in the experimental results, the mobile robot was also tested with the automatic baseline selection technique developed in Chapter 3. The entire operation of the system remained the same except that it automatically determines its baseline before it commenced navigation and readjusting its baseline after every 15m of distance travelled. This was tested in the semi-outdoor environment illustrated in Fig. 6.23. The

automatically selected baseline was consistent with results presented in Fig. 3.13, with the baseline automatically selected at 30cm. Since all the autonomous experiments including those in the same environment were conducted using a fixed baseline of 30cm, the experimental results that include the use of the automatic baseline selection technique were omitted. To truly illustrate the benefit of using the automatic baseline selection technique in a fully autonomous experiment, the mobile robot would be required to navigate in a large and diverse environment containing both cluttered and open spaces. For example, the mobile robot should be navigating from a cluttered indoor corridor to an open semi-outdoor/outdoor environment, which might require it to engage automatic doors, unlevelled ground, recognize man-made pathways (to contain it to travel only along man-made pathways in wide open areas), etc. In addition, the mobile robot should also decide when to execute the automatic baseline selection procedure by detecting changes in the environment (e.g. from cluttered to open spaces or vice-versa) instead of performing this procedure after every 15m of distance travelled.

In the next experiment, the mobile robot was driven around manually in the semi-outdoor environment collecting all the necessary data from its sensors for building a map in an offline manner. Two offline maps were illustrated in Section 6.7.2. Since the location where the mobile robot builds a node in the environment was known and given that the grids on the floor were approximately uniform (60 x 60 $\pm$ 5cm in each dimension), a set of ground truth node locations were calculated and used to assess the quality of the offline maps built by the mobile robot. Again, the topological maps produced were coherent but not highly precise. The subsequent semi-autonomous experiments in Section 6.7.3, which utilized one of the two offline maps presented in the experimental results, showed that the mobile robot was capable of finding its way to a target goal location given that the topological map provided might not be highly accurate.

The rank-based framework was compared to the probabilistic framework in Section 6.7.4, revealing the pros and cons of both frameworks, on the same dataset collected using the rank-based framework in all the experiments (including the supplementary experiments in Appendix E). The probabilistic framework was concluded to be more superior than the rank-based framework although it might require more effort to detect loop closures, depending on the parameters. Having said so, it produces less Type I errors which were considered more risky than Type II errors in general and compensated the longer time required to detect loop closures by having to perform less validations. In addition, the probabilistic framework was concluded to be a more attractive and elegant framework to utilize because it eliminated the requirement of a fixed threshold, could be conveniently adapted to other systems operating on different appearance models and image similarity metrics and for systems requiring different convergence speeds. In the next chapter, the probabilistic framework will be shown to be operating within a metric SLAM system for map-merging purposes.

Lastly, a ground plane detection system using the information returned by the Bumblebee stereovision system was developed based on the work of Agrawal et al. (2007) (thoroughly described in Appendix F). This can possibly add another layer of intelligence for

the mobile robot to make better decisions during navigation and obstacle avoidance. However, the current system is too occupied with the existing algorithms, but this issue can be resolved by replacing the current processor with the next generation multi-core processor. There are many possible extensions that can be made to the current system. An interesting consideration would be to find locations containing unique and viewpoint invariant landmarks to perform loop closing. The ease and accuracy with which an appearance-based system can relocalize itself to this unique and viewpoint invariant landmark are expected to improve. As a result, this will make the overall system more robust. In addition, the notion of "k-reachability" and "k-connectivity" sets in nodal graphs might be a beneficial factor to consider while deciding the target loop closure node. For example, a more connected node does not only imply a higher likelihood of being able to detect loop closure (e.g. due to a better position estimate) but also providing more options for loop closure validation due to its denser connectivity (since the loop closure validation is based on ensuring its "*local context consistency*"). Moreover, as a node is more connected, it makes it more crucial to ensure that the loop closure event is valid since it is expected to inflict greater damage to map if the system commits to an incorrect loop closure due to its dense connections. Finally, besides balancing the mobile robot's efforts amongst loop closing and exploration, it is definitely worthwhile considering ways to include goal seeking into the equation.

6.9 Chapter Summary

Map building and localization are prerequisites for a mobile robot navigating unknown terrains. However, a truly autonomous mobile robot has to decide its next course of action depending on whether it should be maximizing exploration, loop closing, goal seeking or a combination of them. In this chapter, a complete loop closing system, performing both active loop closing detection and validation (tight integration with the path planner), has been described. On the lower level, a reactive obstacle avoidance system is developed as a safety measure for avoiding obstacles that were not detected by the omnidirectional stereovision system, allowing the mobile robot to steer clear of harm's way. In conclusion, the multiple experiments presented in this chapter, using the fully autonomous, semi autonomous and offline modes, have validated the proposed system.

7

Online Map Merging

7.1 Introduction

Map merging is an important but difficult problem in mobile robotics. It is important because it addresses the issue of merging independent maps collected by a team of mobile robots or merging partial maps collected by a single mobile robot on different runs into a globally consistent map. The multi-robot scheme is more suitable for tackling large-scale environments since multiple robots can cooperate to explore the same environment. Nevertheless, the same result can be achieved by using a single robot which performs intermittent exploration at different times. This eliminates the complexity and cost associated with the multi-robot scheme, but with the tradeoff that exploration and mapping of the same environment will take more time to complete. Ultimately, depending on the requirements of the target application (i.e. home vacuum cleaners, lawn mowers, scout robots, search and rescue robots, etc), it provides a more practical solution for the inclusion of mobile robots into our daily lives and ensures its life-long operation. The difficulty of this problem varies depending on whether the initial corresponding starting locations of the multiple robots or the starting locations of a single mobile robot for each independent run are provided, the type of sensors available and the kind of environment where it is required to operate in. Nonetheless, this problem has not been receiving as much attention as the SLAM problem.

The introduction of probabilistic frameworks in SLAM systems, specifically those primarily based on laser rangefinders and perform laser scan matching, have achieved tremendous success in solving the SLAM problem. Since it is a prerequisite for mobile robots to perform SLAM in unknown environments, it is not too surprising that many multi-robot or single robot schemes capable of map merging employ a laser rangefinder as its primary sensing mode following its success in the SLAM problem (Thrun, 2001; Konolige et al., 2003; León et al., 2009). Of course, the current state of the art is not only limited to laser-based SLAM systems. In the past decade, visual SLAM systems have become more

feasible and practical due to the availability of cheap computing power and improvements in image quality. Due to the richness of visual information, the field is currently experiencing a surge of new visual SLAM systems. A popular approach is to identify and track distinctive landmarks/features in the environment. Unfortunately, there are no complete geometric map merging systems using vision as its primary sensor in the literature. In (Jennings et al., 1999), two mobile robots, each equipped with trinocular stereovision, could localize themselves relative to each other at different times by detecting corresponding landmarks between their maps. However, the two separate maps were not merged into a globally consistent one. In another work (Gil et al., 2010), the proposed system performed map building using a multi-robot scheme by taking visual measurements (with landmarks detected using SIFT (Lowe, 2004)) from each mobile robot. A common global map was built for all mobile robots simultaneously instead of performing map merging when overlapping areas were detected or when the mobile robot meets (due to limitations with inter-robot communication). Unfortunately, two rather restrictive assumptions were made for this system. Firstly, it assumed that each mobile robot was able to communicate with a central agent in the system without interruption. Secondly, the relative initial starting positions of the mobile robots was required to be approximately known. These assumptions, particularly the first one, make it unusable in many applications. In a more recent work by Konolige et al. (2010), a purely vision-based solution was outlined for the case of a single mobile robot scheme, where partial maps collected by manually driving it at different times were merged into a globally consistent one when overlap was detected. This system did not fit into the category of geometric map merging systems because its map was made up of a set of non-linear constraints among camera views (perspective), represented as nodes and edges (similar to a topological map with 6DOF relative metric information between nodes) whereby these constraints were derived using the inputs from the visual odometry and place recognition system. Its place recognition system was based on the popular use of visual vocabularies which incorporated geometric information from the matching of stereo views for robustness.

In the literature, there are several notable works for laser-based map merging systems such as the virtual robot approach proposed by Adluru et al. (2008), the use of the Hough Transform by Carpin (2008) and the manifold representation introduced by Howard et al. (2006). Adluru et al. (2008)'s virtual robot approach took local maps built by each individual mobile robot in the environment as range measurements for its virtual robot with the odometry readings of the virtual robot derived from the registration of similar structures in the individual local maps. Local maps from each mobile robot were conveniently merged together by the virtual robot when similarities between them were detected. The virtual robot approach eliminated the need to track the joint trajectories for each individual mobile robot to only the trajectory of the virtual robot making it more scalable (more robots possible) and more robust (only 1 common structure between local scans is sufficient). However, it is also arguable whether it is truly robust to perform map merging when only 1 common structure is available although this might not be an issue since a multiple hypotheses framework was used. The possibility to maintain multiple

hypotheses also raised the issue with the exponential increase in the number of hypotheses to maintain, which has not been addressed. In addition, the issue of communication between multiple robots was not clearly addressed and it seems that communication to a central agent from each mobile robot was required. The experimental results included in the paper illustrated the maps produced by merging local maps from 3 to 10 mobile robots for an offline dataset collected in a maze-like environment using corner features.

On the other hand, the system proposed by Carpin (2008), based on the Discretized Hough Transform, required the mobile robots to exchange information. This implied that the mobile robots must be within the communication range of one another to share its knowledge about the environment. Map merging was performed when an overlap between maps were detected and the relative transformation parameters were recovered by using the cross correlation of the Hough spectrum and x-y spectra of the maps (converted into binary images) to be merged. Although this means that, for larger environments, the mobile robots would have to deliberately plan to meet up; however, this solution seems more plausible than having a central agent (unless the central agent is a mobile robot and the system does not require uninterrupted communication with the central agent). As for the system proposed by Howard et al. (2006), loop closing and map merging were achieved by using a novel map representation. As defined by Howard et al. (2006), maps were taken out of the two dimensional plane and were transformed into a surface embedded in a higher dimensional space using the manifold representation. By facilitating a many to one representation, this representation has the advantage of being self-consistent and resolves issues due to errors accumulated in the localization of the mobile robot. As such, map merging could be delayed until it can confidently decide the corresponding merging point for the map. Initial relative positions of the mobile robot were not required to be known by the system and the presence of another robot in its vicinity was detected by using distinct laser-visual bar codes. However, many of its processes were computationally expensive (laser scan matching, its manifold representation and the batch processing nature of its Maximum Likelihood estimator) and might not be suitable for online map merging systems, since it is less desirable to have the mobile robots being idle for too long while waiting for computations to complete. Zhou and Roumeliotis (2006) proposed a system closely resembling to (Howard et al., 2006) but they replaced the Maximum Likelihood estimator with the Extended Kalman Filter (EKF), performed landmark matching instead of scan matching and did not employ the manifold representation in order to create a suitable online map merging system. Similarly, each mobile robot was capable of detecting the presence of other mobile robots but by purely using visual means. This is by far the only system which does not assume the maps to be overlapping for map merging. To achieve this, they exploited the capability of the mobile robots to detect the presence of another mobile robot in the vicinity which may or may not have any overlapping regions in the local maps.

The map merging problem is actually closely related to the loop closing problem. Both problems attempt to identify whether the mobile robot is at a previously visited location with the additional extension for the map merging problem to stitch corresponding maps,

which exhibit similar locations, into a globally consistent map. As such, many problems associated to the dependency on laser rangefinders or vision systems alone discussed in Section 6.3.2 similarly applies to the map merging problem. The work presented here shares the same idea with Ho and Newman (2007), in the sense that both visual and laser range information were found to be beneficial for improving the robustness of place recognition systems which solve the map merging and loop closing problems. The main limitation of the visual loop closure detection system proposed in (Ho and Newman, 2007) was the lack of a probabilistic framework to allow the system to degrade gracefully when uncertainty becomes prevalent. This limitation was later resolved by Cummins and Newman (2008) (discussed in Chapter 5) who proposed a generalized probabilistic framework for appearance-based mapping and localization. For optimal performance, it required a once-off learning process to build its visual dictionary for the bags-of-words paradigm, which may take hours to perform. Nevertheless, it could still perform sub-optimally when standard dictionaries were used.

To the best of our knowledge, the work presented here is the first to combine a probabilistic Haar-based place recognition system using omnidirectional vision with a SLAM system based on laser scan matching for map merging. Instead of the pan-tilt camera system used in (Ho and Newman, 2007; Cummins and Newman, 2008), an omnidirectional vision system was employed to alleviate the *windowing* problem suffered on systems using perspective cameras and effectively reducing the time required to produce an image with 360^o FOV of the location at the cost of a lower effective resolution of the omnidirectional image. Furthermore, the approach proposed here is algorithmically simple, efficient and does not require any offline processing; facilitating the merging of maps to be executed as an online process. Since the proposed system was tested on a single mobile robot, one or more previously collected maps could be loaded into the system. Subsequently, the mobile robot performs SLAM by associating consecutive laser scans via an EKF framework. Each of these laser scans or local maps would be associated with an omnidirectional image which describes the appearance of the location where the laser scan was taken. Using the Voronoi exploration strategy described in (Tungadi and Kleeman, 2009), the mobile robot performs autonomous exploration. This exploration strategy was found to be highly beneficial for the proposed map merging system since the trajectory taken by the mobile robot was ensured to be roughly the same given a static or partially static environment. As a result, the detection of map merging opportunities was improved. The system would only proceed with map merging via scan matching only if the place recognition system provided a match. Finally, the maps were merged if scan matching succeeded.

The work presented in this chapter is the outcome of the collaboration with a fellow PhD candidate, Fredy Tungadi, at the Intelligent Robotics Research Centre. It is basically an application of the probabilistic Haar-based place recognition system described in Section 5.4.2 combined with Fredy's laser-based scan matching SLAM system with autonomous exploration to solving the map merging problem. Please refer to Section 5.4.2 for details of the place recognition system. This chapter proceeds by describing the research platform used in this work followed by a brief overview of the laser-based scan

matching SLAM system. Subsequently, the map merging algorithm is presented, highlighting how the place recognition system was integrated with the SLAM system. Finally, experimental results conducted in an indoor environment will be presented, followed by discussion, possible future work and conclusion.

7.2 Research Platform

The two wheeled differential drive ActivMedia Pioneer P3-DX (Mobile Robots, 2010) illustrated in Fig. 7.1 was used as the main research platform in this work. It was equipped with two Hokuyo URG-04LX laser range finders, each with a maximum range of 4m, mounted on the front and rear of the mobile robot covering a full 360^o of the plane. In addition, an omnidirectional vision system made up of a PixeLink camera looking upwards to an equiangular mirror designed by Chahl and Srinivasan (1997) was mounted onto the centre of the mobile robot (the same mirror used for the omnidirectional stereovision system in Chapter 2).

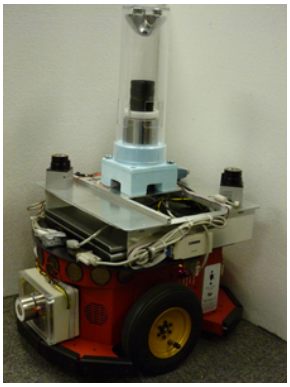


Figure 7.1: The ActivMedia Pioneer P3-DX with two Hokuyo laser rangefinders and an omnidirectional vision system

7.3 Autonomous Exploration and Scan-Matching SLAM

The SLAM algorithm in (Tungadi and Kleeman, 2009) was implemented based on the EKF framework described by Davison (1998), where all map features were included into the SLAM state vector and were updated on each measurement step. The prediction model, based on odometry readings, was derived based on (Kleeman, 2003), where its error model assumed error sources for its wheel separation and the left and right wheel length measurements as additive white noise. As presented in (Tungadi and Kleeman, 2007), scan poses, each associated to a local laser scan, were created for every 0.7m travelled by the mobile robot and were updated using the observation obtained by scan matching. There are many variants of scan-matching available in the literature but Polar Scan Matching (PSM) (Diosi and Kleeman, 2005) was employed by this system due to its faster convergence. PSM operates in the laser scanner's native polar coordinate system; simplifying the search for corresponding points in the scan via bearing matching. The

augmented state vector, containing both the state of the mobile robot (θ_v, x_y, y_v) and its scan poses, is expressed as,

$$X = [\theta_v, x_y, y_v, L_1, L_2, \ldots, L_n] \tag{7.1}$$

with the locations of the scan poses defined by the centroid of the laser scanner's pose in the global coordinate frame where they were created and is defined as $L_i = (x_{Li}, y_{Li}, \theta_{Li})$. The observation model for the pose of a laser scan's coordinate frame with respect to the mobile robot is calculated as follows,

$$H_L(t) = \begin{bmatrix} x_{hi}(t) \\ y_{hi}(t) \\ \phi_{hi}(t) \end{bmatrix} = \begin{bmatrix} (x_{Li}(t) - x_v(t))cos(\theta_v(t)) + (y_{Li}(t) - y_v(t))sin(\theta_v(t)) \\ -(x_{Li}(t) - x_v(t))sin(\theta_v(t)) + (y_{Li}(t) - y_v(t))cos(\theta_v(t)) \\ \phi_{Li}(t) - \theta_v(t) \end{bmatrix} \tag{7.2}$$

A heuristic-based error estimation technique described in (Tungadi and Kleeman, 2007) was used for providing a better estimation of the error covariance of the scan matches to the EKF in corridor environments.

As described in (Tungadi and Kleeman, 2009), the exploration algorithm took advantage of the traversable paths generated by using the Voronoi Graph (perimeter of the Voronoi cells), which enabled the mobile robot to strategically explore the environment by travelling in a loop-like trajectory. The generated trajectory has maximum clearance from obstacles detected in the environment. Paths leading through gaps that were too narrow for the mobile robot were rejected by ensuring the perpendicular distance from the boundary of the Voronoi cells to the closest obstacle to be greater than the radius of the mobile robot. In addition, the Voronoi Graph exhibited loop-paths which helped in guiding the mobile robot to possible loop closing locations during exploration in order to maintain the global consistency of the metric map.

The plane sweep algorithm proposed by Fortune (1997), which provided a simple $O(nlogn)$ solution, was used to generate the Voronoi Graph. The Voronoi graph was then converted into an undirected-weighted graph structure such that it could be used for path planning and exploration purposes. The exploration algorithm works by periodically extracting loop-paths using the loop-path extraction algorithm in (Tungadi and Kleeman, 2009) and subsequently executing them in order of size, progressing from shorter to longer loop paths, until all loop-paths have been executed. A graph-based exploration technique was used to fully map the environment. By using this strategy, a stable partial map creation was ensured before the mobile robot was required to travel longer distances to other parts of the environment. As described previously, the Voronoi Graph would produce approximately the same path given a static or partially static environment. This is an attractive property in path planners for systems using loop closing, place recognition or map merging techniques that are dependent on the visual appearance of the location. If the mobile robot is too far off from its previous trajectory, the visual appearance of the current location may be too different from the previous location. This results in lost

opportunities to detect loop closures and perform map merging. Nevertheless, this is not possible when the environment has changed, such as the inclusion of new obstacles in the environment.

7.4 Fusion of Laser Scan-Matching and Probabilistic Place Recognition for Map Merging

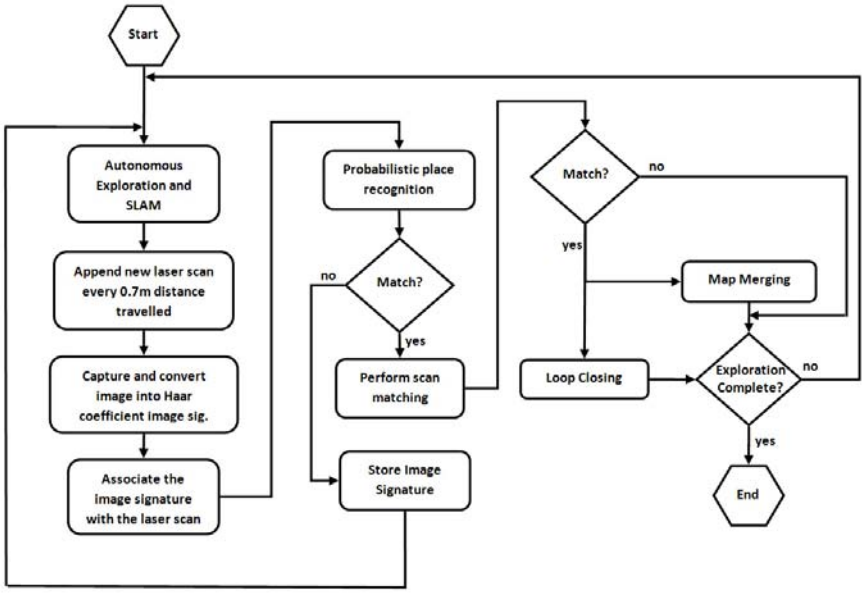


Figure 7.2: Flowchart of the Online Map Merging Algorithm

The mobile robot would start off in an unknown environment without knowing its initial starting position. This initial starting position was assumed to be the origin of the local coordinate system attached to the current map. An observation of the environment was made (laser scan and scan matching), producing a scan pose that was associated to a Haar decomposed image of the current location, for every 0.7m travelled by the mobile robot. Subsequently, the probabilistic Haar-based place recognition system would decide whether the mobile robot was at a previously seen location in the current and/or previously built maps (if loaded into the system initially) or not based on the maximum posterior probability by using the decomposed image (-1 for new location or a previously visited location otherwise). If the system was believed to be at a new location (no match), the system would insert this new scan pose, associated with a local laser scan, into the metric map. In addition, a new node, associated to the new scan pose in the metric map and the corresponding Haar decomposed image signature, would be inserted into the bidirectional graph structure (a topological map). The structure of this bidirectional graph is different from the one used in Chapter 5 since it does not explicitly maintain the spatial relationship between nodes in the map (maintained by the metric map). This graph was purely used to maintain the connectivity between image signatures captured at different locations. On the other hand, if a match was returned, it implied that the place recognition system believed that the mobile robot was at a previously seen location either in the current map

or a previously built map depending on the node being returned. As described in Sect. 5.2, the Haar decomposed image is not rotation invariant (the unwarped image would be column wise shifted for every 10^o equivalent in pixels resulting in 36 decomposed images representing the same place). Due to this reason, the system was able to provide an approximate relative orientation of the current mobile robot's position with respect to the reference image/laser scan up to a resolution of 10^o, making it easier and faster for laser scan matching to converge.

When a matching node was returned by the place recognition system, laser scan matching was performed. Firstly, the coordinate frame of the current scan would be transformed into the coordinate frame of the matching reference scan. Then, using the suggested orientation from the place recognition system, PSM was performed by iteratively minimizing the sum of squared range residual. This would result in the relative pose of the laser scan with respect to the matching reference scan. Subsequently, loop closing or map merging was performed if both the laser scan matching and place recognition system's results were consistent. For the case where loop closure was detected, the scan matching results would be used to update the current map, using the standard EKF update equation in (Davison, 1998). This process does not require any new scene features to be appended into the SLAM state vector. In contrast, if map merging should be performed, the relative pose resulting from scan matching would be used to find the relative transformation matrix of the current map with the corresponding previously built map which contains the matching node. Using this transformation matrix, the current laser scan would be transformed to the reference frame of the previous map. In order to maintain the correct correlation between maps, each of the pose would be appended into the SLAM state vector one by one as in (Davison, 1998) using the following equations,

$$X_{new} = [x_v, L_1, ..., L_n, L_i^{new}] \tag{7.3}$$

$$P_{new} = \begin{bmatrix} P_{xvxv} & P_{xvL1} & \dots & P_{xvLn} & P_{xvxv}(D_L)^T \\ P_{L1xv} & P_{L1L1} & \dots & P_{L1Ln} & P_{L1xv}(D_L)^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{Lnxv} & P_{LnL1} & \dots & P_{LnLn} & P_{Lnxv}(D_L)^T \\ D_L P_{xvxv} & D_L P_{xvL1} & \dots & D_L P_{xvLn} & P_{L_i^{new} L_i^{new}} \end{bmatrix} \tag{7.4}$$

where

$$D_L = (\frac{\partial L_i^{new}}{\partial x_v}) \tag{7.5}$$

$$P_{L_i^{new} L_i^{new}} = D_L P_{xvxv} (D_L)^T + (\frac{\partial L_i^{new}}{\partial H}) R (\frac{\partial L_i^{new}}{\partial H})^T \tag{7.6}$$

and L_i^{new} is the ith pose of the reference scan from the previous map, x_v is the current pose of the robot, H is the measurement of the new scan pose, L_i^{new}, and R is the estimated covariance of H. In addition, the graphs which maintained the topological structure of these maps for the place recognition system were merged accordingly.

For the case when laser scan matching and the place recognition system produced inconsistent results, the mobile robot would overturn the decision to perform loop closing or map merging and proceed with exploration provided that there are still regions in the map to be explored (loop paths from the Voronoi Graph). The entire algorithm is summarized into the flowchart shown in Fig. 7.2.

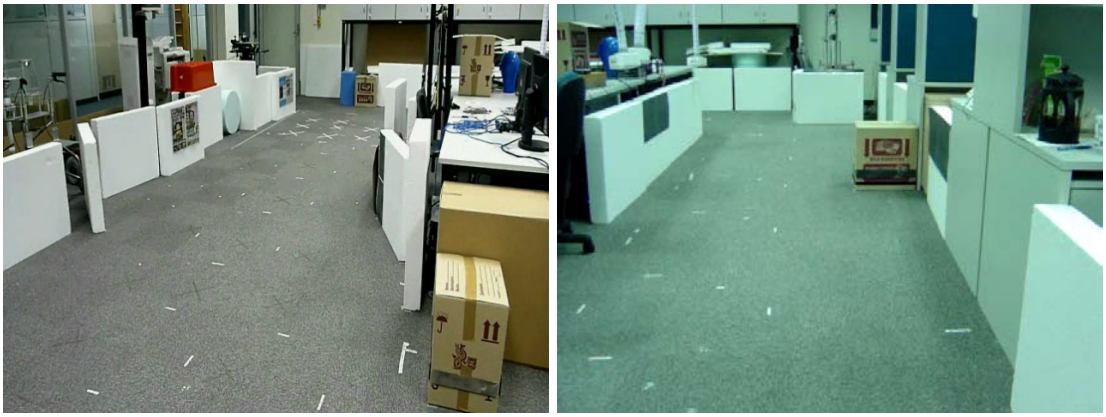
7.5 Results

Experiments were conducted in an indoor lab and office environment at Monash University. In these experiments, a partial map of the environment would be created by the mobile robot before it was moved to another random location in the environment. The system could either be turned off/reset (with the partial map preloaded into the system prior to its subsequent execution) or remain switched on while it was being transported. However, in these experiments, the former case was illustrated in order to simulate a typical indoor mobile robot which is more likely to be switched on and off at different times and days for exploration and mapping rather than doing it as a once-off continuous process.

The omnidirectional vision system presented in Sect. 7.2 was capable of producing images at resolutions of 1280 x 1024. Nevertheless, for these experiments, the omnidirectional images were captured at a resolution of 640 x 512. As the mobile robot performs autonomous exploration and SLAM, unique laser scans, each associated with its corresponding omnidirectional image, were created at intervals of approximately 0.7m. A total of three experiments were conducted with varying difficulty levels and the following illustrates results obtained from each experiment where the place recognition system's parameters were defined as $E = 15$, $F = 15$ and $G = 2$ respectively.

