MULTISENSOR BASED ENVIRONMENT MODELLING AND CONTROL APPLICATIONS FOR MOBILE ROBOTS

DILAN AMARASINGHE

Multisensor Based Environment Modelling and Control Applications for Mobile Robots

© Dilan Amarasinghe, B.Sc (Hons)

A thesis submitted to the

School of Graduate Studies

in partial fullfillment of the

requirements for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

July, 2008

St. John's Newfoundland

Abstract

For autonomous operations of mobile robots, three key functionalities are required: (a) knowledge of the structure of the world in which it operates, (b) ability to navigate to different positions autonomously using path planning algorithms, and (c) ability to precisely localize itself for the task execution. This thesis will address some of the issues related to the first and third requirements. The knowledge of the structure of the environment can be represented in several forms such as: 3D models, 2D wall plan, 2D plan of landmarks, and position and velocity of moving objects. Efficient navigation and obstacle avoidance methods are often aided by information about the structure of the environment in any of the above forms. At the end of each navigation task the robot has to execute an assigned task such as pick and place or park. In most cases these tasks require precise localization of the robot where the degree of precision required depends on the task specification.

Taking these functions into consideration, this thesis addresses the issues of learning the structure of the world by constructing a visual landmark map of static landmarks. Additionally, it provides a solution to the precise localization problem of the mobile robot using a vision based hybrid controller. On the subject of the visual landmark map, the thesis describes a landmark position measurement system using an integrated laser-camera sensor. The traditional laser range finder can be used to detect landmarks that are direction invariant in the laser data. The processes that are dependent on the presence of directional invariant features such as navigation and simultaneous localization and mapping (SLAM) algorithms will fail to function in their absence. However, in many instances, it is possible to find a larger number of landmarks that are visually salient using computer vision. The calculation of depth to a visual feature is non-trivial due to the loss of depth information in the sensor

model. While considering the drawbacks and limitations in laser and camera as a sensor, this thesis proposes a novel integrated sensor method to calculate position of the visual features. In addition, a comprehensive experimental analysis is presented to verify the sensor integration method for the EKF based SLAM algorithm.

For effective operation of a robot's SLAM algorithm, it is necessary to identify dynamic objects in the environment. In order to achieve this objective a novel robust technique for detecting moving objects using a laser ranger mounted on a mobile robot is presented. After initial alignment of two consecutive laser scans, each laser reading is segmented and classified according to object type: stationary, non-stationary or indeterminate. Laser reading segments are then analyzed using an algorithm to maximally recover the moving objects. The proposed algorithm has the ability to recover all possible laser readings that belong to moving objects. The developed algorithm is verified using experimental results in which a walking person is detected by a moving robot.

Finally, a novel vision-based hybrid controller for parking of mobile robots is proposed. Parking or docking is an essential behavioral unit for autonomous robots. The proposed hybrid controller is comprised of a discrete event controller to change the direction of travel and a pixel error driven proportional controller to generate motion commands to achieve the continuous motion. At the velocity control level, the robot is driven using a built-in PID control system. The feedback system uses image plane measurements in pixel units to perform image-based visual servoing (IBVS). The constraints imposed due to the nonholonomic nature of the robot and the limited field of view of the camera are taken into account in designing the IBVS-based controller. The controller continuously compares the current view of the parking station against the reference view until the desired parking condition is achieved. A comprehensive analysis is provided to prove the convergence of the proposed method. Once

the proposed parking behaviour is invoked, the robot has the ability to start from any arbitrary position to achieve successful parking given that initially the parking station is in the robot's field of view. As the method is purely based on vision the hybrid controller does not require any position information (or localization) of the robot. Using a Pioneer 3AT robot, several experiments are carried out to validate the method. The experimental system has the ability to achieve the parking state and align laterally within 1 cm of the target pose.

Acknowledgments

Firstly, I would like to thank my supervisors Dr. George Mann and Dr. Raymond Gosine for their support and continuing encouragement, without which I could not have completed this programme. They have encouraged me throughout my candidature, offered me words of wisdom when needed, and allowed me the freedom to explore new areas. Also I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), Canadian Foundation for Innovation (CFI) and Presidents Fund in Sri Lanka for providing financial assistance for this research.

I would thank every one of my colleagues at the Intelligent Systems Lab (ISLAB): Rajib, Momotaz, Jeffery, Farid and Nusrat, from whom I learnt so much and for their friendship during my stay. I would also like to thank numerous work term students who have worked in the lab and contributed in various ways to build and modify the robots.

Lastly, I appreciate all the care and unwavering support from my parents, brother and sister during the long years that I have been away from home; to whom I dedicate this thesis.

Contents

1 Introduction **1**

1.1 Mobile robot navigation 1

1.2 Problem statement 6

1.2.1 Case for multi-sensor simultaneous localization and mapping (SLAM) 7

1.2.2 Case for moving object detection 8

1.2.3 Case for visual servoing of mobile robots 10

1.3 Research objectives 11

1.4 Contributions of the thesis 12

1.5 Thesis organization 13

2 Simultaneous Localization and Mapping (SLAM) and Related Issues **15**

2.1 Preliminaries 18

2.1.1 Coordinate systems and transformations 18

2.1.2 Mobile robot and sensor model 22

2.2 SLAM Formulations 24

2.2.1 SLAM Problem 25

2.2.2 Extended Kalman filter solution 26

2.2.3 Alternative solutions to the SLAM problem 37
2.3 Applications 40
2.3.1 Application of SLAM 41
2.3.2 Multisensor approach to SLAM 44
2.3.3 SLAM in Dynamic Environments 45
2.4 Conclusion 46

3 Multisensor Landmark Detection and Localization for SLAM 48
3.1 Introduction 48
3.1.1 Related Work 50
3.1.2 Objective 52
3.2 Landmark Detection and Position Estimation Using a Single Sensor . 54
3.2.1 Landmark localization using computer vision 56
3.2.2 Landmark localization using laser data 58
3.3 Calibrated Laser-Vision Sensor 59
3.3.1 Visual Landmark Detection 61
3.3.2 Sensor calibration 63
3.3.3 Measurement Model 64
3.4 EKF based SLAM Algorithm 66
3.5 Experiments and Results 66
3.6 Conclusion 74

4 Modeling of Environmental Dynamics 75
4.1 Background 76
4.1.1 Challenges in Robotics 77
4.2 Moving Objects and SLAM 78
4.3 Laser scan segmentation 81

4.3.1 Definitions 82
4.3.2 Segmentation Algorithm 84
4.3.3 Parameter selection 85
4.3.4 Moving Object Detection 87
4.3.5 Experimental Results 91
4.4 Moving Object Detection and Position Calculation 92
4.4.1 Moving Object Position Calculation 92
4.4.2 Experimental Results 94
4.5 Conclusion 95

5 Active Control and Robot Parking **99**

5.1 Introduction 99
5.1.1 Related Work 101
5.1.2 Motivation 103
5.2 Visual Servoing System 104
5.2.1 Parking Station 104
5.2.2 Robot and camera model 105
5.2.3 Overall Vision Based Control System 106
5.2.4 Assumptions 107
5.3 Control Strategy 108
5.3.1 Finite State Machine 109
5.3.2 Details of the States 110
5.4 Analysis of Convergence 115
5.4.1 Proof of convergence 116
5.5 Experiments and Results 125
5.5.1 Image processing 125

5.5.2 Tuning 126
5.5.3 Results 126
5.6 Conclusion 133

6 Conclusion **135**

6.1 Discussion 135
6.2 Conclusion 137
6.3 Publications Resulting from the PhD Program 138
6.3.1 Journal Papers 138
6.3.2 Refereed Conference Papers 139
6.4 Future Work 139

List of Tables

3.1 The measurement of noise levels of the respective sensors that is used in the SLAM. 70

5.1 System parameters and their selection criteria. 127

5.2 Selected parameter values in parking system. 128

5.3 Summary of the set of experiments. 128

List of Figures

2.1 Coordinate frame labeling and corresponding notations. 19

3.1 Block diagram of (a) the proposed multisensor based SLAM process and (b) the multisor sensor based SLAM process presented in [1]. The proposed method fuses the information from each sensor at a lower level whereas the method in (b) fuses the information at map level. . 53

3.2 Sensitivity of the uncertainty of the feature localization. The uncertainty is quantified by the area of the ellipse representing 95% confidence. $p_{\max} = 1.75mm$, $v = 0.092m/s$ and $\omega = 4 \times 10^{-3} rad/s$. ($\sigma_p = 10.9 \times 10^{-6} m$, $\sigma_{\dot{p}} = 0.3 \times 10^{-4} m/s$, $\sigma_v = 7.8mm/s$, $\sigma_\omega = 10^{-6} rad/s$) 57

3.3 A typical laser reading in an indoor environment where there are not sufficient direction invariant features. 59

3.4 The camera and the laser range sensor used in the experiments. . . . 60

3.5 Coordinate frames of calibrated laser-vision sensor 60

3.6 Line feature detection using artificially generated corner features. . . 62

3.7 Detected line features using Hough transformation and the corner based method. 62

3.8 Calibration curve for mapping between the field of view of the camera and the field of view of the laser range finder. 64

3.9 Interpolation of the range to the line feature. 65
3.10 The curve constructed on the image plane by connecting the laser readings mapped from the laser coordinate frame to image frame. . . 67
3.11 The landmarks detected by the camera and their bearing angle superimposed on laser readings. 69
3.12 Number of landmark features detected by vision and laser system. . . 70
3.13 Results of a localization and mapping of the first robot run: (a) with odometry, and (b) using EKF and vision-laser landmark localization (see the attached video file for incremental map building along with the current image frame). 71
3.14 Results of a localization and mapping of the second robot run: (a) with odometry, and (b) using EKF and vision-laser landmark localization (see the attached video file for incremental map building along with the current image frame). 72
3.15 3σ bounds of the localization errors of the first experiment. 73
3.16 3σ bounds of the localization errors of the second experiment. 73

4.1 Typical laser scans from a stationary robot. The object on the left hand side moves downward in a negative y-direction. 83
4.2 Perfectly separated object positions. 88
4.3 Partially separated object positions. 88
4.4 Object moving away from the scanner. 89
4.5 Partial laterally separated object positions. 90
4.6 The detection of a moving object similar to case 2. (a) Two laser scans. (b) Detected moving object. 92

4.7 The detection of a moving object similar to case 5. (a) Two laser scans. (b) Detected moving object. 93
4.8 (a) The final moving object segment, its centroid of the convex hull (calculated object position) and the bounding rectangle. (b) The actual view of the object. 94
4.9 Track of the walking person at a distance about 1.5m - 2.5m from the robot. The blue bounding boxes represent the detected moving objects while the black stars represent the possible torso position of the person when the scan segments from the two legs are available. 96
4.10 The track of a person walking approximately 5m from the robot. The blue bounding boxes represent the detected moving objects while the black stars represent the possible torso position of the person when the scan segments from the two legs are available. 97

5.1 Reference and actual image of the parking station 105
5.2 Differentially driven robot model. 106
5.3 The control objective and the introduction of the axis. 107
5.4 Overall block diagram of the control system. 107
5.5 Proposed finite state machine 110
5.6 The control scenarios in *forward* state. The corresponding control error is labeled as e. (a) If the robot starts with $x > 0$. (b) If the robot starts with $x < 0$. (the global x-y reference frame is defined in Fig. 5.7) 112
5.7 The effect of oversteering in the forward state on the final position. Q_2 and Q_3 are desirable robot paths while Q_1 and Q_4 are undesirable. . . 113

5.8 The relationship between c and $|f_{12}|$. The allowable maximum for c is defined assuming that the parking station appears symmetric in the images when the robot is at the exact parking condition ($\delta_u = 0$). . . 114

5.9 Typical robot position and oscillation envelope about the y-axis. . . . 118

5.10 Properties of $|\tan(\beta_1) - \tan(\beta_2)|$. (a) The behavior of $|\tan(\beta_1) - \tan(\beta_2)|$ with $d = 0.12$m and $r_{\min} = 0.2$m and (b) the plot of the maximum of θ for the practical limits of d and $r_{\min}$. 122

5.11 The P3AT robot at (a) a typical starting position and (b)the parked position of the robot. 128

5.12 Results of the Experiment I. (a) The x-y trajectory of the robot during parking. (b)The trajectory of the feature positions during parking. (c) The initial and final images acquired during the parking routine. (d, e)The commanded and actual heading and rotational velocities of the robot during parking. 129

5.13 Results of the Experiment II. (a) The x-y trajectory of the robot during parking. (b)The trajectory of the feature positions during parking. (c) The initial and final images acquired during the parking routine. (d, e)The commanded and actual heading and rotational velocities of the robot during parking. 130

5.14 Results of the Experiment III. (a) The x-y trajectory of the robot during parking. (b)The trajectory of the feature positions during parking. (c) The initial and final images acquired during the parking routine. (d, e)The commanded and actual heading and rotational velocities of the robot during parking. 131

5.15 Effects of the c-adaptation on the path of the robot. (a) Successful parking with c-adaptation. In this case the allowed maximum of the c value is 478 and c varies from a c_{min} of 275 to a c_{max} of 470. (b) Parking with a higher static value of c (c) Parking attempt with a low static value of c. 132

5.16 Final parking positions for 20 different arbitrary starting positions. Note: Some readings may overlap each other. 132

List of Symbols

Chapters 2 and 3

$\hat{x}$: mean of the estimated state vector x

$\Sigma_{\mathbf{x}}$: covariance of the estimated state vector x

$\zeta(k)$: innovation of the landmark measurments at time step k

$\mathbf{x}_v(k)$: robot pose at time step k

$v(k)$: robot heading velocity at time step k

$\omega(k)$: robot rotational velocity at time step k

$\mathbf{z}_i(k)$: measurement vector of the i-th landmark at time step k

$r_i(k)$: range to the i-th landmark at time step k

$\theta_i(k)$: bearing to the i-th landmark at time step k

$\mathbf{x}_{ij}$: spatial relationship between coordinate frames i and j

$\oplus$: compounding operator for coordinate frames

$\ominus$: inverse operator for coordinate frames

$\nabla_{\oplus}$: Jacobian of the compounding operator

$\nabla_{\ominus}$: Jacobian of the inverse operator

M : the generic map of the environment independent of the presentation

$Z(1:k)$: history of measurements from the time step 1 to k

$U(1:k)$: history of robot control inputs from the time step 1 to k

Chapter 4

L_C : laser range data from the current scan

L_P : laser range data from the previous scan

A_C : laser readings in current scan that represent stationary objects

A_P : laser readings in previous scan that represent stationary objects

M_C : laser readings in current scan that represent moving objects

M_P : laser readings in previous scan that represent moving objects

N_C : laser readings in current scan that are out of the field of view of the sensor in the previous scan

N_P : laser readings in previous scan that are out of the field of view of the sensor in the current scan

O_C : laser readings in previous scan that will be occluded by the laser readings in the current scan

O_P : laser readings in current scan that will be occluded by the laser readings in the previous scan

X_y^j : j-th reading in the laser reading set X_y

$r(\cdot)$: range to a given laser reading in meters

X_y^s : the set of laser reading segments in the laser reading set X_y

Δt : time interval between the laser readings used in the moving object detection

δt : acquisition rate of the laser scanner

δd_c : closest point detection threshold

$V_{\min}$: minimum detectable relative velocity of a moving object

Chapter 5

P_i^r : i-th feature in the reference image

P_i : i-th feature in the actual image

u_i^r : position of the i-th feature in the reference image

u_i : position of the i-th feature in the actual image

A : width of the parking station in the actual image

A^r : width of the parking station in the reference image

E_A : error in the width of the parking station

e_i : error in the position of the i-th feature

W : overall width of the image

δA : threshold values that define the acceptable limits of the error, E_A

δu : threshold values that define the acceptable limits of the errors, e_i

v : the heading velocity of the robot

ω : the rotational velocity of the robot

X : the set of states in the state machine

Σ : the set of events associated with the state machine

α : the set of transition functions in the state machine

x_0 : initial state of the state machine

X_m : marked or terminating states in the state machine

S_t : *start* state

F : *forward* state

R : *reverse* state

S_p : *stop* state

ϵ_i : i-th event in the set Σ

v_r and v_c: forward and reverse heading velocities

f_1 : the distance between second and first features in the actual image

f_2 : the distance between third and second features in the actual image

f_{12} : the difference between f_1 and f_2

x and $\dot{x}$: robot position in x-axis and its rate of change

y and $\dot{y}$: robot position in y-axis and its rate of change

ϕ : robot orientation

λ : focal length of the camera

p : number of pixels per meter in the camera sensor

c : the distance from the edge of the image that a feature is allowed to travel

β_1 : actual angle swept by the second and third features

β_2 : actual angle swept by the first and second features

$A_{\min}$: the minimum allowed width of the parking station

$r_{\max}$: the furthest that the robot is allowed to travel from the center feature during the parking process

$c_{\max}$: the maximum practical value of the c

$d\theta$: angle that defines a marginal region around the y-axis

B : actual physical width of the parking station in meters

List of Abbreviations

CML	:	Concurrent Mapping and Localization
EKF	:	Extended Kalman Filter
FSM	:	Finite State Machine
GNN	:	Global Nearest Neighbor
GPS	:	Global Positioning System
ICNN	:	Individual Compatibility Nearest Neighbor
IC	:	Individual Compatibility
IBVS	:	Image Based Visual Servoing
JC	:	Joint Compatibility
JCBB	:	Joint Compatibility Branch and Bound
MDA	:	Multi-Dimensional Assignment
NEES	:	Normalized Estimation Error Squared
NMEE	:	Normalized Mean Estimation Error
NIS	:	Normalized Innovation Squared
PBVS	:	Position Based Visual Servoing
SLAM	:	Simultaneous Localization and Mapping
SEIF	:	Sparse Extended Information Filter
SLMOT	:	Simultaneous Localization and Moving Object Tracking
SLAMMOT	:	Simultaneous Localization, Mapping and Moving Object Tracking
SIFT	:	Scale Invariant Feature Transform
SJPDAF	:	Sample-based Joint Probabilistic Data Association Filter

Chapter 1

Introduction

1.1 Mobile robot navigation

During recent years, mobile robotic research has been expanded to include many industrial and service applications and the navigation of robots has been studied and implemented in a range of applications, such as: navigation in well structured buildings such as homes, offices and warehouses; exploration in harsh environments such as underwater, deserts, abandoned mines, volcanos, battlefields, space; consumer applications such as robotic vacuum cleaners, lawn mowers, golf carts [2, 3, 4, 5, 6, 7]. Although robots provide some unique benefits to humans in exploration and working in harsh environments, in the future robotic applications will be expanded to large scale environments with multiple robots. Health care applications and servicing robots in airports are some targeted future applications. In general indoor service mobile robotic applications, the robot will respond to a user request and navigate in a dynamic environment, where other moving and unexpected obstacles are envisaged. In such a scenario the robot performs tasks at specific places, returns back to its parking station, and waits in idle mode until it receives a new assignment. In such

a scenario the users can be humans or other robots that may require the service of a particular robotic agent. When a robot is introduced into an indoor environment that it has never seen and has no other information such as maps, it requires some fundamental features and capabilities to be adaptive and also to perform its tasks in a useful manner. In addition to general automation requirements of a mobile robot, this thesis considers the following three groups of major capabilities. These requirements have been identified at a higher level, aside from the requirement of having robust low-level intelligent controllers of robots maneuvering.