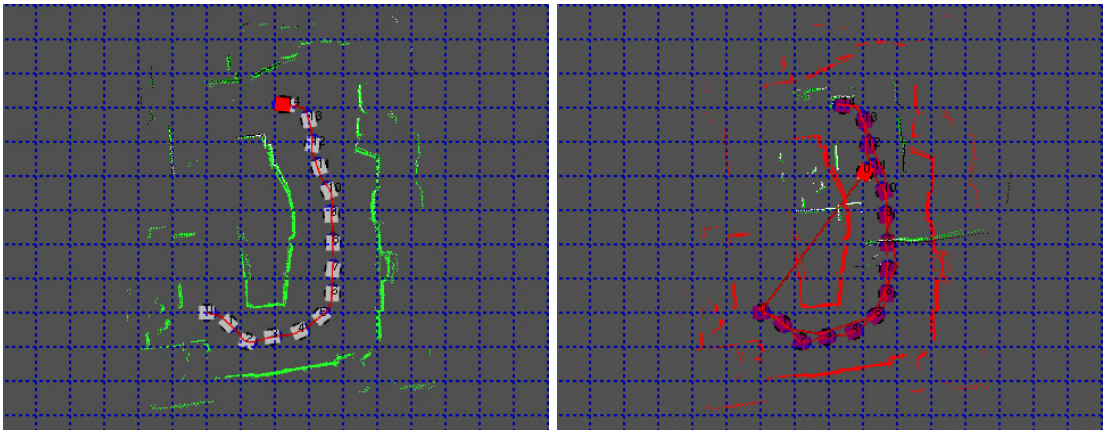
a First Experiment - Map Merging in Lab G15

In the first experiment, the mobile robot was initially required to build a partial map of Lab G15. Lab G15 is shown in Fig. 7.3 and the partial map built is shown in Fig. G.2. It was then switched off, restarted at a random location in Lab G15, preloaded with the previously built partial map and headed off in the opposite direction of travel (as compared to the direction of travel taken for the partial map) for exploration (Fig. G.3). In Fig. G.4, the place recognition system found a matching node in the previously collected partial map (map merging) which was later verified by laser scan matching. The mobile robot continued to traverse in the environment and Fig. 7.4(g) shows the locations where loop closures (yellow circles) and false negatives (blue circles) were detected by the place recognition system. No false positives were reported by the system in this experiment.

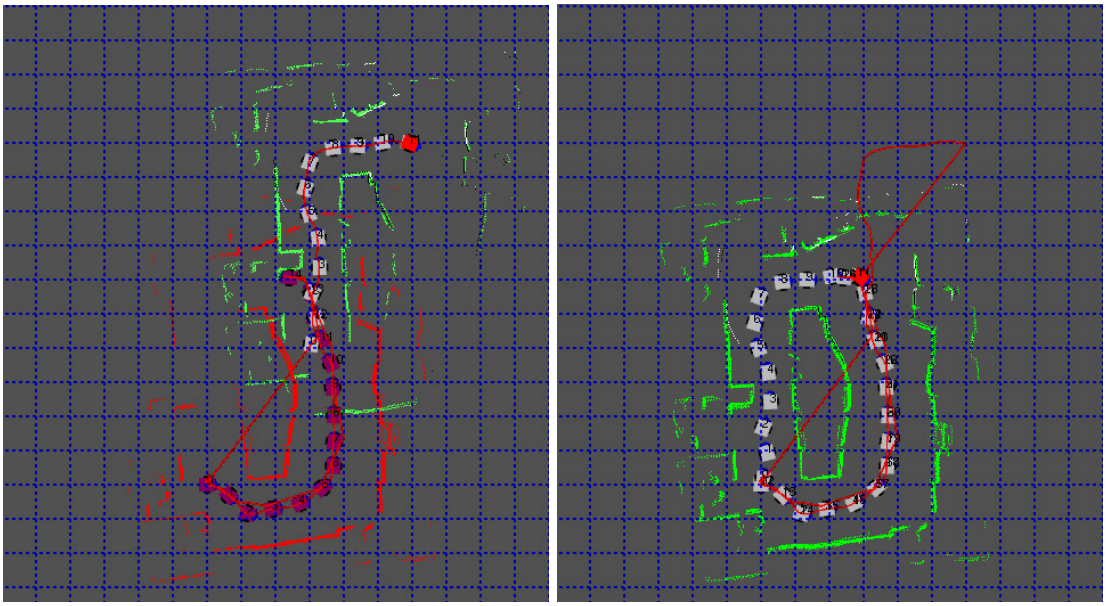


(a) (b)

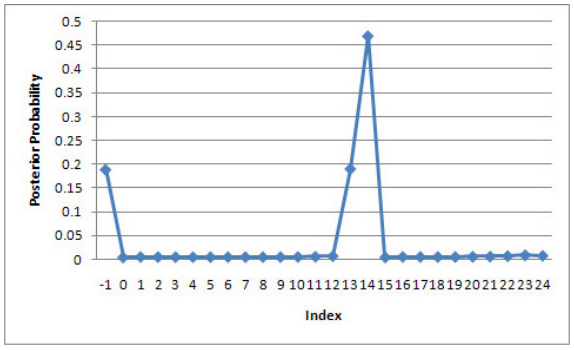
Figure 7.3: Lab G15



(a) Partial Map (b) Restart Location Overlaid with Partial Map (Red)

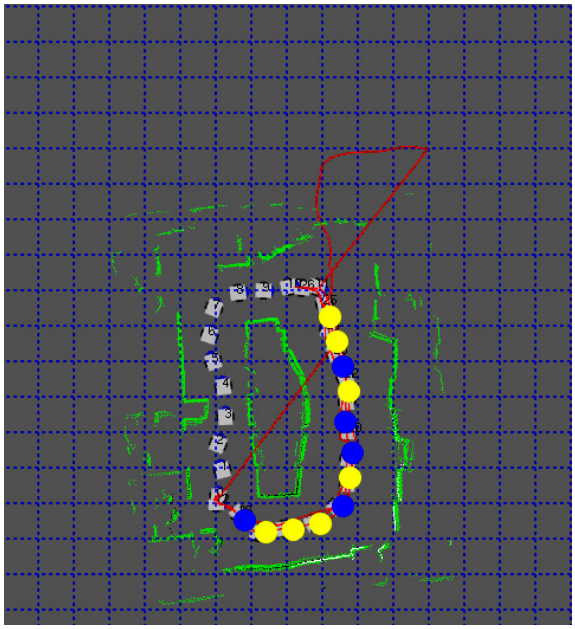


(c) Map Merging Detected (d) Merged Map



(e) Posterior Probability at Merging Point

(f) Omnidirectional Images



(g) Merged Map - Loop Closures After Map Merging

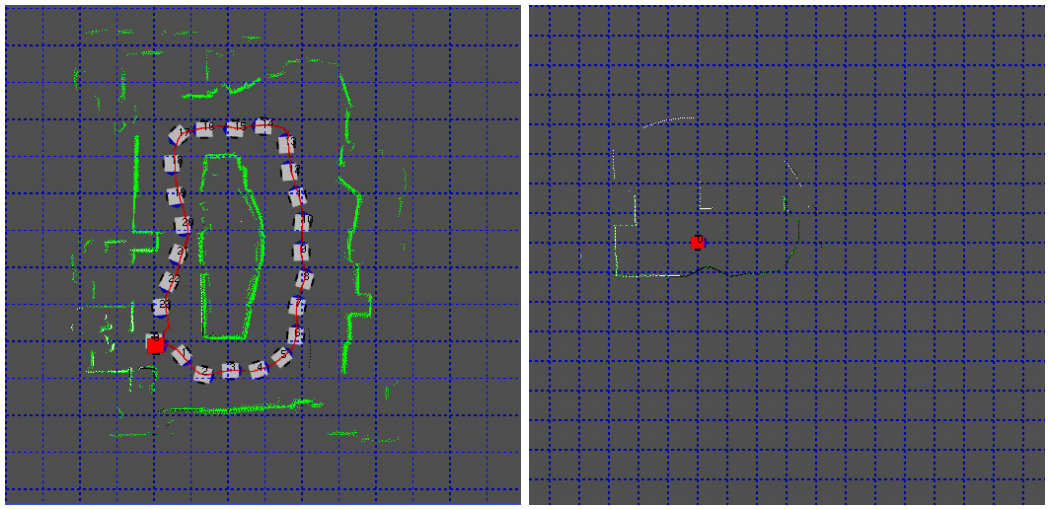
Figure 7.4: Map Merging Experiment 1 (Grid Size is 1x1m)

b Second Experiment - Labs G10 and G15

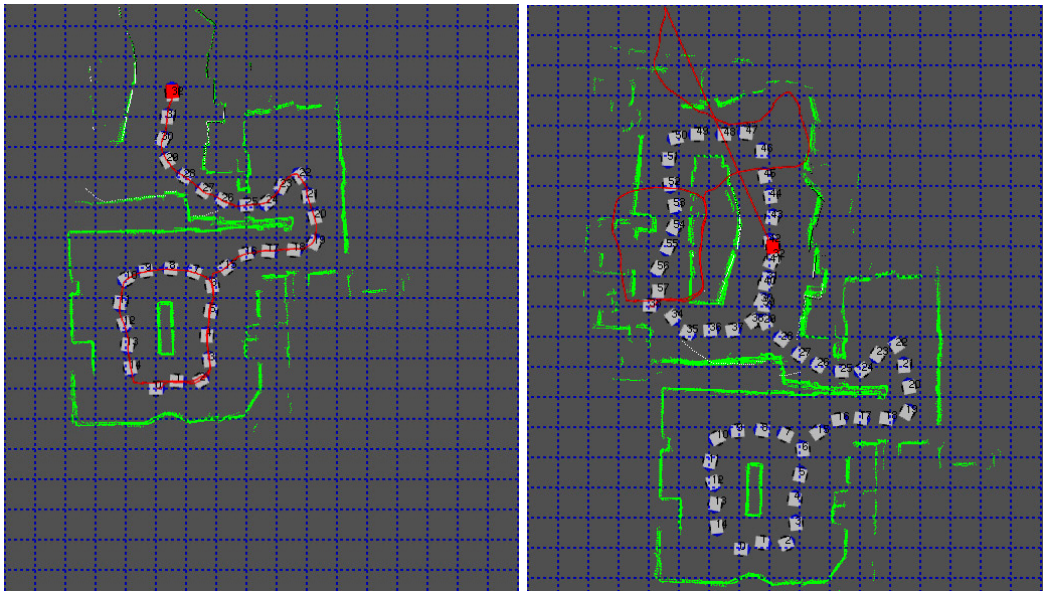
In the second experiment, the robustness of the system was tested in a more challenging setting. The mobile robot was initially required to explore and complete the map of Lab G15 (shown in Fig. 7.6(a)). It was then switched off, restarted at a random location in its neighbouring Lab G10 with the previously acquired map of Lab G15 preloaded into the system. Lab G10 is shown in Fig. 7.5 and the initial state of the mobile robot when it was being restarted at a random location in Lab G10 is shown in Fig. 7.6(b) (without the previously acquired map of Lab G15 overlaid on top of the current map this time). The mobile robot completed the exploration of Lab G10, revisited some nodes on the way out to the corridor before it detected a map merging location shown in Fig. 7.6(c) in Lab G15. The mobile robot continued to traverse in the environment and Fig. 7.6(d) shows the locations where loop closures (yellow circles) and false negatives (blue circles) were detected by the place recognition system. Similarly, no false positives were reported by the system in this experiment.

(a) Lab G10 (b) Corridor of Labs G10 and G15

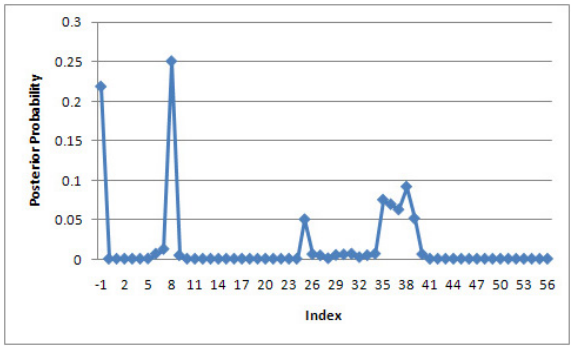
Figure 7.5: Lab G10



(a) Partial Map (b) Restart Location

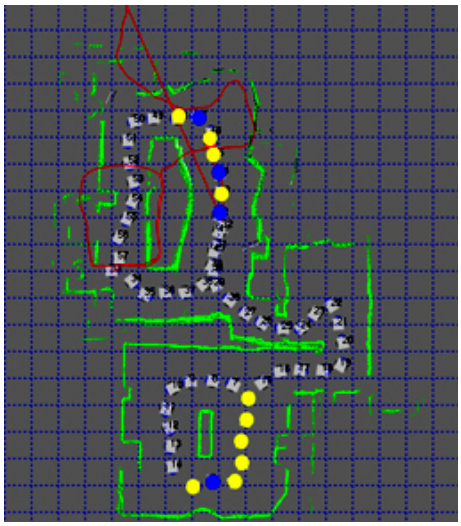


(c) Map Merging Detected (d) Merged Map



(e) Posterior Probability at Merging Point

(f) Omnidirectional Images



(g) Merged Map - Loop Closures After Map Merging

Figure 7.6: Map Merging Experiment 2 (Grid Size is 1x1m)

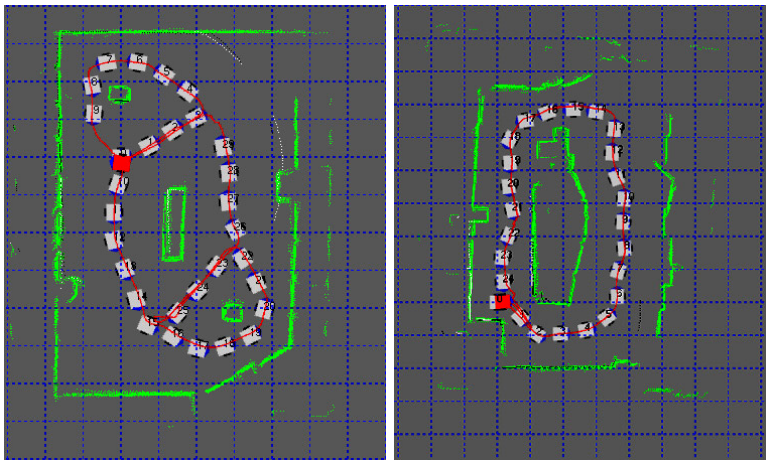
c Third Experiment - Labs G10 & G15, Room G13 and High Voltage Lab

The third experiment was similar to the second experiment except that the mobile robot was required to explore a larger environment and merge the current map of the mobile robot with 3 previously acquired maps. The mobile robot was preloaded with 4 non-overlapping maps as shown in Fig. 7.8. In this experiment, it would not be possible for the mobile robot to perform map merging with the High Voltage Lab since it has been deliberately denied access to it. Nevertheless, the map of the High Voltage Lab was still loaded into the system in order to create more ambiguity for the place recognition system since it would now have a larger database of images to compare with. Take note that Lab G10 in this experiment was made up of a much larger area as compared to the one demonstrated in the previous experiment. The mobile robot, randomly placed in Lab G10, was restarted and performed autonomous exploration. Experimental results demonstrating the mobile robot performing map merging at Lab G10, Room G13 and Lab G15 are shown in Fig. 7.9(a)-7.9(f). The posterior probability of the probabilistic place recognition system at various map merging locations are shown in Fig. 7.9(g)-7.9(i) with some sample omnidirectional images taken during the experiment shown in Fig. 7.9(j).

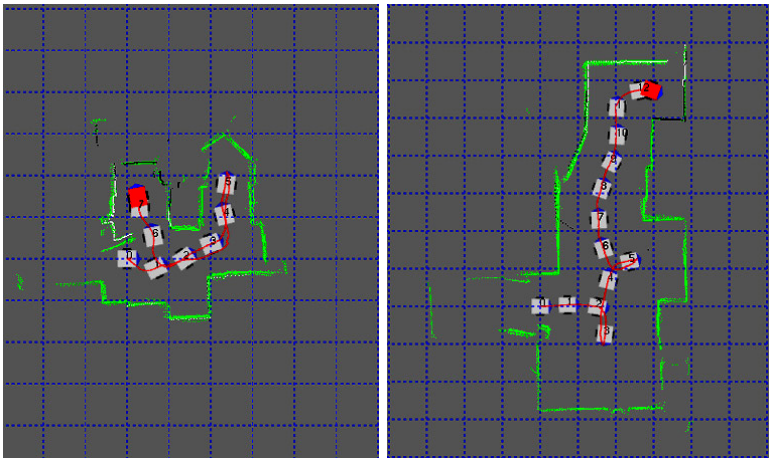
The final map is illustrated in Fig. 7.9(k) which includes the locations where loop closures (yellow circles), false negatives (blue circles) and false positives (magenta circles) were detected. The two false negatives reported by the place recognition system were rejected by laser scan matching.

(a) (b)

Figure 7.7: (a)Room G13 and (b)High Voltage Lab

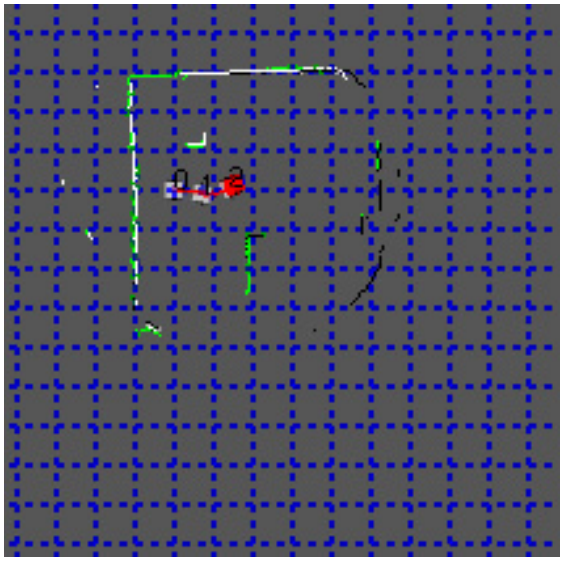


(a) Lab G10 (b) Lab G15

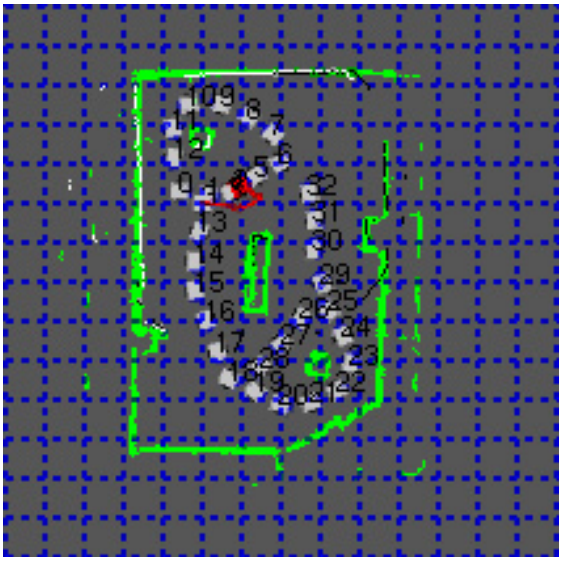


(c) Room G13 (d) High Voltage Lab

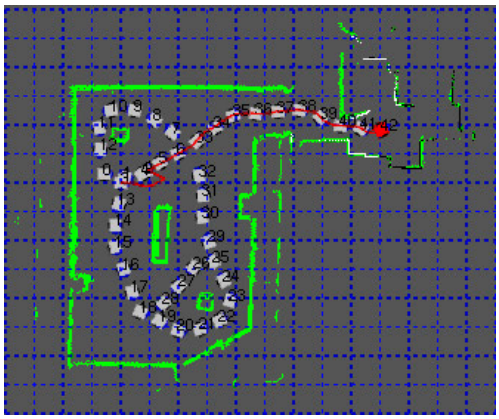
Figure 7.8: Partial Maps for the Third Map Merging Experiment



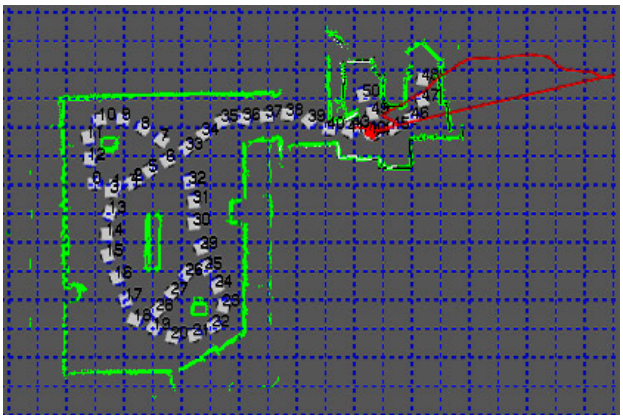
(a) Map Merging Detected at Lab G10



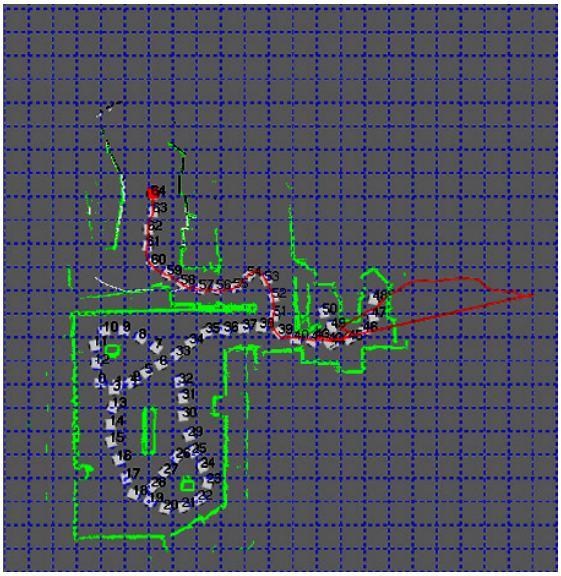
(b) Current Map Merged with Map of Lab G10



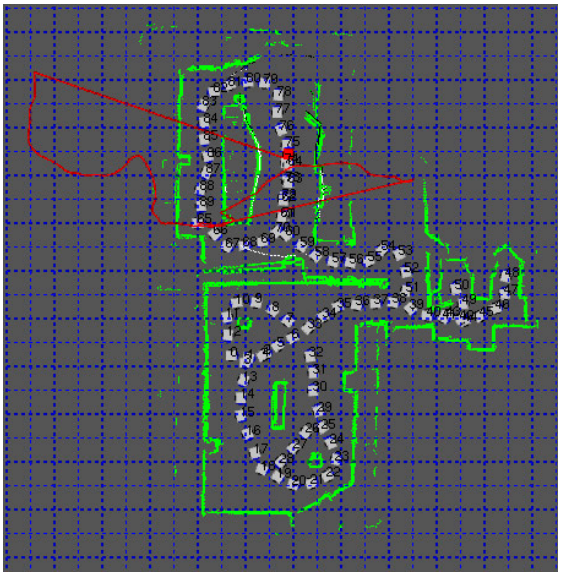
(c) Map Merging Detected at Room G13



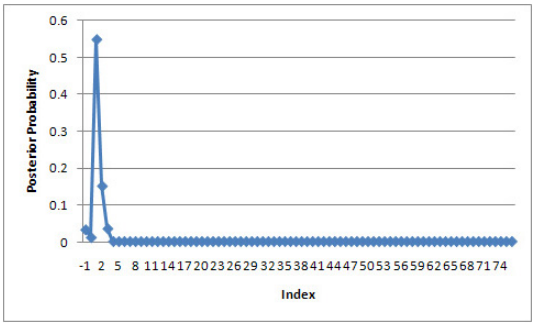
(d) Current Map Merged with Map of Room G13



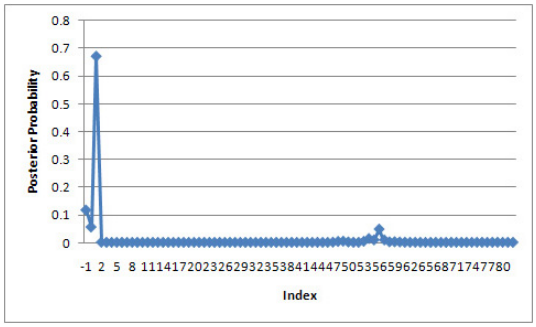
(e) Map Merging Detected at Lab G15



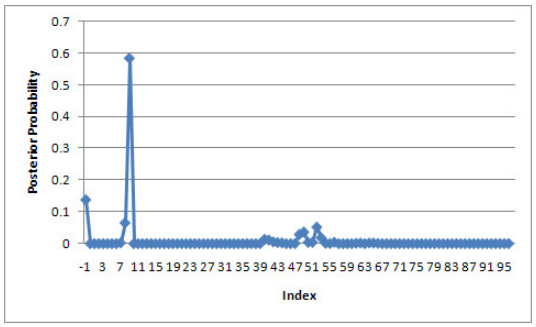
(f) Current Map Merged with Map of Lab G15



(g) Posterior Probability at Merging Point at Lab G10

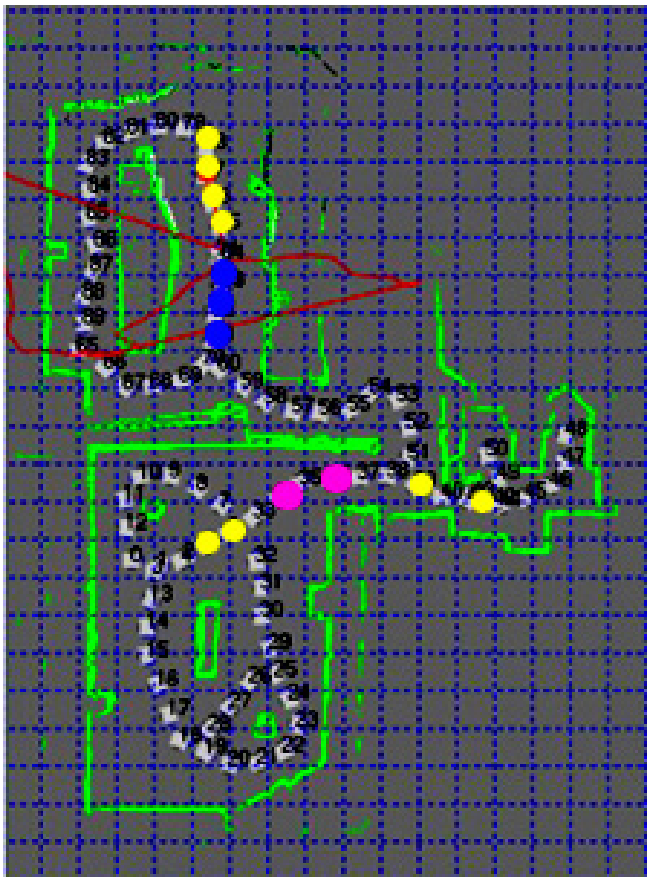


(h) Posterior Probability at Merging Point at Room G13



(i) Posterior Probability at Merging Point at Lab G15

(j) Omnidirectional Images



(k) Final Map - Loop Closures After Map Merging

Figure 7.9: Map Merging Experiment 3 (Grid Size is 1x1m)

7.6 Discussion

The experimental results clearly showed that the proposed online map merging system could robustly perform map merging in challenging environments featuring geometrically similar corner and junctions. The loop closure detection and map merging processes have also been made more efficient since the search area for scan matching was significantly reduced when the place recognition system provided it with the matching node and an approximate relative orientation with respect to its reference scan. The graph in Fig. 7.10 provides a comparison of the total time required to perform exhaustive scan matching as opposed to image querying on a laptop with a 1.6GHz AMD processor with 1Gb RAM. Furthermore, the image querying process could be further optimized by replacing the current image querying process described in (Jacobs et al., 1995) with a kd-tree implementation.

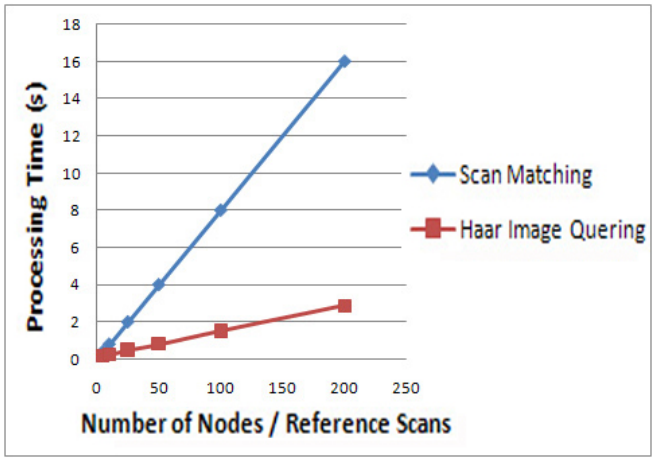


Figure 7.10: Comparison of Image Querying and Scan Matching Processing Time

As described previously, the experiments were conducted with the parameters of the place recognition system defined as $E = 15$, $F = 15$ and $G = 2$ respectively. Referring to the comparison between the rank-based and probabilistic frameworks in Sect. 6.7.4, this set of parameters produced the least amount of Type I errors at the cost of requiring more effort in detecting loop closures since a higher E value required a more discriminative score to be produced when comparing the appearance model at the current location of the mobile robot with the true matching node relative to other nodes in the map for the probabilistic place recognition to believe it was indeed revisiting a previously seen location. Nevertheless, this was found suitable for this system since the combination of the metric SLAM system with the Voronoi Graph for autonomous exploration ensured the mobile robot to traverse in approximately the same path when it revisited the same location given a static or partially static environment. Since the mobile robot's location would always be close to where the reference image was taken, this produced on average a much more discriminative score with the appearance model associated with the reference location. Also, by keeping away from objects, the views were less dominated by very close objects and collision avoidance was easier (more tolerance). In these experiments, the place

recognition system illustrated the following properties desirable on map merging systems: (a) the gradual build up of its belief that it was indeed in a previously seen location (depending on how discriminative the image matching scores are) and (b) dampening the effect of perceptual aliasing which might lead to more false positive detections. Of course, there were instances when the system's belief to be at a previously seen location builds up rather quickly if the scores were highly discriminative. This was not surprising since a highly discriminative matching score provided a very strong indication to the system that there was a high probability that it was indeed revisiting this location in the map based on the appearance of the image.