1. **Localization and mapping** : Robotic mapping is one of the key capabilities a robot requires to operate in environments when prior information about the structure of the environment is unavailable. In such circumstances a robot is required to explore and acquire information about the structure of the operating workspace from a series of sensor measurements collected by the robotic system or a map. In order to build an accurate map of a relatively large workspace, the robotic map building process requires an accurate estimate of the robot pose with respect to its initial start position or with reference to an already established map. The latter process is called self-localization. The mapping capability removes the requirement of having a pre-built infrastructure for navigation where the robot has the capability to select and choose landmarks or features for alignment and navigation. This capability is regarded as one of the important features required in developing an autonomous robot [8]. In the absence of a GPS, for operation in indoor, underground or underwater applications, one can integrate the odometry and steering angle data obtained from the encoders of the wheeled robot to generate discrete localization information for mapping - the process known as dead reckoning. Usually, accurate odometry

data based localization is not possible due to associated errors (both deterministic and otherwise) in the sensor data. These errors can accumulate due to finite sensor resolution and other errors introduced by skidding/slipping, kinematical errors in wheels such as wheel misalignments, wear of wheels or traveling in uneven terrains [9]. Therefore accurate localization is a challenging issue and as a result, without having accurate localization information the robot will be unable to generate a consistent map. Therefore, the two problems, map-building and self-localization have to be solved simultaneously. Due to this dual nature of the estimation problem, it is often termed Concurrent Mapping and Localization (CML) or more commonly Simultaneous Localization and Mapping (SLAM), a term first coined by Leonard and Durrant-Whyte [10]. SLAM has been an active research area in mobile robotics since the late 80's and is largely motivated by the seminal paper on uncertain spatial relationships in robotics by Smith et al [11]. The Kalman filter based developments later became the standard method for many SLAM implementations although there are other methods available such as expectation maximization [12], scan matching [13] and genetic algorithms [14] that have been devised later for solving the SLAM problem. Since the Kalman filter based method constructs and updates the map while providing an estimate of the robot position in a uniform manner by accommodating sensor uncertainties, this method became the most popular among SLAM researchers. Although it remains extremely popular to this date, various drawbacks of the Kalman filter method, such as linearization errors, its inability to represent non-gaussian probability distributions of the variables and computational efficiency when building large maps, have been identified. In order to overcome these drawbacks several variations and new methods such as Sparse Extended Information Filters (SEIF) [15] and FastSLAM [16] came

into the forefront.

2. **Modeling environmental dynamics**: General purpose robots that operate in highly dynamic environments are required to share a common workspace with other agents. Identification of static objects and accommodating them into the existing map can be automatically performed in a SLAM filter. An accurate and complete description of both moving and static objects has two key advantages in mobile robotics. Firstly, it provides information necessary for the robot control algorithm to avoid collisions with those unexpected (unmapped) obstacles. Rather than considering them as static obstacles in each time step, by detecting and classifying them as dynamic objects, the robot can move efficiently and intelligently avoid those moving objects through planning ahead. Secondly, the identified moving objects can be removed from the robotic mapping process in order to generate an accurate map. Most mapping algorithms, including existing SLAM techniques, assume a static environment during the mapping process. Although this assumption would hold for mapping robots that are operating only when the environment is free from any moving objects, in more general applications the robot is required to build and maintain a map of the environment during its operational lifetime. Such an assumption often leads to inconsistencies in the map and also in the robot localization. After the proper identification of the sensor data that corresponds to the moving objects, they can be removed from the mapping and localization process leading to a more accurate map. The static obstacles can be identified using range sensors and avoidance of them can be accomplished using techniques such as potential field methods in behaviour based control [17]. However, identification of moving objects by a moving robot is quite a challenging task. In the past several

techniques have been developed using vision [18, 19] and laser ranger [20, 21, 22]. The primary goal of these moving object detection methods is to identify sensory data that corresponds to a moving object.

3. **Precise alignment of the robot against landmarks**: Visual servoing has been a popular and important research area in both manipulator and mobile robot control [23, 24]. Use of traditional visual servoing methods is more challenging due to the varying types of mobile robot platforms and the lesser number of degrees of freedoms as compared to manipulators. The typical service robot performs a variety of tasks, such as material handling, where the robot is required to align itself to a specific configuration to accomplish the assigned tasks. In another application, the robot can navigate while aligning itself with respect to a set of specific features observed by the robot. Wall following [17] and optical guidance [25] are typical examples where robots are required to align during navigation. These locations are usually identified by some special landmarks, such as the target object or artificial landmarks placed by the user. Generally, these landmarks are arbitrarily placed without any reference to the robot's internal map and can be changed by users at any time. During its operation when the robot is in the vicinity of the landmark, it should be able to identify the landmark as well as align itself so that it will be able carry out the intended function. Computer vision can be easily adapted to identify visually distinct objects or landmarks. In case of robot alignment, the robot controller has to overcome the nonholonomic constraints of the robot and the field of the view constraints of the camera. Most vision based solutions to the robot alignment problem can be categorized into: conventional controllers with smooth velocity control [26, 27] and hybrid controllers [28, 29, 30, 31, 32]. Other controllers us-

ing intelligent techniques such as fuzzy logic and neural networks [33, 34, 35, 36] show improved performances. However, generally they have to be trained on each landmark before they can be deployed.

All these tasks primarily depend on sensory information for their operation. Additionally, each of those tasks can mutually benefit from each other. As an example the landmarks on the detected moving objects can be removed from the mapping process yielding more accurate maps. Therefore, this research will focus on the development of efficient methods for unified localization and mapping and moving object detection to better utilize the sharable information between modules rather than each module operating on its own and collaborating at a higher level. Additionally, this thesis describes an accurate and highly adaptable robot alignment algorithm.

1.2 Problem statement

Various types of sensors are being used and researched in mobile robotics. It has been well established that autonomous operation of a general purpose service robot requires several types of sensor modalities. Thus, by using a more distributed and integrated multi sensor approach rather than a single sensor for each task, the performance of each task can be further enhanced resulting in an efficient utilization of the onboard resources and improved performance. In this thesis several essential aspects of indoor autonomous robotics are explored and discussed leading to contributions in each aspect in improving performance while mitigating some drawbacks of previous methods.

1.2.1 Case for multi-sensor simultaneous localization and mapping (SLAM)

Accurate localization and map building are essential tasks for autonomous operation of mobile robots. These operations take on extra significance when a robot is introduced into a new environment where it doesn't have any *prior* information. In such cases the robot essentially starts with a blank memory. As the robot explores and navigates in a new environment it will gradually build the map and localize itself against the newly created map. In subsequent revisits to a previously mapped area of the environment, the robot can update the map to reflect any changes to the map since the time it was last built. Additionally, the robot can refine the map for better accuracy and fidelity when it revisits an already mapped area. Localization and mapping has been an extremely active field of research in mobile robotics during the last decade. Hence there are numerous techniques for map building and localization. Among them landmark based methods have been the most widely explored.

Various types of sensors such as laser range scanners [37], millimeter wave radar [38], sonar [39] and cameras [40, 41] have been used in landmark based SLAM implementations. Due to the high degree of precision offered by the the laser range scanner [42], it is the most popular sensor that is used in majority of the reported SLAM methods. However, in most indoor environments, finding an adequate number of landmarks using a laser range scanner is difficult and sometimes impossible. It is known that computer vision provides a rich set of visual information about the robot's environment that can be used to identify visually salient landmarks (e.g. corner features [43, 44], lines [45], etc.). However, most reported results in vision based SLAM suffer from several drawbacks, which include (1) the requirement of a large number of landmarks [46, 47], (2) use of specific visual patterns [48, 40], and (3) small

scale navigation (or camera movement) [46]. However, in real world scenarios these conditions are rarely satisfied as robots are required to navigate long distances and sometimes in areas where there are a low number of recognizable landmarks. The main drawback in vision based landmark localization is the large uncertainties associated with vision based measurement of landmark positions. This is primarily due to the loss of depth information in the sensing model. The alternative solution is to fuse information from laser and vision sensors. Although fusion of information in the cartesian space in existing multi-sensor methods [1] improves the number as well as quality of the detected landmarks, this does not include all the salient visual landmarks but only those which are supported by laser data. However through effective integration of both vision and laser sensor calibration, the complementary properties of both sensors can be exploited to localize the landmarks more accurately and build a map that is rich in landmarks.

1.2.2 Case for moving object detection

Robots usually operate in dynamic environments with humans and other robots moving around. Thus, in mobile robot navigation the dynamic nature of the environment has to be considered in path planning and navigation algorithms. Additionally, when the robot needs to map the environment, the sensory data corresponding to the moving objects have to be removed from the mapping process. Thus, in order to support path planning and mapping in dynamic environments it is important to have robust moving object detection capability built into the robot. Additionally, the moving object detection can support new functionality such as following a moving object or monitoring a moving object. Most of the existing moving object detection methods [49, 50, 20, 21, 22, 51, 52, 53] use scan registration of time elapsed laser data

prior to analyzing the differences between the respective laser scans. While most of these methods are developed as general moving object trackers, some are explicitly treated as moving people detectors and employ special properties that are related to properties of scans related to moving people. One key problem in moving object detection in registered scan data is that, when the laser data is highly contaminated with data from moving objects, most conventional registration methods [13] would fail to register the scans accurately. This low precision in registration will lead to highly erroneous moving object detection, i.e. the accuracy of the moving object detection method is as good as the accuracy of the laser scan registration algorithm. Additionally, most moving object trackers do not employ techniques to capture the complete moving objects [51, 53, 22]. This would normally lead trackers to wrongly classify moving objects and further, when incomplete, to locate them at an erroneous position. Therefore, any laser based moving object tracker has to accomplish the following:

1. It should be able to detect moving objects in highly cluttered environments. In most highly cluttered environments, moving objects are typically humans sharing the same workspace as the robot. Therefore the algorithms should be able to detect randomly moving nonuniform objects.

2. It should be able to extract all sensor data corresponding to each moving object. Through the correct identification of all the data, while it is possible to support the first condition it will also be able to support the stationary landmark detection for SLAM.

1.2.3 Case for visual servoing of mobile robots

In general vision based robot control [23, 24], the robot is controlled using the difference (error) between the measured or estimated values and the desired value. In visual servoing these quantities are either directly expressed in image plane coordinates or calculated or estimated using measurements made in image plane coordinates, usually in cartesian coordinates. Hence, visual servoing can be categorized as either direct image-based visual servoing (IBVS), when the errors are expressed in image plane units (usually in pixels); or position-based visual servoing (PBVS) when the errors are expressed in cartesian coordinates. In PBVS, as the error quantities are expressed in cartesian coordinates, traditional path planning and control techniques can be used to achieve the control objectives. However, in PBVS methods the error has to be calculated from the image plane quantities through coordinate transformations. It is well known that these coordinate transformations usually introduce errors due to calibration errors. Due to this reason the IBVS method is popular as a more robust and flexible control technique.

Differentially driven mobile robots have nonholonomic constraints [24, 54]; i.e. a robot cannot move sideways. Further, the limited field of view in vision systems generally imposes an additional constraint on the control law. Thus, image based visual servoing of mobile robots is a challenging task given the limited number of degrees of freedom (usually two) and the limited field-of-view available in the vision system. For nonholonomic robots visual servoing can be applied for path following [25, 55, 56] or it can be used to align the robot with a given pose. In vision based robot alignment techniques, the robot is aligned against a fixed set of features, so that the robot will satisfy a predefined control objective [28, 31]. Robot parking controllers generally belong to either conventional smooth controllers or hybrid controllers. In

conventional controllers [26, 27] the robot is aligned to a set of features seen by the camera using smooth control of robot velocity. Hybrid controllers for vision based robot control [28, 29, 30, 31, 32] allow the robot velocities to be controlled in a discontinuous manner. In hybrid controllers, a finite state machine is used to define a set of states to reflect multiple operational contexts in a robotic task.

The typical issues in the continuous controllers include: the convergence of the solutions when starting from an arbitrary robot pose [26], and the inability to obtain a unique final position [27]. Some improvements have been reported in vision-based parking controllers using intelligent control techniques such as fuzzy logic and neural networks [33, 34, 35, 36]. Generally, fuzzy logic and neural network based controllers perform satisfactorily, but they do not guarantee convergence. Hybrid controllers equipped with a state machine have different control algorithms for each state. Hence, multiple switching control algorithms give rise to discontinuous control of the robot velocities. Most vision based hybrid controllers [28, 29, 30, 31, 32] use this property to overcome nonholonomic and field of view constraints [54] of the system. Therefore in this research hybrid control strategy was chosen in the design of a new parking controller. Lyapunov techniques [57] have been widely adopted in hybrid closed-loop parking controllers [28, 29, 30, 32]. Limitations of the past hybrid methods include: the rapid switching behaviour around the parking position (zeno behaviour) [28, 31]; partial utilization of the available field of view of the camera [32, 31]; and not explicitly addressing the field of view constraints of the camera [29, 30].

1.3 Research objectives

The main focus of this research is to develop a multisensor simultaneous localization and mapping technique and other necessary functions for moving object detection

and visual servoing.

Objective I: Simultaneous localization and mapping: to develop a multisensor SLAM implementation that exploits the ability of the computer vision to reliably detect landmarks and the accuracy of the laser scanners to measure the range to the detected landmarks.

Objective II: Moving object detection: to identify complete moving objects in the environment using a laser range scanner mounted on a moving robot.

Objective III: Visual servoing of mobile robots: to develop a hybrid controller for visual servoing with ability to efficiently overcome the nonholonomic nature of the robot and the constraints imposed by the limited field of view of the off-the-shelf cameras.

1.4 Contributions of the thesis

The resulting contributions of this thesis can be highlighted as follows:

1. **Large scale multi–sensor SLAM**: The large scale landmark based EKF based SLAM was developed by fusing data from a monocular vision system and laser range scanner. The camera images are used to detect the visual landmarks and then the laser range scanner is used to measure the range to the detected landmarks. The proposed landmark detection method, while providing a higher number of landmarks, will enable high repeatability in the detection of landmarks in the presence of sensor noise.

2. **Moving object detection**: A novel moving object detection algorithm was developed using the segmentation of laser range data. The algorithm first registers two subsequent laser scans and then segments the scan data. Finally,

the laser scan segments will be classified into three categories: moving objects, stationary objects, or indeterminate.

3. **Hybrid controller for image based visual servoing of a mobile robot**: A novel image based visual servoing method was developed to control nonholonomic mobile robots against a set a visual features. The proposed controller avoids the oscillatory behavior (zeno behavior) that is exhibited by the traditional controllers while intelligently adapting the controller parameters for the maximum use of the available field of view.

1.5 Thesis organization

Chapter 1 highlights the main areas of research in mobile robotics and associated issues. This chapter also provides a list of issues that is addressed in this thesis and its contributions. Chapter 2 provides an overview of the SLAM problem and the state of the art solution that has been proposed to solve the problem. Additionally chapter 2 highlights some of the areas of the SLAM that can be improved using multisensor implementations for landmark and moving object detection. Chapter 3 details the proposed multisensor landmark detection and localization method for SLAM. Chapter 4 provides the details of the moving object detection algorithm that can be used to identify moving objects. This type of moving object identification can be used to improve the robustness of SLAM through the elimination of the non-stationary components of the map. In almost every SLAM implementation the final robot position and map estimates will have some bounded uncertainty. Chapter 5 introduces a visual servoing method for a nonholonomic mobile robot that can be used to achieve accurate positioning using additional visual features and thus overcome the bounded uncertainty of the localization in SLAM. Chapter 6 draws the conclusions

of this research and discusses future directions.

Chapter 2

Simultaneous Localization and Mapping (SLAM) and Related Issues

In autonomous robot navigation, localization or the question "where am I?", with respect to a *priori* map is a fundamental, well known, and widely studied problem. Direct calculation of the position of the robot from odometry data is highly susceptible to sensor noise and unpredictable movements such as slippage of the robot. These errors accumulate over time and can very quickly render the calculated position of the robot useless. Thus, in order to reliably localize a robot, an accurate map of the environment should be used as the reference. A map representing the robot's workspace can take many different forms, such as: 2D line map, landmark position map, view (appearance) map etc. Robot localization becomes a challenging task when the map of the operating area is (1) completely unknown, (2) partially known, or (3) if the robot is operating in a highly dynamic environment with many moving objects. Thus it is essential for the robot to have sufficient ability to map the environment

robustly by itself. This is more important as it enables the robot to operate in new workspaces without any human intervention, thus allowing for greater flexibility in their deployment. In order to build a map from sensor data acquired at different poses in the workspace, the robot should have a reliable knowledge about the differences between relative poses of the robot at which the sensor readings are taken. This is essentially the relative localization problem. However in order to solve the mapping problem the localization problem is also required to be solved simultaneously using the current up to date map. This dual problem is widely known in robotics research as the simultaneous localization and mapping (**SLAM**) problem or concurrent mapping and localization (**CML**).

SLAM has been addressed using many different algorithms as discussed later in this chapter. There are some characteristics of the problem that have to be addressed by any algorithm in order for it to be useful.

Nonlinearity of the robot model Most mobile robot models used today (e.g. unicycle model) are inherently non-linear. Therefore the estimation frameworks that are used for linear systems such as the Kalman filter have limitations in solving the SLAM problem. Usually the robot model is linearized at the current pose, and velocities or other techniques such as sample based methods (e.g. particle filter) that can handle the nonlinearities have to be used.