The experiments conducted clearly showed how the refined probabilistic framework described in Sect. 5.4.2 could be easily adapted to the map merging problem. The set of weights used by the image querying function for this system was the same set of weights used in all other experiments in the previous chapters (Table 5.1) even when a different camera (PixeLink camera) was used. The robustness of the proposed online map merging system could be further improved by incorporating an active validation strategy, similar to the one described in Sect. 6.3.3, such that ambiguous matches could be validated. This is very beneficial since an incorrect map merging decision causes greater irreversible damage to the map as compared to an incorrect loop closure decision when a single hypothesis framework was used. On the other hand, if a multiple hypotheses framework was used, an active validation strategy would also be useful to reduce the number of hypotheses to maintain in the system.

The current issue with the system is its inability to navigate in environments featuring long and straight corridors without any geometrical differences since it is impossible for scan matching to validate the match provided by the place recognition system even if the environment is rich in visual information. Nevertheless, it is possible to resolve this issue by tracking distinctive visual features (e.g. SURF) in the scene on the unwarped omnidirectional image and estimating its 3D location by using the proposed calibration method described in Chapter 2. Interestingly, where there is considerable visual complexity, the Haar-based signature is more discriminative. In the same situation, the laser range data is probably more reliable if the 3D space is not too cluttered and rich in vertical geometric features. Thus, an ideal scenario where this combination works best is when the space is visually rich, is not too cluttered and contains vertical geometric features.

7.7 Chapter Summary

This is the first system to combine a probabilistic Haar-based place recognition system using omnidirectional images with laser scan matching to perform map merging. By using an omnidirectional vision system instead of a pan-tilt camera unit, it alleviates the *windowing* problem suffered on systems using perspective cameras and reduces the time required to produce an image with 360^o FOV. In conclusion, the experiments conducted validated the proposed system and demonstrated how the probabilistic place recognition system could be easily adapted onto different systems and be used as a solution to the map merging problem.

8

Conclusion and Future Work

8.1 Conclusion

Autonomous mobile robots have long been envisioned to roam the surface of the Earth, with the idea conveyed through science fiction novels, blockbuster movies and prototypes in research laboratories. Despite the advancements made in the past decades, the current state-of-the-art is still far from those being depicted in the context of its intelligence and reliability. Research has been conducted from several directions in order to further our understanding towards the mystery behind the general intelligence exhibited in human beings from psychology to neuroscience to artificial intelligence to engineering. Due to the complexity of the human brain, it is often hard to make clear conclusions from experimental results which makes the studying of smaller but yet intelligent, non-human primates like the macaques (Rosa and Tweedale, 2005) or small organisms like bees (Dyer et al., 2005; Srinivasan et al., 2009), ants (Phan and Russell, 2009), rodents (Milford and Wyeth, 2010), etc, an attractive approach.

In Chapter 1, perception was identified as the main bottleneck in intelligent robotics with visual information being the primary sensing mode which forms the way we model and understand the world. Perception is part of intelligence and it allows us to reason, learn, understand and solve problems. Our visual perception system is not flawless. It is prone to various kinds of optical illusions, unable to detect large scene changes when the change coincides with visual disruptions arising from a saccade or brief obscuration of the scene, context dependency based on priors for object recognition in low resolution images, etc.

Despite the various challenges, there are strong expectations that intelligent robotics will emerge as the "next PC industry" as predicted by the founder of Microsoft Corporation (Gates, 2007). It might be true that a great amount of research effort and technological breakthroughs are required to achieve the kind of general intelligence as displayed in robots such as C-3PO or R2-D2 featured in the popular Star Wars films. However, task specific

robots for every household such as those envisioned in (Gates, 2007) (e.g. laundry, lawn-mowing, vacuum cleaning, food and medicine dispensing robots) are already starting to emerge. In a way similar to that which applied to the PC industry, where it was initially used in a very limited set of applications to being used as a general tool in every industry 30 years later through research and development, it is anticipated for the robotics industry to pick up in the same way, starting from task specific and evolving into general robots.

In this thesis, a mobile robot capable of navigating in unknown environments was thoroughly described. The author has explored and integrated several higher level, visually perceived information sources such as range estimation, egomotion estimation and place recognition into a mobile robotic system that performs simultaneous localization and topological map building. The onboard navigation system of the mobile robot utilizes this information to decide its next course of action, performs path planning and executing the selected path, as it balances its effort amongst loop closing and exploration.

Range estimation was achieved primarily using the omnidirectional stereovision system described in Chapter 2. In that chapter, a camera calibration technique suitable for omnidirectional vision systems using equiangular mirrors was described. Stereo correspondences were established via a local area-based method and depth estimation was achieved via a triangulation process that utilizes the information from the camera calibration process. The establishment of stereo correspondences and triangulation were both offloaded to the Graphics Processing Unit (GPU) with details provided in that chapter. The omnidirectional stereovision system was further extended from a standard to a multi-baseline omnidirectional stereovision system in Chapter 3, with the aim to reduce the ambiguity in depth estimates via a voxel voting process. In addition, that chapter details an automatic baseline selection algorithm, which can be used by both the standard or multibaseline versions of the system to dynamically adjust the vertical separation between the top and bottom omnidirectional catadioptric vision systems according to the environment. A secondary pseudo range information (disparity map) was returned by a Bumblebee stereovision camera described in Chapter 6, functioning as a reactive obstacle avoidance system, to steer the mobile robot away from obstacles which the omnidirectional stereovision system has failed to detect. The two sensors complement one another due to the vertical stereo setup for the primary sensor and the horizontal stereo setup for the secondary sensor, where the former is more sensitive towards horizontal features whereas the latter is better for vertical features.

In Chapter 4, the egomotion estimation or visual odometry system was introduced. It provides a 3DoF motion estimate of the mobile robot by combining the distance travelled estimated using a ground plane optical flow tracking algorithm, with the bearing/heading estimated by the panoramic visual compass algorithm. This system had been rigorously evaluated in different environments and an average positional drift of 5.64% of the total distance travelled was reported for a semi-outdoor environment. Equipped with the visually percevied range and egomotion information together with the place recognition system detailed in Chapter 5, the mobile robot was capable of building a globally consistent topological map (with relative spatial information between nodes preserved in the links of the

graph) by applying a global relaxation algorithm as it performs simulataneous localization and mapping (manually driven in the environment). In Chapter 5, the Haar-based place recognition system was presented in two frameworks (e.g. rank-based and probabilistic), where the robustness of using the Haar decomposed image as the appearance model for unique locations in the map was evaluated on a dataset of semi-outdoor and outdoor omnidirectional images. The proposed probabilistic framework was demonstrated to exhibit many desirable properties for a place recognition system with one of the key advantages being its flexibility to allow the user to decide what is deemed as a discriminative score and for adjusting its convergence speed depending on the target application and system. With the ability to recognize previously visited locations and an algorithm to recover the local relative metric information between the current location of the mobile robot with the location of the matching node (since producing a match does not require the mobile robot to be exactly but just close to the it), the mobile robot was able to perform loop closing.

Deliberate loop closing was demonstrated to be advantageous in Chapter 5 to partially rectify errors in the map building process and to contain the errors (positional drift) accumulated in the localization process over time. The importance of performing loop closing regularly was reemphasized in Chapter 6 in the context of autonomous navigation. Besides loop closure detection, which was presented as the place recognition system in Chapter 5, Chapter 6 argued that a complete loop closing system should exhibit the following properties: (a) it should actively search for a loop closure instead of just waiting for loop closures to happen by chance, (b) there should be ways of measuring the validity of a detected loop closure, (c) it should validate an ambiguous loop closure by actively validating it in the future (gathering new evidence) if past and present information were not capable of resolving its ambiguity and (d) there should be some way of undoing the modifications made to the map for a system maintaining a single hypothesis or for the case of a multi-hypotheses system, pruning down the number of hypotheses that it has to maintain based on the validation results. To achieve this, the complete loop closing system was tightly integrated with the path planner where the mobile robot was required to perform and balance its effort between exploration and loop closing. Results from experiments conducted in unknown environments, ranging from indoors to outdoors, demonstrated the performance of the system; combining the various components summarized in the previous paragraphs together with a local reactive obstacle avoidance system to perform autonomous navigation by using only visually perceived information. Besides the fully autonomous experiments, the system was illustrated to operate in a semi-autonomous goal-seeking mode and an offline topological map building mode. Towards the end of Chapter 6, the two frameworks described for the place recognition system in Chapter 5 were compared using the actual data collected from the experiments.

Although vision might be our primary sensing mode, it is clearly subjected to ambiguity. For humans, vision is complemented through our other senses (hearing, touching, tasting and smelling). Similarly, mobile robots should be equipped with a rich set of complementary sensors in face of challenges in the real world. In Chapter 7, the author

explored how laser range data can be combined with an omnidirectional vision system to further improve the robustness of the place recognition system. The indoor mobile robot described in this chapter is a laser-based metric SLAM system capable of autonomous navigation. The probabilistic place recognition system was integrated into this system with the aim to perform map merging as the system recognizes previously visited places in other disjoint maps using a single robot configuration. This resulted in a robust online indoor map merging system that is beneficial for the life-long operation of a mobile robot. Besides the utilization of laser range information, some preliminary work was conducted to explore how an inexpensive GPS receiver could be utilized by the mobile robot in Appendix G. The combination of visually perceived information with the global coordinates from a GPS receiver is deemed an attractive approach for mobile robot navigation since vision algorithms work well in textured and cluttered environments whereas GPS works better in open environments but might fail in urban environments with dense high rise buildings, areas with tall trees (a thick canopy) and in indoor environments.

Although the feasibility of the system has been demonstrated in this thesis, there are still open problems to be addressed with many possible improvements. This thesis concludes with possible future work in the following section.

8.2 Future Work

In this thesis, a brief outline of possible extensions was provided towards the end of each chapter, specific to its context. Given the scope of this thesis, there are many possible avenues for further improvement and future research. In this section, future work that may lead to important research results and further advance the field, is proposed.

A Dynamic Multisensor Framework

In Chapter 7, a robust online map merging system based on a combination of laser scan matching and a probabilistic appearance-based place recognition system was presented. In addition, Appendix G presented preliminary results of the integration with an inexpensive GPS receiver. Despite that visual perception offers many advantages that may not be available in other sensing modes, it does have its downsides. Ideally, a mobile robot should have a rich set of complementary sensors that will further improve the robustness of the system. For example, a place recognition system benefits greatly when both structural information from the laser rangefinder and visual information from the camera are made available to the system to deal with problems such as perceptual aliasing and structural similarities. Similarly, the integration of GPS information on a vision-based mobile robot will also improve the localization of the mobile robot in outdoor environments as they complement one another (as discussed in the previous section). What is proposed here is not merely to extend the current vision-based system to a multisensor framework, but to develop a dynamic system that is able to adjust the priorities and weights of sensor information available to the system (e.g. which sensor information is more reliable and how to mix evidence), parameters of algorithms (e.g. image resolution, stereo correspondence

parameters, single or multi baseline stereovision), etc, based on the current state of the mobile robot and understanding of the environment. For example, the system should place less priority on GPS information when the mobile robot is perceived to be in a dense urban environment with many high rise buildings since GPS information will not be accurate due to multipaths. In some cases, it might be worthwhile for the mobile robot to spend more time to perform multibaseline stereovision in situations where the accuracy of stereo correspondences may be compromised or even when long range depth perception is deemed beneficial.

Key Frame Detection for Appearance-based Mapping and Localization Systems

In Chapter 5, an appearance-based mapping and localization system was presented with the core components being an appearance-based place recognition system for reliable loop closure detection, a relaxation algorithm for maintaining a globally consistent map and a method to recover relative metric information between matching locations during loop closure (besides visual odometry and omnidirectional stereovision presented in other chapters). The autonomous experiments, using the appearance-based mapping and localization system conducted in Chapter 6, captures an omnidirectional image when the mobile robot is $\geq$1m away from the previous node. The system then decides whether a new node is inserted into the topological map to represent this location based on the place recognition system's results. Although this approach was shown to work well in the experiments, this can be further improved by incorporating a key frame detection system.

We define that a key frame detection system's main responsibility is to decide when an image from a video camera should be passed to the place recognition system to validate whether the current location of the mobile robot is a previously seen or new location. It achieves this by comparing the current input image relative to the image associated to the most recent node inserted to the topological map. Based on some image similarity measure, the key frame detection system should find an image frame that has illustrated sufficient difference in terms of its appearance as compared to its reference image at the point in time. Eventually, this will replace the fixed distance threshold (e.g. $\geq$1m away from the previous node) used by the current system to detect key frames. This appearance-based approach to determine key frames based on the detection of sufficient change in its image content has been studied (Sivic and Zisserman, 2003; Nourani-Vatani and Pradelier, 2010; Zhang et al., 2010), but not extensively. The most difficult problem here is to devise a framework to evaluate the performance of approaches using different appearance models since there is no clear justification of the correct number of key points and where they should be for a given environment (ground truth). The proposed extension here includes further investigation on devising a proper performance evaluation framework for such approaches and also to investigate the notion of being able to predict where a key frame should be considering both the planned motion of the mobile robot and the structure of the environment.

Towards Large Scale Autonomous Experiments

In Chapter 6, experiments conducted in various environments operating in the fully autonomous, semi-autonomous and offline modes were presented. The next eventual step is to prepare the system towards large scale autonomous experiments in urban environments. The experiment should ideally challenge the mobile robot to transit from indoors to outdoors, from cluttered to open spaces and from static to dynamic environments. To prepare the mobile robot for such experiments, various upgrades are required in terms of hardware and software.

On the hardware side, a more suitable platform such as the all-terrain, all-weather Seekur Jr. mobile platform (Mobile Robots, 2010) is required to allow the mobile robot to engage in typical outdoor environments (e.g. Fig. 6.46) and yet being small enough to navigate along indoor hallways and powerful enough to handle the payload of the additional sensors and computers. Another possible improvement is to redesign the current mechanical system of the camera elevation device (based on a combination of a DC motor, gears and threaded rods) such that the total time it requires to change from one baseline to another is reduced to make it more attractive and less costly (in terms of time) to execute the automatic baseline selection technique or to combine multiple stereo pairs taken at different baselines regularly (probably by borrowing ideas used in the design of cable-based elevators). In addition, it might be worthwhile to upgrade the current Canon Powershot S3 IS cameras used by the omnidirectional catadioptric vision systems to current HD web cameras (e.g. Logitech C910). The Canon Powershot S3 IS was chosen at the early stages of the research due to it being able to focus well to the equiangular mirror at short distances, flexibility to capture high resolution still images (up to 6MP) and providing a live image sequence (30Hz, 320 x 240 in resolution) at a reasonably cheap price. However, current HD web cameras such as the Logitech C910 (Logitech, 2010), are able to focus well to close objects, provide a live image sequence at resolutions of 1280 x 720 and are now available at very affordable prices. These new cameras will support the development of the proposed key frame detection system, facilitate real-time omnidirectional stereovision at higher resolutions (desirable due to the lower effective image resolution of our omnidirectional vision system) and provides an option to track distinctive visual features that are beneficial to the localization of the mobile robot.

In terms of software, there may be an exhaustive list of possible improvements that may require more research depending on the complexity of the task. The ideal experiment described previously that requires the mobile robot to transit from indoors to outdoors, from cluttered to open spaces and from static to dynamic environments may be the ultimate test for mobile robots but may also be too ambitious at this point in time. Nevertheless, this goal can be achieved by incrementally solving its sub problems. Practically, the immediate experiment is to challenge it to transit from a semi-outdoor to an open outdoor environment containing man-made footpaths. For the vision-based system, a footpath detection system will be required to restrict the mobile robot to traverse only along the detected footpaths. This will then facilitate feasible experiments for the proposed vision-based system. With this configuration, the system can be extended to include additional

sensors such as a GPS receiver that will be beneficial for certain locations in the environment. An additional interesting consideration would be to explore ways to allow the mobile robot to balance its efforts among loop closing, exploration and goal seeking.

Map Maintenance for Long-term Operations

In many applications, mobile robots may need to operate in the same environment intermittently for days, weeks, months or years instead of just a once-off operation that lasts for a few hours. Vision-based mobile robots have to deal with severe lighting differences at different times of the day (Milford and Wyeth, 2010), visual appearances of objects in the scene at different months of the year (e.g. seasonal changes) (Valgren and Lilienthal, 2010), modifications to the scene (e.g. new objects included or previously observed objects removed), etc. In Chapter 7, a robust online map merging (combining laser scan-matching and omnidirectional vision) was presented with the aim to provide the mobile robot with the ability to merge individual maps collected at different times into a globally consistent map as it performs exploration. Nevertheless, a truly robust vision-based system targeted for long-term operations has to deal with the previously identified issues. The first issue relates to lighting invariance that is still a huge problem yet to be resolved. The second issue relates to scene variations due to seasonal changes. For example, the visual appearance of the same location may be very different in winter when the ground is covered in snow as compared to during autumn when most trees are bare with thick layer of leaves on the ground. The last issue relates to the degree of tolerance of the proposed appearance model to deal with modifications in scene (large or subtle) when previously observed objects are removed or new objects are present in the scene. One possible approach is to allow multiple representations for the same location in the environment. Specific to the proposed vision-based system in this thesis, this implies that the topological map should allow multiple appearance-models to represent the same node in the map. Essentially, this translates to several problems such as the higher computational requirements, maintenance of the map (e.g. when to delete old and insert new appearance models) and a feasible way to evaluate the system. Another interesting idea is the notion of identifying unique, viewpoint invariant landmarks that may help in the first two problems but not if this key landmark was removed from the scene, which may require the mobile robot to validate the removal of this key landmark and update its system accordingly.

The system described in this thesis demonstrates a feasible vision-based solution for a fully autonomous mobile robot. This was achieved through the research and development of a variable single/multi baseline omnidirectional stereovision system, visual odometry system and place recognition system into a complete autonomous navigation system. The results presented have also reinforced the fact that autonomous navigation can be achieved even if its internal representation of the environment is not up to millimeters in precision, as long as the map is coherent and with only reasonable accuracy. In conclusion, the vision of having reliable task specific robots is definitely achievable in the near future.

With more research and development efforts, it will not be too far before the dream of realizing a fully autonomous robot with general intelligence can be fulfilled.

A

Multimedia DVD Contents

A multimedia DVD, containing videos that help to better describe methods or results presented in Chapters 2-7 and Appendices E-F, accompanies this thesis. Videos for each chapter can be found in the folders, **ChapterX**, where X corresponds to the respective chapter number and videos for the appendices can be found in the folders, **AppendixY** where Y corresponds to the respective appendix letter. The videos provided were tested to run properly on Windows Media Player or VLC media player. The following details the videos included for the relevant chapters.

Chapter 2: Omnidirectional Catadioptric Stereovision

The following video can be found in the folder **Chapter2** of the multimedia DVD.

- OmniStereo_SO.avi - 3D visualization of omnidirectional stereovision in semi-outdoor environment 1 (Fig. 2.12). Also available on Youtube: `http://www.youtube.com/watch?v=bj2IJBzQO5o`

Chapter 3: Multibaseline Omnidirectional Stereovision

The following video can be found in the folder **Chapter3** of the multimedia DVD.

- Eye-FullTowerBaselineAdjustment.avi - Shows the baseline variation process of the Eye-Full Tower when it was mounted on the ActivMedia Pioneer P3-AT.

Chapter 4: Visual Odometry

The following videos can be found in the folder **Chapter4** of the multimedia DVD.

- PanVisCompass_Outdoor.avi - The panoramic visual compass experiment in an outdoor environment in Fig. 4.5. Also available on Youtube: `http://www.youtube.com/watch?v=k6Qu98TrKQI`

- IndoorVisualOdo.avi - 10 out of 59 randomly selected experimental runs to evaluate the performance of the visual odometry system in an indoor environment (Fig. 4.18(a)). Also available on Youtube: `http://www.youtube.com/watch?v=r8JKCSc5__g`

- SOVisualOdo.avi - 10 out of 22 randomly selected experimental runs to evaluate the performance of the visual odometry system in a semi-outdoor environment (Fig. 4.18(b)). Also available on Youtube: `http://www.youtube.com/watch?v=Y_rXRWD7eOI`

Chapter 5: Mobile Robot Localization and Mapping

The following videos can be found in the folder **Chapter5** of the multimedia DVD.

- IndoorLoopClosing.avi - 10 randomly selected experimental runs in an indoor environment. This is the same video available in the visual odometry chapter but this video highlights the effect of loop closing.

- SemiOutdoorLoopClosing.avi - 10 randomly selected experimental runs in a semi-outdoor environment. This is the same video available in the visual odometry chapter but this video highlights the effect of loop closing.

Chapter 6: Autonomous Vision-based Topological SLAM

The following videos can be found in the folder **Chapter6** of the multimedia DVD.

- ObstacleAvoidance.avi - Reactive obstacle avoidance using disparity maps from the Bumblebee (Fig. 6.16). Also available on Youtube: `http://www.youtube.com/watch?v=IoSMa_eavu0`

- IndoorExp.avi - Indoor autonomous experiment (Fig. 6.20-6.22). Also available on Youtube: `http://www.youtube.com/watch?v=z077PZpPjnI`

- SOExp1.avi - Semi-outdoor autonomous experiment 1 (Fig. 6.25-6.27).

- SOExp2.avi - Semi-outdoor autonomous experiment 2 (Fig. 6.28-6.30). Also available on Youtube: `http://www.youtube.com/watch?v=JKWDhsLfgJs`

- SOExp3.avi - Semi-outdoor autonomous experiment 3 (Fig. 6.31-6.33). Also available on Youtube: `http://www.youtube.com/watch?v=4u5cBLzqitY`

- SOExp4.avi - Semi-outdoor autonomous experiment 4 (Fig. 6.34-6.36). Also available on Youtube: `http://www.youtube.com/watch?v=oe7k2TODrlE`

- SOExp5.avi - Semi-outdoor autonomous experiment 5 (Fig. 6.37-6.39). Also available on Youtube: `http://www.youtube.com/watch?v=YXzYGkdjLiE`
- SOExp6.avi - Semi-outdoor autonomous experiment 6 (Fig. 6.40-6.42). Also available on Youtube: `http://www.youtube.com/watch?v=KLuramqN-2Q`
- SOExp7.avi - Semi-outdoor autonomous experiment 7 (Fig. 6.43-6.45). Also available on Youtube: `http://www.youtube.com/watch?v=mIVwhsa-5ss`
- OutdoorExp.avi - Outdoor autonomous experiment (Fig. 6.46-6.49). Also available on Youtube: `http://www.youtube.com/watch?v=LaFFzRYEWGQ`
- Offline.avi - Example of offline data collection process (Section 6.7.2).
- SemiAutoExp1.avi - Semi-autonomous experiment 1 (Fig. 6.58-6.59). Also available on Youtube: `http://www.youtube.com/watch?v=UU_DwkmowFo`
- SemiAutoExp2.avi - Semi-autonomous experiment 2 (Fig. 6.60-6.61).
- Autobaseline.avi - Automatic baseline selection in semi-outdoor environment. Also available on Youtube: `http://www.youtube.com/watch?v=KxAsp8a3KlQ`

Chapter 7: Online Map Merging

The following videos can be found in the folder **Chapter7** of the multimedia DVD.

- MapMergingExp1.avi - First map merging experiment in Lab G15 (Fig. 7.3-7.4). Also available on Youtube: `http://www.youtube.com/watch?v=LymgfkVpwLs`
- MapMergingExp2.avi - Second map merging experiment in Labs G10 and G15 (Fig. 7.5-7.6). Also available on Youtube: `http://www.youtube.com/watch?v=dQemNJX3kAY`
- MapMergingExp3.avi - Third map merging experiment in Labs G10 G15, Room G13 and High Voltage Lab (Fig. 7.7-7.9). Also available on Youtube: `http://www.youtube.com/watch?v=GXlWLdit5TM`

Appendix E: Supplementary Experimental Results

The following videos can be found in the folder **AppendixE** of the multimedia DVD.

- SupExp1.avi - Supplementary Experiment 1 (Fig. E.1-E.3).
- SupExp2.avi - Supplementary Experiment 2 (Fig. E.4-E.6).
- SupExp3.avi - Supplementary Experiment 3 (Fig. E.7-E.9).
- SupExp4.avi - Supplementary Experiment 4 (Fig. E.10-E.11).

Appendix F: Ground Plane Detection using Stereovision

The following videos can be found in the folder **AppendixF** of the multimedia DVD.

- Indoor_GroundPlane.avi - Ground Plane Detection in indoor environment.
- SemiOutdoor_GroundPlane.avi - Ground Plane Detection in semi-outdoor environment.
- Outdoor_GroundPlane.avi - Ground Plane Detection in outdoor environment.

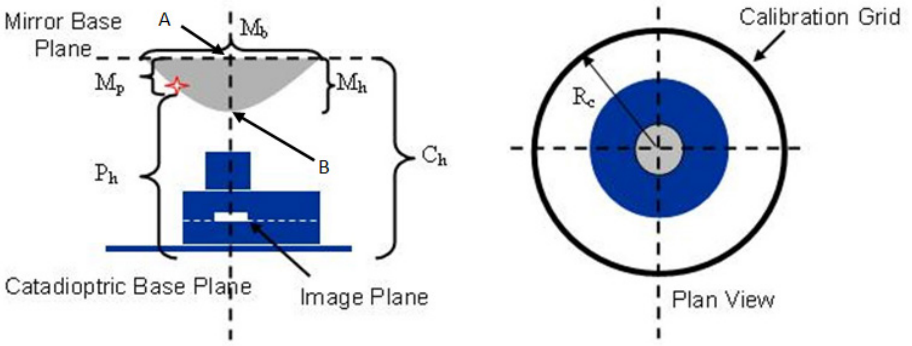
B

Calibrating a Non-central Equiangular Catadioptric System

This appendix thoroughly describes the proposed calibration technique for a non-central equiangular catadioptric system. The only additional tool required for the proposed calibration process is a hollow metal cylinder, with its inner surface fully covered with accurately sized black and white checkerboard patterns which is referred to as the calibration bin. The catadioptric system is placed inside this hollow metal cylinder during the calibration process as shown in Fig. B.1(c). To ensure that errors are minimized, the configuration of the catadioptric system and the hollow metal cylinder should adhere to the three assumptions made for the proposed calibration process; (1) The base of the mirror is parallel to the image plane and catadioptric base plane, (2) the centre of the curved mirror (anywhere along the points between point A and B indicated in Fig. B.1(a)) is aligned to the centre of the catadioptric base and hollow metal cylinder and (3) the hollow metal cylinder is assumed to be a perfect cylinder and wraps perfectly around the catadioptric structure. Subsequently, by making use of the available geometrical information of the catadioptric system labelled in Fig. B.1, the following equations can be used to compute the angle of elevation for the pixel of interest in the image (in this case, the angle of elevation of the point indicated by the red star) which happens to fall on the epipolar line propagating outwards from the centre of the mirror towards the mirror rim in the image.(For reference, the epipolar line at $0^o/360^o$ falls on the same line which propagates vertically from the centre of the mirror to the mirror rim in the image).