Noisy sensors Sensor measurements are inherently noisy. Some sensors are noisier than others and even the same type of sensors from different manufacturers or units in the same batch may show different noise levels. Therefore the key issue in addressing the sensor noise is the development of a reliable statistical model of the sensor that represents a wide spectrum of noise processes.

Data association When a sensor reading is taken, the robot will identify landmarks

or other key features from sensor data. In the subsequent sensor readings with the same field of view, the same objects will be visible. The SLAM algorithm should be able to associate the same objects that were seen in two sensor readings as the same features and establish their correspondences by assigning a unique identity. This is of primary importance, because without the correspondence the SLAM process will risk initializing new features in the map when in essence they are the same features already in the map.

Loop closing Loop closing is a problem closely related to data association but at a more global level. When a robot revisits an area that it has seen before, it should have the capability to identify the corresponding features and estimate the pose of the robot correctly. Loop closing in most SLAM methods is an inherently built in property, but there are other methods that explicitly address the loop closing problem [58].

Consistency Consistency is an important statistical property that is required to maintain the quality of the map. In principle the uncertainty estimate of the map and robot pose (or path) should represent the actual statistical uncertainty. It has been observed that due to many assumptions including the inaccurate linearization models, the SLAM algorithm may fail to produce consistent results in longer robot runs [59, 60].

Scalability In most indoor environments the robot will explore a relatively small area. The size of the areas that robots should map can vary from single rooms to whole floors of large buildings or warehouses. Moreover, in outdoor applications such as in exploration, the size of the mapping environment can grow without any bounds. The most common method used to efficiently map large environments is to build smaller sub maps and merge them together at suitable

intervals [61].

The characteristics listed above provide metrics to compare and contrast SLAM algorithms and evaluate new methods. The most commonly used method in SLAM, the extended Kalman filter (**EKF**) will be discussed in detail in this chapter. A discussion of the methods used for evaluating consistency in EKF based SLAM is followed by a discussion of other alternative solutions to SLAM, emphasizing the additional key advances that those methods bring in over the EKF based SLAM. SLAM is widely researched area. There is a large body of literature that discusses various aspects of the problem ranging from sensors to algorithms. In this chapter only the key contributions in the SLAM are highlighted along with some important descriptions of current state of the art applications. First in the processing section, notation and the key relationships between different coordinate frames are described.

2.1 Preliminaries

In order to facilitate the formulation of EKF based SLAM formulation, the following key definitions are introduced. They include the robot model and sensor model along with the relationship between coordinate systems.

2.1.1 Coordinate systems and transformations

Figure 2.1 shows the relative coordinate frames that are used in the mobile robot mapping and localization. In the Figure 2.1 the frames $< i >$ and $< j >$ represent the world and robot coordinate frames respectively. The frame $< l >$ is the landmark frame. When a landmark is represented as a single point then this representation will become redundant. However, the notation will be used for completeness. If the

landmark represents a complex object, such as a line or a composite of lines, then the representation in frame $< l >$ can be used to describe its location and orientation.

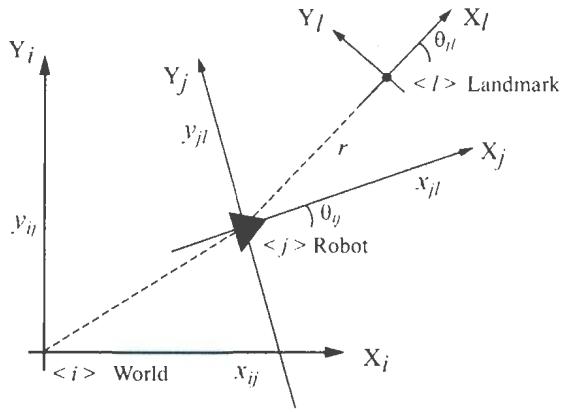


Figure 2.1: Coordinate frame labeling and corresponding notations.

In the proceeding section the formulation of the uncertain spatial relationships that is commonly used in mobile robotics is discussed. These relationships are reported in the EKF based SLAM algorithm description in Smith et al [11]. Also it is important to note that all the state variables, $\mathbf{x}$, discussed hereafter can generally have arbitrarily complex probability distributions.

Let $\mathbf{x}_{ij}$ be the spatial relationship between coordinate frames i and j, which is described by the vector $[x_{ij}\ y_{ij}\ \theta_{ij}]^T$. The covariance matrix of $\mathbf{x}_{ij}$ is denoted as $\Sigma_{\mathbf{x}_{ij}}$ which is a 3×3 matrix.

Compounding relationship [11]

The landmarks in the environment are first observed in the robot coordinate frame. The final map, however, is constructed in a world coordinate frame. Therefore, in order to realize a correct representation in the world coordinate frame, measured landmark positions and their uncertainty are *compounded* with the robot position and with associated uncertainty of the robot position. The $\mathbf{x}_{ij}$, relationship between

frame i and j can be compounded with $\mathbf{x}_{jl}$ using:

$$\mathbf{x}_{il} = \mathbf{x}_{ij} \oplus \mathbf{x}_{jl} = \begin{bmatrix} x_{jl}\cos\theta_{ij} - y_{jl}\sin\theta_{ij} + x_{ij} \\ x_{jl}\sin\theta_{ij} + y_{jl}\cos\theta_{ij} + y_{ij} \\ \theta_{ij} + \theta_{jl} \end{bmatrix} \tag{2.1}$$

where $\oplus$ is the compounding operator. The mean of the compounding operation can be represented as the compounding between two means.

$$\hat{\mathbf{x}}_{il} \approx \hat{\mathbf{x}}_{ij} \oplus \hat{\mathbf{x}}_{jl}. \tag{2.2}$$

Since the compounding relationship is nonlinear, the equation is approximated with first order Taylor series approximation. Similarly, first order approximation of the covariance of the compounding relationship can be expressed as:

$$\Sigma_{\mathbf{x}_{il}} \approx \nabla_{\oplus} \begin{bmatrix} \Sigma_{\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{ij}\mathbf{x}_{jl}} \\ \Sigma_{\mathbf{x}_{ij}\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{jl}} \end{bmatrix} \nabla_{\oplus}^{T} \tag{2.3}$$

where the Jacobian of the compounding operation is defined by:

$$\nabla_{\oplus} = \frac{\partial(\mathbf{x}_{ij} \oplus \mathbf{x}_{jl})}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jl})} = \begin{bmatrix} 1 & 0 & -(y_{il} - y_{ij}) & \cos\theta_{ij} & -\sin\theta_{ij} & 0 \\ 0 & 1 & (x_{il} - x_{ij}) & \sin\theta_{ij} & \cos\theta_{ij} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

In some practical situations the two spatial relationships are independent of each other ($\Sigma_{\mathbf{x}_{ij}\mathbf{x}_{jl}} = 0$) e.g. the landmark observations and their uncertainty is independent of the robot pose and its uncertainty. In such cases the covariance can be expressed as:

$$\Sigma_{\mathbf{x}_{il}} \approx \nabla_{1\oplus}\Sigma_{\mathbf{x}_{ij}}\nabla_{1\oplus}^{T} + \nabla_{2\oplus}\Sigma_{\mathbf{x}_{jl}}\nabla_{2\oplus}^{T} \tag{2.5}$$

where $\nabla_{1\oplus}$ and $\nabla_{2\oplus}$ are the left and right hand side of $\nabla_{\oplus}$, respectively.

Inverse relationship [11]

The useful relationship is used to map the inverse of a relationship. For example, rather than describing the robot pose with respect to the world frame, the world frame can be described with respect to the robot frame. Therefore the inverse of $\mathbf{x}_{ij}$ is expressed as:

$$\mathbf{x}_{ji} = \ominus(\mathbf{x}_{ij}) = \begin{bmatrix} -x_{ij}\cos\theta_{ij} - y_{ij}\sin\theta_{ij} \\ x_{ij}\sin\theta_{ij} - y_{ij}\cos\theta_{ij} \\ -\theta_{ij} \end{bmatrix} \tag{2.6}$$

where $\ominus$ is the inverse operator. Similar the compounding operator, the mean of the inverse can be expressed as:

$$\hat{\mathbf{x}}_{ji} \approx \ominus(\hat{\mathbf{x}}_{ij}). \tag{2.7}$$

The first order covariance estimate can be calculated by:

$$\Sigma_{\mathbf{x}_{ji}} \approx \nabla_{\ominus}\Sigma_{\mathbf{x}_{ij}}\nabla_{\ominus}^{T} \tag{2.8}$$

where the Jacobian of the inverse operation, $\nabla_{\ominus}$ is expressed as:

$$\nabla_{\ominus} = \frac{\partial(\mathbf{x}_{ji})}{\partial(\mathbf{x}_{ij})} = \begin{bmatrix} -\cos\theta_{ij} & -\sin\theta_{ij} & y_{ji} \\ \sin\theta_{ij} & -\cos\theta_{ij} & -x_{ji} \\ 0 & 0 & -1 \end{bmatrix}. \tag{2.9}$$

Composite relationships [11]

The two primary operators (compounding and inverse) can now be used to define other useful composite relationships. One such useful relationship is the tail to tail relationship. When a robot navigates in the environment, it will have to perform path planning for obstacle avoidance. The obstacles are generally represented in the map with respect to the world frame, $\mathbf{x}_{il}$ and it is efficient to represent those objects in the map with respect to the robot frame, $\mathbf{x}_{jl}$, using the inverse of the robot pose estimate $\mathbf{x}_{ij}$.

$$\mathbf{x}_{jl} = (\ominus \mathbf{x}_{ij}) \oplus \mathbf{x}_{il} = \mathbf{x}_{ji} \oplus \mathbf{x}_{il} \tag{2.10}$$

and the first order approximation of the mean and covariance can be easily calculated from the corresponding equations of the compounding and the inverse relationships.

2.1.2 Mobile robot and sensor model

Although mobile robots can have a large variety of models, the differentially driven robot model is most commonly used in robotic research. This is specially the case for robots destined for indoor operation. The differentially driven robot can be easily and accurately modeled using the unicycle model.

System model

A robot model is used to transform the robot pose from $\mathbf{x}_r(k)$ to $\mathbf{x}_r(k+1)$ using the measured velocity inputs $[v(k),\ \omega(k)]$ during Δt, where $\cdot(k)$ denotes the value of any quantity at the end of k-th time step. This transformation can be written in a general form:

$$x_r(k+1) = f(x_r(k),\ v(k),\ \omega(k)), \tag{2.11}$$

which can be expanded to

$$x_r(k+1) = x_r(k) + \Delta t v(k) \cos\left(\theta_r(k) + \frac{\Delta t \omega(k)}{2}\right) \tag{2.12}$$

$$y_r(k+1) = y_r(k) + \Delta t v(k) \sin\left(\theta_r(k) + \frac{\Delta t \omega(k)}{2}\right) \tag{2.13}$$

$$\theta_r(k+1) = \theta_r(k) + \Delta t \omega(k). \tag{2.14}$$

The current covariance of the robot pose $\Sigma_r(k)$ can be propagated through the robot model equations to the covariance of the robot after the velocity commands $v(k)$ and $\omega(k)$ during Δt, $\Sigma_r(k+1)$. The equations for the propagation of the uncertainty in the odometry measurements through the robot model are discussed in the next section. The robot model $f(.)$ can be expanded to accommodate the augmented state vector by including the landmark locations in $f(.)$. Since landmarks are considered stationary, the landmark model is a stationary process which for i-th landmark is simply:

$$x_i(k+1) = x_i(k)$$

$$y_i(k+1) = y_i(k).$$

Measurement model

Range and bearing measurements to a landmark is the most commonly used sensor model. In some cases (mostly computer vision based techniques) a bearing only model is also employed. The nonlinear observation model for the i-th landmark, h_i

can be expressed as:

$$\mathbf{z}_i(k) = \begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = h_i(\mathbf{x}(k)) + w, \tag{2.15}$$

where r_i and θ_i can be calculated using:

$$r_i = ((x_i - x_r) + (y_i - y_r))^{\frac{1}{2}} \tag{2.16}$$

$$\theta_i = \arctan(y_i - y_r, x_i - x_r) - \theta_r, \theta_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{2.17}$$

where x_i and y_i are the landmark position expressed in the world frame and x_r, y_r and θ_r describe the robot pose in the world frame.

2.2 SLAM Formulations

There are numerous popular SLAM implementations where the localization objective of all algorithms remains the same, i.e. to obtain the best estimate for the vehicle pose x_r, y_r and θ_r in the world frame with respect to the initial starting position. The mapping methods differ from each other based on the representation used in building the map. They are (1) landmark based maps, (2) maps based on direct sensor data, and (3) occupancy grid based maps. In type (1) the features in the environment are identified as landmarks and their positions and uncertainties in the positions are calculated from the raw sensor data. This information is used in SLAM to build a map of landmarks. In type (2) representation, direct sensor data are used for SLAM without any intermediate representations. The map built by incremental laser scan registration is an example of type (2) representation. Type (3) representation

segments the environment into a finite number of regularly shaped blocks. Using sensor data, the occupancy of these blocks that make up the environment can be probabilistically inferred. All three types of representations have been popular in SLAM, and specifically, the landmark based mapping method has been widely used. The following section characterizes the SLAM problem in a generic way.

2.2.1 SLAM Problem

The algorithms in SLAM can be broadly categorized into online and full SLAM problems. Online SLAM problem can be defined as the concurrent estimation of the current robot pose and the map of the environment using all past control inputs and measurements. The probability of the estimated current robot position, $\mathbf{x}_r(k)$ and map, M can be expressed as:

$$p(\mathbf{x}_r(k), M|Z(1:k), U(1:k)) \tag{2.18}$$

where, $Z(1:k)$ and $U(1:k)$ represent all the sensor data and the measurement data history of the robot. In contrast, the full SLAM problem is defined as the estimation of the map along with the complete pose history (path), $\mathbf{x}_r(1:k)$ of the robot which can be probabilistically expressed as:

$$p(\mathbf{x}_r(1:k), M|Z(1:k), U(1:k)). \tag{2.19}$$

Naturally the full SLAM is a much more difficult estimation problem than the online version, due to its high dimensionality of the parameter space and the data association problem, with a large number of pose-feature associations. For these reasons the online SLAM problem has been explored more than its counterpart. Therefore

the algorithms developed in this thesis focus on using the online approach to the SLAM problem. The most popular algorithm for the online SLAM problem is the extended Kalman filter. In the next section the general formulation of the Kalman filter based solution to the online SLAM problem is explained.

2.2.2 Extended Kalman filter solution

The Extended Kalman filter (EKF) [62] has been widely used in solving the SLAM problem since the pioneering work by Smith et al [11]. In the EKF based solution the information from sensors are integrated into a single covariance matrix. The state vector of the filter represents the current estimate of the robot pose and the landmark positions. Although robot pose history can be maintained in the state vector, in most implementations the current robot pose is kept in the state vector by updating the previous state. To start the discussion on EKF based SLAM algorithm, the state vector and covariance matrix are defined.

State vector and covariance matrix

State vector with N landmarks can be defined as:

$$\mathbf{x}(k) = [\ \mathbf{x}_r(k) \mid \mathbf{x}_1(k) \ldots \mathbf{x}_i(k) \ldots \mathbf{x}_N(k)]^T \tag{2.20}$$

where $\mathbf{x}_r(k)$ $(= [\ x_r\ \ y_r\ \ \theta_r\ ]^T)$ is the robot pose and $\mathbf{x}_i(k)$ $(= [\ x_i\ \ y_i\ ]^T)$ is the position of the i-th landmark in the world frame at time k.

The covariance matrix of the state vector $\Sigma(k)$ is composed of the covariance of vehicle pose, $\Sigma_{rr}(k)$, covariance of the map, $\Sigma_{mm}(k)$ and cross covariance between vehicle pose and map, $\Sigma_{rm}(k)$ and can be expressed as:

$$\Sigma(k) = \begin{bmatrix} \Sigma_{rr}(k) & \Sigma_{rm}(k) \\ \Sigma_{rm}^T(k) & \Sigma_{mm}(k) \end{bmatrix}. \tag{2.21}$$

Prediction

The Kalman filter has three steps: prediction, observation and update [38, 63]. According to the general notation in the Kalman filter, $\hat{\mathbf{x}}^-(k)$ and $\hat{\mathbf{x}}^+(k)$ are predicted and updated estimates of the state vector, respectively. $\Sigma^-(k)$, $\Sigma^+(k)$ and $\mathbf{W}(k)$ are *priori* covariance matrix, *posteriori* covariance matrix and Kalman gain, respectively.

Robot pose after a time interval is predicted using (2.12) and the rest of the state vector remains unchanged (as features remain stationary in the world frame). The covariance matrix can be updated by the new information by propagating the previous covariance matrix through the system model and adding the contribution of the uncertainty from the odometry readings using the first order linearization of (2.12). The covariance of the predicted state, $\Sigma^-(k)$ can be calculated by [63]:

$$\Sigma^-(k) = \nabla_x f(k) \Sigma^+(k-1) \nabla_x f^T(k) + \nabla_u f(k) \Sigma_u \nabla_u f^T(k), \tag{2.22}$$

where $\Sigma_u = \text{diag}[\sigma_v^2 \;\; \sigma_\omega^2]$ is the covariance matrix of the measurement noise in the odometry data. The matrix Σ_u can be considered diagonal assuming that v and ω are independent measurements. The Jacobians $\nabla_x f(k)$ and $\nabla_u f(k)$ are defined as:

$$\nabla_x f(k) = \frac{\partial f}{\partial \mathbf{x}} \tag{2.23}$$

$$\nabla_u f(k) = \frac{\partial f}{\partial \{v, \omega\}}. \tag{2.24}$$

The Jacobians $\nabla_x f(k)$ and $\nabla_u f(k)$ can be represented in the following matrix

form:

$$\nabla_x f(k) = \begin{bmatrix} \frac{\partial f_r}{\partial\{x_r, y_r, \theta_r\}} & \mathbf{0}_{rm} \\ \mathbf{0}_{mr} & \mathbf{I}_{mm} \end{bmatrix} \tag{2.25}$$

$$\nabla_u f(k) = \begin{bmatrix} \frac{\partial f_v}{\partial\{v,\omega\}} \\ \mathbf{0}_{m\times 2} \end{bmatrix}. \tag{2.26}$$

Therefore, $\Sigma^-(k)$ can be efficiently implemented by

$$\Sigma^-(k) = \begin{bmatrix} F_x \Sigma_{rr}(k) F_x^T + F_u \Sigma_u F_u^T & F_x \Sigma_{rm}(k) \\ (F_x \Sigma_{rm}(k))^T & \Sigma_{mm}(k) \end{bmatrix} \tag{2.27}$$

where $F_x = \frac{\partial f_r}{\partial\{x_r, y_r, \theta_r\}}$ and $F_u = \frac{\partial f_v}{\partial\{v,\omega\}}$.

Observation and update

When the robot's sensors detect a landmark it can be either a landmark that has been perceived by the robot previously and has been included in the state vector, or a new landmark encountered by the robot, that will be added to the state vector. In the latter case, it is required that the landmark should be initialized into the filter. The initialization procedure is described in the next section. In the former case, the new information should be integrated into the filter by updating the system. First, using (2.15), predicted observations $\hat{\mathbf{z}}(k)$ of the landmarks are calculated using the predicted state vector. The *innovation*, which is the error in the expected observation and innovation covariance matrix can be calculated from:

$$\zeta(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k) \tag{2.28}$$

$$S(k) = \nabla_x h_i(k) \Sigma^-(k) \nabla_x h_i^T(k) + \mathbf{R}(k) \tag{2.29}$$

where $\mathbf{R}(k)$ is the measurement covariance matrix; and Jacobian $\nabla_x h_i(k)$ can be calculated using:

$$\nabla_x h_i(k) = \begin{bmatrix} -\frac{\Delta x}{d} & -\frac{\Delta y}{d} & 0 & \cdots & \frac{\Delta x}{d} & \frac{\Delta x}{d} & \cdots \\ \frac{\Delta y}{d^2} & -\frac{\Delta x}{d^2} & -1 & \cdots & -\frac{\Delta y}{d^2} & \frac{\Delta x}{d^2} & \cdots \end{bmatrix} \tag{2.30}$$

where $d = ((x_i - x_r) + (y_i - y_r))^{\frac{1}{2}}$, $\Delta x = x_i - x_r$, and $\Delta y = y_i - y_r$.

Finally, the estimated position of the robot and the feature locations are updated using:

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{W}(k)\ \zeta(k) \tag{2.31}$$

$$\Sigma^+(k) = \Sigma^-(k) - \mathbf{W}(k)\ \mathbf{S}(k)\ \mathbf{W}^T(k) \tag{2.32}$$

where, $\mathbf{W}(k) = \Sigma^-(k)\ \nabla_x \mathbf{h}^T(k)\ \mathbf{S}^{-1}(k)$.

Map management [63]

During exploration of the environment the robot observes new landmarks and they are added to the state vector, and consequently the state covariance matrix will be updated accordingly. Similarly when a landmark is no longer observable or deemed not worthy of maintaining in the map, it will be removed from the state vector by the removal of the corresponding elements in the state vector and corresponding rows and columns in the state covariance matrix.

The newly observed landmark, z_i can be augmented into the state vector $\mathbf{x}(k)$ using:

$$\mathbf{x}_a(k) = f_i(\mathbf{x}(k), z_i) = \begin{bmatrix} \mathbf{x}(k) \\ g_i(\mathbf{x}_r(k), z_i) \end{bmatrix} \tag{2.33}$$

where $g_i(\mathbf{x}_r(k), z_i) = [x_r + r_i \cos(\theta_r + \theta_i), y_r + r_i \sin(\theta_r + \theta_i)]^T$. The state covariance matrix is first augmented with the measurement covariance.

$$\Sigma^*(k) = \begin{bmatrix} \Sigma_{rr}(k) & \Sigma_{rm}(k) & 0 \\ \Sigma_{rm}^T(k) & \Sigma_{mm}(k) & 0 \\ 0 & 0 & \Sigma_z(k) \end{bmatrix} \tag{2.34}$$

Since the true covariance of the new landmark depends on the vehicle pose covariance, it is important to update the augmented state covariance matrix $\Sigma^*(k)$ by propagating it through the system model for a new landmark, z_i.

$$\Sigma_a(k) = \nabla_{x_a} z_i \Sigma^*(k) (\nabla_{x_a} z_i)^T \tag{2.35}$$

where Jacobian $\nabla_{x_a} z_i$ is given by:

$$\nabla_{x_a} z_i = \begin{bmatrix} \mathbf{I}_{vv} & 0 & 0 \\ 0 & \mathbf{I}_{mm} & 0 \\ \nabla_{\mathbf{x}_r} g_i & 0 & \nabla_{z_i} g_i \end{bmatrix} \tag{2.36}$$

where,

$$\nabla_{\mathbf{x}_r} g_i = \begin{bmatrix} 1 & 0 & -r_i \sin(\theta_r + \theta_i) \\ 0 & 1 & r_i \cos(\theta_r + \theta_i) \end{bmatrix} \tag{2.37}$$

and

$$\nabla_{z_i} g_i = \begin{bmatrix} \cos(\theta_r + \theta_i) & -r_i \sin(\theta_r + \theta_i) \\ \sin(\theta_r + \theta_i) & r_i \cos(\theta_r + \theta_i) \end{bmatrix}. \tag{2.38}$$

Due to the sparse nature of $\nabla_{x_a} z_i$ the $\Sigma_a(k)$ can be implemented more efficiently using:

$$\Sigma_a(k) = \begin{bmatrix} \Sigma_{rr}(k) & \Sigma_{rm}(k) & \Sigma_{rr}(k)(\nabla_{\mathbf{x}_r} g_i)^T \\ \Sigma_{rm}^T(k) & \Sigma_{mm}(k) & \Sigma_{rm}^T(k)(\nabla_{\mathbf{x}_r} g_i)^T \\ (\nabla_{\mathbf{x}_r} g_i)\Sigma_{rr}(k) & (\nabla_{\mathbf{x}_r} g_i)\Sigma_{rm}(k) & (\nabla_{\mathbf{x}_r} g_i)\Sigma_{rr}(k)(\nabla_{\mathbf{x}_r} g_i)^T + (\nabla_{z_i} g_i)\Sigma_z(k)(\nabla_{z_i} g_i)^T \end{bmatrix}. \tag{2.39}$$

When a landmark is no longer observable by the robot it can be removed from the state vector by deleting the corresponding elements and can be removed from the state covariance matrix by simply deleting the corresponding rows and columns. If and unobservable landmark is not removed it will remain the map without any effect to the overall map but it will act as a unnecessary phantom landmark in path planning.

Data Association

When the robot makes an observation of a landmark, one of two things can be true. It can either be a landmark that the robot has seen previously and which has been already integrated into the map, or it is a new landmark that is suitable for inclusion in the map. The latter case is related to the map management. In the former case the corresponding landmark in the map is required to be identified before the update. This correspondence identification problem is commonly known as the data association in SLAM. The most common aspects of a landmark used in data association are

its global position in the map together with the uncertainty bounds of the current estimate, the appearance of the landmark, and its position in the layout of a set of landmarks. The appearance based method requires additional information about the landmark, such as extra features indicating its uniqueness. Computer vision based methods often augment this additional information in their data association as a landmark can be easily characterized by its visual appearance [64].

The most common method for data association is the global nearest neighbor (**GNN**) [65] which, is also known as individual compatibility nearest neighbor (**ICNN**) [66]. The current landmark is associated with the i-th landmark if the following condition based on the Mahalanobis distance holds:

$$D^2 = (\mathbf{z}_i - \hat{\mathbf{z}}_i)^T \Sigma_i (\mathbf{z}_i - \hat{\mathbf{z}}_i) < G \tag{2.40}$$

where D is the Mahalanobis distance between the measured feature and i-th landmark in the map, $\mathbf{z}$ is the current measurement, $\hat{\mathbf{z}}$ is the estimated measurement, and Σ_i is the covariance matrix of the i-th landmark all in the robot frame. The test condition is often chosen to be a gate G of suitable value while other works prefer to use $\chi^2_{d,\alpha}$ where $d = \dim(\mathbf{z})$ and α is the confidence level. The main difference between a simple gating test against the chi-squared test is that the former offers greater flexibility. The nearest neighbor method can be successfully applied under the following conditions. (1) The global uncertainty of the robot is relatively smaller than the distance between the closest landmark pair and (2) the sensor in the robot has an adequately low level of noise such that there won't be an excessively high number of spurious landmark observations.

When above conditions do not hold true, especially the first condition, where it can give rise to a single observation being associated with two or more landmarks, a

more sophisticated data association method has to be used. The joint compatibility (**JC**)[66] can be used to test the consistency of the landmark-observation associations. A tree based search method based on JC known as joint compatibility branch and bound (**JCBB**) [61] is used to exploit the information in the layout of the landmarks present in the map. In JCBB, each observation is tested for individual compatibility (**IC**) against each landmark. Once it passes the IC test, the observation is tested for joint compatibility between all the associated landmarks-measurement pairs established thus far. Both IC and JC based methods take the hard-data association decisions where they are not allowed to reverse or be discarded in the future, even if there is any new and more convincing evidence. New evidence can be incorporated by expanding the search for associations in time (across multiple sensor frames). Multi-dimensional assignment (**MDA**) based technique has been used to achieve this objective [37]. Other techniques in SLAM, such as in FastSLAM [16, 67] (described in more detail in section 2.2.3) maintain multiple hypotheses for the robot path. Each of these paths also maintains a data association hypothesis on its own, yielding an implicit multi hypothesis scenario for data association. When the path does not fit the current state of observations (through wrong data association or otherwise) it has a higher chance of being discarded in the resampling stage.

Consistency Analysis

In EKF based SLAM, consistency of the filter is one very important consideration. This is mainly due to many assumptions that have been made during the formulation of the filter. Practically, once the filter becomes inconsistent it becomes worthless to continue any further as the constructed map is bound to become highly inaccurate. Generally the filter is said to be inconsistent if the current estimate is overly optimistic (i.e. the statistics of the filter represent the system as having a lower uncertainty than

is the actual case). Consistency is especially important in data association and loop closing. Basic criteria for a consistent filter are [65]:

1. The state errors should have a zero mean and have a magnitude that is within the acceptable limits of the covariance as calculated by the filter.

2. Innovations should also follow the first criteria.

3. Innovations should be acceptable as white.

The first and second criteria can be formally expressed as:

$$E[\mathbf{x}(k) - \hat{\mathbf{x}}^{+}(k)] = 0 \text{ and } \Sigma(k|k) \geq \Sigma(k) \tag{2.41}$$

$$E[\mathbf{z}(k) - \hat{\mathbf{z}}(k)] = 0 \text{ and } S(k|k) \geq S(k) \tag{2.42}$$

where $\Sigma(k|k)$ is the current estimated covariance and $\Sigma(k)$ is the true covariance. The ideal condition is when the estimated covariance is equal to the true covariance. When the estimated covariance is greater than the true values, then the filter can be characterized as conservative. When the estimated covariance values are less than the true values the filter becomes optimistic and the magnitude of the system error runs the risk of exceeding the covariance bounds. The first criterion can only be tested for simulated filters when the true value of the estimated variables is available, while the second and third can be tested for real filter estimates.

Following are the commonly used methods for testing the consistency of the filter [65]:

1. Normalized estimation error squared (**NEES**) test: The NEES $\epsilon(k)$ is defined as:

$$\epsilon(k) = (\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))^T \Sigma_{k|k}^{-1} (\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)). \tag{2.43}$$

Under the assumption that the filter is consistent and linear-gaussian, $\epsilon(k)$ is chi-square distributed with $\dim(\mathbf{x}_k)$ degrees of freedom [65]. Therefore the average value for $\epsilon(k)$ will be equal to the $\dim(\mathbf{x}_k)$. For a simulated filter a Monte Carlo test with N robot runs can be used to calculate the average value of $\epsilon(k)$ by

$$\bar{\epsilon}(k) = \frac{1}{N} \sum_{i=1}^{N} \epsilon(k). \tag{2.44}$$

Then the mean of the $N\bar{\epsilon}(k)$ has a chi-squared distribution with a mean of $N\dim(\mathbf{x}_k)$. For the Monte Carlo tests a two sided test for the 95% probability region can be carried out to evaluate the consistency. If the average NEES is higher than the upper bound of the confidence interval then the filter is highly optimistic. Although the lower bound does not have any significance relating to consistency it can be used to test the conservativeness of the filter.

2. Normalized mean estimation error (**NMEE**) test: If there is a large bias in the estimation error then the NEES value will also be higher. Therefore if the average NEES test fails it is customary to test the mean estimation error of the j-th component of the state vector for N robot runs using:

$$\bar{\mathbf{x}}_j(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{x}_j(k) - \hat{\mathbf{x}}_j(k|k)}{\sqrt{\Sigma_{jj}(k|k)}}. \tag{2.45}$$

The average NMEE, $\bar{\mathbf{x}}_j(k)$ should ideally have a $N(0, 1/N)$ distribution. Therefore the the value $\bar{\mathbf{x}}_j(k)$ can be easily tested to lie within the 95% confidence bounds.

3. Normalized innovation squared (NIS) test: Similar to the average NEES test, if the filter is consistent the normalized innovation squared:

$$\epsilon_\nu(k) = (\mathbf{z}(k) - \hat{\mathbf{z}}(k|k))^T S_{k|k}^{-1}(\mathbf{z}(k) - \hat{\mathbf{z}}(k|k)) \tag{2.46}$$

should have a chi-squared distribution with n_z degrees of freedom, where n_z is the dimension of the measurement.

4. Whiteness test using sample autocorrelation: If the innovations are zero mean and white then the autocorrelation of the each component (for N runs) of the innovation should have a zero mean and a variance of $1/N$.

The maps produced by EKF based SLAM implementations are inherently inconsistent [60, 68]. This fact has been experimentally shown using stationary and moving robot experiments [60, 59]. The consistency can be improved by inflating the standard deviations of robot motion and landmark measurements. This will extend the duration that the filter will be statistically consistent. But in the short term, the filter will produce highly conservative estimates. Another more general solution used to circumvent the inconsistencies arising in mapping of large areas is sub mapping [69]. The principle concept of sub mapping is to build smaller consistent maps and then join them to produce a larger global map, which will be less inconsistent.

Scalability

The two main implementation issues in EKF based SLAM are the high computational complexity in relatively large environments and the data association problem. The computational complexity of the EKF is largely dominated by the update stage of the filter and it is known to be in the order $\mathcal{O}(n^2)$, where n is the number of land-

marks in the state vector. The computational complexity of the EKF based method is circumvented to a large extent by using the compressed version of the EKF. The compressed EKF filter improves the computational efficiency by updating only the local landmarks that are currently visible to the robot. The robot performs the single global update when the robot travels into a new local region. The gain in the computational cost of the compressed filter increases when the number of landmarks in the global map is larger compared to the local map. Improvements in the computational complexity of SLAM are achieved through the use of various methods such as particle filters (FastSLAM), information filters (sparse extended information filter (SEIF)), etc.

2.2.3 Alternative solutions to the SLAM problem

In this section two alternative solutions to the SLAM problem are reviewed. The first method, FastSLAM, is based on the particle filter and offers an elegant solution that can be viewed as a solution to both full SLAM and online SLAM. The second is an online algorithm based on the extended information filter and it mainly exploits the sparse nature of the information matrix. Both solutions are computationally more efficient than the EKF and under certain conditions they offer comparable performance.

FastSLAM [16]

The structure of the SLAM problem has a property where if the true path of the robot is known then the landmark position estimation problems are independent of each other [70]. Due to the uncertainty of the current robot pose in the EKF solution, landmark estimation problems are dependent on each other. This property

of conditional independence is exploited using Rao-Blackwellized particle filter [16] where each particle represents a hypothetical path of the robot [16]. FastSLAM algorithm has several key advantages over other methods.

1. The very fact that it uses a particle filter in which the robot pose is represented by a single particle means the nonlinearity in the robot motion model is accurately captured, instead of using the linearized versions. The gain in the accuracy of the motion model becomes significant when the robot motion model is highly nonlinear.

2. Since the data association is calculated on a per particle basis, the robot maintains several hypotheses of data associations rather than a single-best data association. This makes the FastSLAM solution much more robust to the errors resulting from wrong data associations.

3. Another advantage of the FastSLAM algorithm is that it has a time complexity of $\mathcal{O}(\log(N))$.

4. Also the FastSLAM solution solves the full and online version of the SLAM problem. This is a result of the fact that FastSLAM calculates the posterior robot path with which it is possible to claim conditional independence. Further, the algorithm calculates the path of the robot with one pose at a time, making it an online solution as well.

Each particle in the filter represents the current robot pose, the mean position of the landmarks and their variances. When the robot progresses to a new pose, the pose of each particle is updated using the robot motion model. At the update stage, the observed landmark information is integrated to that of the previous time step using an EKF. The unobserved landmarks are appended to the particle without any changes.

The particle set in hand at this stage has a proposal distribution that is dominated by the control inputs to the robot, when it should be conditioned on the observed landmark locations. In order to make this adjustment in the final distribution, the usual resampling step is performed where each particle is appended a weight based on the ratio of the target (final) distribution and the proposal distribution. These weights are used in drawing a new set of particles where the particles which are in most agreement with the measurements will survive. In an improved version of FastSLAM, FastSLAM 2.0 [71], the measurements are taken into consideration to obtain a better proposal distribution. Experimental results have shown that both versions of FastSLAM perform comparably when using a large number of particles. However, the second version performs better when the measurement noise is low. One key drawback of the FastSLAM algorithm is in its loop closing ability. Due to the lack of correlations, the additional information from loop closing cannot be propagated through the entire map. Due to this reason it is imperative in FastSLAM to maintain a reasonably rich particle diversity. This can be done by either adopting the second version, having a large number of particles or both.

Sparse Extended Information Filters (SEIF) [15]

An extended version of the information filter (EIF)[15] is a popular estimation process similar to EKF. The only difference is that in EKF one can extract mean and variance of an estimated variable without any additional computations. In contrast EIF requires an additional set of computations to derive the mean and variance of a variable. The information matrix is the central data structure in the EIF (equivalent to the inverse of the covariance matrix). The information matrix of the SLAM process shows interesting properties that are exploited in the derivation of SEIF. The elements in the information matrix represent either links between robot pose and

landmarks or the landmarks themselves. Also, inspection of the normalized information matrix (inverse covariance matrix) shows that it is generally a sparse matrix (a great number of elements in the matrix will have very small values which can be assumed zero for practical purposes). This is due to the fact that only landmarks that are close together show higher correlations. At any given time the robot will only be able to observe a limited number of landmarks known as, active landmarks. Since the elements in the information matrix represent the links between robot and landmarks and between landmarks, when the robot makes observations or when the robot moves, only a few elements corresponding to the robot pose and active landmarks are affected. Thus, motion and measurement updates can be achieved in an information additive manner. The key advantages of SEIF is that the measurement update time is independent of the total number of landmarks in the map and only depends on the number of active landmarks currently in view. SEIF is known to perform at a lower accuracy than EKF implementations; specially with a lower number of active landmarks due to the approximations introduced to maintain the sparsity.