The following equation computes the parameter, P_h, which measures the height of the point of interest on the mirror with respect to the catadioptric base,

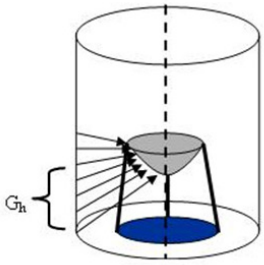
$$P_h = C_h - (M_h - M_p) \tag{B.1}$$



(a) Cross-section of catadioptric system

(b) Plan view of calibration with catadioptric system

(c) Physical Calibration Setup



(d) 3D Model of Calibration Setup

Figure B.1: Camera Calibration Setup

where M_p is the height of the point measured from the mirror base which can be computed by solving the following equation which describes the profile of the curved mirror when the elevation angular magnification factor is $\alpha = 7$,

$$\left(x^2 + y^2\right)^2 - 8x^2y^2 = r_o^4 \tag{B.2}$$

where (x, y) describes the profile of the mirror in Cartesian coordinates, with y chosen to represent the horizontal direction of the mirror profile while x represents the vertical direction, and $r_o = 79.4$ for this mirror. The general quartic equation can then be described as,

$$a_o x^4 + a_1 x^3 + a_2 x^2 + a_3 x + a_4 = 0 \tag{B.3}$$

If $a_3 = a_1 = 0$, the general quartic equation will become a biquadratic equation with the general form as follows,

$$a_o x^4 + a_2 x^2 + a_4 = 0 \tag{B.4}$$

By simplifying equation (B.2) into the form of (B.4) and by letting $z = x^2$ in (B.2), the roots can be obtained by solving the simple quadratic equation,

$$z^2 - 6y^2 z + y^4 = r_o^4 \tag{B.5}$$

where y, representing the horizontal direction of the mirror profile, can also be defined as the actual radial distance (in mm) where the incident light ray reflects off the mirror which is measured from the centre of the mirror and can be computed as

$$y = \frac{\sqrt{(P_x - M_{cx})^2 + (P_y - M_{cy})^2}}{\sqrt{(M_{rx} - M_{cx})^2 + (M_{ry} - M_{cy})^2}} \times \frac{M_b}{2} \tag{B.6}$$

where (M_{cx}, M_{cy}) is the pixel location of the centre of the mirror in the image, (M_{rx}, M_{ry}) is the pixel location of the mirror rim in the image and (P_x,P_y) is the location of the point of interest in the image. The roots, z_1 and z_2, of equation (B.5) can then be obtained by substituting appropriate values of y and r_o into the equation. Since $z = x^2$, there will be four possible solutions for x and the following solution defines the parameter M_p,

$$M_p = +\sqrt{z_1} \tag{B.7}$$

where

$$z_1 = \frac{6y^2 + \sqrt{(6y^2)^2 - 4\,(y^4 - r_o^4)}}{2} \tag{B.8}$$

and

$$z_2 = \frac{6y^2 - \sqrt{(6y^2)^2 - 4\,(y^4 - r_o^4)}}{2} \tag{B.9}$$

This is a manual calibration technique and it requires the manual selection of points on the calibration grid by the user as shown in Fig. B.2. Points are selected based on the location of the vertices of the square grids in the image and are automatically refined by analyzing a small window surrounding the selected point. Since the square grids are uniform in size, it is then relatively easy to determine the parameter, G_h, which measures the actual height of the selected point on the calibration grid in the image from ground (assuming that both the calibration grid and catadioptric camera structure are placed on the ground). With all this information available, the angle of elevation, φ, of a particular point in the image can be determined by,

$$\varphi = tan^{-1}\frac{R_c - y}{|P_h - G_h|} \quad \text{if} \quad P_h - G_h >= 0 \tag{B.10}$$

$$\varphi = tan^{-1}\frac{|P_h - G_h|}{R_c - y} + 90 \quad \text{if} \quad P_h - G_h < 0 \tag{B.11}$$

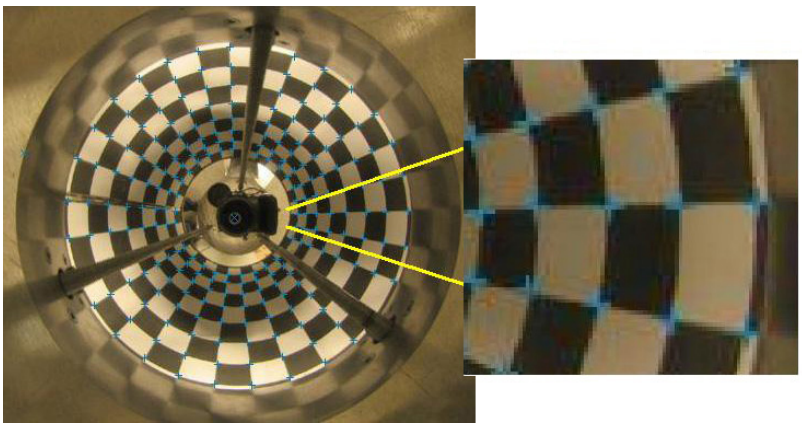
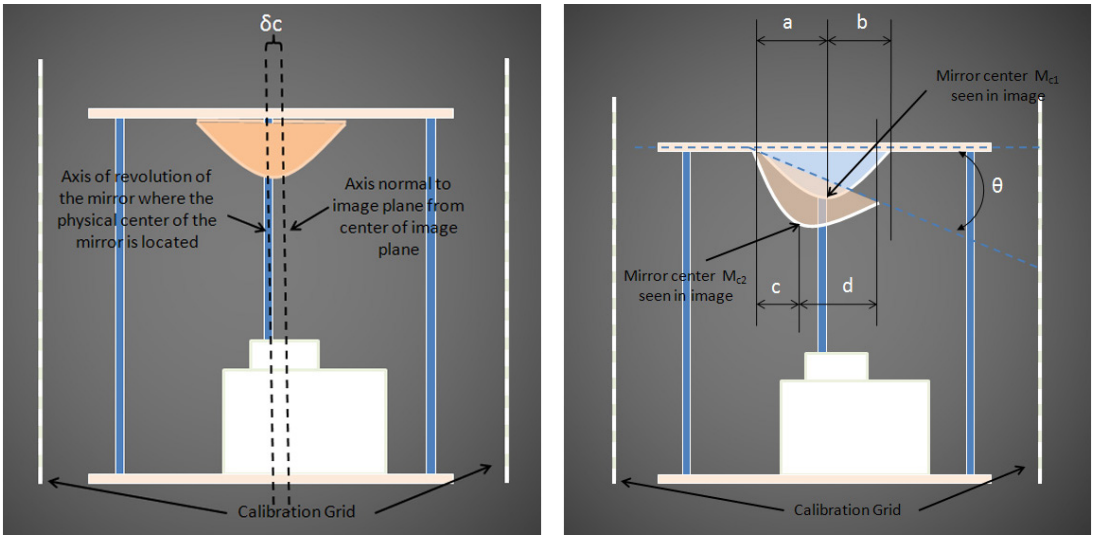


Figure B.2: Manually selected calibration points (marked in blue)

As mentioned earlier, a total of three assumptions were made for the calibration process. If all these assumptions were satisfied, the calibration process could be further generalized and simplified as the angles of elevation computed for selected points on a particular line at a particular angle could be applied to all epipolar lines at angle θ_e. Unfortunately, this is often not the case as it is quite difficult to perform a very precise manual setup. As such, the following discussion will be related to the impact of the violation of these assumptions.

The first assumption requires the base plane of the mirror to be parallel with the camera image plane and catadioptric base plane. To begin with, the catadioptric base plane is calibrated such that it becomes parallel to the ground plane with the assumption that calibration is performed on a planar surface. Then, the subsequent task is to ensure that the base plane of the mirror and camera image plane are parallel to the ground plane by firstly calibrating the base plane of the mirror with respect to ground plane and subsequently the camera image plane with respect to the base plane of the mirror. Due to the design of the system, one of the main sources of error depends on how accurately the base plane of the mirror is calibrated with respect to ground plane. In reality, depending on the point of rotation, this will result in both translational and rotational differences in 3D space (6 DOF). To simplify the analysis process, it is assumed that the translational difference is negligible. In order to further simplify the problem, the analysis will focus on only 1 DOF of rotation and assumes that the point of rotation is on the left hand border of the mirror as illustrated in Fig. B.3(b). As illustrated, the ideal orientation of the base plane of the mirror will actually result in equal portions of the left and right halves of the mirror (indicated by a and b) in the image whereas a non-parallel base plane of the

mirror will have unequal portions (indicated by c and d) measured from the centre of the mirror, M_{c1} and M_{c2}, seen in the image. This will directly affect the estimated radial distance y in equation (B.6) since this relies on the fact that the base plane of the mirror should be parallel to the image plane. However, the total error depends on the amount of rotational difference, θ, which has been exaggerated here in order to illustrate the impact of the violation of this assumption. In reality, by precisely fabricating the components of the catadioptric system and with the aid of a precise digital level, the amount of rotational difference can be rendered very small. Subsequently, a circle, with a user defined radius and its centre location coinciding with the centre of the mirror in the image, is used as a tool to ensure that the image plane is parallel to the base plane of the mirror. This requirement is satisfied when this circle fits perfectly on the border of the mirror in the image as illustrated in Fig. B.4.



(a) Misalignment of the centre of the mirror with the centre of the image plane

(b) Base of mirror is not parallel with the image plane

Figure B.3: Violation of the Camera Calibration Assumptions

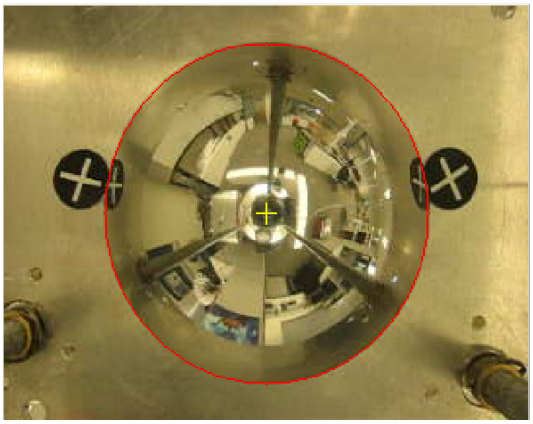


Figure B.4: Calibrating the image plane with respect to the base plane of the mirror (center of the mirror marked with the crosshair and border of the mirror with the red circle)

The second assumption requires the centre of the curved mirror to be aligned with the centre of the image plane. If the first assumption is fully satisfied, the centre of the curved mirror can lie anywhere along the line between the points A and B as illustrated in Fig. B.1(a) because this is the same when it is viewed in 2D. However, since the system can never be ideal, the centre of the image plane is defined to be at point B since this is the point which can be seen in the image. The violation of this assumption will result in a pure translational difference in 3D space between the locations where the two centres will be aligned with the current physical location of the two centres. As a result, it becomes a 2 DOF problem since a change in the vertical direction, which affects the parameter C_h, can be accounted for by making precise measurements. For simplicity, the following analysis is performed by fixing one of the remaining 2 DOFs and assumes that this is a 1 DOF problem. As illustrated in Fig. B.3(a), a shift of δc will eventually cause a shift of the position of the mirror in the warped omnidirectional image. During the calibration process, the user will be required to define the centre and rim of the mirror in the image (without subpixel accuracy). As such, even if this problem is extended back to 2 DOF, this will only result in very small errors (due to the subpixel accuracy) in the determination of y in equation (B.6), M_p in equation (B.7) and φ in equations (B.10) and (B.11) as long as the centre of the calibration tool is aligned with the centre of the mirror. Last but not least, the third assumption which requires the hollow cylinder to be a perfect cylinder is a more trivial matter not because the errors resulting from an imprecise cylinder are negligible but because it has no dependency on any components of the catadioptric system and the fabrication of precise cylinders are achievable with the current technology.

In order to improve the accuracy of the estimated 3D points in the scene, angles of elevation for epipolar lines at regular intervals (defined by the intervals of the square grids of the calibration grid in the image) are computed as shown in Fig. B.2. The angles of elevation (degrees) with respect to radial distance (pixels) from the centre of the mirror in the image along epipolar lines at different angles will be stored in a lookup table. To compute the angle of elevation of a point located on an epipolar line at angle θ_{e1} and radial distance R_{d1}, linear interpolation/extrapolation of the information stored in the lookup table will be performed. Linear interpolation/extrapolation is deemed suitable as it is found that the relationship between the angle of elevation and radial distance of the point in the image is approximately linear (also mentioned in Section 2.3.1 and validated in Fig. 2.6). Once stereo correspondences are established, the angles of elevation can be computed based on the locations of the stereo pair in the image and the 3D position of the scene point can be calculated by performing triangulation.

C

Camera Motion Estimaton using Ground Plane Optical Flow

Assumptions: Camera plane parallel to ground plane and axis of rotation of mobile robot aligns with the camera's optical axis.

A pair of corresponding points in image sequence tracked using optic flow is expressed as,

$$P_1 = [x_1, y_1, 0, 1] \tag{C.1}$$

$$P_2 = [x_2, y_2, 0, 1] \tag{C.2}$$

Two pairs of corresponding points

$$P_c^{'} = P_c\phi \tag{C.3}$$

$$\begin{bmatrix} x_1^{'} \\ y_1^{'} \\ x_2^{'} \\ y_2^{'} \end{bmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta \\ sin\theta \\ dx \\ dy \end{bmatrix} \tag{C.4}$$

Solving 4 unknowns with a system of 4 linear equations,

$$x_1^{'} = x_1 cos\theta - y_1 sin\theta + dx \tag{C.5}$$

$$y_1^{'} = y_1 cos\theta + x_1 sin\theta + dy \tag{C.6}$$

$$x_2^{'} = x_2 cos\theta - y_2 sin\theta + dx \tag{C.7}$$

$$y_2^{'} = y_2 cos\theta + x_2 sin\theta + dy \tag{C.8}$$

From C.5,

$$x_1^{'} = x_1 cos\theta - y_1 sin\theta + dx \quad \text{(C.9)}$$
$$y_1 sin\theta = x_1 cos\theta + dx - x_1^{'} \quad \text{(C.10)}$$
$$sin\theta = \frac{x_1 cos\theta + dx - x_1^{'}}{y_1} \quad \text{(C.11)}$$

From C.6,

$$y_1^{'} = y_1 cos\theta + x_1 sin\theta + dy \quad \text{(C.12)}$$
$$x_1 sin\theta = y_1^{'} - y_1 cos\theta - dy \quad \text{(C.13)}$$
$$sin\theta = \frac{y_1^{'} - y_1 cos\theta - dy}{x_1} \quad \text{(C.14)}$$

From C.7,

$$x_2^{'} = x_2 cos\theta - y_2 sin\theta + dx \quad \text{(C.15)}$$
$$y_2 sin\theta = x_2 cos\theta + dx - x_2^{'} \quad \text{(C.16)}$$
$$sin\theta = \frac{y_2^{'} - y_2 cos\theta - dy}{x_2} \quad \text{(C.17)}$$

From C.8,

$$y_2^{'} = y_2 cos\theta + x_2 sin\theta + dy \quad \text{(C.18)}$$
$$x_2 sin\theta = y_2^{'} - y_2 cos\theta - dy \quad \text{(C.19)}$$
$$sin\theta = \frac{y_2^{'} - y_2 cos\theta - dy}{x_2} \quad \text{(C.20)}$$

C.11 = C.17,

$$\frac{x_1 cos\theta + dx - x_1^{'}}{y_1} = \frac{x_2 cos\theta + dx - x_2^{'}}{y_2} \quad \text{(C.21)}$$
$$x_1 y_2 cos\theta + y_2 dx - y_2 x_1^{'} = x_2 y_1 cos\theta + y_1 dx - x_2^{'} y_1 \quad \text{(C.22)}$$
$$(x_1 y_2 - x_2 y_1)\, cos\theta = y_1 dx - x_2^{'} y_1 + x_1^{'} y_2 - y_2 dx \quad \text{(C.23)}$$
$$cos\theta = \frac{y_1 dx - x_2^{'} y_1 + x_1^{'} y_2 - y_2 dx}{x_1 y_2 - x_2 y_1} \quad \text{(C.24)}$$

C.14 = C.20,

$$\frac{y_1^{'} - y_1 cos\theta - dy}{x_1} = \frac{y_2^{'} - y_2 cos\theta - dy}{x_2} \quad \text{(C.25)}$$

$$x_2 y_2^{'} - x_2 y_1 cos\theta - x_2 dy = x_1 y_2^{'} - x_1 y_2 cos\theta - x_1 dy \quad \text{(C.26)}$$

$$(x_1 y_2 - x_2 y_1)\, cos\theta = x_1 y_2^{'} - x_1 dy + x_2 dy - x_2 y_1^{'} \quad \text{(C.27)}$$

$$cos\theta = \frac{x_1 y_2^{'} - x_1 dy + x_2 dy - x_2 y_1^{'}}{x_1 y_2 - x_2 y_1} \quad \text{(C.28)}$$

C.24 = C.28,

$$y_1 dx - x_2^{'} y_1 + x_1^{'} y_2 - y_2 dx = x_1 y_2^{'} - x_1 dy + x_2 dy - x_2 y_1^{'} \quad \text{(C.29)}$$

$$(y_1 - y_2) dx = dy(x_2 - x_1) + x_1 y_2^{'} - x_2 y_1^{'} + x_2^{'} y_1 - x_1^{'} y_2 \quad \text{(C.30)}$$

$$dx = \frac{dy(x_2 - x_1) + x_1 y_2^{'} - x_2 y_1^{'} + x'_2 y_1 - x_1^{'} y_2}{y_1 - y_2} \quad \text{(C.31)}$$

From C.5,

$$x_1^{'} = x_1 cos\theta - y_1 sin\theta + dx \quad \text{(C.32)}$$

$$x_1 cos\theta = x_1^{'} + y_1 sin\theta - dx \quad \text{(C.33)}$$

$$cos\theta = \frac{x_1^{'} + y1 sin\theta - dx}{x_1} \quad \text{(C.34)}$$

From C.6,

$$y_1^{'} = y_1 cos\theta + x_1 sin\theta + dy \quad \text{(C.35)}$$

$$y_1 cos\theta = y_1^{'} - x_1 sin\theta - dy \quad \text{(C.36)}$$

$$cos\theta = \frac{y_1^{'} - x_1 sin\theta - dy}{y_1} \quad \text{(C.37)}$$

From C.7,

$$x_2^{'} = x_2 cos\theta - y_2 sin\theta + dx \quad \text{(C.38)}$$

$$x_2 cos\theta = x_2^{'} + y_2 sin\theta - dx \quad \text{(C.39)}$$

$$cos\theta = \frac{x_2^{'} + y_2 sin\theta - dx}{x_2} \quad \text{(C.40)}$$

From C.8,

$$y_2^{'} = y_2 cos\theta + x_2 sin\theta + dy \quad (C.41)$$

$$y_2 cos\theta = y_2^{'} - x_2 sin\theta - dy \quad (C.42)$$

$$cos\theta = \frac{y_2^{'} - x_2 sin\theta - dy}{y_2} \quad (C.43)$$

C.34 = C.40,

$$\frac{x_1^{'} + y_1 sin\theta - dx}{x_1} = \frac{x_2^{'} + y_2 sin\theta - dx}{x_2} \quad (C.44)$$

$$x_1^{'} x_2 + x_2 y_1 sin\theta - x_2 dx = x_1 x_2^{'} + x_1 y_2 sin\theta - x_1 dx \quad (C.45)$$

$$(x_2 y_1 - x_1 y_2) sin\theta = x_1 x_2^{'} - x_1 dx + x_2 dx - x_1^{'} x_2 \quad (C.46)$$

$$sin\theta = \frac{x_1 x_2^{'} - x_1 dx + x_2 dx - x_1^{'} x_2}{x_2 y_1 - x_1 y_2} \quad (C.47)$$

C.37 = C.43,

$$\frac{y_2^{'} - x_1 sin\theta - dy}{y_1} = \frac{y_2^{'} - x_2 sin\theta - dy}{y_2} \quad (C.48)$$

$$y_1^{'} y_2 - x_1 y_2 sin\theta - y_2 dy = y_1 y_2^{'} - x_2 y_1 sin\theta - y_1 dy \quad (C.49)$$

$$(x_2 y_1 - x_1 y_2) sin\theta = y_1 y_2^{'} - y_1 dy + y_2 dy - y_1^{'} y_2 \quad (C.50)$$

$$sin\theta = \frac{y_1 y_2^{'} - y_1 dy + y_2 dy - y_1^{'} y_2}{x_2 y_1 - x_1 y_2} \quad (C.51)$$

C.47 = C.51,

$$x_1 x_2^{'} - x_1 dx + x_2 dx - x_1^{'} x_2 = y_1 y_2^{'} - y_1 dy + y_2 dy - y_1^{'} y_2 \quad (C.52)$$

$$(x_2 - x_1) dx = dy(y_2 - y_1) + y_1 y_2^{'} - y_1^{'} y_2 + x_1^{'} x_2 - x_1 x_2^{'} \quad (C.53)$$

$$dx = \frac{dy(y_2 - y_1) + y_1 y_2^{'} - y_1^{'} y_2 + x_1^{'} x_2 - x_1 x_2^{'}}{x_2 - x_1} \quad (C.54)$$

C.31 = C.54,

$$\frac{dy(x_2 - x_1) + x_1 y_2^{'} - x_2 y_1^{'} + x_2^{'} y_1 - x_1^{'} y_2}{y_1 - y_2} = \frac{dy(y_2 - y_1) + y_1 y_2^{'} - y_1^{'} y_2 + x_1^{'} x_2 - x_1 x_2^{'}}{x_2 - x_2} \quad (C.55)$$

$$\text{LHS: } dy(x_2^2 - 2x_1 x_2 + x_1^2) + [x_1 y_2^{'} - x_2 y_1^{'} + x_2^{'} y_1 - x_1^{'} y_2](x_2 - x_1) \quad (C.56)$$

$$\text{RHS: } dy(y_2 y_1 - y_2^2 - y_1^2 + y_2 y_1) + [y_1 y_2^{'} - y_1^{'} y_2 + x_1^{'} x_2 - x_1 x_2^{'}](y_1 - y_2) \quad (C.57)$$

$$dy = \frac{[y_1 y_2^{'} - y_1^{'} y_2 + x_1^{'} x_2 - x_1 x_2^{'}](y_1 - y_2) - [x_1 y_2^{'} - x_2 y_1^{'} + x_2^{'} y_1 - x_1^{'} y_2](x_2 - x_1)}{x_2^2 - 2x_1 x_2 + x_1^2 + y_2^2 - 2y_1 y_2 + y_1^2} \quad (C.58)$$

D

Modelling the State Transition Probabilities and Likelihood Voting Scheme

As highlighted in Section 5.4.2, there are several major differences in terms of the system characteristics when comparing our system with the one proposed by Angeli et al. (2008). These differences are listed as follows,

1. Haar coefficients of omnidirectional images were used to discriminate between the images instead of the bag-of-visual words approach using perspective cameras.
2. Omnidirectional images were captured intermittently and only a single omnidirectional image is associated with a node in the topological map instead of using a continuous stream of video images
3. Topological relationship between these images were maintained in a bidirectional graph structure

These differences in system characteristics have major implications to how the state transition probabilities and likelihood voting scheme were modelled. Due to (1), the virtual image, which is originally created using the occurrences of words available in its evolving dictionary, cannot be created the same way when Haar coefficients were used. The original idea for having this virtual image to represent all unmapped locations is such that it is statistically more likely for this virtual image to match the incoming query image if the mobile robot is currently at an unexplored location. For our system, the score of the virtual image is calculated by subtracting the mean of all scores, μ_a, with G times the standard deviation, σ which is expressed in Equation 5.19.

Due to (2) and (3), our system violates the assumption of the original framework, which can be visualized through the example illustrated in Fig. D.1. According to the original framework, all images from $I_0 ... I_8$ are retained and compared against, resulting in a duplicate at I_0/I_8. This assumption is violated in our system since a unique image is associated with each node in the topological map. For the original framework, I_5 and I_4 can always be assumed as adjacent locations of I_6 since an image sequence was used. However, in our system, adjacent locations of I_7 are defined by the connecting nodes at the end of links originating from I_6.

Finally, the combination of (1), (2) and (3) results in the complete redefinition of the state transition probabilities and likelihood voting scheme since the scores returned by the place recognition system using Haar coefficients differed from the bag-of-visual words approach and the way

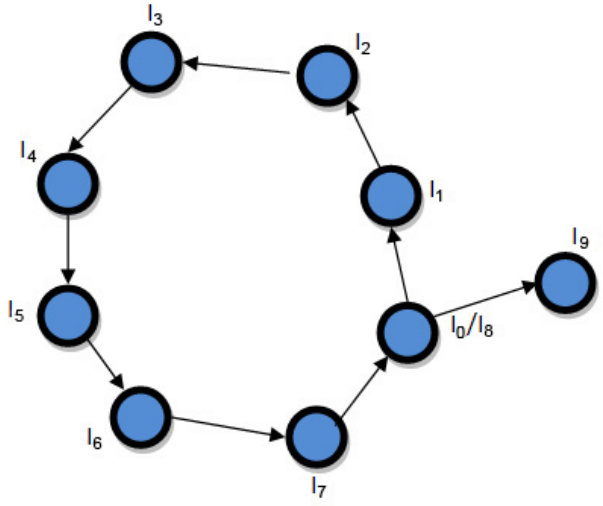


Figure D.1: Trajectory of an Image Sequence

that the system was capturing images intermittently affected the convergence speed. The following describes how the state transition probabilities and likelihood voting scheme were modelled and how it ended up with the expressions in Section 5.4.2.

D.1 State Transition Model

The original state transition model as defined by Angeli et al. (2008) was modelled according to the following expressions,

- $p(S_t = -1|S_{t-1} = -1) = 0.9$ - the probability that no loop closure has occurred at time t is high, given that none has occurred at time $t-1$
- $p(S_t = j|S_{t-1} = -1) = \frac{0.1}{(t-e)+1.0}$ with $i \in [0, t-e]$ - the probability that loop closure is low at time t, given that none has occurred at time $t-1$
- $p(S_t = -1|S_{t-1} = k) = 0.1$ with $k \in [0, t-e]$ - the probability that no loop closure is low at time t, given that loop closure has occurred at time $t-1$
- $p(S_t = j|S_{t-1} = k)$ = a Gaussian on the distance between j and k whose sigma value is chosen so that it is non-zero for exactly 4 neighbours (i.e. $j = k-2, ..., k+2$) with i and $j \in [0, t-e]$.

As described previously, our system captures images intermittently and each node in the topological map is associated with a unique omnidirectional image and appearance model. This implies that for $p(S_t = j|S_{t-1} = k)$, the neighbours of the node at index j are not $j = k-2, ..., k+2$ but rather nodes that are connected to the node at index j defined by the links in the topological map. In addition, the probabilities defined for different state transitions in the original framework affects the convergence speed. The likelihood voting scheme was also redefined in order to shorten the total time required for convergence. This will be described after the new state transition model is presented.