2.3 Applications

The theoretical analysis of algorithms in the recent past has led to a greater understanding of SLAM as well as the development of efficient algorithms that can be used to serve practical purposes. Developments in the application areas involve development of new sensor technologies such as compact sensors, formulation of computationally efficient methods in sensor data processing such as in stereo vision, and development of multisensor approaches such as laser-camera-based methods. The following sections offer a review of the application areas of SLAM and the related sensor technologies.

2.3.1 Application of SLAM

SLAM has been applied to a wide variety of tasks in many different types of environments. The primary applications have been in indoor mobile robotics, while there is a considerable application base in outdoor environments. The following itemized descriptions provide a summary of the significant state of the art experimental contributions in SLAM. They are categorized according to the type of sensor used, the type of environment, and the scale of the application, followed by other important remarks.

Laser and Sonar based SLAM applications

- [15], [61], [72] Laser / Outdoor / Large 150m x 150m / The Victoria park, Sydney dataset. The tree trunks in the park have been extracted from the laser range finder data and treated as landmarks. This is one of the pioneering and complete datasets for large scale outdoor SLAM and has been used as a benchmark for testing several SLAM algorithms.

- [37] Laser / Outdoor / Large 60m x 60m / Uses laser data to identify natural outdoor features in a campus environment such as poles, trees, etc. The application scenario described in the paper closely resembles a semi-urban environment and demonstrates the applicability of the EKF based SLAM using laser range finder in cluttered environments.

- [73] Laser / Outdoor / City wide several km / Is similar to [37]above, but larger scale with heavy clutter (moving vehicles and people) in a real urban setting.

- [69] Laser / Indoor / Large 100m x 70m / Line features in the laser range data

are used as landmarks.

- [74] Imaging Sonar / Underwater / Medium / The SLAM was performed in a semi-structured environment where there are several straight ridges that can be extracted as lines in Sonar images.

- [39] Imaging Sonar / Underwater / Large / Artificially placed sonar targets are observed as 2D landmarks using imaging Sonar.

Computer vision based SLAM applications

- [75] Monocular Vision / Airborne / Large / 6 DOF system was used with artificially placed landmarks on the ground. The images are captured at 50Hz.

- [76] Monocular Vision / Airborne / Large / SIFT features are used as landmarks and the vehicle is flown at a constant altitude simulating a 2D SLAM problem.

- [77] Sonar / Indoor / Medium small office environment / Landmarks such as walls and corners are detected as points and lines using time series data from multiple sonar sensors.

- [78] Stereo Vision / Indoor / Large / The artificially placed markers on the floor are detected and located using stereo cameras.

- [79] Trinocular Vision / Indoor-3D / Medium / 3D SLAM implementation uses a trinocular camera system in a small office environment.

- [80] Ceiling Vision / Indoor / Medium small office environment / Line features in the ceiling tiles and the lighting elements are used as landmarks.

- [81] Ceiling Vision / Indoor / Small / Special lighting fixtures on the ceiling are selected as landmarks.

- [46] Trinocular Vision / Outdoor-3D / Small / EKF based SLAM application is implemented using the stereo located corner features of the images as the landmarks. The method provides true 6DOF camera movements.

- [48, 82] Stereo Vision / Indoor / Small / Very first attempts at vision based SLAM, uses a artificial set of features. Although the sensor rig could calculate 3D positions, landmarks are mapped to 2D space.

- [83] Monocular Vision / Indoor/3D / Small / Corner features are detected as landmarks and use a free camera motion model yielding a 3D SLAM system.

- [84, 85] / Monocular Vision / Indoor / Small / Natural Landmarks (vertical lines).

- [86] Monocular Vision / Indoor-3D / Small / Shows a bearing only application of SLAM using a single camera.

- [87] Monocular Vision / Indoor-Outdoor-3D / Small / Natural corner features are selected as landmarks in a 3D SLAM implementation.

- [1] Laser-Vision / Indoor / Medium / Uses the line features from laser range data. Additionally obtains redundant information about the corner features to label corners and semiplanes in the line landmarks detected from laser data.

As can be seen from the above list SLAM has been experimentally applied in many different types of environments, from small lab size experiments to city wide SLAM. Traditionally, laser range finder has been the sensor of choice for SLAM. The typical landmarks found in laser data include line objects (corresponding to flat wall-like objects), small isolated objects (tree trunks, poles), corners (intersection of line

objects), etc. Laser data is also used as a dense point cloud in scan registration [13] that would allow SLAM to operate without landmark detection.

Computer vision has found wide adoption as the primary sensor technology in SLAM. The popularity can be mainly attributed to the highly detailed information that the sensor provides about its environment. In computer vision landmarks are often identified as a wide variety of objects that can be easily detected from their background (markers, lines, patterns on ceilings). Among them corner features are the easiest to detect and the most general type of landmark that is used in SLAM using computer vision.

2.3.2 Multisensor approach to SLAM

It is apparent that a single sensor modality would not be able to perform all the tasks assigned to a mobile robot. This makes the case for multi sensor implementation of mobile robots. In [88] information extracted from vision and sonar have been used in the development of EKF based SLAM method. The landmarks in this method are represented as lines and points. Lines in the environment are first detected in both sonar and vision data using Hough transform and the corresponding pairs have been identified using data association algorithms. The data association algorithm essentially performs the data fusion at landmark level (appearance based). Experimental data shows that the error quantities in localization are less than that in the odometry only navigation. EKF based robot localization has been achieved using a combined laser and vision sensor [89, 90]. In above implementations the vertical lines in the images and the horizontal lines extracted from the laser data are used as features in the environment. These methods do not perform any data fusion between the two sensors, rather they use the extracted features from each sensor independently in the

localization process. In contrast, Castellanos et al. [1] describe a multisensor SLAM implementation using the laser and vision, where individual maps are derived from all sensor data and fused using sensor calibration information. The vertical line positions from the camera are used as redundant information to locate corners and semiplanes in the laser data. In [1] the visual features are used as a labeling mechanism of landmarks identified from the laser data. However, it is quite apparent that given the high level of textural information that vision provides, it can be used to serve as a primary sensor to identify the landmarks. Once the laser and camera are calibrated, the precise laser data can be used to locate the landmarks identified by the camera data.

2.3.3 SLAM in Dynamic Environments

In most implementations of SLAM algorithms the sensor data is assumed to be from an environment that is clutter free. However, this is rarely the case in most uncontrolled environments such as public facilities and outdoor environments (urban environments). In those cases at any given time there could be any number of moving objects present in the sensor data. The landmarks identified in these moving objects are required to be separately processed from the stationary landmarks.

The first option is to model the movement of these objects and identify the motion model of the landmark and then incorporate the motion model of the moving landmark in the system equations [50, 37]. The estimation of the accurate motion model of a randomly moving object is a difficult task and can lead to a highly uncertain estimate. The incorporation of such moving landmarks with an uncertain motion model to the map could lead to the corruption of the system and ultimate failure. The problems with the moving landmarks could be compounded by the difficulties in data

association. The simple and more prudent option is to disregard all the landmarks attached to the moving objects. In order to remove sensor data corresponding to the moving objects, first the moving objects are required to be accurately identified.

Moving object detection methods based on laser data [50, 20, 21, 51] can be employed to remove the laser data corresponding to the moving objects. While above methods offer solution to the moving object detection, they do not provide a comprehensive solution to the problem. A more detailed treatment of the current state of the art in laser based moving object detection is detailed in Chapter 4.

2.4 Conclusion

The EKF based SLAM basics and the issues in theory and implementation have been presented in this chapter. Major issues in implementations of EKF based SLAM, including complexity and consistency have been described. From the literature survey it is evident that from a performance perspective the EKF is the best solution to the SLAM problem, barring the inconsistency problems arising due to the nonlinearities in the system model. This epitomizes the optimal nature of the Kalman filter. In many practical implementations where robots operate in large environments a pure EKF solution has severe drawbacks in computational efficiency. The current state of the art alternative solutions offer attractive computational performances while providing close to optimal performance.

It is evident that performance of the EKF can be improved by obtaining high quality landmark data through low level sensor fusion. Higher quality landmarks can be obtained by the fusion of many attributes of landmarks that are being extracted from each sensor, rather than with individual sensors. Also there is a notable vacuum in the literature for a systematic algorithm for moving object detection in the domain

of SLAM in dynamic environments. Availability of a robust moving object detection method will greatly advance the SLAM with resulting contributions in mapping of dynamic environments, to simultaneous localization and moving object tracking (SLMOT).

Chapter 3

Multisensor Landmark Detection and Localization for SLAM

Landmark detection and localization are the primary tasks of the sensor in the implementation of any SLAM technique. In this chapter a multisensor landmark detection and localization algorithm is explored for the EKF based SLAM implementation. The proposed method integrates the laser range data and vision data to detect the maximum possible number of landmarks and then localizes them to the best possible accuracy. The chapter starts by introducing the multisensor landmark detection and localization schemes. The rest of the chapter develops the proposed multisensor approach and presents results of the landmark localization and the SLAM implementation using EKF based SLAM.

3.1 Introduction

Among various sensors used in solving the SLAM problem in robotics, laser range scanners have received most attention, mainly due to their response behaviour and

ability to accurately scan a wider field of view. Laser range finders can precisely locate landmarks in environments having directional variant features, such as protruding edges in walls, edges of objects located in the field of view such as chairs, or tables, and also moving objects such as humans [91]. However, when such features are unavailable, such as corridors having flat walls, long empty rooms and halls, the laser data will contain a minimum number of features that can be detected as landmarks.

Recently, computer vision received much attention for SLAM [92, 93, 94] and has the ability to extract visually salient features even in flat walls or corridors in buildings. However, there are many drawbacks in vision based sensors. Monocular SLAM implementations require the features to be present in the field of view for a longer duration to facilitate proper convergence of the feature position estimate. However, stereo vision has the ability to overcome this issue in single camera systems, but requires a significant computational overhead, particularly for calibration and 3D estimates. Thus, it is possible to use the features of both sensors, laser and camera, to overcome the drawbacks of each. Hence this work demonstrates a novel application of a single sensor based on a laser-vision model. Early work of the laser-vision model uses two sensor readings separately and fuses the SLAM data in the post processing stage to estimate robot pose [1]. In contrast, the method proposed in this chapter performs feature extraction at the sensor level while using a laser-vision model as a single sensor for detecting and locating landmarks. Therefore this chapter presents the following key contributions. First, the work demonstrates effective integration of laser and camera as a single sensor for general purpose robot navigation. Secondly the work demonstrates how the integrated laser-camera model can be used effectively to solve the SLAM problem. The sensor also has the ability to either work as a laser only sensor or vision only sensor.

3.1.1 Related Work

The research in computer vision based simultaneous localization and mapping (**SLAM**) can be broadly categorized into two areas. They are: appearance based methods and feature based methods. In appearance based localization and mapping, image features are collectively used to describe a scene. These feature based descriptions are used to compare and contrast the images that the robot acquires along the way. Hence when a robot revisits an environment, the localization algorithm will be able to measure the similarity between the images of the current scene and the images that are registered in a database. In most cases this type of qualitative localization and mapping can only generate topological representations of the environment. Although it provides a viable and more natural mapping and localization procedure, the qualitative algorithms do not provide detailed information about the environment. Details in such a map may be inadequate, especially when robots require accurate information about the structure of the environment for tasks such as path planning. Although appearance based methods have been used in SLAM [95, 96, 97], they are mostly used in the re-localization of the robots [98, 99, 100].

In contrast to the appearance based methods, feature based methods uniquely identify visually salient landmarks in the environment and calculate their position with respect to the robot. Such measurements can be used in estimators to build the map of the visual landmarks while localizing the robot. The primary advantage of the feature based methods is the higher fidelity of the map. The feature based methods can be classified based on the method that they use to calculate the range and bearing to the features. The most common method is the use of stereo cameras [40, 48, 46, 101, 64]. Other methods used to calculate the feature position include: single camera based feature position estimation [83, 87] and optical flow based calculation [47]. Although

computer vision based SLAM methods show significant advances, they exhibit one or more of the following drawbacks with respect to general SLAM applications.

1. The methods were only demonstrated to work in small scale environments [40, 83, 87].

2. It is necessary to have a large number of features in the environment for the SLAM algorithms to properly converge [46, 47].

These issues can be primarily attributed to the large uncertainties associated with the vision based feature position calculation. Further, in stereo and other vision based feature position calculation methods, uncertainty of the feature position increases with increased distance. Additionally, a regular camera lens provides only a limited field of view. This severely limits the amount of time that a feature is actively observed in the SLAM process, especially if the robot is moving at relatively high speeds.

On the contrary, the laser range finder provides excellent range measuring capabilities and has been widely used in SLAM implementations. Landmarks that are generally invariant to the direction of scanning (such as chair and table legs, corners, tree trunks, poles, etc.) can be identified in laser range data. However, typical indoor environments with corridors, walls and other structured shapes either do not have any features or have only very few features. During the estimation process, when landmarks are absent in the environment, uncertainty of the estimator rapidly grows. The landmarks that will be encountered with a higher robot uncertainty will have a higher uncertainty bound (Theorem 3 in [38]). This will lead to possible inconsistent data associations when the robot revisits the same area. Hence frequent featurelessness in the environment will lead to a highly unstable SLAM process. However, computer vision can be used to detect visually salient features on walls and other places where it is not possible to use a laser range finder to detect landmarks, and the

laser range finder can be used to measure the range to the visually salient landmarks. On multi sensor SLAM, Castellanos et. al. [1] have presented a laser-camera based method that fuses landmark information from laser range finder data with image data. The method presented in [1] detects landmarks using data from each sensor and calculates the individual and joint compatibility between them. From the laser range finder it locates the line segments, corners and semiplanes. Using camera data it obtains redundant information about the landmarks that were observed by the laser range finder. Thus this method only provides the laser based landmarks with additional redundant information about the corners and semiplanes from vision data. In contrast, the proposed method uses vision as the primary sensor to obtain vertical edge features and then uses data from the laser range finder to measure the range to those landmarks.

3.1.2 Objective

The main objective of this chapter is to introduce a novel integrated laser-camera sensor that can be readily used in landmark based simultaneous localization and mapping algorithms. In contrast to the other notable works in multisensor SLAM [1] the proposed method fuses the information in the sensor domain, rather than fusing map information that is being built using each sensor, as shown in Fig. 3.1. In the proposed work a camera is mounted on a laser range finder and the coordinate transformations are obtained through an experimental calibration process [102]. The vertical lines in the environment are detected using the image data (bearing information) and the range to the vertical lines can be then interpolated using the laser readings and the coordinate transformation between the laser and the camera. These located features are then used in the extended Kalman filter based SLAM formulation.

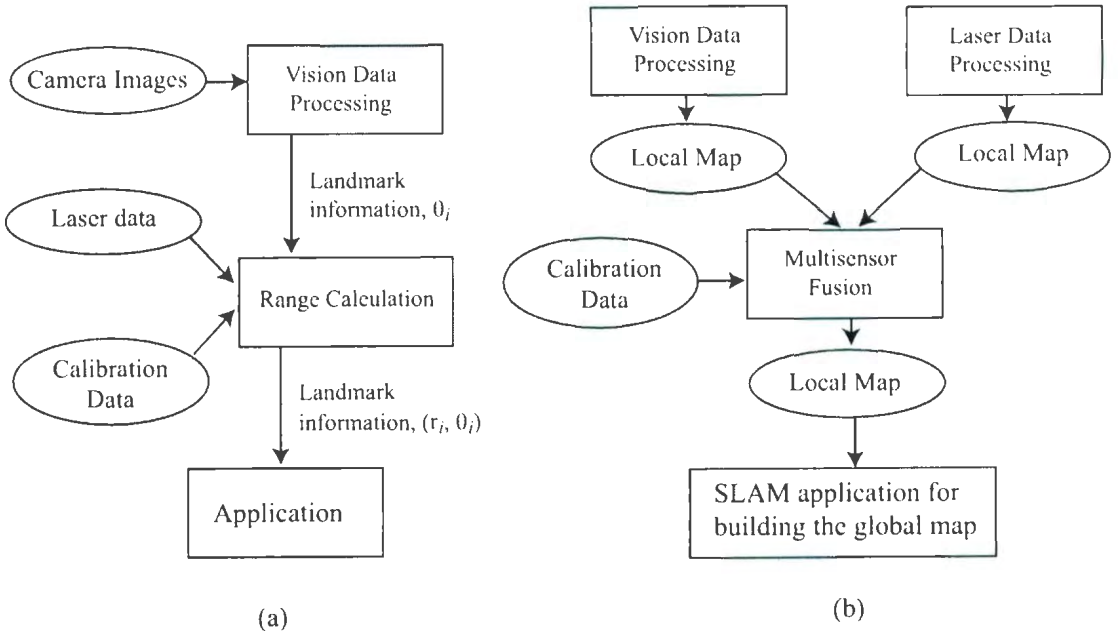


Figure 3.1: Block diagram of (a) the proposed multisensor based SLAM process and (b) the multisor sensor based SLAM process presented in [1]. The proposed method fuses the information from each sensor at a lower level whereas the method in (b) fuses the information at map level.

3.2 Landmark Detection and Position Estimation Using a Single Sensor

This section explores the applicability of each sensor for landmark detection and position estimation. In robotics, the camera and the laser range finder are the most commonly used sensors for environment sensing. Computer vision based solutions have long been proposed for detection and in many cases for position estimation of visually salient landmarks. The most important advantage of using computer vision for landmark detection is that it can detect visually salient landmarks with a high degree of details that can later be used for tracking or association. For example, scale invariant feature transform (SIFT) uses rich visual information to derive a multi dimensional descriptor of visual features [103]. This type of rich description is useful for associating features in stereo vision [44] and for homography estimation [104]. Due to the inherent sensor model, computer vision can only capture the bearing to a feature. Therefore in computer vision, stereo vision is the most popular method for direct landmark position estimation. On the contrary, a laser range finder scans its field of view to measure the distances to closest objects. Usually, the measurements are taken at very small angular resolution and a higher range accuracy than any of the other range sensors, providing a high resolution depth plan of the field of view of the scanner. Next, the issues relating to the landmark detection and position measurement using a single sensor are addressed.

Monocular vision has been widely used in visual landmark detection in bearing only SLAM. Starting from the initial works of Andrew Davison [48] the research in vision based SLAM has moved to realtime monocular SLAM [83, 105, 106] implementations. In [83, 105, 106] the position (depth) of the visual landmarks is estimated using repeated observation of the landmark, and when the estimation converges it is

initialized into the map. This type of feature initialization requires landmarks to be present in the field of view of the camera until the depth estimates converge to an acceptable level. Although these are pioneering methods in vision based SLAM, in typical application scenarios the landmarks cannot be guaranteed to remain in the field of view for a specific duration. In other methods the optical flow of a landmark, along with the robot velocities, can be used to calculate its position with respect to the robot frame. However, due to the high sensitivity to noise in robot velocity measurements, the uncertainty of the final calculated values can be extremely large and the resulting position calculations will be of limited use. This uncertainty problem in the position calculation is magnified at low robot velocities. Further, the optical flow based method cannot directly calculate the object position when the robot is making pure translational motion as shown in the next section 3.2.1.

In the detection of landmarks based on the laser range finder data, the corner and line (planes in the real world) features are the mostly used features [107, 1]. The landmarks that can be represented by a point in the map are often preferred over the line features, which can only be localized with a higher degree of freedom when the complete line segment is in the field of view of the scanner. The corner features that are invariant to the direction of the laser scan arise in the laser data due to objects such as corners in walls and other objects that have protrusions similar to legs of tables. However, in some cases these types of corner features may not be available in environments such as long corridors. Nevertheless, in most cases there are patterns on walls and other features that can be easily detected using computer vision. In addition, due to the differences in the appearance of surfaces under lighting, the corner features would usually appear as visually salient features. In the rest of this section two attempts in localizing features using computer vision and laser range data are discussed with their limitations. The next section introduces an integrated

laser-vision sensor that exploits the above mentioned properties of the visual features with the high accuracy of the laser based measurement.

3.2.1 Landmark localization using computer vision

Landmark localization using only monocular vision has been achieved using two main methods: bearing only localization and optical flow based localization. Bearing only localization requires multiple wide baseline frames to infer the 2D position of a landmark. Therefore, the position estimation and the accuracy of the estimation of a landmark using bearing only readings are highly dependant on the movement of the camera and the number of sensor frames. In contrast, the optical flow based feature localization can be used to calculate the landmark position as soon as accurate optical flow data becomes available. Thus, in this thesis, for monocular vision based landmark localization, only the optical flow based method was investigated.

From the six degrees of freedom general model, the horizontal velocity of features (optical flow) ($\dot{p}$) on the image plane can be derived from the horizontal feature position (p), heading velocity (v), rotational velocity (ω), and focal length of the camera (λ) as follows [23]:

$$\dot{p} = \frac{pv}{Z} - \frac{\omega}{\lambda}(\lambda^2 + p^2) \tag{3.1}$$

where Z is the distance to the feature in the direction of the heading velocity. Using the above equation and the camera model ($p/\lambda = X/Z$) where X is the perpendicular distance from the feature to the heading direction, the feature position with respect to the robot can be calculated by:

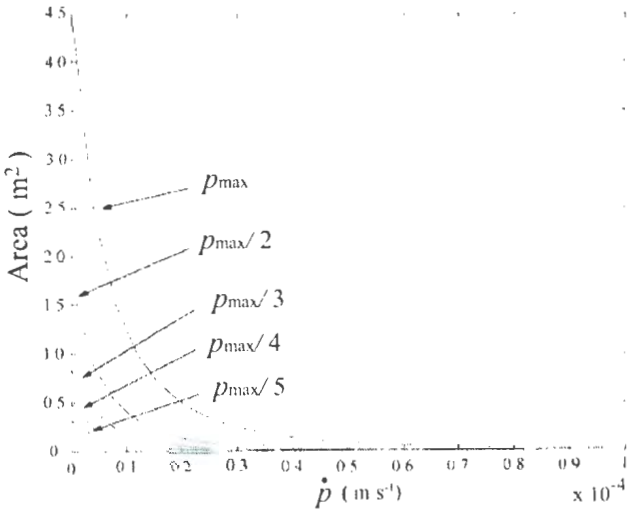


Figure 3.2: Sensitivity of the uncertainty of the feature localization. The uncertainty is quantified by the area of the ellipse representing 95% confidence. $p_{\max} = 1.75mm$, $v = 0.092m/s$ and $\omega = 4 \times 10^{-3}rad/s$. ($\sigma_p = 10.9 \times 10^{-6}m$, $\sigma_{\dot{p}} = 0.3 \times 10^{-4}m/s$, $\sigma_v = 7.8mm/s$, $\sigma_\omega = 10^{-6}rad/s$)

$$\begin{aligned} X &= \frac{p^2 v\lambda}{\dot{p}\lambda + (\lambda^2 + p^2)\omega} \\ Z &= \frac{pv}{\dot{p}\lambda + (\lambda^2 + p^2)\omega} \end{aligned} \tag{3.2}$$

The covarianceof the calculated position can be found using the first order Taylor expansion of the feature position $[X, Y]^T$. The covariance matrix of the position calculation can be obtained from

$$\Sigma_{X,Z} = J \cdot \mathrm{diag}\left[\sigma_p, \sigma_{\dot{p}}, \sigma_v, \sigma_\omega\right] \cdot J^T \tag{3.3}$$

where

$$J = \begin{bmatrix} \frac{\partial X}{\partial p} & \frac{\partial X}{\partial \dot{p}} & \frac{\partial X}{\partial v} & \frac{\partial X}{\partial \omega} \\ \frac{\partial Z}{\partial p} & \frac{\partial Z}{\partial \dot{p}} & \frac{\partial Z}{\partial v} & \frac{\partial Z}{\partial \omega} \end{bmatrix}$$

and σ_p, $\sigma_{\dot{p}}$, σ_v, and σ_ω are the standard deviations of the horizonal feature position on the image, horizontal optical flow, heading velocity and rotational velocity of the robot, respectively. The measurement covariance matrix is considered diagonal assuming that each of the measurements are independent from others. The uncertainty of the calculated locations can be evaluated by comparing the area of the ellipsoid defined by the 95% confidence interval. The uncertainty comparison for varying optical flows and feature positions is shown in Figure 3.2. From Figure 3.2 it is clear that at low optical flows the uncertainty increases regardless of the feature position on the image. Moreover, as the feature moves closer to the edge of the image, the uncertainty increases even for the same optical flow value. Generally, a robot encounters many combinations of robot velocities and feature positions which could give rise to high covariance values in the feature position calculations. The limitations in the usable range of optical flow and feature position make the optical flow based feature position calculation method unsuitable for SLAM applications.

3.2.2 Landmark localization using laser data

The direction invariant features in the laser data can be identified as unique landmarks using the minimum points in the laser data plot [22] which appear as peaks from the robot direction when the laser data is connected with line segments. These landmarks generally remain in the laser data regardless of the direction of scan. In addition to the convex features that appear as minimum points in the laser data, concave points such as sharp corners can be reliably detected in the laser data. However, as shown in Figure 3.3, in certain environments such as in long corridors, there might not be any directional invariant features. In such cases feature based laser only SLAM implementations will not be possible unless higher level features such as lines are

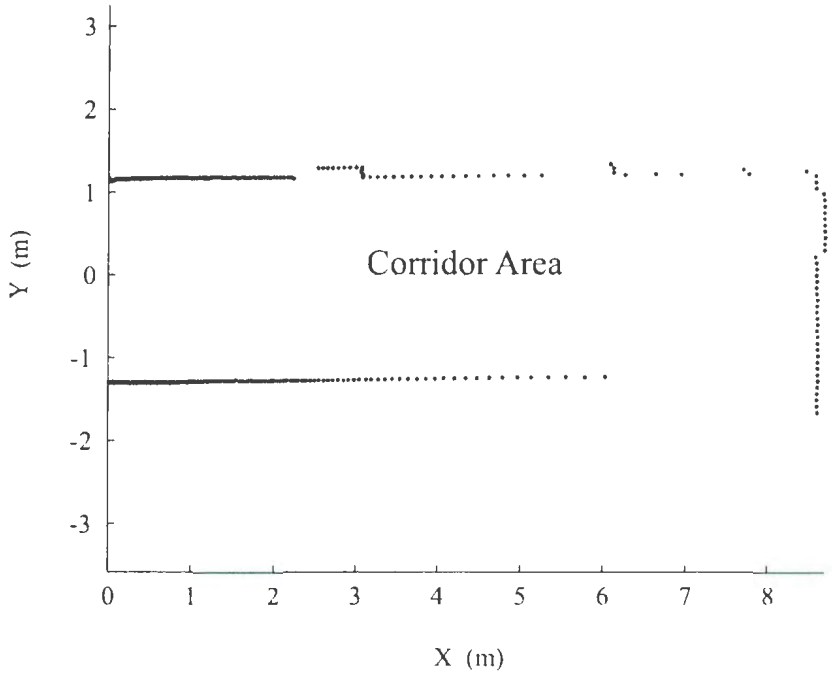


Figure 3.3: A typical laser reading in an indoor environment where there are not sufficient direction invariant features.

used.

3.3 Calibrated Laser-Vision Sensor

A camera is mounted on the laser range finder using a custom made bracket as shown in Fig. 3.4. The camera is mounted at the center of the laser range finder to maintain the coordinate transformation between the laser scanning plane and the camera coordinate system as simple as possible. The coordinate frames are defined as shown in Fig. 3.5. In the real setup the axes z_l and z_c coincide with each other (i.e. $a = 0$) and $b = 19$cm.

Figure 3.4: The camera and the laser range sensor used in the experiments.

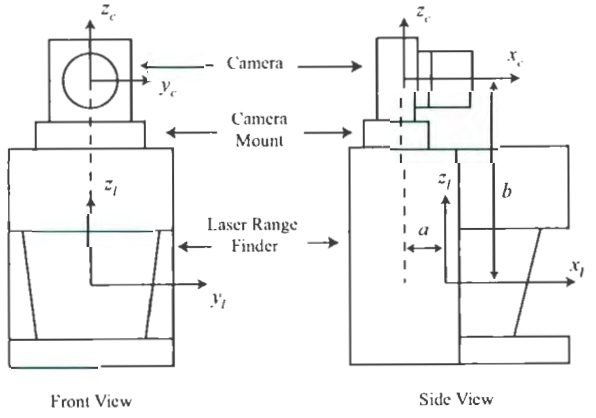


Figure 3.5: Coordinate frames of calibrated laser-vision sensor

3.3.1 Visual Landmark Detection

Landmarks in the camera images can take several forms. The most common landmarks are the visually distinct corner features. Other visually salient landmarks include lines, arcs, and user defined objects. In the proposed method visually salient vertical line features were detected in the captured images. Although the choice of vertical lines as the exclusive landmark type restricts the applicability of the method in diverse environments, it offers a trade off between simplicity in landmarks the applicability in most structured indoor environments. Line features are robust in terms of detection accuracy and repeatability compared to corner points and much easier to describe and detect than complex composite objects. In this work two algorithms have been evaluated for the detection of vertical lines in the images.

1. Hough transform based method.

2. Artificial corner feature based method.

Line detection algorithms based on the Hough transformation are most popular in computer vision and pattern recognition. Hough transformation typically accumulates the votes for line configurations based on their support in the binary image. Since it is of interest to detect only the vertical (or close to vertical) lines, the search space can be restricted to compute the angle values in the vicinity of zero, thus reducing the computational cost. In addition to the Hough transform based method, a simpler and computationally efficient corner based method was tested for vertical line detection. Initially, a set of horizontal lines were superimposed on the original image as shown in Fig 3.6. Then, all the resulting corner features were detected using a Harris corner detector [43] and are indicated by the white circles in Fig. 3.6.

This list of corner features is then searched for sets of features that are vertically aligned. If the number of features in a set is greater than the threshold value, then

Figure 3.6: Line feature detection using artificially generated corner features.

the average horizontal position of the features is identified as a consistent vertical line. Identified lines are marked with white line stubs at the bottom of the image frame shown in Fig. 3.6. A comparison of the two methods is shown in the Fig. 3.7 for three typical images that are taken during a robot run. The lines in the top part of the image are the ones detected using Hough transformation and the lines in the bottom part are detected using the corner based method. It is evident from the images that on average Hough transform returns more line images than the corner based method. This can be attributed to the fact that it accumulates the evidence for lines in the whole region rather than for some sampled points in the image, as in the case with the corner based method.

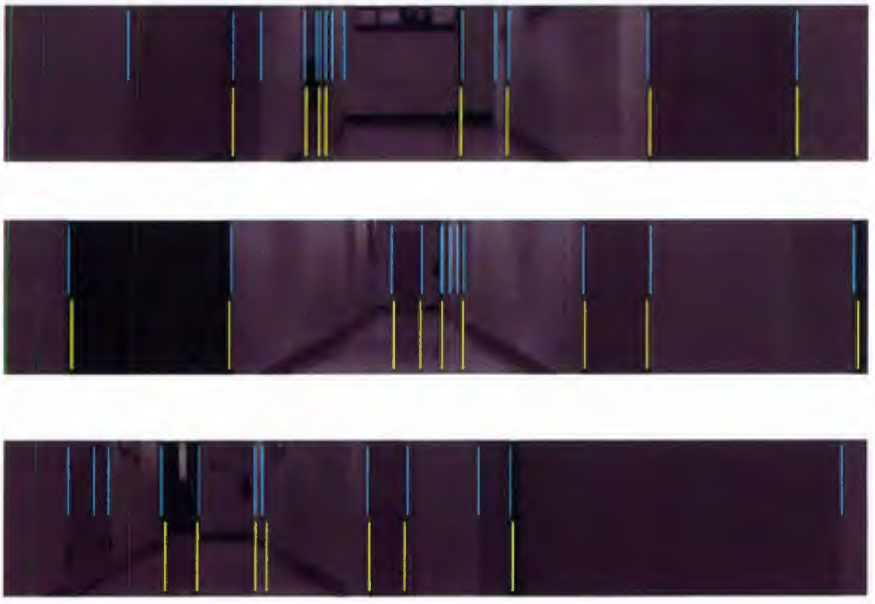


Figure 3.7: Detected line features using Hough transformation and the corner based method.

3.3.2 Sensor calibration

In order to measure the distances to the visual landmarks using the laser range finder, the coordinate transformations of the two sensors have to be accurately calibrated. There are two possible sources for errors in the calibration information: the errors in the alignment of the frames of the sensors (parameters a and b in Fig. 3.5) and the errors in camera calibration. Although the camera is calibrated using standard camera calibration techniques[1], the distortions especially at the edge of the images, contribute significantly to the errors.

The main objective of the sensor calibration method is to accurately map the field of view of the camera to that of the laser range finder. In order to achieve that objective, a 'v' shaped target with black and white faces is placed in front of the robot. In a series of image and laser data with the 'v' shaped object placed to span the field of view of the camera (since the field of view of the camera is less than that of the laser range finder), the angle to the tip of 'v' is measured from the center of each sensor. In the camera images it is measured in degrees from the optical axis (θ_c) and in the laser range finder it is measured from the central laser scan (θ_l). Thus, the error in the calibration can be calculated from $e = \theta_l - \theta_c$. As shown in Fig. 3.8, the error e is approximated using a higher order polynomial $e(\theta_c)$ with respect to θ_c. Thus for any new measurement in the image θ_c, the corresponding mapping angle in the laser range finder can be calculated from $\theta_c + e(\theta_c)$. Similarly, the reverse mapping, the mapping of a reading in laser data onto the image, is also possible with the same data with a new calibration curve of $e(\theta_l)$ vs. θ_l.

[1]MATLAB toolbox for camera calibration, http://www.vision.caltech.edu/bouguetj/calibdoc/

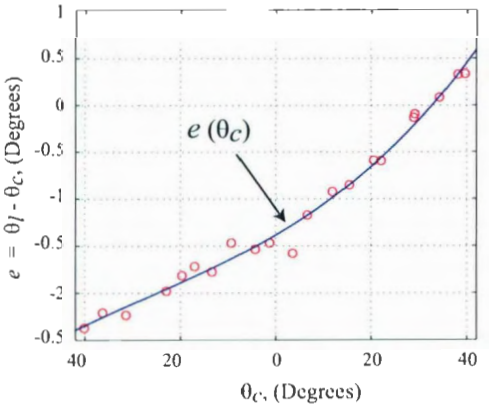


Figure 3.8: Calibration curve for mapping between the field of view of the camera and the field of view of the laser range finder.

3.3.3 Measurement Model

The goal of defining a measurement model is to calculate the range to the landmark that has been detected by computer vision and then define its uncertainty. The bearing angle (θ_c) of the detected landmarks (line features) can be calculated using a camera model with sub pixel accuracy. A laser ranger provides a set of scanned readings that provides the range to the objects in the laser scan plane. The scanner is able to operate in a field of view of 180° with a half a degree resolution. Therefore, using the coordinate transformation between the camera and the laser range finder along with the calibration information, the range to the line features can be calculated. Due to the resolution constraints in the laser data, the range value has to be interpolated from the data to increase its accuracy. This process of range interpolation is shown in Fig. 3.9. It should be noted that the coordinate frame of the laser range data and the camera coincide with each other as the calibration is already applied to the bearing angle of the camera. Thus, in Fig. 3.9 bearing angle can be explicitly expressed as in the laser coordinate frame.