The new state transition model is expressed as,

$$p(S_t = -1|S_{t-1} = -1) = P_{nc}^{S_{t-1}=-1} \quad \text{(D.1)}$$

$$p(S_t = j|S_{t-1} = -1) = \frac{P_c^{S_{t-1}=-1}}{t-e+1.0} \text{ with } j \in [0, t-e] \quad \text{(D.2)}$$

$$p(S_t = -1|S_{t-1} = k) = P_c^{S_t=-1} \text{ with } k \in [0, t-e] \quad \text{(D.3)}$$

$$p(S_t = j|S_{t-1} = k) = \begin{cases} P_{nc,m}^{S_t=-1} & : \text{ if } j = k \\ p_{nc,1}^{S_t=-1} & : j \in \text{all neighbours of node } k \end{cases} \quad \text{(D.4)}$$

where

$$P_{nc}^{S_{t-1}=-1} = \frac{A}{(t-e+A)} \begin{cases} A = 2.0 & : SI = -1 \\ A = 1.5 & : SI > -1 \end{cases} \quad \text{(D.5)}$$

$$P_c^{S_{t-1}=-1} = 1 - P_{nc}^{S_{t-1}=-1} \quad \text{(D.6)}$$

$$P_{nc}^{S_t=-1} = \begin{cases} \frac{N_k+1}{N_k+2} & : SI = -1 \\ 0.2 & : SI > -1 \end{cases} \quad \text{(D.7)}$$

$$P_c^{S_t=-1} = 1 - P_{nc}^{S_t=-1} \quad \text{(D.8)}$$

$$P_{nc,m}^{S_t=-1} = P_{nc}^{S_t=-1} * \frac{C}{N_k + C} \quad \text{(D.9)}$$

$$P_{nc,1}^{S_t=-1} = \frac{P_{nc}^{S_t=-1} - P_{nc,m}^{S_t=-1}}{N_k} \quad \text{(D.10)}$$

where N_k is the total number of neighbours of node k, C is a constant set at 2.0 and SI is the selected index (largest probability) of the prior posterior, $p(S_{t-1}|I^{t-1})$. These expressions have several implications. Firstly, $p(S_t = -1|S_{t-1} = -1) \approx Ap(S_t = 0|S_{t-1} = -1)$ given that t is sufficiently large (depending on the value of e) while still ensuring that the condition $p(S_t \geq -1|S_{t-1} = -1) = 1.0$ remains true. Similarly, $P_{nc,m}^{S_t=-1} = CP_{nc,1}^{S_t=-1}$ while the condition, $p(S_t \geq -1|S_{t-1} = k) = 1.0$, remains true.

D.2 Likelihood Voting Scheme

In the original likelihood voting scheme defined by Angeli et al. (2008), a subset of images, $H_t \subseteq I^{t-e}$, is selected if $\frac{D_i-\mu_a}{\mu_a}$ (coefficient of variation - c.o.v.) is greater than $\frac{\sigma}{\mu}$ (standard c.o.v.). The belief at time t, $p(S_t|I^{t-1})$, is then multiplied by the difference between the particular c.o.v. of I_i with the standard c.o.v., plus 1. Thus, the likelihood is expressed as,

$$\mathcal{L}(S_t|I_t) = \begin{cases} \frac{D_i-\mu_a}{\mu_a} - \frac{\sigma}{\mu_a} + 1 = \frac{D_i-\sigma}{\mu_a} & : S_i \geq \mu_a + \sigma \\ 1.00 & : otherwise \end{cases} \quad \text{(D.11)}$$

where μ_a, D_i and σ denotes the mean of the scores, the score of the image and standard deviation of all scores. However, by taking a closer look at $p(S_t = j|S_{t-1} = -1)$, which defines the probability that the mobile robot is detecting a previously visited node at time t, given that none occurred at time $t-1$ in the state transition model, reveals that its probability reduces as the number of nodes in the topological map increases as illustrated in Fig. D.2 (assuming that A is fixed at a value of 2.0). The total time it takes the original likelihood voting scheme to converge is already slow due to the way that images were captured intermittently on our system. However, the effect of the

reduction in probability for this particular state transition as the total number of nodes increases in the topological map indicates that the mobile robot is required to proceed with navigation and remain in a previously visited area for a longer period of time before it can build up its belief and transits from no loop closure at time $t-1$ to loop closure at time t. In order to reduce the total convergence time, the likelihood, $\mathcal{L}(S_t|I_t)$, is computed by finding a subset $H_t \subseteq I^{t-e}$ of images whose score, D_i $(i \in [-1, t-e]))$, is smaller than the threshold computed using the mean of all scores, μ_a, minus its standard deviation, σ (smaller Haar scores representing a better match). At the same time, the mean of all inliers, μ_{in}, which represents the mean score of those which is larger than the threshold, is computed. The likelihood function is redefined as,

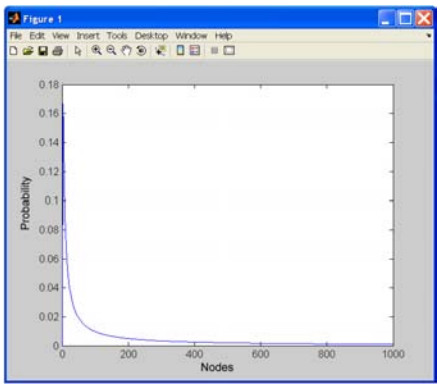


Figure D.2: Probability of $p(S_t = j|S_{t-1} = -1)$ as Total Nodes in Map Increases

$$\mathcal{L}(S_t|I_t) = \begin{cases} \frac{(D_i - \mu_{in})^B + \mu_{in}}{\mu_{in}} & : D_i \leq \mu_a - \sigma \\ 1.00 & : otherwise \end{cases} \tag{D.12}$$

Equation D.12 basically amplifies the difference between the score of an image with respect to the mean score of all inliers by varying the power factor B, if $D_i \leq \mu_a - \sigma$ is true. The value of the power factor B is modelled according to the probability of the state transition, $p(S_t = j|S_{t-1} = -1)$, as the total nodes increases in the topological map. Given that if $D_i - \mu_{in} >= E$ is considered a discriminative score for a matching node and also that $E^B p(S_t = j|S_{t-1} = -1) = F$, then B can be expressed as,

$$B = \frac{log(F/p(S_t = j|S_{t-1} = -1))}{log(E)} - H \tag{D.13}$$

where H is basically an offset of the power factor B when a loop closure was detected in the previous time step. If the loop closure detected was indeed a true event, there is a higher probability for new locations in the vicinity exhibiting similar appearance. Due to the previously detected loop closure, there is also a higher probability for the system to be matching to an existing node in the map (e.g. possibly neighbour of the previously detected loop closure) given sufficient similarity in the location's visual appearance. As such, this offset ensures that false positives are reduced for such scenarios and a discriminative matching score is required for the system to remain in the state of loop closure detected from the previous to the current time step. H is currently defined as,

$$H = \begin{cases} 0.3 & \text{:j=-1 and SI>-1} \\ 0 & \text{: otherwise} \end{cases} \tag{D.14}$$

In Equation D.13, the value of E defines what is deemed as a highly discriminative score for a matching node and the combination of E and F affects the convergence speed of this framework. The response of B when $E = F = 15.0$ and with $SI = -1$ or $j > -1$ is as shown in Fig. D.3. The power factor B varies according to the total nodes in the topological map such that consistent amplification is performed. As expressed in the new state transition model, the value of A changes from 2.0 to 1.5 and $P_{nc}^{S_t=-1}$ becomes 0.2 when loop closure was detected for the previous time step. Similarly, the value of B is subtracted with 0.3 when loop closure was detected for the previous time step. The justification behind this is such that the state transition from loop closure at time $t-1$ to no loop closure at time t converges quicker in order to avoid producing false positives. For this system to remain at the state of loop closure from time $t-1$ to t will require a decent score for the input query image at time t with the images in the database. Since this is a previously visited location, this is naturally being taken care of as the input query image will definitely be highly similar with an existing image in the database. Finally, the score of the virtual image is calculated as,

$$\mathcal{L}(S_t = -1|I_t) = \mu_a - G\sigma \tag{D.15}$$

which makes it statistically more likely for this virtual image to match the incoming query image if the mobile robot is currently at an unexplored location. To the best of my knowledge, this is the first work to incorporate the use of Haar coefficients as the appearance model into a probabilistic framework for a place recognition system. Nevertheless, there is still room for further validation and refinement of the proposed model.

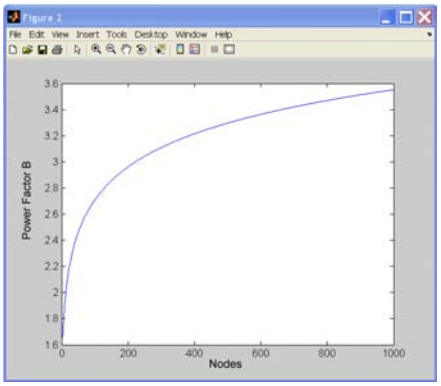
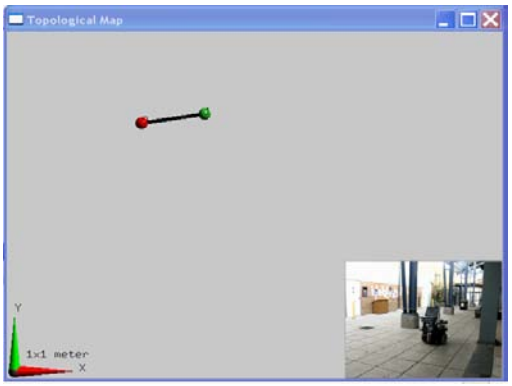


Figure D.3: The response of B from 0 to 1000 nodes

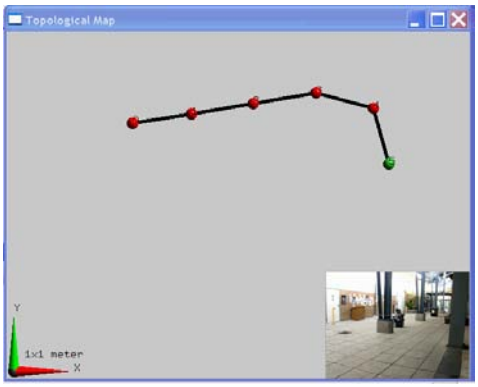
E

Supplementary Experimental Results

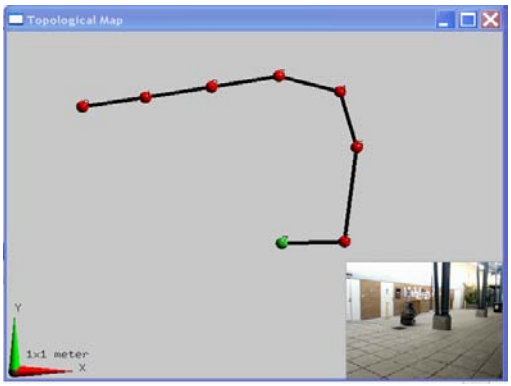
Supplementary Experiment 1 This is a simple experiment showing the mobile robot switching from the pure exploration to the loop closing and exploration mode. The experiment ended at $t = 13$ when the mobile robot detected a loop closure. The resulting topological map was properly scaled, rotated and superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. E.2. The matching omnidirectional image pair which raises the loop closing event at $t = 13$ is illustrated in Fig. E.3.



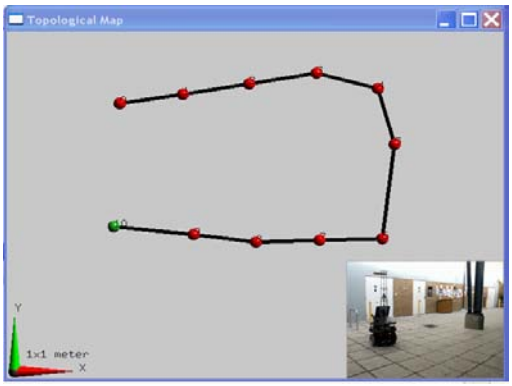
(a) t=1



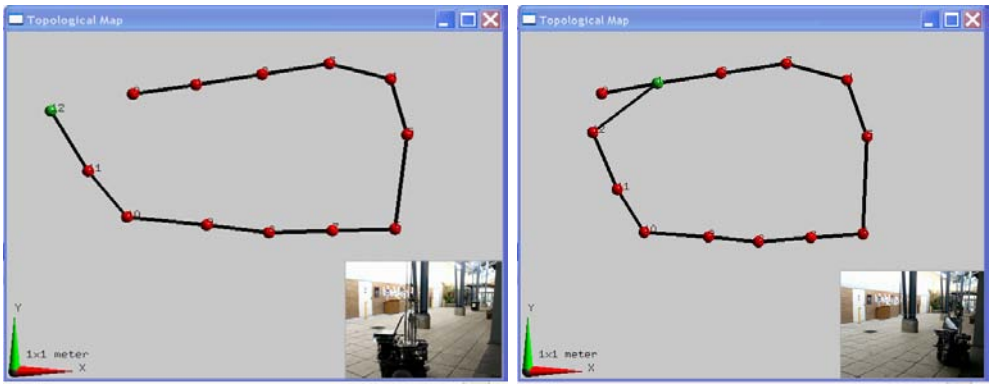
(b) t=5



(c) t=7



(d) t=10



(e) t=12 (f) t=13

Figure E.1: Supplementary Semi Outdoor Experiment 1

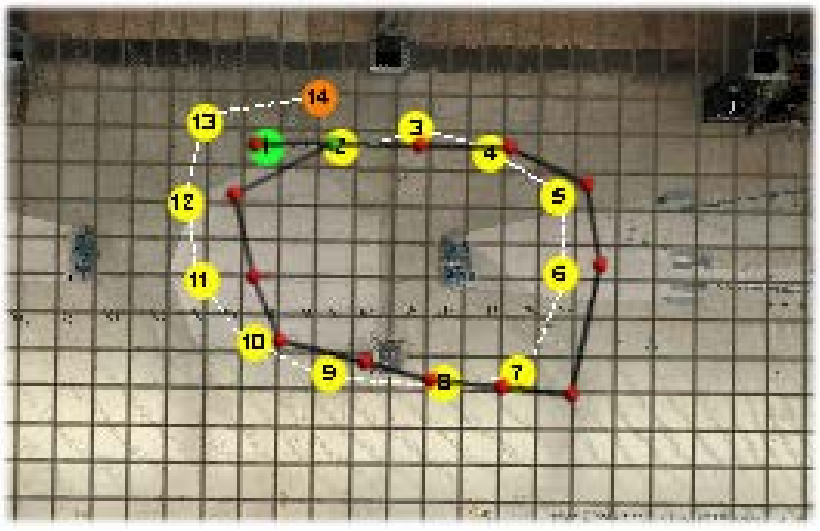
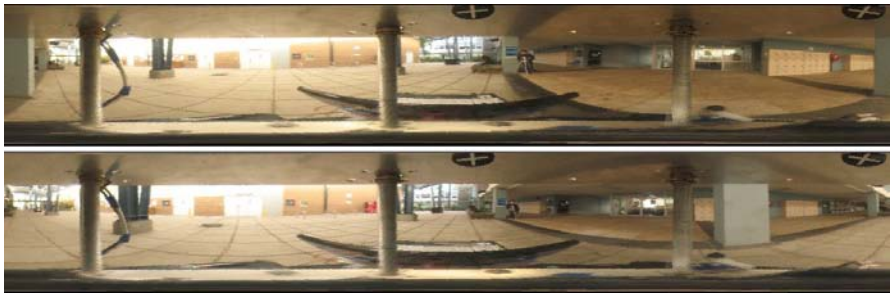


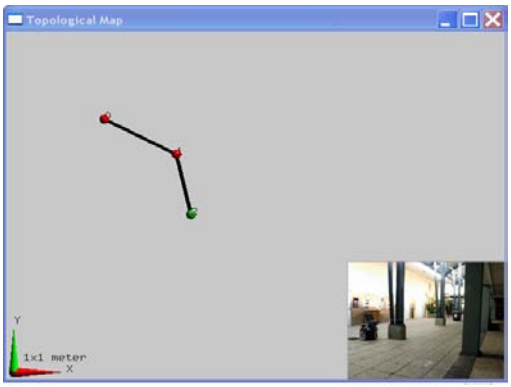
Figure E.2: Supplementary Semi Outdoor Exp.1 - Comparing Against Ground Truth



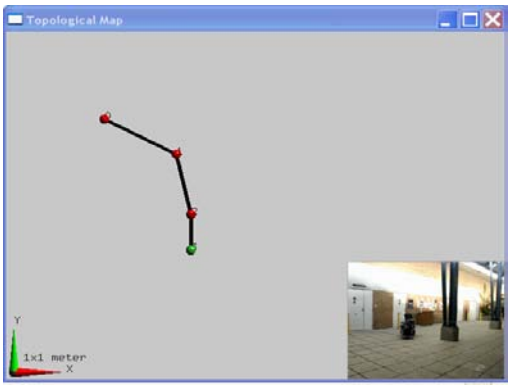
(a) t=13

Figure E.3: Matching Omnidirectional Image Pairs for Supplementary Semi-Outdoor Exp. 1 (Loop Closure Detection)

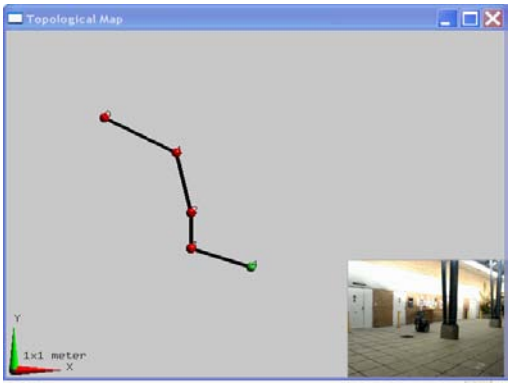
Supplementary Experiment 2 In this experiment, the mobile robot detected a loop closure at $t = 5$ but was further invalidated by a subsequent loop closure event at $t = 6$ that prompted system restoration. After system restoration, the mobile robot detected loop closures at $t = 10$ and $t = 17$. It proceeded with exploration, detecting a few more loop closures on the way until it returned to node 13 in the topological map at $t = 29$. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. E.5. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. E.6. The following are node pairs (node numbers based on the ground truth trajectory) which raise loop closure events in this experiment: 7-5, 9-5, 11-6, 12-10, 18-2, 19-17, 21-17, 23-17, 24-20, 29-26, 30-24.



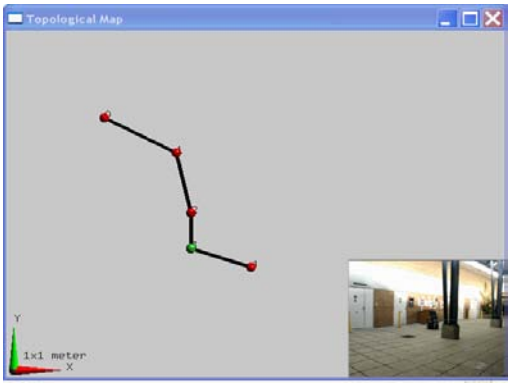
(a) t=2



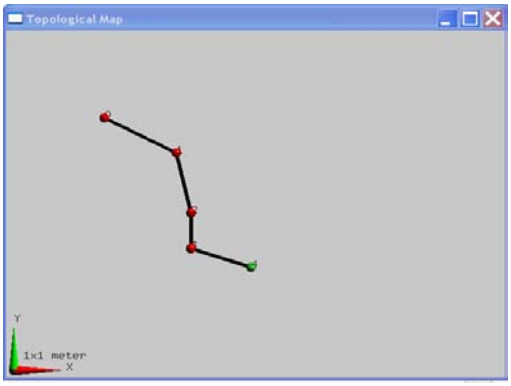
(b) t=3



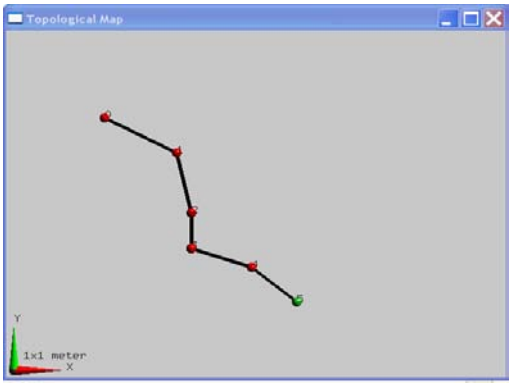
(c) t=4



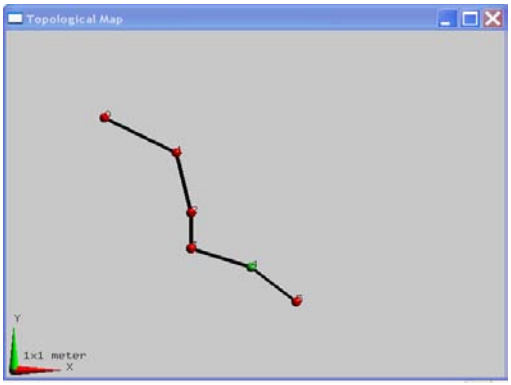
(d) t=5



(e) t=6 (Recovery)

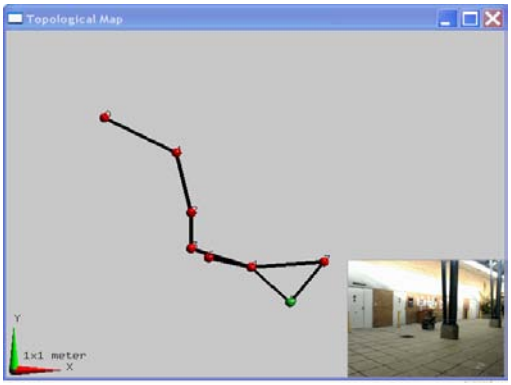


(f) Recovery

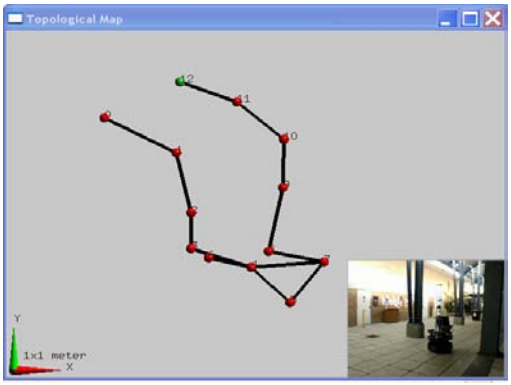


(g) Recovery

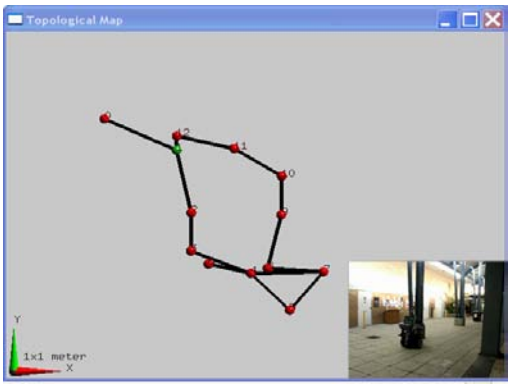
(h) t=9



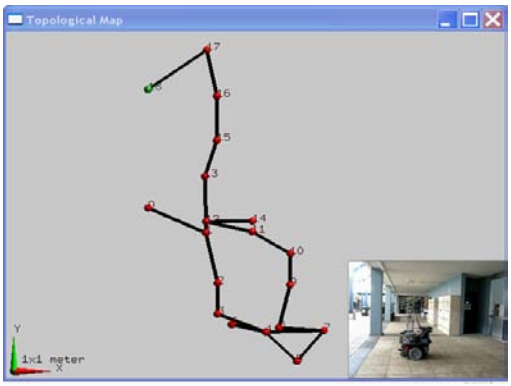
(i) t=10



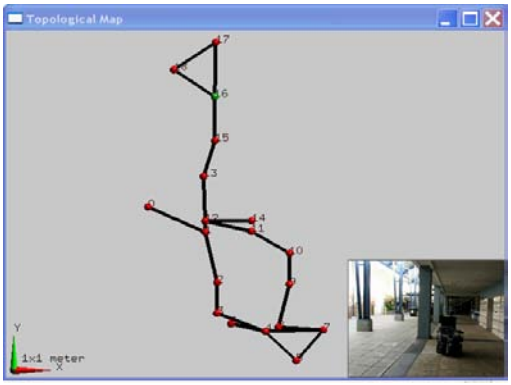
(j) t=16



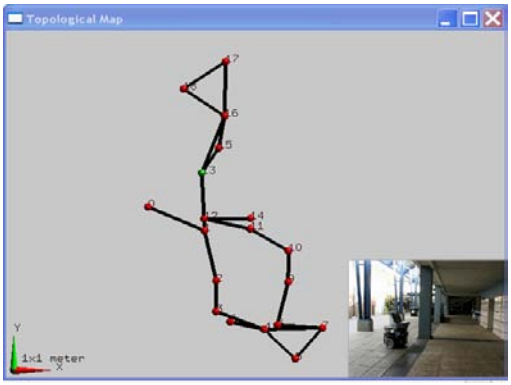
(k) t=17



(l) t=27



(m) t=28



(n) t=29

Figure E.4: Supplementary Semi Outdoor Experiment 2

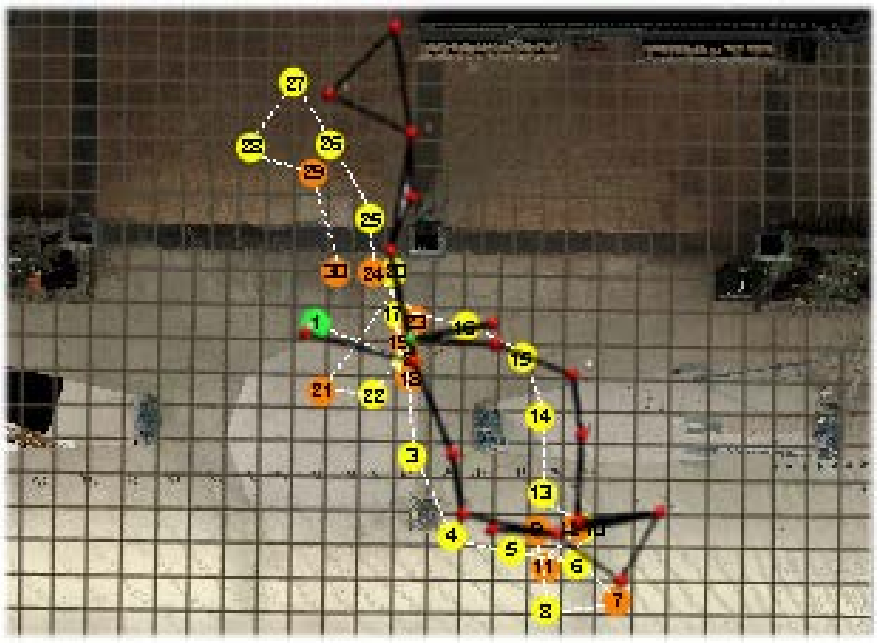


Figure E.5: Supplementary Semi Outdoor Exp.2 - Comparing Against Ground Truth

(a) t=10

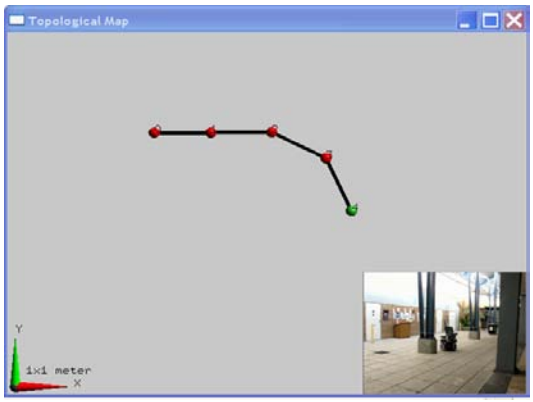
(b) t=17

(c) t=28

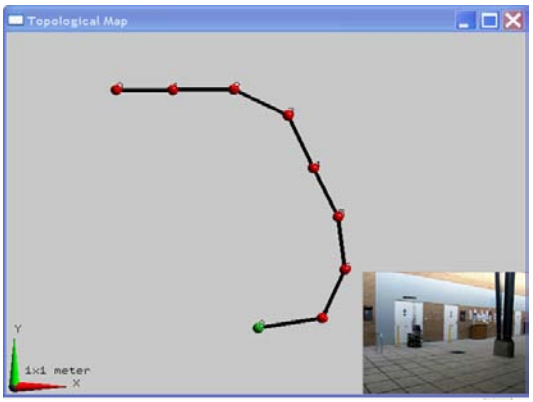
(d) t=29

Figure E.6: Matching Omnidirectional Image Pairs for Supplementary Semi-Outdoor Exp. 2 (Loop Closure Detection)

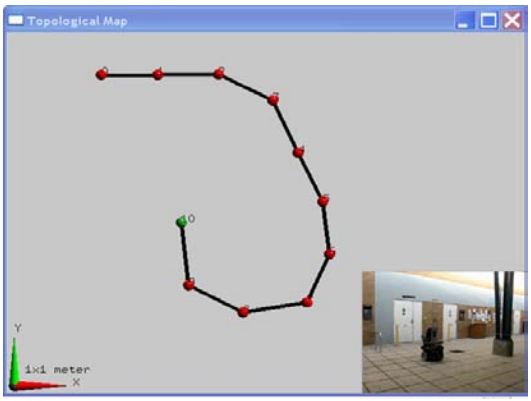
Supplementary Experiment 3 The mobile robot travelled in a loop-like trajectory until it made a detour after $t = 11$ but managed to get back on track to close the loop at $t = 17$ (on the way detecting loop closures at $t = 13 - 15$). From there, it wandered off for further exploration. The scale and rotationally corrected topological map was superimposed onto the plan view of the stitched laser scans with ground truth as shown in Fig. E.8. A subset of matching omnidirectional image pairs which raises loop closure events are provided in Fig. E.9. The following are node pairs (node numbers based on the ground truth trajectory) which raise loop closure events in this experiment: 14-11, 16-11, 18-4.



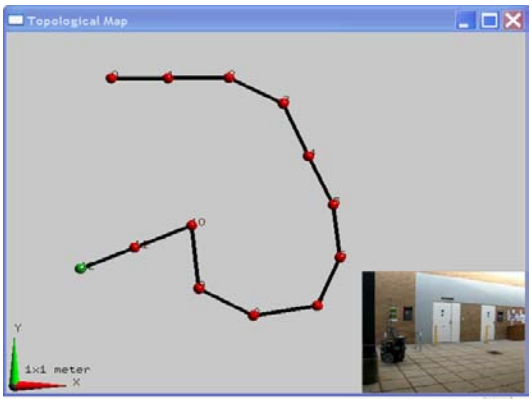
(a) t=4



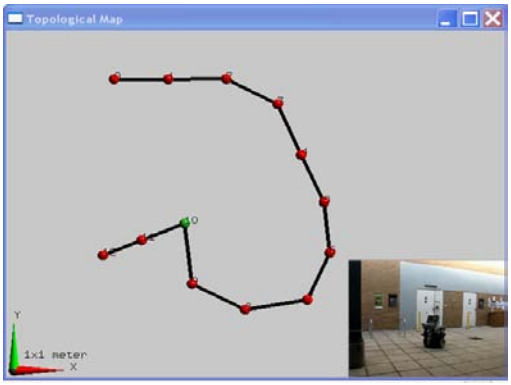
(b) t=8



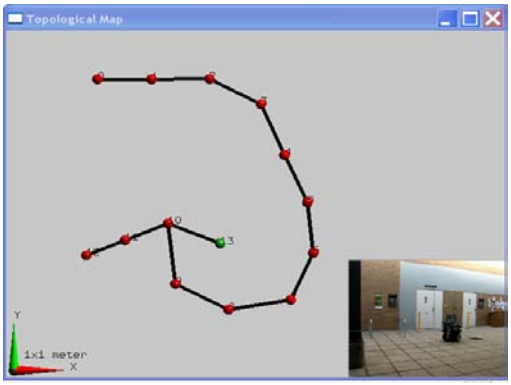
(c) t=10



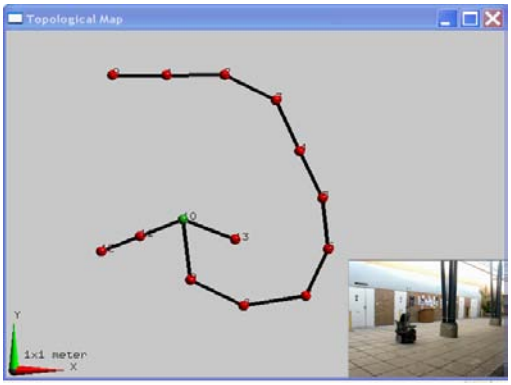
(d) t=12



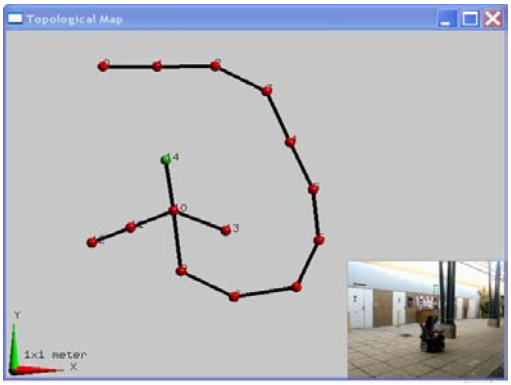
(e) t=13



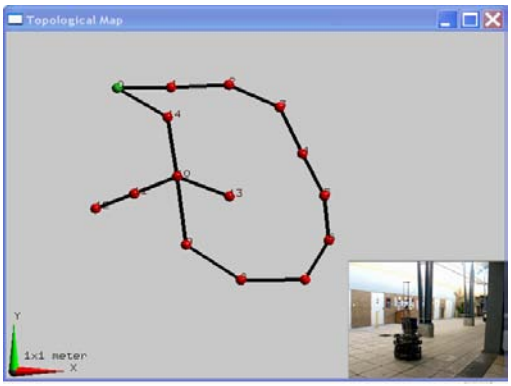
(f) t=14



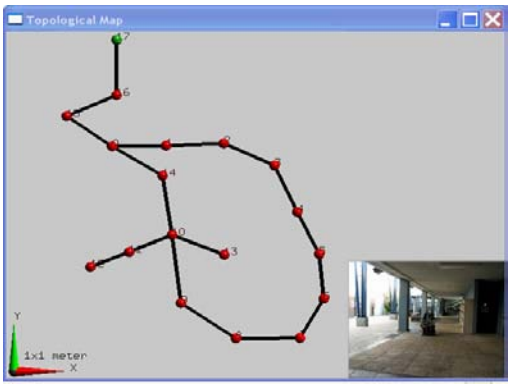
(g) t=15



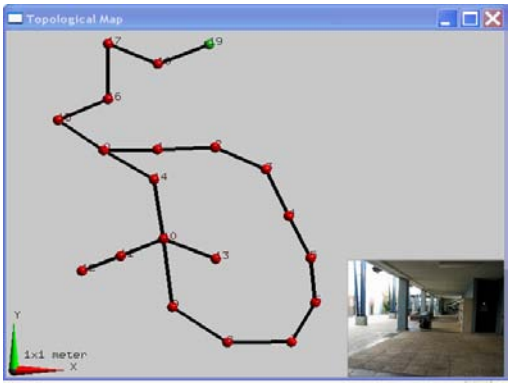
(h) t=16



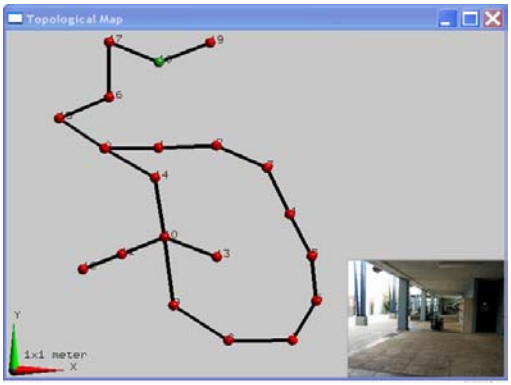
(i) t=17



(j) t=20



(k) t=22



(l) t=23

Figure E.7: Supplementary Semi Outdoor Experiment 3

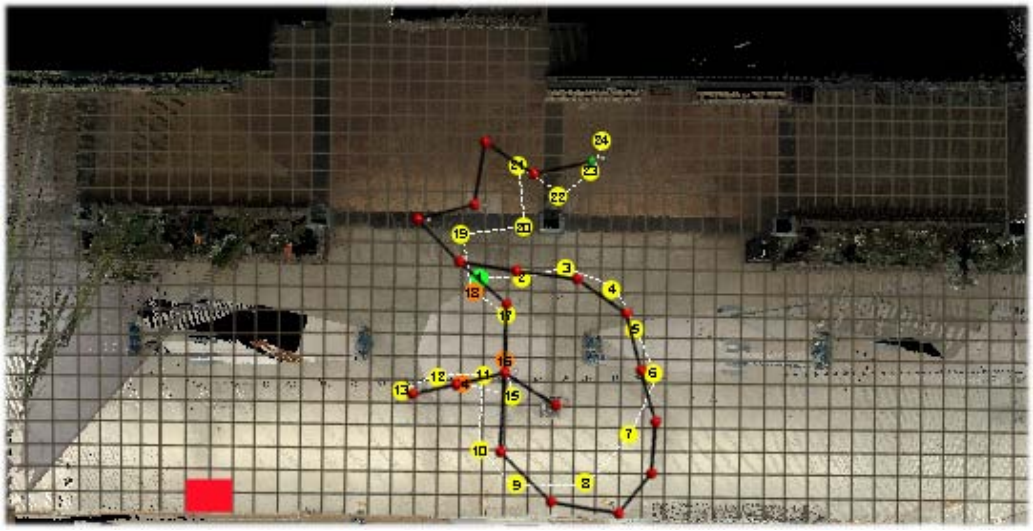
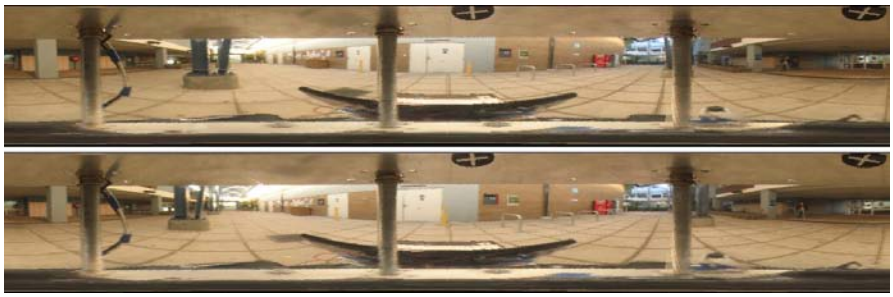


Figure E.8: Supplementary Semi Outdoor Exp.3 - Comparing Against Ground Truth

(a) t=13



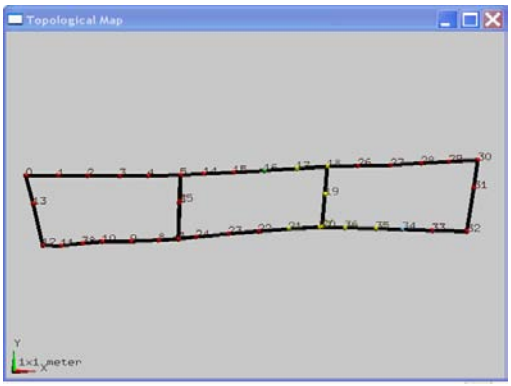
(b) t=15

(c) t=17

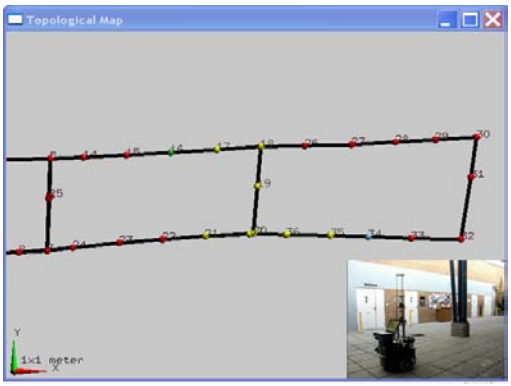
(d) t=23

Figure E.9: Matching Omnidirectional Image Pairs for Supplementary Semi-Outdoor Exp. 3 (Loop Closure Detection)

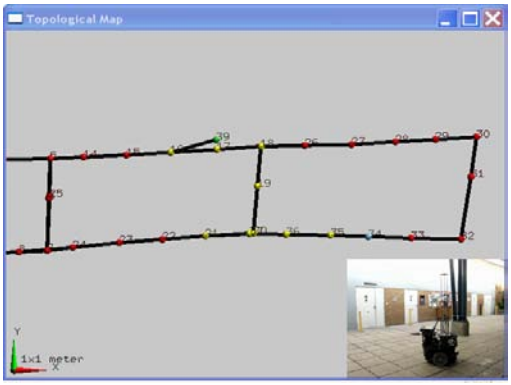
Supplementary Experiment 4 (Semi-Autonomous) Similar to the other experiments in Section 6.7.3, the map created by offline experiment 1 was loaded into the system. The starting and goal positions were defined as nodes 16 and 34 respectively. At $t = 1$ and $t = 2$, the mobile robot seems to be trying to execute the planned path. However, it was slightly deviated from the actual position resulting in no matches detected at both locations. At $t = 3$, it detected a loop closure event. The matching omnidirectional image pair can be found in Fig. E.11. Although the match is correct, however, the system was uncertain with its decision to loop closure and was invalidated later at $t = 4$. Nevertheless, it managed to reach its intended target destination at $t = 12$.



(a) Map Loaded

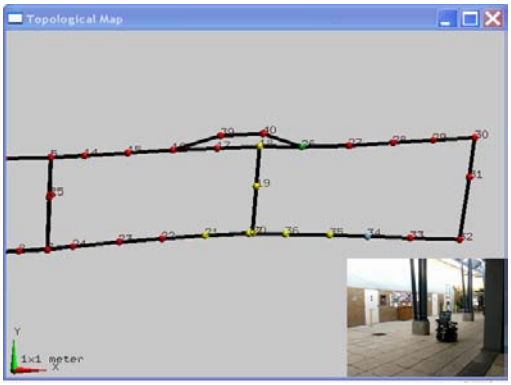


(b) t=0

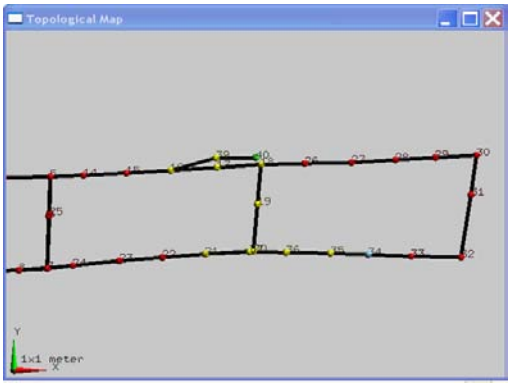


(c) t=1

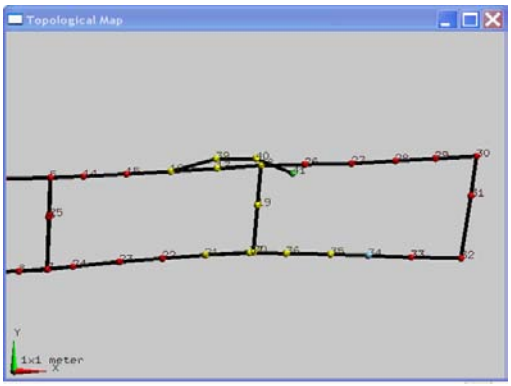
(d) t=2



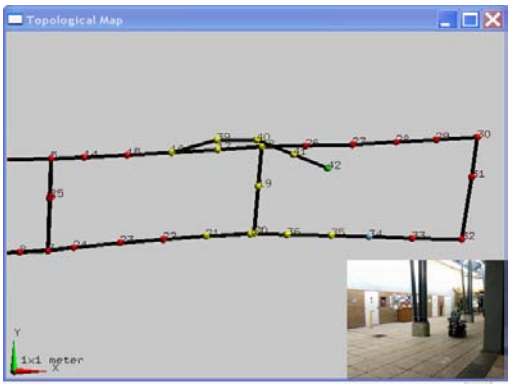
(e) t=3



(f) t=4 (Recovery)

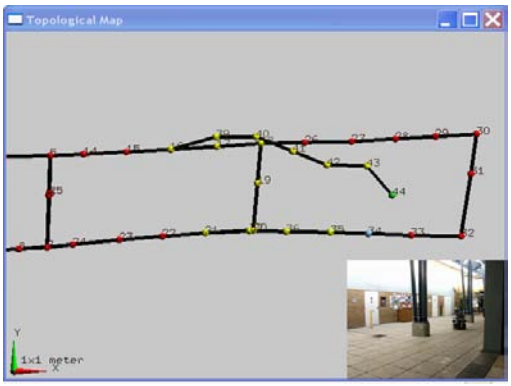


(g) Recovery

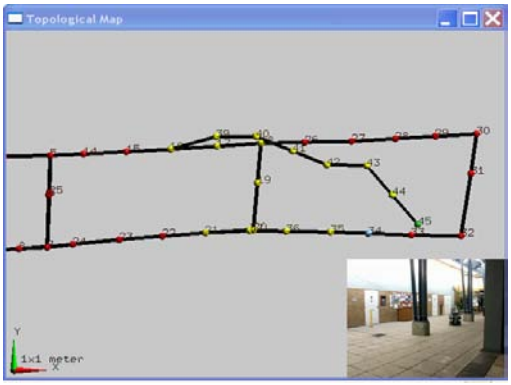


(h) t=5

(i) t=6

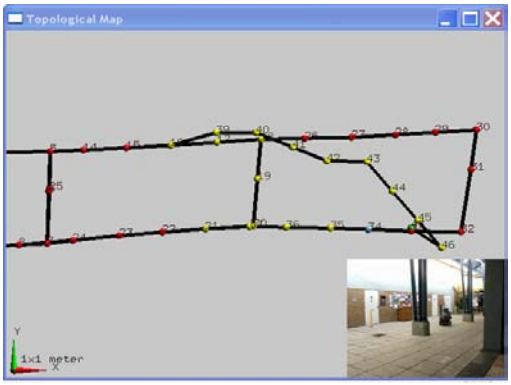


(j) t=7

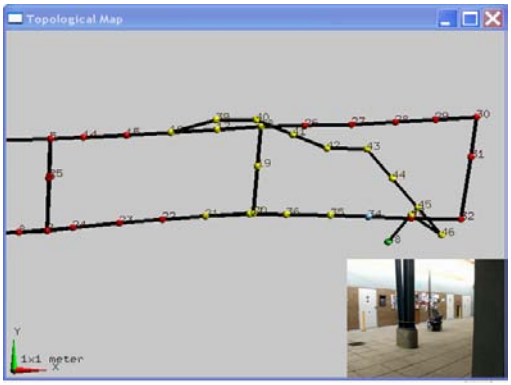


(k) t=8

(l) t=9



(m) t=10

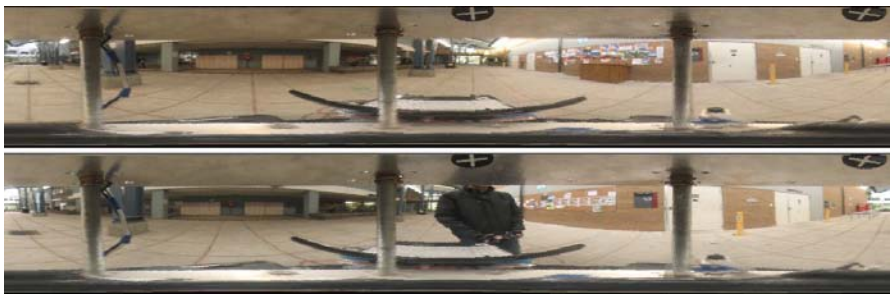


(n) t=11

(o) t=12

Figure E.10: Supplementary Experiment 4 (Semi-Autonomous)

(a) t=3



(b) t=12

Figure E.11: Matching Omnidirectional Image Pairs for Supplementary Experiment 4 (Loop Closure Detection)

F

Ground Plane Detection using Stereovision

The proposed ground plane detection technique was developed based on the work in Agrawal et al. (2007) which detects the ground plane in stereovision data. In this work, the stereovision data returned by the Bumblebee stereovision camera were used for generating plane hypotheses (any 3 non-collinear 3D points define a plane hypothesis). Plane hypothesis that were too slanted were rejected. In addition, the selection of points was biased to those having a Euclidean distance of less than 5 meters from the mobile robot for the generation of plane hypotheses since this is the range where estimated 3D points should be most accurate for the Bumblebee stereovision system. Finally, the best plane hypotheses was determined via a RANSAC procedure which selects the plane that fits the most number of points.

According to (Weisstein, 2010a,b), the plane passing through three 3D points $P_1 = (x_1, y_1, z_1)$, $P_3 = (x_2, y_2, z_2)$ and $P_3 = (x_3, y_3, z_3)$ can be defined as the set of all points (x, y, z) that satisfy the following determinant equations,

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = \begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x - x_2 & y - y_2 & z - z_2 \\ x - x_3 & y - y_3 & z - z_3 \end{vmatrix} = 0 \tag{F.1}$$

This plane can also be described in the form of $ax + by + cx + d = 0$ with the following system of equations,

$$ax_1 + by_1 + cz_1 + d = 0 \tag{F.2}$$
$$ax_2 + by_2 + cz_2 + d = 0 \tag{F.3}$$
$$ax_3 + by_3 + cz_3 + d = 0 \tag{F.4}$$

which can be resolved using Cramer's Rule and basic matrix manipulations,

$$d = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \quad a = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \quad b = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix} \quad c = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{F.5}$$

A collinearity test was conducted on the 3 randomly selected points in order to ensure that these 3 points do not lie on the same straight line. Three or more points $P_1, P_2, P_3, ...$ are said to be collinear if they lie on a straight line L. Three points are collinear if the ratios of the distances satisfy,

$$x_2 - x_1 : y_2 - y_1 : z_2 - z_1 = x_3 - x_1 : y_3 - y_1 : z_3 - z_1 \tag{F.6}$$

A slightly more tractable condition is obtained by noting that the area of a triangle determined by the 3 points is zero if they are collinear (including the degenerate cases of two or all three points being concurrent),

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0 \tag{F.7}$$

or in expanded form,

$$x_1(y_2 - y_3) + x_2(y_3 + y_1) + x_3(y_1 - y_2) = 0 \tag{F.8}$$

The first method based the ratios of distances to determine the collinearity of the 3 randomly selected points was used. By enforcing this condition, the 3 randomly selected points were ensured to generate a valid hypotheses. This plane hypotheses was then subjected to another test to ensure that it is not too slanted. This could be measured using the normal unit vectors of the plane (perpendicular to the plane). The normal vector at a point (x_{0y0}) on a surface $z = f(x, y)$ is given by,

$$N = \begin{bmatrix} f_x(x_0, y_0) \\ f_y(x_0, y_0) \\ -1 \end{bmatrix} \tag{F.9}$$

where $f_x =$ and $f_y =$ are partial derivatives. A normal vector to a plane specified by

$$f(x, y, z) = ax + by + cz + d = 0 \tag{F.10}$$

is given by

$$N = \Delta f = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{F.11}$$

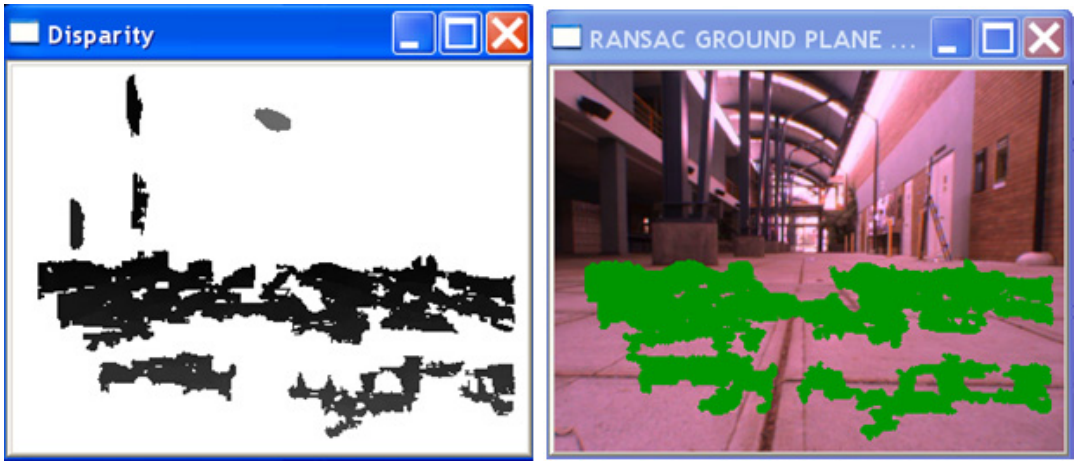
where Δf denotes the gradient. The equation of a plane with a normal vector $n = (a, b, c)$ passing through the point (x_0, y_0, z_0) is given by,

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} = a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \tag{F.12}$$

Since the main objective of this technique was to detect the ground plane in stereovision data, the ideal plane hypothesis should have normal unit vectors of $b = 1$ or $b = -1$, a=0 and c=0. To determine whether a plane is too slanted, thresholds were specified as ± 0.2 from the desired ideal values. The best plane hypothesis was determined using the total number of points fitting this plane hypothesis based on some error function. Given x_1 and z_1 for a point, the height y of this point on the hypothesized plane was calculated and the error, $e = \sqrt{(y_1 - y)^2}$, was then computed. A point would be considered to fit the hypothesized plane if $e < T_e$. This technique can be summarized as,

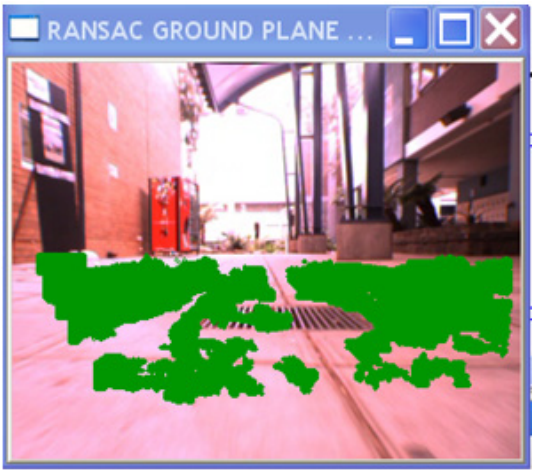
1. Randomly select three 3D points returned by the Bumblebee stereovision system (select those that have a Euclidean distance of less than 5m from the mobile robot.
2. Check for collinearity. Repeat from Step (1) if points are collinear.
3. Generate plane hypothesis. Ensure that plane is not too slanted. Repeat from Step (1) if plane is too slanted.
4. Compute the number of points which fits the plane hypothesis. Store this plane hypothesis if this is the hypothesis which fits the most number of points.
5. If the total number of points for this plane hypothesis is greater than T_n or the total number of iterations i is greater than T_i, return the best plane hypothesis. Else, repeat from Step (1) with $i = i + 1$.

Fig. F.1 shows the detected ground planes in different environments. On the left is the disparity map returned by the Bumblebee stereovision system whereas on the right, the detected ground plane is overlaid on the original image in green. The detected ground plane is similar in shape to parts of the disparity map since we utilize the 3D data from the stereovision system and is restricted to locations in the image where a valid disparity value is available.

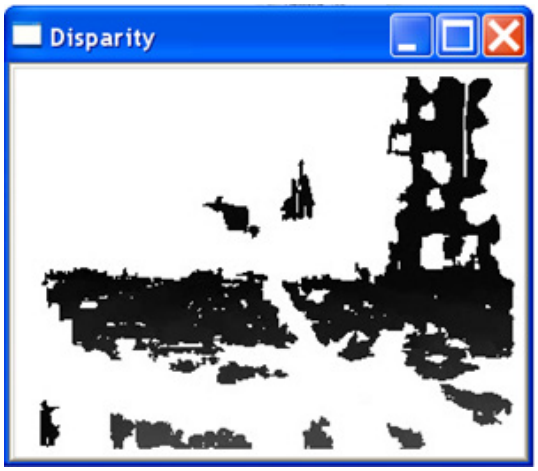


(a) Disparity Map1 (b) Detected Ground Plane1

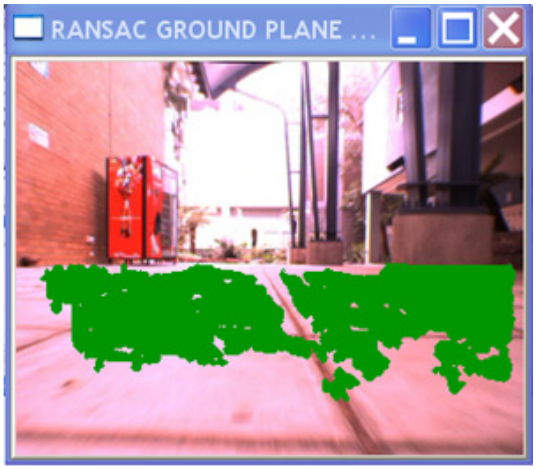
(c) Disparity Map2



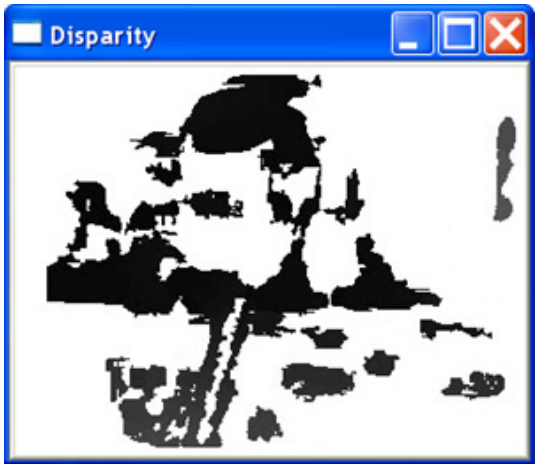
(d) Detected Ground Plane2



(e) Disparity Map3



(f) Detected Ground Plane3



(i) Disparity Map4

(j) Detected Ground Plane4

Figure F.1: Ground Plane Detection Results

G

Global Positioning System (GPS)

G.1 Introduction

In 1996, the NAVSTAR GPS, developed by the US Department of Defence, was the only fully functional Global Navigation Satellite System (GNSS) available for both military and civilian use. At present, China has been expanding its existing regional Beidou-1 system into a GNSS system, referred to as Beidou-2/Compass; Russia has been reviving its GLONASS system for both military and civilian applications, which was once fully functional as a GNSS for a short time in 1995 but was only intended for military applications since its deployment in 1982 and the European Galileo, despite experiencing delays due to a number of factors, has been making progress and is expected to complete in the coming years. For more details on the development of each system, please refer to (Hegarty and Charte, 2008; Sweet, 2008; Chiang et al., 2010). At the time of writing, NAVSTAR GPS still remains as the only fully functional GNSS but this will not be true in the near future.