Assuming the resolution of the laser range scanner is at 0.5°, the range to the line

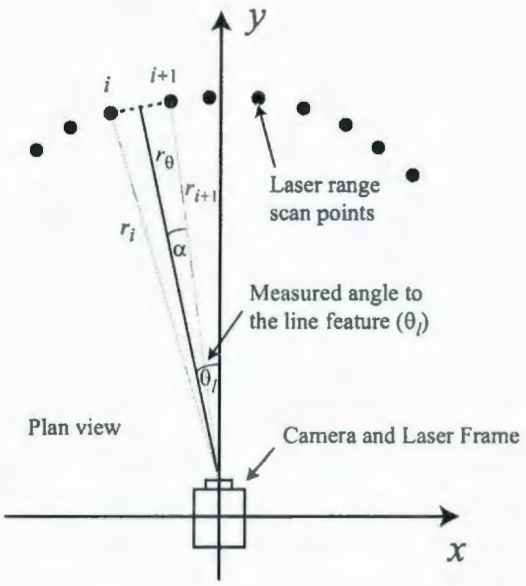


Figure 3.9: Interpolation of the range to the line feature.

feature can be calculated using the following interpolation:

$$r_{\theta_l} = \frac{r_{i+1}\cos(\theta_l - \alpha) + r_i \cos(0.5^0 - \alpha + \theta_l)}{2\cos(\theta_l)} \tag{3.4}$$

Since the bearing to the feature is measured using the camera model, and the range is measured using the interpolated range data, the uncertainty of the measurements also have to be calculated using the characteristics of each sensor. The uncertainty in the bearing angle increases with it as the effects due to the increasing effects of the lens distortion as the angle move towards the periphery of the image. However, it impossible to quantify the uncertainty without the exact lens distortion to compare with the current image. Therefore an estimated worst case value is used for the bearing uncertainty. The uncertainty in the range measurements is approximated with the possible uncertainty at the typical maximum range limits due to unavailability of a systematic error model from the manufacturer. Thus, the covariance matrix of the measurements can be expressed as:

$$R = \text{diag}[\ \sigma_r^2 \quad \sigma_\theta^2\ ] \tag{3.5}$$

where σ_r and σ_θ are the standard deviations of the range and bearing measurement errors, respectively.

3.4 EKF based SLAM Algorithm

The EKF based SLAM algorithm described in Chapter 2 has been used for SLAM. The algorithm is summarized in Algorithm 1. The equations are elaborated in detail in Chapter 2.

Algorithm 1 `EKF based SLAM Algorithm`

1: Initialize the robot pose and covariance to zero.
2: **while** Robot runs **do**
3: Move the robot to the next location.
4: Predict:
$x_r(k+1) = f(x_r(k),\ v(k),\ \omega(k))$
$\Sigma^-(k) = \nabla_x f(k)\Sigma^+(k-1)\nabla_x f^T(k) + \nabla_u f(k)\Sigma_u \nabla_u f^T(k)$
5: Observe the landmarks.
6: Perform data association.
7: Update:
$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{W}(k)\ v(k)$
$\Sigma^+(k) = \Sigma^-(k) - \mathbf{W}(k)\ \mathbf{S}(k)\ \mathbf{W}^T(k)$
8: Perform map management.
9: **end while**

3.5 Experiments and Results

This section provides information about experiments that have been carried out to validate the suitability of the integrated sensor. Before the description of the experiments and their results, a key step in the selection of the detected visual landmarks

Figure 3.10: The curve constructed on the image plane by connecting the laser readings mapped from the laser coordinate frame to image frame.

has to be explained. In some cases, the visual landmark (the vertical line) would not cross the plane of the scanning laser. As an example, there could be visual features on protruding (or retracted) walls or objects on top of tables. In such cases the range to those landmarks cannot be guaranteed to be accurate. Thus the landmarks that do not intersect with the laser plane have to be removed from the list of detected landmarks before the calculation of the range to the landmarks. After the visual landmarks have been detected, the first step in the detection of such landmarks is the reverse mapping of the laser points onto the image using reverse sensor calibration (as described in the section 3.3.2) and coordinate transformation. Once the field of view of the two sensors is calibrated, the horizontal and vertical position (p and q) of the laser point in the image can be calculated by using:

$$p = \lambda r_i \tan(\theta_l)$$

$$q = \frac{\lambda b}{r_i \cos(\theta_l)}$$

where θ_l is the angle to the laser point from the vertical plane through the optical axis, b is the vertical displacement of the camera and laser coordinates and r_i is the laser reading that is being mapped. Fig. 3.10 shows the curve constructed from mapped laser readings.

In the next step the intersecting points between the vertical lines and the curve of the mapped laser readings are found. The vertical gradient in the neighborhood of

each intersecting point can be calculated by a suitable gradient detector. Then the vertical landmarks corresponding to points with weak total vertical gradients can be dropped. Although this method is able to remove most of the landmarks that do not intersect the laser plane, in rare cases two vertical aligned landmarks that belong to objects with different ranges could yield erroneous range information.

Two SLAM experiments were carried out to evaluate the fitness of the multi-sensor landmark detection and measurement method. In the first experiment the robot was driven through a regular office environment where it encountered narrow corridors, open office areas, and regular object clutter that are typical to an office environment. The robot travelled approximately 67 m making two loops through the office environment. In the second, longer experiment the robot was driven through the main corridors in a university building where the corridors were considerably wider compared to the first experiment. The robot travelled approximately 148 m while looping one and half times in the same environment. The experiments were carried out using a Pioneer 3AT robot equipped with a SICK laser range finder and a camera with a regular off the shelf lens. During this experiment the laser range data, images from the camera and odometry data were logged at regular spatial intervals (20 cm or 2° apart, whichever occurs first). The noise levels that have been used in the map estimation and localization are listed in Table 3.1.

Figure 3.11 shows the process of feature detection and localization using an integrated sensor for a typical set of image and laser scan data. As can be seen from Figure 3.11 the laser range finder can only detect the landmark at location C (using the intersection of two lines) while computer vision can be effectively used to detect other landmarks that can only be detected using a camera (at locations A and B).

As discussed previously, the protruding features in the laser data can be detected as landmarks in the laser data. Figure 3.12 shows a comparison between the number

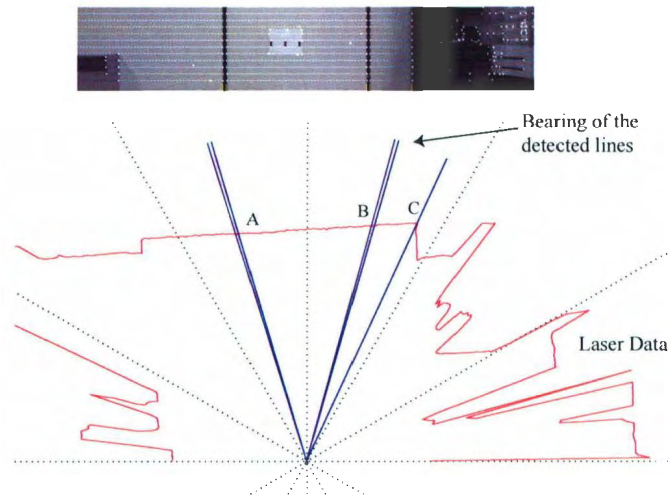


Figure 3.11: The landmarks detected by the camera and their bearing angle superimposed on laser readings.

of landmarks that can be detected in laser data and in image data during the first experiment. It is clearly evident that there are significant periods when image features outnumber the laser based landmarks. Further it should be noted that when there is a low number of visual features there is a significantly higher number of laser based landmarks. Although the results are purely specific to a given environment, the total number of landmarks can be improved using the proposed method in addition to the laser based landmarks.

After the landmarks are detected and located using laser data and images, the data is processed off-line using the EKF method outlined in Chapter 2. The Joint Compatibility Branch and Bound (JCBB)[66] algorithm was used for the data association. In the first experiment a map consisting of 71 landmarks has been built at the end of the run (Figure 3.13(b)). Figure 3.13(a) shows the robot path using pure odometry data. The 95% confidence bounds of the errors in the robot pose estimate are shown in Figure 3.15. In Figure 3.15 it is possible to observe the rapid decrease in the uncertainty of the robot position estimation due to the loop closing around

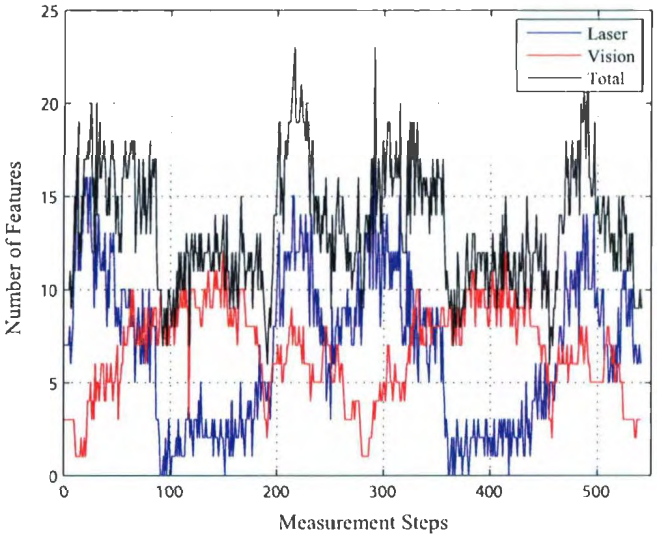


Figure 3.12: Number of landmark features detected by vision and laser system.

the midway point of the robot run. In the second experiment the robot constructed a map (as shown in Figure 3.14) that contains 271 landmarks. Although the robot travels a considerably longer distance in a different environment compared to the first experiment, a similar pattern can be observed in the performance of the EKF based SLAM in the second experiment. The accuracy of the EKF based SLAM algorithm was enough to robustly close the loop in the long run, but during the initial steps of the loop closing there were erroneous data associations. The ability of the algorithm to recover from the initial errors data association can be mainly attributed to the large size of the map compared to the number of erroneous data associations.

Table 3.1: The measurement of noise levels of the respective sensors that is used in the SLAM.

Quantity	Measurement noise
Range to the Landmark (cm), σ_r	5.0
Bearing to the Landmark (Degrees), σ_θ	1.0
Robot heading velocity (cm/sec), σ_v	0.5
Robot rotational velocity (Degrees/sec), σ_ω	0.025

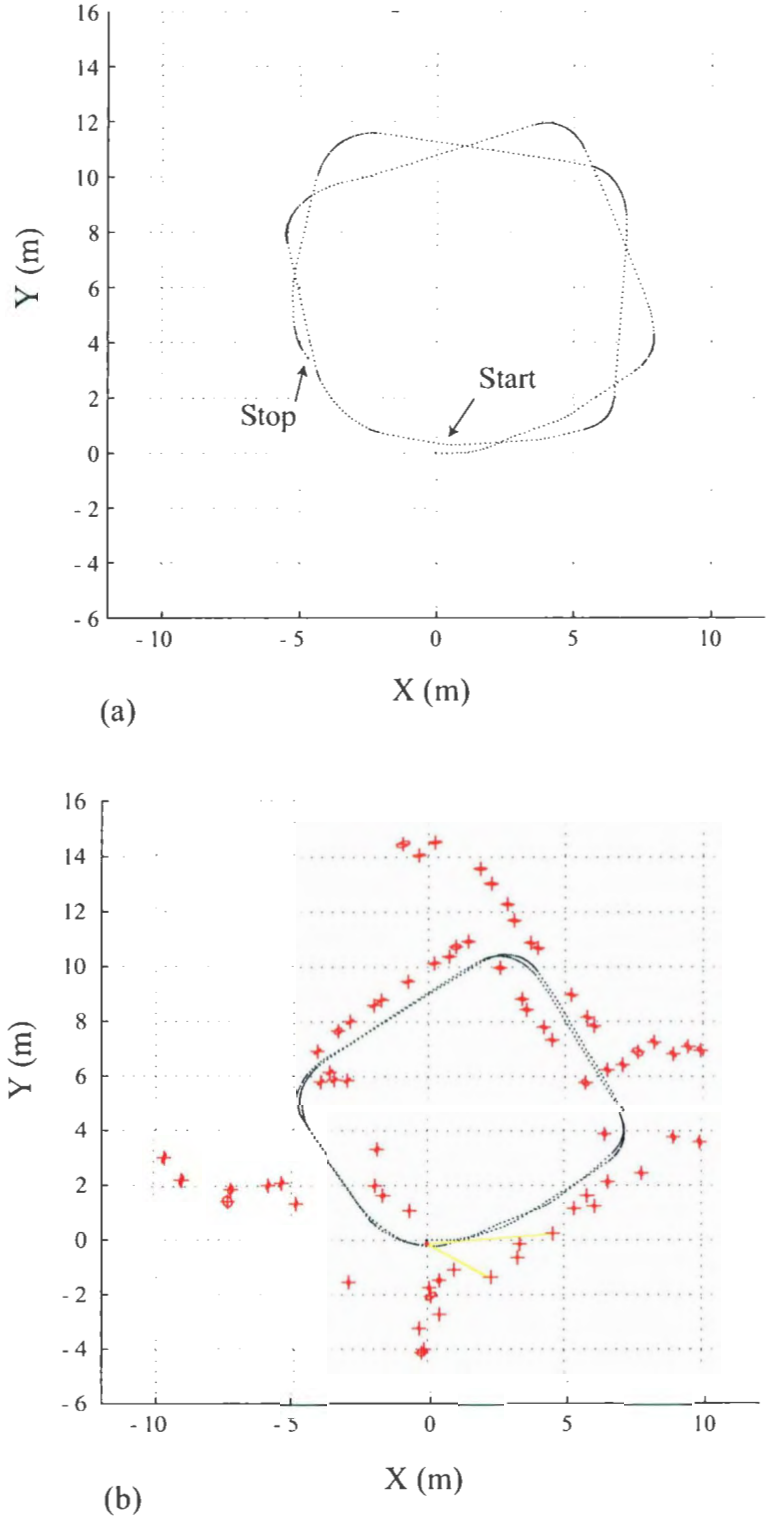


Figure 3.13: Results of a localization and mapping of the first robot run: (a) with odometry, and (b) using EKF and vision-laser landmark localization (see the attached video file for incremental map building along with the current image frame).

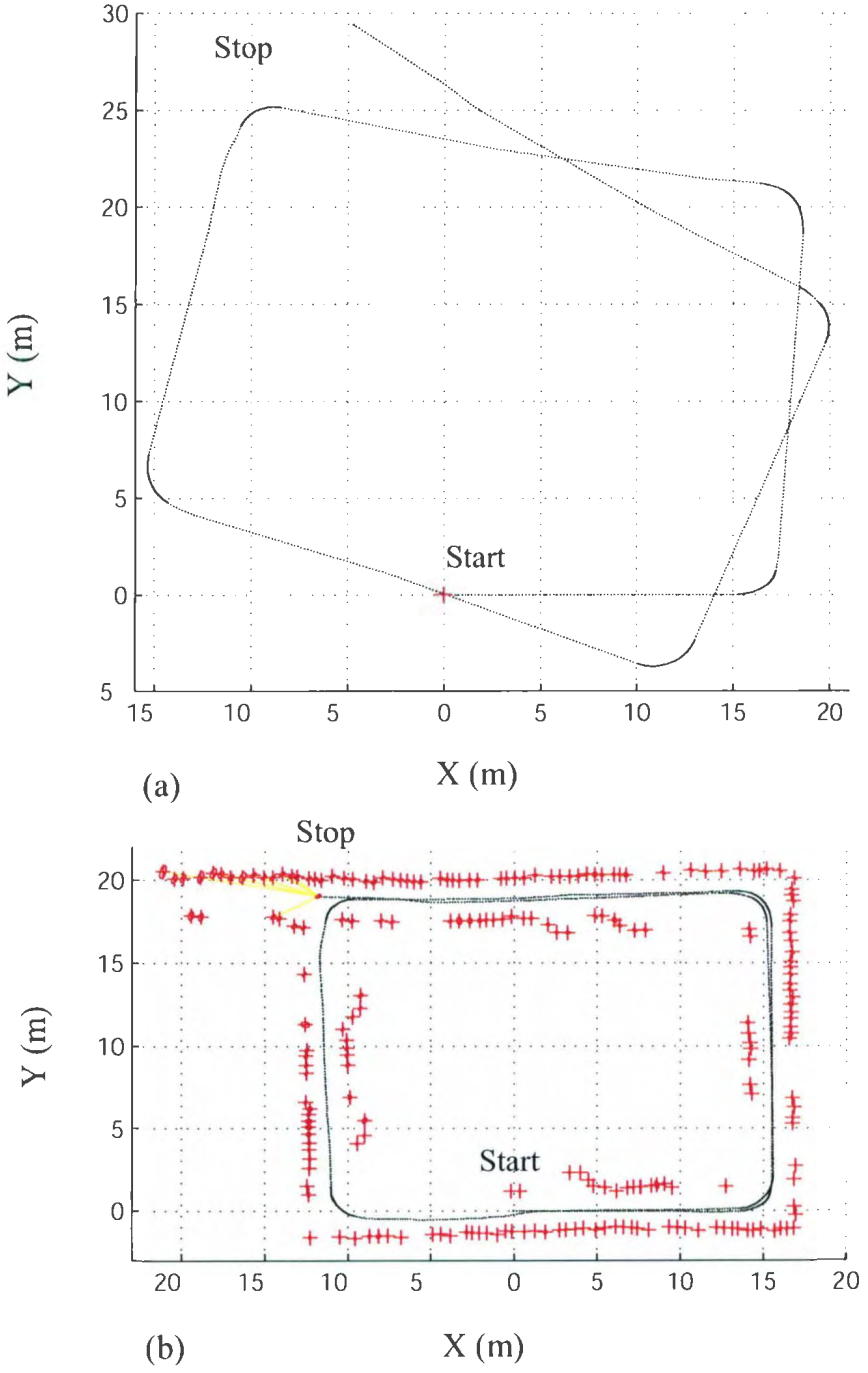


Figure 3.14: Results of a localization and mapping of the second robot run: (a) with odometry, and (b) using EKF and vision-laser landmark localization (see the attached video file for incremental map building along with the current image frame).

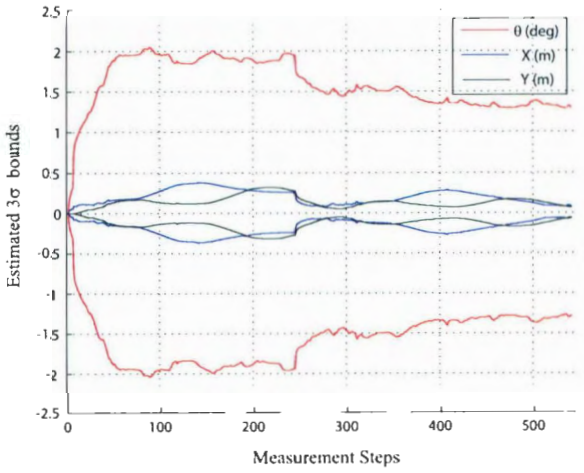


Figure 3.15: 3σ bounds of the localization errors of the first experiment.