The NAVSTAR GPS consists of 24 satellites, divided into 6 orbits, with 4 satellites in each orbit. At the time when the system was made available for civilian use, the GPS signal only contains two frequency components; the L1 which has a centre frequency of 1575.42MHz and L2 at 1227.6MHz. The L1 transmits both coarse/acquisition (C/A) and precision (P) codes whereas the L2 only transmits P codes. The P codes were intended for military applications where a valid decryption key was required to access the information. The NAVSTAR GPS has been consistently maintained and upgraded over the years. Currently, a new military signal referred to as the M code is transmitted on both L1 and L2, a new civil signal known as the L2C is transmitted on L2 and a new carrier frequency, known as L5, which operates at 1176.45MHz is made available.

Prior to 2^{nd} of May, 2000, the accuracy of the C/A codes, made publicly available to navigation systems, were intentionally incorporated with random errors up to a hundred meters through a feature known as Selective Availability (SA). SA was disabled following the development of a new system which could deny access to GPS information by hostile forces within a specific area without affecting the rest of the world. At present, civilian accessible GPS information have an accuracy of 10-15 meters (not taking into account of effects due to atmospheric conditions, multipaths, etc). Higher accuracy can be achieved with GPS receivers that utilizes information from the Wide Area Augmentation System (WAAS) satellites. WAAS uses a network of ground-based reference stations which tracks small variations of the satellites in the western hemisphere. This information

Figure G.1: Holux GPSlim 240

is sent to geostationary WAAS satellites through master stations and the correction information is made available to WAAS enabled receivers. Unfortunately, this information is only useful for people living in the US. For the rest of the world, higher accuracy can be achieved by utilizing the Differential Global Positioning System (DGPS). The operating principle of DPGS is similar to WAAS with the main distinction that correction signals for DGPS are sent via ground transmitters instead of the orbiting satellites used by WAAS. For more information, please refer to (Tsui, 2001).

Preliminary experiments were conducted using an inexpensive GPS receiver. The Holux GPSlim 240 (2010) illustrated in Fig. G.1 is a compact device powered by the SiRF III which can operate up to 8 hours with a single charge. It interfaces to a computer using serial communication via Bluetooth with the data transmitted based on the National Marine Electronics Association (NMEA) 0183 protocol. Based on (SiRF, 2005), a small program was written to communicate with and extract the incoming data from the GPS receiver . The GPS receiver returns useful information such as speed, direction, time and most importantly, position. This global position is returned in the latitude-longitude format. The latitude is measured from the equator, with positive values going north and negative values going south (-90 to 90) and longitude is measured from the Prime Meridian (Greenwich, England) with positive values going east and negative values going west. However, for mobile robot localization purposes, it is more convenient if the global position is represented using Cartesian coordinates instead.

The two commonly adopted coordinate systems, which can be used to convert the global position from the latitude-longitude format to Cartesian coordinates, are the Universal Transverse Mercator (UTM) and State Plane Coordinate System (SPCS). The UTM coordinate system uses 60 zones, each 6 degrees of longitude wide. On the other hand, the SPCS form zones based on political boundaries and is only used in the US. SPCS is 4 times more accurate than UTM through the use of relatively small zones. Since the accuracy of the conversion is only true within the same zone, the SPCS is not suitable for regional, national or global mapping tasks. The UTM coordinate system was employed in the preliminary experiments. Despite the fact that UTM is less accurate than SPCS but its accuracy of 1:2500 (interpreted as a maximum of 1 unit error for every 2500 units measured) is deemed tolerable. In addition, it is also attractive since its implementation is much easier. Please refer to (Dutch, 2003) for extra information on the UTM coordinate system and (Stem, 1989) for extra information on the SPCS.

G.2 Preliminary Experiments

The first set of preliminary experiments conducted were aimed at testing the quality of GPS information collected under different weather conditions and locations. Since it was inconvenient to carry a laptop around while collecting data from the GPS receiver, an alternative setup was used. GPS data was logged on a smartphone (Dopod 595) using a freeware known as GPSVP (2010). The logged GPS data was saved into the OziExplorer ".PLT" file format and was uploaded to GPSLib (2010) in order to convert it into the ".KML" file format which is compatible with Google Earth. GPS data was collected while walking around the Engineering Faculty in Monash University (Fig. G.2), on a bus ride from Monash University to Chadstone Shopping Centre (Fig. G.3-G.4) and while walking home from the Engineering Faculty to Rusdenhouse (Fig. G.5-G.7) showing the effect of different weather conditions on the quality of GPS data.

Figure G.2: Walk Around the Engineering Faculty

Figure G.3: Chadstone Bus Ride

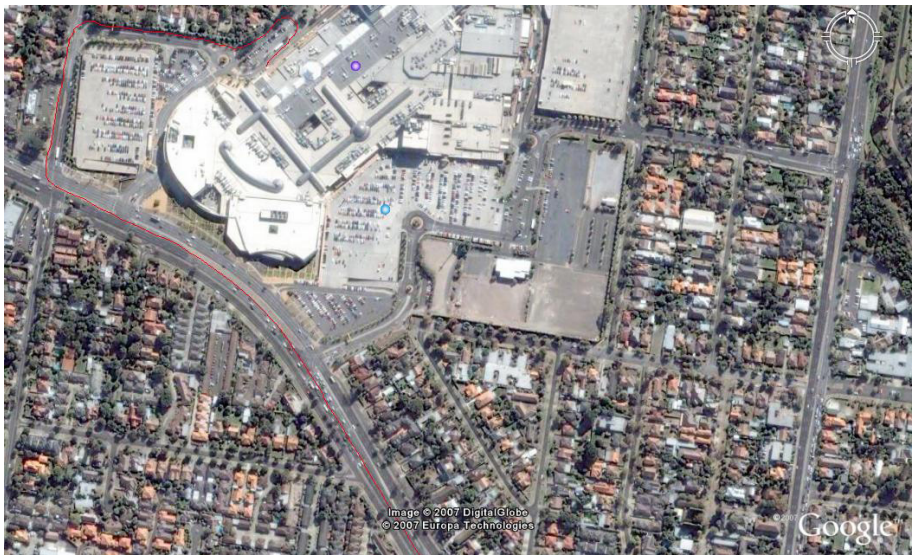


Figure G.4: Chadstone Bus Ride (Close Up View)

(a) Clear Day (b) Cloudy Day

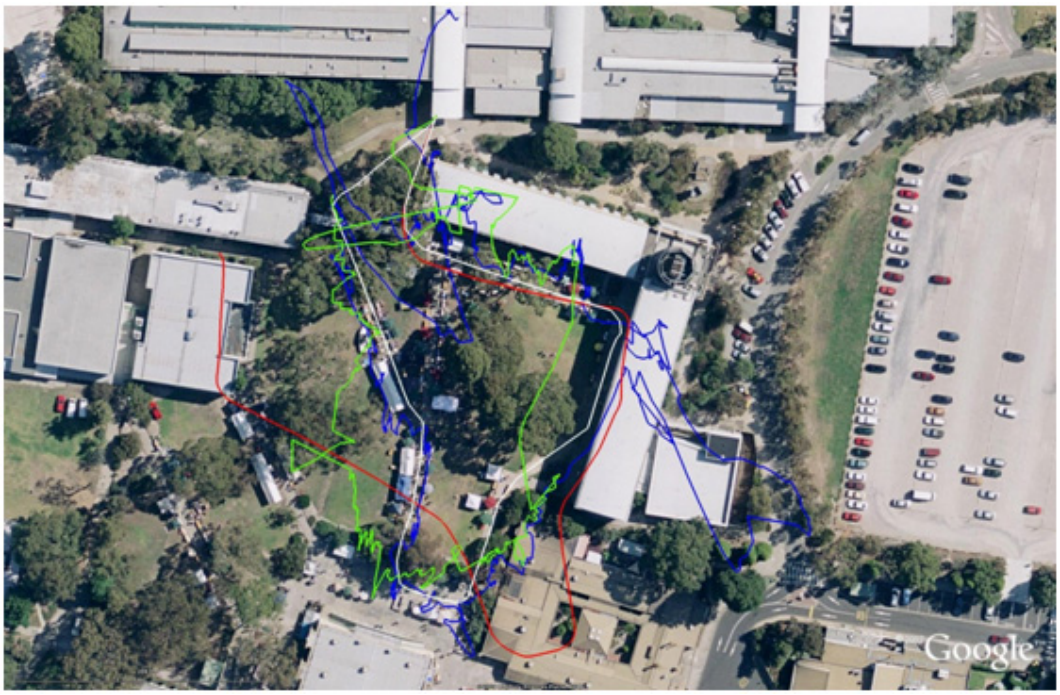
Figure G.5: Different Weather Conditions

Figure G.6: Walk from Engineering Faculty to Rusdenhouse (Clear Day)

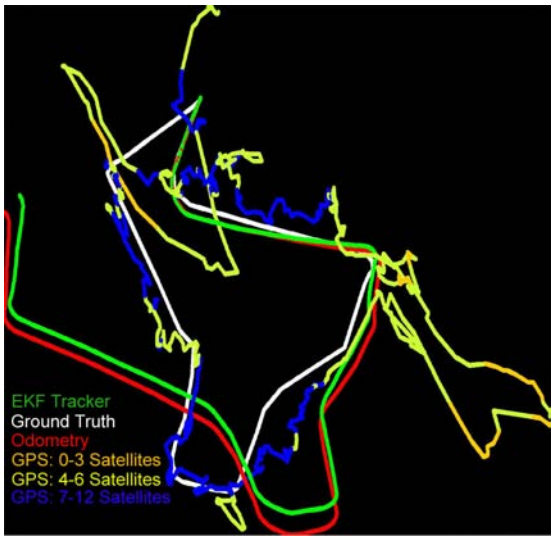
Figure G.7: Walk from Engineering Faculty to Rusdenhouse (Cloudy Day)

In the second experiment, the GPS receiver was connected to a laptop which controls and receives odometry information from the ActivMedia Pioneer P3-AT. GPS information and the mobile robot's odometry is fused using the Extended Kalman Filter (EKF) framework described by Thrapp et al. (2001). The odometry error model proposed by Tur (2007) and the GPS pose error model proposed by Tur and Borja (2007) have been employed for use in this work. The error covariance matrix for GPS pose estimates were estimated by collecting GPS data from various environments under varying weather conditions (in order to use a different error covariance matrix depending on the number of satellites in view or the positional dilution of precision (PDOP) value) The mobile robot was manually driven in an outdoor environment where the GPS data (global position, PDOP, number of satellites in view) and odometry information (x,y,θ) were logged.

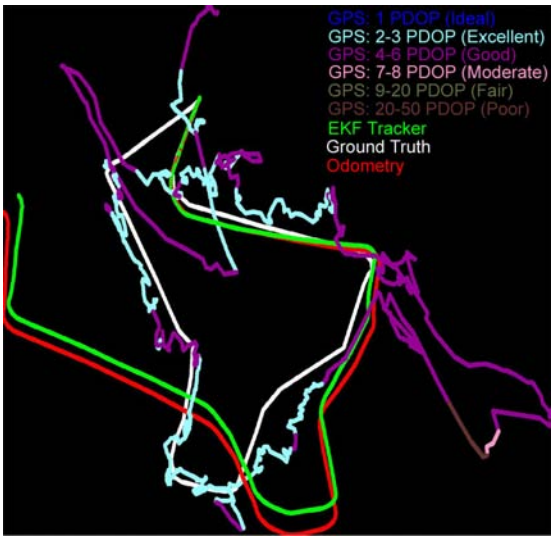
Experimental data were collected from two environments. The mobile robot was manually driven in approximately the same trajectory for two times in the same environment, once during a clear day and another during a cloudy day. The data collected from each run were processed offline using the EKF framework and the results are illustrated in Fig. G.8-G.11. A total of 3 figures are available for each experiment; (a) showing the final results where the various trajectories were overlaid on top of the map of the environment in Google Earth, (b) primarily showing the number of satellites in view to the GPS receiver at different locations and (c) the PDOP value of its estimated position at different locations. Take note that the trajectory produced by the EKF framework in (b) and (c) for each experiment is different from the one shown in (a). The trajectory (EKF framework) in (b) and (c) was produced by using the estimated error covariance matrix based on the method in (Tur and Borja, 2007) whereas the trajectory shown in (a) was produced by using two empirically determined error covariance matrices; one for the case when the total number of satellites in view is 7 or more and another when there are less than 7 satellites in view.



(a) Final Results (Ground Truth - White, Odometry - Red, GPS - Blue, EKF Tracker - Green)



(b) Number of Satellites

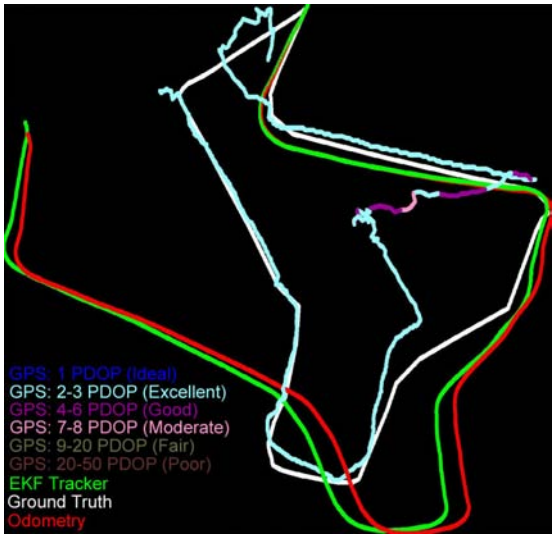


(c) PDOP

Figure G.8: Environment 1 - Poor GPS Position Fix (Longer Trajectory)

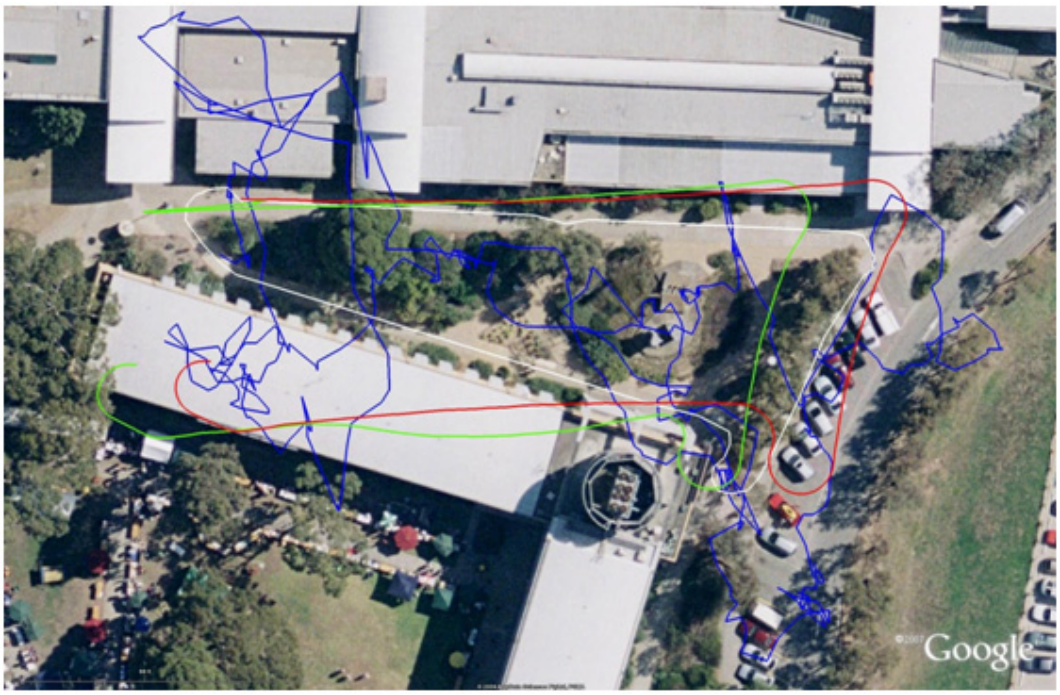
(a) Final Results (Ground Truth - White, Odometry - Red, GPS - Blue, EKF Tracker - Green)

(b) Number of Satellites

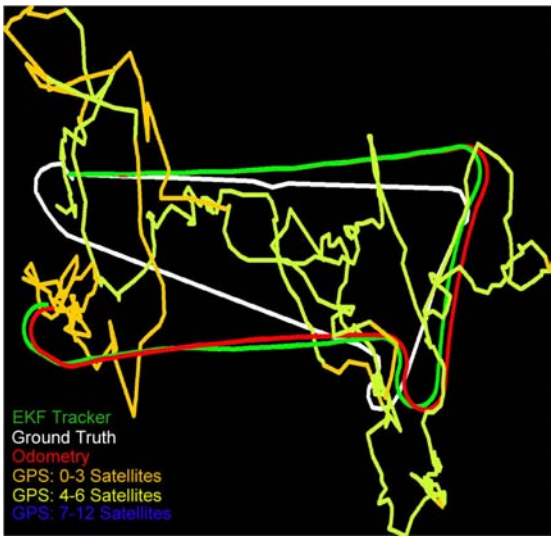


(c) PDOP

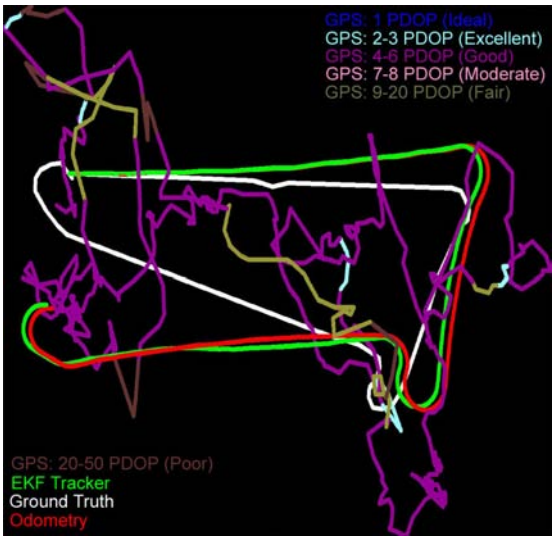
Figure G.9: Environment 1 - Good GPS Position Fix (Longer Trajectory)



(a) Final Results (Ground Truth - White, Odometry - Red, GPS - Blue, EKF Tracker - Green)



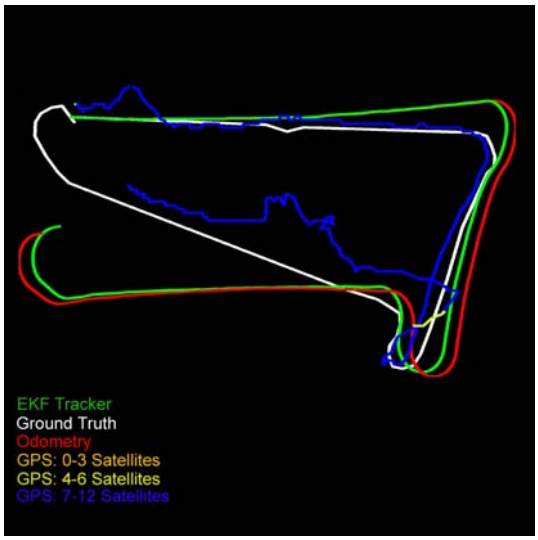
(b) Number of Satellites



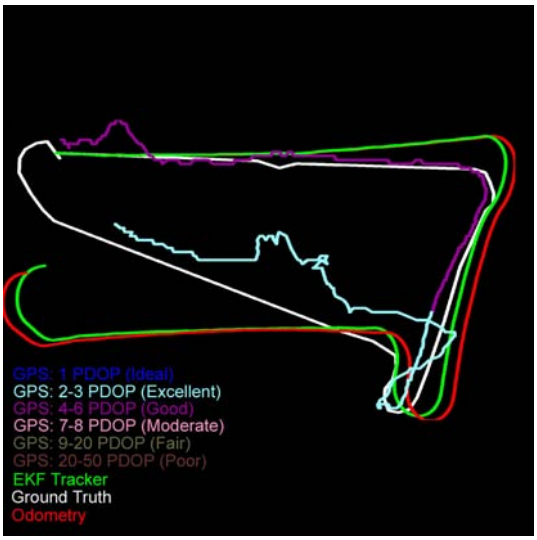
(c) PDOP

Figure G.10: Environment 2 - Poor GPS Position Fix (Longer Trajectory)

(a) Final Results (Ground Truth - White, Odometry - Red, GPS - Blue, EKF Tracker - Green)



(b) Number of Satellites



(c) PDOP

Figure G.11: Environment 2 - Good GPS Position Fix (Longer Trajectory)

G.3 Discussion and Conclusion

The accuracy of the GPS data, to a great extent, relies on the number of satellites in view and as well as the geometry of the satellites. When only 3 or less satellites are in view, it can only provide a 2D positional fix. For better accuracy, 4 or more satellites are required such that a 3D positional fix is possible. In addition, the Dilution of Precision (DOP) can be used as a gauge for the accuracy of the returned global position. There are several measures of DOP such as the horizontal, vertical, positional and time DOP. Nonetheless, all these values are based on the positions of the satellites. The accuracy of the returned global position can be improved by minimizing DOP, which is achieved by maximizing the volume contained by the satellites. Of course, other factors such as ionospheric conditions, multipaths, distance from reference receivers (depending on the type of GPS employed) and quality of the GPS receiver play a role in its accuracy as well. Ultimately, this work aims to utilize the information to enable the mobile robot to make the best decision based on the quality of the data being received.

As illustrated in the preliminary results, the trajectory produced by the EKF framework using the estimated error covariance matrix shown in (b) and (c) of each experiment was almost entirely disregarding the GPS information due to its relatively large uncertainties with its pose estimates defined by the estimated error covariance matrix. On the other hand, the empirically determined error covariance matrices (dependent on the number of satellites in view) produced a relatively better estimate of the mobile robot's location but was still not good enough in general. The reason that the EKF framework was not performing as well as the systems described in (Thrapp et al., 2001) or (Tur and Borja, 2007) was because that the GPS used in those systems were much more accurate as compared to the Holux GPSlim 240. To properly fuse GPS data from an inexpensive GPS receiver with odometry information, it might be worthwhile considering some of the factors discussed in (Agrawal and Konolige, 2006) which only incorporates GPS data into its Kalman filter framework only when it has a 3D positional fix, the mobile robot is travelling 0.5m/s or faster to limit the effect of velocity noise and after a considerable amount of distance has been traversed by the mobile robot since the last GPS reading. In this case, the role of the GPS receiver is to reduce the drift accumulated in the estimated position of the mobile robot intermittently when GPS data is deemed reliable between two distant locations. This is highly beneficial for vision-based mapping and localization systems since vision systems work well in texture rich and cluttered environments whereas GPS works best in open environments.

References

ABB Robotics (2010). Abb product guide.
URL: *http://www.abb.com/product/us/9AAC910011.aspx*

Adluru, N., Latecki, L. J., Sobel, M. and Lakaemper, R. (2008). Merging maps of multiple robots, *IEEE International Conference on Pattern Recognition*, pp. 1–4.

Agrawal, M. and Konolige, K. (2006). Real time localization in outdoor environments using stereo vision and inexpensive GPS, *IEEE International Conference on Pattern Recognition*.

Agrawal, M. and Konolige, K. (2007). Rough terrain visual odometry, *International Conference on Advanced Robotics*.

Agrawal, M., Konolige, K. and Bolles, R. C. (2007). Localization and mapping for autonomous navigation in outdoor terrains : A stereo vision approach, *IEEE Workshop on Application of Computer Vision*, Austin Texas.

Altendorfer, R., Moore, N., Komsuoğlu, H., Buehler, M., Jr., H. B., d. McMordie, Saranli, U., Full, R. and Koditschek, D. (2001). RHex: A biologically inspired hexapod runner, *Autonomous Robots* **11**: 201–213.

Amato, N. M. and Wu, Y. (1996). A randomized roadmap method for path and manipulation planning, *IEEE International Conference on Robotics and Automation*, pp. 113–120.

Angeli, A., Doncieux, S., Meyer, J.-A. and Filliat, D. (2009). Visual topological SLAM and global localization, *IEEE International Conference on Robotics and Automation*.

Angeli, A., Filliat, D., Doncieux, S. and Meyer, J.-A. (2008). A fast and incremental method for loop-closure detection using bags of visual words, *IEEE Transactions on Robotics, Special Issue on Visual SLAM* **24**(5): 1027–1037.

Arican, Z. and Frossard, P. (2007). Dense disparity estimation from omnidirectional images, *Advanced Video and Signal Based Surveillance*, pp. 399–404.

Australian Bureau of Statistics (2008). 3222.0 - population projections, australia, 2006 to 2101.
URL: *http://www.abs.gov.au/Ausstats/abs@.nsf/mf/3222.0*

Bajracharya, M., Maimone, M. W. and Helmick, D. (2008). Autonomy for mars rovers: Past, present and future, *Computer* **41**(12): 44–50.

Baker, S. and Nayar, S. K. (1999). A theory of single-viewpoint catadioptric image formation, *International Journal of Computer Vision* **35**(2): 175–196.

Bay, H., Tuytelaars, T. and Gool, L. V. (2006). SURF: Speeded up robust features, *Proceedings of Ninth European Conference on Computer Vision*.

Beevers, K. R. and Huang, W. H. (2005). Loop closing in topological maps, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 4368–4372.

Benosman, R. and Kang, S. B. (2001). *Panoramic Vision: Sensors, Theory and Applications*, Springer-Verlag.

Bhattacharya, P. and Gavrilova, M. (2007). Voronoi diagram in optimal path planning, *International Symposium on Voronoi Diagrams in Science and Engineering*.

Boor, V., Overmars, M. H. and van der Stappen, A. F. (1999). The gaussian sampling strategy for probabilistic roadmap planners, *IEEE International Conference on Robotics and Automation*, Vol. 2, Detroit, Michigan, pp. 1018–1023.

Caglioti, V., Taddei, P., Boracchi, G., Gasparini, S. and Giusti, A. (2007). Single-image calibration off-axis catadioptric cameras using lines, *IEEE 11th International Conference on Computer Vision*.

Campbell, J., Sukthankar, R., Nourbakhsh, I. and Pahwa, A. (2005). A robust visual odometry and precipice detection system using consumer-grade monocular vision, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain.

Carpin, S. (2008). Merging maps via hough transform, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 1878–1883.

Chahl, J. and Srinivasan, M. (1997). Reflective surfaces for panoramic imaging, *Applied Optics* **36**(31): 8275–8285.

Cheng, Y., Maimone, M. W. and Matthies, L. (2006). Visual odometry on the mars exploration rovers: A tool to ensure accurate driving and science imaging, *IEEE Robotics and Automation Magazine* **13**(2): 54–62.

Chiang, K.-W., Huang, Y.-S., Tsai, M.-L. and Chen, K.-H. (2010). The perspective from asia concerning the impact of compass/beidou-2 on future gnss, *Survey Review* **42**(315): 3–19.

Chong, K. S. and Kleeman, L. (1997). Accurate odometry and error modelling for a mobile robot, *IEEE International Conference on Robotics and Automation*, pp. 2783–2788.

Collins, R. T. (1996). A space-sweep approach to true multi-image matching, *IEEE Computer Society Comference on Computer Vision and Pattern Recognition*.

Comport, A., Malis, E. and Rives, P. (2010). Real-time quadrifocal visual odometry, *The International Journal of Robotics Research* **29**(2-3): 245–266.

Cummins, M. and Newman, P. (2008). FAB-MAP probabilistic localization and mapping in the space of appearance, *International Journal of Robotics Research* **27**: 647–665.

Davison, A. J. (1998). *Mobile Robot Navigation using Active Vision*, PhD thesis, University of Oxford.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* **1**: 269–271.

Diosi, A. and Kleeman, L. (2005). Laser scan matching in polar coordinates with application to SLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Duckett, T. (2000). *Concurrent Map Building and Self-localisation for Mobile Robot Navigation*, PhD thesis, Department of Computer Science, University of Manchester.

Duckett, T., Marsland, S. and Shapiro, J. (2000). Learning globally consistent maps by relaxation, *IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 3841–3846.

Dutch, S. (2003). Converting utm to latitude and longitude (or vice versa).
URL: *http://www.uwgb.edu/dutchs/UsefulData/UTMFormulas.HTM*

Dyer, A. G., Neumeyer, C. and Chittka, L. (2005). Honeybee (apis mellifera) vision can discriminate between and recognize images of human faces, *Journal of Experimental Biology* **208**: 4709–4714.

Dyer, C. R. (2001). *Volumetric Scene Reconstruction from Multiple Views*, Vol. Chapter 16, Kluwer, Boston.

Ekstrom, A. D., Kahana, M. J., Caplan, J. B., Fields, T. A., Isham, E. A., Newman, E. L. and Fried, I. (2003). Cellular networks underlying human spatial navigation, *Nature* **425**: 184–188.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation, *Computer* **22**(6): 46–57.

Engels, C., Stewenius, H. and Nistér, D. (2006). Bundle adjustment rules, *Photogrammetric Computer vision*.

Fernandez, D. and Price, A. (2004). Visual odometry for an outdoor mobile robot, *IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, pp. 816–821.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applicatons to image analysis and automated cartography, *Communications of ACM: Graphics and Image Processing* **24**(6): 381–395.

Fjerdingen, S. A., Liljebäck, P. and Transeth, A. A. (2009). A snake-like robot for internal inspection of complex pipe structures (PIKo), *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St Louis, Missouri, USA, pp. 5665–5671.

Fortune, S. (1997). *Voronoi Diagrams and Delaunay Triangulations*, CRC Press, Inc.

Fua, P. (1991). Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities, *International Joint Conference on Artificial Intelligence*, pp. 1292–1298.

Gage, D. W. (1995). Ugv history 101: A brief history of unmanned ground vehicle (ugv) development efforts, *Unmanned Systems Magazine* **13**(3): 1–9.

Gallup, D., Frahm, J.-M., Mordohai, P. and Pollefeys, M. (2008). Variable baseline/resolution stereo, *IEEE International Conference on Pattern Recognition*.

Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q. and Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions, *IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, pp. 1–8.

GASS CUDA.NET (2008). Online.
URL: *http://www.gass-ltd.co.il/en/products/cuda.net*

Gates, W. G. (2007). A robot in every home, *Scientific American* pp. 58–65.

Geyer, C. and Danilidis, K. (2001). Catadioptric projective geometry, *International Journal of Computer Vision* **45**(3): 223–243.

Gil, A., Óscar Reinoso, Ballesta, M. and Juliá, M. (2010). Multi-robot visual slam using a rao-blackwellized particle filter, *Robotics and Autonomous Systems* **58**(1): 68–80.

Gluckman, J. and Nayar, S. K. (1998). Ego-motion and omnidirectional cameras, *IEEE International Conference on Computer Vision and Pattern Recognition*.

Golfarelli, M., Maio, D. and Rizzi, S. (1998). Elastic correction of dead-reckoning errors in map building, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada, pp. 905–911.

Gomes, P. (2010). Surgical robotics: Reviewing the past, analysing the present, imagining the future.

Goncalves, L., Bernardo, E. D., Benson, D., Svedman, M., Ostrowski, J., Karlsson, N. and Pirjanian, P. (2005). A visual front-end for simultaneous localization and mapping, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 44–49.

Gonçalves, N. and Araújo, H. (2004). Projection model, 3D reconstruction and rigid motion estimation from non central catadioptric images, *2nd Symposium on 3D Data Processing, Visualization and Transmission*.

Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J. and Maisonnier, B. (2009). Mechatronics design of NAO humanoid, *IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 769–774.

GPSLib (2010). Oziexplorer track (.plt) to google.earth path (.kml) converter.
URL: *http://gpslib.net/services/OziTrack-to-GoogleEarth-Track/*

GPSVP (2010). Gps navigation software for smartphones, pdas and pcs.
URL: *http://code.google.com/p/gpsvp/*

Gränstrom, K., Callmer, J., Ramos, F. and Nieto, J. (2009). Learning to detect loop closure from range data, *IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 15–22.

Hansen, P., Corke, P. and Boles, W. (2010). Wide-angle visual feature matching for outdoor localizaton, *International Journal of Robotics* **29**(2-3): 267–297.

Hart, P. E., Nilsson, N. J. and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions of System, Science and Cybernetics* **SSC-4**(2): 100–107.

Hartley, R. I. and Zisserman, A. (2000). *Mutliple View Geometry in Computer Vision*, Cambridge University Press.

He, L., Luo, C., Geng, Y., Zhu, F. and Hao, Y. (2007). *Reliable Depth Map Regeneration Via a Novel Omnidirectional Stereo Sensor*, Lecture Notes on Computer Science 4841, Srpinger-Verlag.

Hecht, E. and Zajac, A. (1987). *Optics 2nd ed.*, Addison-Wesley.

Hegarty, C. J. and Charte, E. (2008). Evolution of the global navigation satellite system (gnss), *Proceedings of the IEEE* **96**(12): 1902–1917.

Ho, K. L. and Newman, P. (2007). Detecting loop closure with scene sequences, *International Journal of Computer Vision* **74**(3): 261–286.

Ho, N. (2010). Personal website.
URL: *http://nghiaho.com/*

Ho, N. and Jarvis, R. (2008). Vision based global localisation using a 3D environmental model created by a laser range scanner, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France.

Holux GPSlim 240 (2010). Holux.
URL: *http://www.holux.com/JCore/en/products/products*$_c$*ontent.jsp?pno* = 253

Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France.

Howard, A., Sukhatme, G. S. and Matarić, M. (2006). Localization and mapping using manifold representations, *Proceedings of the IEEE* **94**(7): 1360–1369.

Hsieh, P.-C. and Tung, P.-C. (2009). A novel hybrid approach based on sub-pattern technique and whitened PCA for face recognition, *Pattern Recognition* **42**(5): 978–984.

Huang, Y. and Ahuja, N. (1992). A potential field approach to path planning, *IEEE Transactions on Robotics and Automation* **8**(1): 23–32.

International Federation of Robotics (2010). World robotics 2010: Executive summary.
URL: *http://www.worldrobotics.org/downloads/2010*$_E$*xecutive*$_S$*ummary*$_r$*ev.pdf*

Intuitive Surgical (2010). da vinci surgical system.
URL: *http://www.intuitivesurgical.com/index.aspx*

Ishiguro, H. (2007). Scientific issues concerning androids, *International Journal of Robotics Research* **26**(1): 105–117.

Jacobs, C. E., Finkelstein, A. and Salesin, D. H. (1995). Fast multiresolution image querying, *Computer Graphics* **29**(Annual Conference Series): 277–286.

Jarvis, R. (1983). Growing fast polyhedral obstacles for planning collision-free paths, *The Australian Computer Journal* **15**(3): 103–111.

Jarvis, R. (1985). Collision-free trajectory planning using distance transforms, *Mechanical Engineering Transactions, Journal of the Institution of Engineers* **ME10**(3): 187–191.

Jarvis, R. (1992). Optimal pathways for road vehicle navigation, *IEEE Region 10 International Conference Technology Enabling Tomorrow: Computers, Comunications, and Automation Towards the 21st Century*, Melbourne, Australia, pp. 876–880.

Jarvis, R. (1994). *On Distance Transform Based Collision Free Path Planning for Robot Navigation in Known, Unknown and Time Varying Environments*, World Scientific Publishing Co. Pty. Ltd., pp. 3–31.

Jarvis, R. (2003). *A Go Where You Look Tele-autonomous Rough Terrain Mobile Robot*, Vol. Experimental Robotics VIII, STAR 5, Springer Verlag.

Jarvis, R. (2008). Sensor rich teleoperational mode robotics bush fire fighting, *Proceedings of the EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment*, Benicassim, Spain.

Jarvis, R., Gupta, O., Effendi, S. and Li, Z. (2009). An intelligent robotic assistive living system, *International Conference on Pervasive Technologies Related to Assistive Environments*, Corfu, Greece.

Jarvis, R. and Marzouqi, M. S. (2005). Robot path planning in high risk fire front environments, *IEEE Region 10 International Conference Technology Enabling Tomorrow: Computers, Comunications, and Automation Towards the 21st Century*, Melbourne, Australia, pp. 1–6.

Jennings, C., Murray, D. and Little, J. J. (1999). Cooperative robot localization with vision-based mapping, *International Conference on Robotics and Automation*, Detroit, Michigan, pp. 2659–2665.

K-Team Mobile Robotics (2010). Beyond minature technology, Online.
URL: *http://www.k-team.com/*

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Transactions of ASME - Journal of Basic Engineering* **82**(Series D): 35–45.

Kang, J.-G., An, S.-Y., Kim, S. and Oh, S.-Y. (2010). Sonar-based simultaneouss localization and mapping using a neuro-evolutionary optimization, *Advanced Robotics* **24**(8-9): 1257–1289.

Kang, S. B., Webb, J. A., Zitnick, C. L. and Kanade, T. (1995). A multibaseline stereo system with active illumination and real-time image acquisition, *IEEE International Conference on Computer Vision*, pp. 88–93.

Kawewong, A., Tongprasit, N., Tangruamsub, S. and Hasegawa, O. (2010). Online and incremental appearance-based slam in highly dynamic environments.

Kim, J. and Brambley, G. (2007). Dual optic-flow integrated inertial navigation, *Australiasian Conference on Robotics and Automation*.

Klaus, A., Sormann, M. and Karner, K. (2006). Segment-based stereo matching using belief propagation and a self adapting dissimilarity measure, *IEEE International Conference on Pattern Recognition*.

Kleeman, L. (2003). Advanced sonar and odometry error modeling for simultaneous localisation and map building, *IEEE/RSJ International Conference on Inteligent Robots and Systems*, Las Vegas, pp. 699–704.

Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V. and Fua, P. (2010). View-based maps, *The International Journal of Robotics Research* **29**(8): 941–957.

Konolige, K., Fox, D., Limketkai, B., Ko, J. and Stewart, B. (2003). Map merging for distributed robot navigation, *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Kröse, B., Bunschoten, R., Vlassis, N. and Motomura, Y. (1999). Appearance based robot localization, *in* G. Kraetzschmar (ed.), *IJCAI-99 Workshop Adaptrive Spatial Representations of Dynamic Environments*, pp. 53–58.

Kuffner, J. J. and LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning, *IEEE International Conference on Robotics and Automation*, Vol. 2, San Francisco, CA USA, pp. 995–1001.

KUKA (2010).
URL: *http://www.kuka.com/*

Labrosse, F. (2007). The visual compass: Performance and limitations of an appearance-based method, *Journal of Field Robotics* **23**(10): 913–941.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning, *Technical Report TR 98-11*, Iowa State University.

LaValle, S. M. and Kuffner, J. J. (1999). Randomized kinodynamic planning, *IEEE International Conference on Robotics and Automation*, number 1, Detroit, MI USA, pp. 473–479.

León, A., Barea, R., Bergasa, L., López, E., Ocana, M. and Schleicher, D. (2009). Slam and map merging, *Journal of Physical Agents* **3**(1): 13–23.

Logitech (2010). Logitech hd pro webcam c910.
URL: *http://www.logitech.com/en-us/webcam-communications/webcams/devices/6816*

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm, *Computer Graphics* **21**(4): 163–169.

Lowe, D. (2004). Distinctive image features from scale invariant keypoints, *International Journal of Computer Vision* **60**(2): 91–110.

Lozano-Perez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of ACM* **22**(10): 560–570.

Lu, F. and Millios, E. (1997). Globally consistent range scan alignment for environment mapping, *Autonomous Robots* **4**: 333–349.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision, *International Joint Conference on Artificial Intelligence*, pp. 674–679.

Lui, W. L. D. and Jarvis, R. (2008). An omnidirectional vision system for outdoor mobile robots, *International Workshop on Omnidirectional Robot Vision (Workshop Proceedings of SIMPAR 2008)*, Venice, Italy, pp. 273–284.

Lui, W. L. D. and Jarvis, R. (2010a). An active visual loop closure detection and validation system for topological slam, *Australasian Conference on Robotics and Automation*, Brisbane, Australia.

Lui, W. L. D. and Jarvis, R. (2010b). Eye-Full Tower: A GPU-based variable multibaseline omnidirectional stereovision system with automatic baseline selection for outdoor mobile robot navigation, *Robotics and Autonomous Systems* **58**(6): 747–761.

Lui, W. L. D. and Jarvis, R. (2010c). A pure vision-based approach to topological SLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, pp. 3784–3791.

Maladen, R. D., Ding, Y., Li, C. and Goldman, D. I. (2009). Undulatory swimming in sand: Subsurface locomotion of the sandfish lizard, *Science* **325**(5938): 314–318.

Meguro, J., Takiguchi, J., Amano, Y. and Hashizume, T. (2007). 3Dreconstruction using multibaseline omnidirectional motion stereo based on GPS/dead-reckoning compound navigation system, *International Journal of Robotics Research* **26**(6): 625–636.

Mei, C. and Rives, P. (2007). Single view point omnidirectional camera calibration from planar grids, *IEEE International Conference on Robotics and Automation*, pp. 3945–3950.

Milford, M. and Wyeth, G. (2010). Persistent navigation and mapping using a biologically inspired slam system, *The International Journal of Robotics Research* **29**(9): 1131–1153.

Mičušík, B. and Pajdla, T. (2004). Autocalibration & 3D reconstruction with non-central catadioptric cameras, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 58–65.

Mičušík, B. and Pajdla, T. (2006). Structure from motion with wide circular field of view cameras, *IEEE Transactions on Pattern Recogniton and Machine Intelligence* **28**(7): 1135–1149.

Mobile Robots (2010). Autonomous mobile robot cores, bases and accessories, Online.
URL: *http://www.mobilerobots.com/Mobile$_R$obots.aspx*

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffman, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A. and Thrun, S. (2008). Junior: The stanford entry in the urban challenge, *Journal of Field Robotics* **25**(9): 569–597.

Nagatani, K., Okada, Y., Tokunaga, N., Yoshida, K., Kiribayashi, S., Ohno, K., Takeuchi, E., Tadokoro, S., Akiyama, H., Noda, I. and Yoshida, T. (2009). A mutli-robot exploration for search and rescue missions: A report of map building in robocuprescue 2009, *IEEE International Workshop on Safety, Security & Rescue Robotics*, Denver, CO, pp. 1–6.

Nakabo, Y., Mukai, T., Hattori, Y., Takeuchi, Y. and Ohnishi, N. (2005). Variable baseline stereo tracking vision system using high-speed linear slider, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 1567–1572.

National Institute on Aging, National Institutes on Health and U.S. Department of Health and Human Services (2007). Why population aging matters: A global perspective.
URL: *http://www.nia.nih.gov/NR/rdonlyres/9E91407E-CFE8-4903-9875-D5AA75BD1D50/0/WPAM.pdf*

Nebot, E. M. (2007). Surface mining: Main research issues for autonomous operations, *in* S. Thrun, R. Brooks and H. Durrant-Whyte (eds), *Robotics Research*, Vol. STAR 28, pp. 268–280.

Ng, T. (2003). The optical mouse as a two-dimensional displacement sensor, *Sensors and Actuators A* **107**: 21–25.

Nilsson, N. J. (1969). A mobile automation: An application of artificial intelligence techniques, *International Joint Conference on Artificial Intelligence*, Washington D.C.

Nilsson, N. J. (1971). *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill.

Nistér, D., Naroditsky, O. and Bergen, J. (2007). Visual odometry for ground vehicle applications, *International Journal of Field Robotics* **23**(1): 3–20.

Nourani-Vatani, N. and Pradelier, C. (2010). Scene change detection for vision-based topological mapping and localization, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, pp. 3792–3797.

Nvidia CUDA Zone (2008). Online.
URL: *http://www.nvidia.com/*

O'Keefe, J. and Nadel, L. (1978). *The Hippocampus as a Cognitive Map*, Oxford University Press.

Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(4): 353–363.

Ollis, M., Herman, H. and Singh, S. (1999). Analysis and design of panoramic stereo vision using equi-angular pixel cameras, *Technical Report CMU-RI-TR-99-04*, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Oxford Dictionaries (2010). The world's most trusted dictionaries.
URL: *http://oxforddictionaries.com/?attempted=true*

Paz, L. M., Tardós, J. D. and Neira, J. (2008). Divide and conquer: EKF SLAM in O(n), *IEEE Transactions on Robotics* **24**(5): 1107–1120.

Phan, T. A. and Russell, R. A. (2009). Quantitative information in sematectonic stigmergy for swarm robots, *Australasian Conference on Robotics and Automation*, Sydney, Australia.

Point Grey (2010). Products and services, Online.
URL: *http://www.ptgrey.com/products/stereo.asp*

Radke, R. J., Andra, S., Al-Kohafi, O. and Roysam, B. (2005). Image change detection algorithms: A systematic survey, *IEEE Transactions on Image Processing* **14**: 294–307.

Ragot, N., Rossi, R., Savatier, X., Ertaud, J.-Y. and Mazari, B. (2008). 3D volumetric reconstruction with a catadioptric stereovision sensor, *In IEEE Symposium on Industrial Electronics*, pp. 1306–1311.

Raibert, M., Blankespoor, K., Nelson, G., Playter, R. and the Bigdog Team (2008). BigDog, the rough-terrain quadruped robot, *Proceedings of the 17th World Congress The international Federation of Automatic Control*, Seoul, Korea.

Rosa, M. G. and Tweedale, R. (2005). Brain maps, great and small: lessons from comparative studies of primate visual cortical organization, *Philosophical Transactions of the Royal Society B* **360**(1456): 665–691.

Rosenfeld, A. and Pfaltz, J. L. (1966). Sequential operations in digital picture processing, *Journal of the Association of Computing Machinery* **13**(4): 471–494.

Rusu, R. B., Holzbach, A., Diankov, R., Bradski, G. and Beetz, M. (2009). Perception for mobile manipulation and grasping using active stereo, *Humanoids*.

Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N. and Fujimura, K. (2002). The intelligent ASIMO: System overview and integration, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, EPFL, Lausanne, Switzerland, pp. 2478–2483.

Sato, T., Kanbara, M., Yokoya, N. and Takemura, H. (2002). 3-D modeling of an outdoor scene by multi-baseline stereo using a long sequence of images, *16th International Conference on Pattern Recognition*.

Scaramuzza, D., Martinelli, A. and Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras, *IEEE International Conference on Vision Systems*.

Scaramuzza, D. and Siegwart, R. (2008). Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles, *IEEE Transactions on Robotics* **24**(5): 1015–1026.

Scharstein, D. and Szeliski, R. (2008). Middlebury college stereovision research page, Online.

Scharstein, D., Szeliski, R. and Zabih, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision* **47**(1/2/3): 7–42.

Se, S. and Jasiobedzki, P. (2006). Photo-realistic 3D model reconstruction, *IEEE International Conference on Robotics and Automation*, Orlando Florida, pp. 3076–3082.

Se, S., Lowe, D. and Little, J. (2002). Mobile robot localizaton and mapping with unvertainty using scale-invariant visual landmarks, *International Journal of Robotics Research* **21**(8): 735–758.

Segway Robotics (2010). Powerful, modular, flexible, simple, Online.
URL: *http://rmp.segway.com/*

Shi, J. and Tomasi, C. (1994). Good features to track, *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle.

Sim, R. and Roy, N. (2005). Global a-optimal robot exploration in slam, *IEEE International Conference on Robotics and Automation*.

SiRF (2005). Nmea reference manual.
URL: *http://www.sparkfun.com/datasheets/GPS/NMEAManual1.pdf*

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos, *IEEE International Conference on Computer Vision*, Vol. 2, Nice, France, pp. 1470–1477.

Spero, D. J. and Jarvis, R. (2005). A new solution to the simultaneous localisation and map building (slam) problem, *Technical Report MECSE-27-2005*, Monash University.

Srinivasan, M. V., Thurrowgood, S. and Soccol, D. (2009). Competent vision and navigation systems, *IEEE Robotics and Automation Magazine* **16**(3): 59–71.

Stachniss, C., Hähnel, D. and Burgard, W. (2004). Exploration with active loop-closing for fast-slam, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp. 1505–1510.

Stam, J., Gallup, D. and Frahm, J.-M. (2008). Stereo imaging with CUDA.

Stefano, L. D., Marchionni, M., Mattoccia, S. and Neri, G. (2004). A fast area-based stereo matching algorithm, *Image and Vision Computing* **22**(12): 983–1005.

Stem, J. E. (1989). *State Plane Coordinate System of 1983*, National Oceanic and Atmospheric Administration.

Stentz, A. (1994). Optimal and ecient path planning for partially known environments, *IEEE International Conference on Robotics and Automation*, pp. 3310–3317.

Stollnitz, E. J., DeRose, T. D. and Salesin, D. H. (1995). Wavelets for computer graphics: A primer, part 1, *IEEE Computer Graphics and Applications* **15**(3): 76–84.

Svoboda, T. and Pajdla, T. (2002). Epipolar geometry for central catadioptric cameras, *International Journal of Computer Vision* **49**(1): 23–37.

Swaminathan, R., Grossberg, M. D. and Nayar, S. K. (2001). Caustics of catadioptric cameras, *IEEE International Conference on Computer Vision*.

Sweet, W. (2008). Loser: Geopositioning - no payoff for galileo navigation system, *IEEE Spectrum* **45**(1): 48–49.

Szeliski, R. and Scharstein, D. (2004). Sampling the disparity space image, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(3): 419–425.

Takashi, M., Suzuki, T., Shitamoto, H., Moriguchi, T. and Yoshida, K. (2010). Developing a mobile robot for transport applications in the hospital domain, *Robotics and Autonomous Systems* **58**(7): 889–899.

Taylor, C. J. and Kriegman, D. J. (1993). Exploration strategies for mobile robots, *IEEE International Conference on Robotics and Automation*, pp. 248–253.

The Japan Society of Mechanical Engineers (2007). Contents.
URL: *http://www.jsme.or.jp/English*

Thrapp, R., Westbrook, C. and Subramanian, D. (2001). Robust localization algorithms for an autonomous campus tour guide, *IEEE International Conference on Robotics and Automation*, pp. 2065–2071.

Thrun, S. (1993). Exploration and model building in mobile robot domains, *IEEE International Conference on Neural Networks*, Vol. 1, San Francisco, CA USA, pp. 175–180.

Thrun, S. (2001). A probabilistic on-line mapping algorithm for teams of mobile robots, *The International Journal of Robotics Research* **20**(5): 335–363.

Thrun, S. (2010). Toward robotic cars, *Communications of ACM* **53**(4): 99–106.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*, MIT Press.

Tighe, P. J., Badiyan, S., Luria, I., Boezaart, A. P. and Parekattil, S. (2010). Robot-assisted regional anesthesia: A simulated demonstration, *Anesthetic and Anagelsia* **111**(3): 813–816.

Toko, Y., Debenest, P., Fukushima, E. F. and Hirose, S. (2004). Robotic system for humanitarian demining, *IEEE International Conference on Robotics and Automation*, Vol. 2, New Orleans, LA, pp. 2025–2030.

Tomatis, N., Nourbakhsh, I. and Siegwart, R. (2002). Hybrid simultaneous localization and map building closing the loop with multi-hypotheses tracking, *IEEE International Conference on Robotics and Automation*, Washington, DC USA, pp. 2749–2754.

Tsagarakis, N., Metta, G., Sandini, G., Vernon, D., Beira, R., Becchi, F., Righetti, L., Santos-Victor, J., Ijspeert, A., Carrozza, M. and Caldwell, D. (2007). iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research, *Advanced Robotics* **21**: 1151–1175.

Tsui, J. B.-Y. (2001). *Fundamentals of Global Positioning System Receivers*, John Wiley & Sons, Inc.

Tully, S., Kantor, G., Choset, H. and Werner, F. (2009). A multi-hypothesis topological slam approach for loop closing on edge-ordered graphs, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO USA, pp. 11–15.

Tungadi, F. and Kleeman, L. (2007). Multiple laser polar scan matching with application to slam, *Australasian Conference on Robotics*.

Tungadi, F. and Kleeman, L. (2009). Loop exploration for SLAM with fusion of advanced sonar features and laser polar scan matching, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, Missouri , USA, pp. 388–394.

Tungadi, F., Lui, W. L. D., Kleeman, L. and Jarvis, R. (2010). Robust online map merging system using laser scan matching and omnidirectional vision, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan.

Tur, J. M. M. (2007). Onto computing the for the odometry pose estimate of a mobile robot, *Conference on Emerging Technologies and Factory Automation*, pp. 1340–1345.

Tur, J. M. M. and Borja, C. A. (2007). Outdoor robot navigation based on a probabilistic data fusion scheme, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA USA, pp. 3733–3738.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition, *Journal of Cognitive Neuroscience* **3**(1): 71–86.

Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization, *IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 1023–1029.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(4): 376–380.

Valgren, C. and Lilienthal, A. J. (2010). Sift, surf & seasons: Appearance-based long-term localization in outdoor environments, *Robotics and Autonomous Systems* **58**(2): 149–156.

Čapek, K. (1920). Rossum's universal robots.

Weisstein, E. W. (2010a). Collinear, From MathWorld–A Wolfram Web Resource.
URL: *http://mathworld.wolfram.com/Collinear.html*

Weisstein, E. W. (2010b). Plane, From MathWorld–A Wolfram Web Resource.
URL: *http://mathworld.wolfram.com/Plane.html*

Wheatstone, C. (1838). Contributions to the physiology of vision. part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision, *Philosophical Transactions of the Royal Society of London* **128**: 371–394.

Williams, S., Pizarro, O., Webster, J., Beaman, R., Mahon, I., Johnson-Roberson, M. and Bridge, T. (2010). Autonomous underwater vehicle assisted surveying of drowned reefs on the shelf edge of the great barrier reef, australia, *Journal of Field Robotics* **27**(5): 675–697.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, USA, pp. 146–151.

Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M. and Nistér, D. (2006). Real-time global stereo matching using hierarchical belief propagation, *British Machine Vision Conference*.

Yang, R. and Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware, *IEEE International Conference on Computer Vision and Pattern Recognition*.

Yershova, A., Jaillet, L., Siméon, T. and LaValle, S. M. (2005). Dynamic-domain rrts: Efficient exploration by controlling the sampling domain, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 3856–3861.

Zelinsky, A. (1992). A mobile robot exploration algorithm, *IEEE Transactions on Robotics and Automation* **8**(6): 707–717.

Zhang, A. M. and Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments, *International Journal of Robotics Research* **28**(3): 331–356.

Zhang, H., Li, B. and Yang, D. (2010). Keyframe detection for appearance-based visual slam, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, pp. 2071–2076.

Zhou, X. S. and Roumeliotis, S. I. (2006). Multi-robot slam with unknown initial correspondence: The robot rendezvous case, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 1785–1792.

Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R. and Sawhney, H. S. (2007). Ten-fold improvement in visual odometry using landmark matching, *IEEE International Conference on Computer Vision*, pp. 1–8.