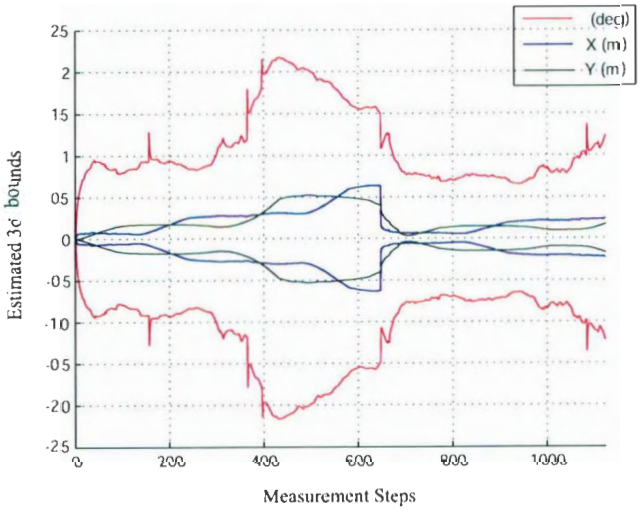


Figure 3.16: 3σ bounds of the localization errors of the second experiment.

3.6 Conclusion

In this chapter it has been shown that computer vision and a laser range scanner can be used to accurately detect and measure the visually salient landmarks in the environment. Further, such measurements can be readily integrated into the EKF based SLAM method to build maps of typical indoor environments. From the results it is evident that using a calibrated laser-vision sensor, a higher number of landmarks can be detected than with each sensor. Future extensions of this work include the use of more accurate sensor uncertainty modeling especially in the case of bearing angle to the landmark and experimentation in large looping environments with possible sub-mapping. Some visual landmarks are present in the form of wide vertical strips, in which two side edges are detected as vertical lines. Thus they are recognized as two landmarks and the SLAM algorithm will attempt to initialize them as such in the map. However, since they are often physically close together only one of them will be initialized into the map. Further, when the robot is away from visual features as described above, the line detection algorithm will often detect a single line due to the limitation in the resolution of the camera. However, as the robot gets closer to the object, it will appear as two landmarks and the data association algorithm will have to decide the best edge to be assigned to the feature that is already in the map. Thus a better sensor model that can handle this type of composite objects is required.

Chapter 4

Modeling of Environmental Dynamics

So far this thesis has focused on navigating a robot in environments where the surrounding objects and perceived landmarks for SLAM have been assumed to be static. The SLAM development based on laser or vision sensors always attempts to identify stationary landmarks. When objects move at relatively high velocities compared to the robot, the SLAM filter has the capacity to ignore such objects through data association. However, when objects move at a relatively low velocity, the data relating to moving landmarks may be associated with existing landmarks leading to faulty mapping in the SLAM. Therefore the objective of this chapter is to develop a technique to detect and isolate moving objects from the laser scan taken by the moving robot and finally to accommodate the SLAM process to be effective in dynamic environments.

4.1 Background

Moving object detection and tracking has been employed in a wide variety of applications ranging from tracking of ships, airplanes, submarines, projectiles and vehicles to people. The common objective in all of these applications is to identify the sensor data corresponding to the concerned object and to estimate its position for tracking purposes. In many applications such as maritime and airborne target detection and tracking, the objects can be easily isolated from the radar data. This is mainly due to the fact that radar reflected from the object will have uniquely defined features in the workspace with respect to the background. In contrast, moving object detection is a challenging problem in environments where there is no such wide difference between the object and the background. In such cases simple detection would not suffice and one has to closely explore the properties of the moving object with respect to its background.

There are numerous types of sensors that are used in observing the environment, known as extroceptive sensors. The most commonly used sensors are radar, laser, sonar, and vision [9]. Radar sensors are popularly used to detect objects that are a few meters to several hundred kilometers away, whereas laser range finders are used to detect objects that are only tens of meters away, but with a higher accuracy than radar. Sonar sensors use bursts of acoustic impulses to measure the distances to the objects using the time of flight and phase shift of the returning signal. While sonar is much cheaper than laser or radar, it has only limited applicability due to its accuracy limitations. Sonar is the most widely used underwater sensor due to the higher performance of acoustic signals in denser media such as water. Computer vision is a versatile sensor that can retrieve a large amount of information compared to other sensors. In contrast with the other range measuring sensors, computer vision

lacks the ability to directly observe the structure of the environment. There are numerous techniques for recovering the depth to an object from vision data, stereo vision being the most commonly used one.

In mobile robotics these sensors are mainly used for navigation and path planning. Additionally, the data gathered from the extroceptive sensors are increasingly used in real time mapping and localization [8]. In mapping and localization, it is assumed that the environment remains stationary during the operation of the robot. However, this condition could be violated in most practical scenarios, especially in uncontrolled outdoor environments. Thus there should be adequate methods to identify and classify moving objects by moving robots, in order to produce an accurate map that is entirely composed of stationary objects. Moving object detection is important for two main reasons. The information about the moving objects can be used in safe robot navigation (obstacle avoidance). If a robot can make a sufficiently reliable estimate about the velocity of a moving object as early as possible, its path planning algorithm can use this information to efficiently circumvent the obstacle. Secondly, moving object detection can function as a filtering process in which the data corresponding to moving objects can be removed from the sensor data in order to provide the SLAM algorithm with data from only stationary objects.

4.1.1 Challenges in Robotics

The laser range sensor is the most popular sensor in indoor and some outdoor operation scenarios for mobile robots. It has been used in moving object detection in mostly trivial scenarios where simple free space consistency is used to detect the motion in objects [21]. However, in many other situations, moving object detection has been found to be non-trivial. Some details of the identified challenges in moving

object detection are summarized below.

Low relative velocities When the relative velocities between the robot and the object are sufficiently large, then the object can be completely separated in the laser data in the world frame. This would yield a trivial moving object detection case, which can be achieved through direct closest point elimination. However, at low relative velocities the object separation will be at a minimum. In such cases the complete moving object detection will be a complex task, as the laser data that represents the moving object may overlap in several different ways, depending on the direction of the relative velocity.

Complexity of the objects Objects in the environment can take arbitrary shapes and forms. The shape can be of fixed nature or change with time through either deformation or rotation. Thus the object can appear in many geometric forms during the lifetime of a track. Therefore the moving object detection algorithm should be robust enough to detect objects under many different scenarios. Additionally the moving object detection system should be capable of detecting multiple objects. When the robot moves, the areas that were previously occluded but stationary will become visible to the laser and thus the detection algorithm must have sufficient capabilities to identify these occluded areas to prevent them from being classified as moving objects.

4.2 Moving Objects and SLAM

Moving object tracking is a popular and widely researched topic in computer vision[1]. Computer vision based methods use color and shape features of objects for detection

[1] http://iris.usc.edu/Vision-Notes/rosenfeld/contents.html

and employ numerous estimation techniques for tracking. Computer vision based tracking of moving objects by moving robots (or by moving platform, in general) still remain a significant research challenge. Examples of some of the attempts made to solve the problem of computer vision based moving object detection from moving robots are shown in [108, 109, 110, 111]. In comparison to laser range based methods, computer vision based methods exhibit some drawbacks. Among others, they include low precision in position estimation, susceptibility to lighting conditions, and reduced field of view when regular lenses are used. In contrast laser range finders provide accurate range data of the environment in a wider field of view.

The Simultaneous Localization, Mapping and Moving Object Tracking (SLAM-MOT) method [49, 50] uses two different rules, to detect approaching and leaving objects. Although the rule related to approaching objects is straightforward, the rule related to leaving objects requires more than two laser scans to identify the moving object. Moreover this method has limitations in detecting the complete object when the object is moving sideways in the field of view. The people detection method in [20] uses a laser scanner and a camera to specifically search for the two leg positions and skin colour. Although [20] provides interesting work on sensor fusion, it can only detect people when they are facing the laser. Lindstrom and Eklundh [21] provide another moving object detection method based on the static world assumption, which provides interesting results of human tracking by a moving robot. In their method all laser readings and the robot itself form a closed polygon, which is also the free space "seen" by the robot. In subsequent scans, the violations of this free space are monitored and such violations are detected as moving objects. While this method can detect objects that are approaching the robot, there is a possibility of not detecting the objects that behave otherwise, since they don't violate the free space condition. Mendes *et al.* [51] provide a target tracking system with a laser

scanner, which assumes that all the objects (or scan segments) are within a certain predefined range from the scanner (there is no distinction between moving objects and stationary objects) and are classified into a known set of objects based on the shape features extracted from the laser data. Further, the presented method will fail to identify moving objects which cannot be classified into the known set of objects. An interesting occupancy grid based moving object detection method is presented by Schulz et al [22]. In their application of the sample-based joint probabilistic data association filter (SJPDAF) they have compared occupancy grid maps of two subsequent laser scans to generate the difference map. The generated difference map is then compared with the minimum points in the current laser scan to remove any stationary features in the map, and the resulting feature map will only contain the moving objects. Also, when a moving object moves sideways at low velocities, the method in [22] will fail to completely recover the moving object. Montemerlo *et al.* [52] provide a multi robot localization and people tracking method based on particle filtering. Laser readings are segmented into clusters and then fitted against cylindrical models, which approximate a model human torso. Fod *et al* [53] propose a people tracker using multiple laser scanners. [53] adopt a blob and object model to combine laser segments from each scanner (blob) to a single object (object). The blobs are identified using a foreground method where all laser scans that belong to furthest objects are assumed to represent stationary objects. In [112] a model based people tracking algorithm using the laser range finder is presented. The algorithm first eliminates the closest points in two subsequent laser range scans and then attempts to identify the clusters of laser readings which fit a model that resembles the cross section of the legs of a person. This algorithm is only suitable for detecting human motion where the visible parts of the legs conform to a given model. The limited applicability in the model based detection is further increased when a person

is moving at a distance from the laser where the number of readings that corresponds to the legs would be minimal. Prassler, Scholz and Elfes [113] present a real time occupancy grid based method for detection and tracking of multiple moving objects using *time-stamp* maps. The method establishes moving object hypothesis from the difference between two subsequent occupancy maps created using laser data. These initially detected moving objects are further filtered by the size of the clusters. The inability to detect slow moving objects (aperture problem) is a limitation of any difference based detection method, owing to the resolution of the sensor and the noise content. When an occupancy grid based method is used as opposed to the use of direct laser range data, the resolution of the laser readings further decreases, and thus the ability to detect even relatively slow moving objects that otherwise are visible in direct laser data, also decreases.

In this chapter a systematic algorithm is proposed to maximally recover the moving objects from laser range scans. The proposed method can recover multiple moving objects regardless of their direction of movement with respect to the robot. The proposed moving object detection algorithm has two distinct steps: (1) laser scan segmentation, and (2) detection of the moving objects in the laser scan segments and the calculation of their position.

4.3 Laser scan segmentation

The objective of a laser scan segmentation algorithm is to identify the laser scans corresponding to the moving objects. At any given time the two subsequent laser range readings are defined as L_P and L_C, where subscripts C and P stand for the current and previous laser scans, respectively. L_C represents a set of range readings returned by the scanner in a single scan. Each reading is represented by the superscripts i or

j, which is a 2D position vector. Two sample laser scans are shown in Fig. 4.1.

In this algorithm it is assumed that initially two laser scans are perfectly aligned with all their stationary objects. This implies that in each laser scan there should be a significant amount of scan points that belong to stationary objects. This method will not suffice for environments that are highly cluttered with moving objects, because there will not be adequate data to properly align any two subsequent scans. Laser scan alignment is a heavily researched area in robotic mapping and localization. A suitable method can be found in [13].

4.3.1 Definitions

The two sets of laser readings can be divided into different mutually exclusive sets, depending on their physical representation, as shown below.

$$L_P = A_P \cup O_P \cup M_P \cup N_P \tag{4.1}$$

$$L_C = A_C \cup O_C \cup M_C \cup N_C \tag{4.2}$$

where A_C and A_P are the laser readings that represent the same stationary objects in the two scans. O_P are the readings in L_P that will be occluded by the readings of L_C, when the robot moves to the current position. O_C are the readings that have been occluded by the readings of L_P, when the robot is in the previous position. M_C and M_P are the readings belonging to the moving object in the respective laser scan, but not occluded by the other. N_C and N_P are the readings that are out of the field of view of each scan when the robot is at the other position. Fig. 4.1 shows the regions in the scans that belong to the corresponding sets.

The following observations can be made regarding the range reading sets presented above.

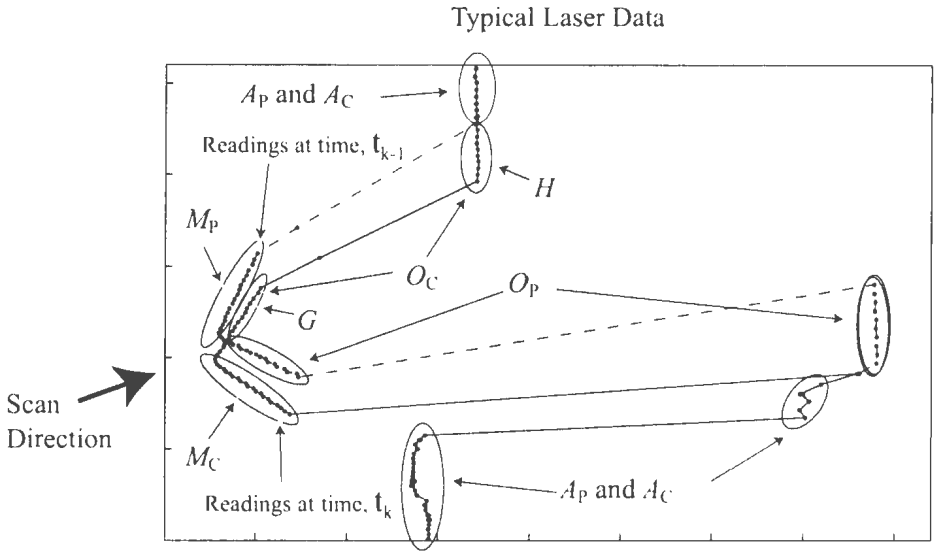


Figure 4.1: Typical laser scans from a stationary robot. The object on the left hand side moves downward in a negative y-direction.

1. The laser scans that are spatially close to each other (after proper alignment) belong to A_C and A_P. Therefore A_C and A_P can be identified by searching for the spatially closest points in two laser scans, L_C and L_P.

2. M_P^i is on or close to the scan line, which emanates from the laser when the robot is at the current position resulting in O_C^j. Similarly, O_P^i is on or close to the scan line, which emanates from the laser when the robot is at the previous position resulting in M_C^j. Apart from yielding different sets, this relationship would also yield a point to point correspondence between the pairs (M_P^i, O_C^j) and (M_C^i, O_P^j).

3. In the point to point correspondences identified according to observation 2, the following is always true for the range values of the corresponding pairs of laser

readings.

$$\begin{aligned} r(O_C^j) &> r(M_P^i) \\ r(O_P^i) &< r(M_C^j) \end{aligned}$$

where $r(\cdot)$ is the range value of the corresponding laser reading.

4.3.2 Segmentation Algorithm

The main objective of the segmentation algorithm is to classify the laser readings into sets, A_C, M_C and O_C. The algorithm has three main stages. These are (1) identification of A_C and A_P, (2) separation of M_C and O_C, and (3) segmentation of identified sets. These three stages are discussed below.

1. Through an element by element comparison the closest points of the two laser scans can be identified and removed. This operation can be described as a set operation as described in (4.3), assuming that the closest elements are common elements in the sets L_C and L_P.

$$\begin{aligned} (L_C \cup L_P) - (L_C \cap L_P) = & \underbrace{O_C \cup M_C \cup N_C}_{B_C} \\ & \cup \underbrace{O_P \cup M_P \cup N_P}_{B_P} \end{aligned} \tag{4.3}$$

2. Algorithm 2 can be used to further identify the sets M_C and O_C from B_C. This algorithm uses the second observation in section 4.3.1 to identify M_C, O_C, M_P and O_P from B_C and B_P. The rest of scan data in B_C and B_P that does not belong to those four sets can be classified into N_C and N_P, respectively.

3. The identified sets M_C, O_C and A_C may have zero (empty set) or more con-

Algorithm 2 Algorithm to identify M_C in B_C

Require: $B_C \neq \emptyset$ and $B_P \neq \emptyset$

Initialize M_C, M_P, N_C, N_P, O_C and $O_P = \emptyset$
for Each element B_C^i, in B_C **do**
  if $\exists$ a B_P^j in B_P that is close to scan line of B_C^i **then**
    if $B_C^i < B_P^j$ **then**
      $O_P \leftarrow O_P + B_P^j$ and $M_C \leftarrow M_C + B_C^i$
    end if
    if $B_C^i > B_P^j$ **then**
      $M_P \leftarrow M_P + B_P^j$ and $O_C \leftarrow O_C + B_C^i$
    end if
  else
    $N_C \leftarrow N_C + B_C^i$
  end if
end for
$N_P \leftarrow B_P - O_P - M_P$

tinuous segments of readings. A continuous segment is a string of consecutive readings. Usually in a laser scan, a continuous segment represents a single object. During this step continuous segments within each set are identified. For example, in Fig. 4.1, G and H are continuous segments of the reading set O_P. The segmented sets will be represented by the superscript s and it can have zero or more continuous segments. For example, $O_P^s = \{G,\ H\}$.

4.3.3 Parameter selection

The following parameters have to be carefully chosen for proper operation of the moving object detection algorithm.

The effective time interval between laser data, Δt: The data acquisition time from the laser range finder is denoted as δt, which is a constant for a given sensor and the computer. The Δt can be chosen to be $n\delta t$ (n is any positive integer), where n has to be chosen according to the minimum relative velocity

that has to be detected, as defined in (4.5).

Closest point detection threshold, Δd_c: In order to identify the stationary objects, the laser data points that are closer to each other have to be detected. The closest points can be easily defined as follows: if a point in the current scan is closer to a data point in the previous scan by a threshold Δd_c, then the points are identified as representing stationary points in their respective laser scans. However, due to the projective nature of the laser beam, the distance between two consecutive laser points at different ranges changes linearly with the range. Therefore a fixed threshold would not suffice for the detection of the closest points, as the points that are further away have greater separation than the points that are closer to the scanner. Thus a variable value for the Δd_c is chosen based on:

$$\Delta d_c = k \tan(\pi/360) r \tag{4.4}$$

where $\pi/360$ is the resolution of the laser, r is the range to the first laser point and k is a suitably chosen tuning parameter to counter the noise levels in the scanner readings. Once the stationary scan points have been identified, the laser readings have to be grouped in segments. A series of consecutive laser readings that is spaced by less than a threshold with each of it neighbors is identified as a segment. Since the same spacing properties as above are applied in selecting a threshold, a similar variable threshold is chosen for segmentation with a different tuning parameter k.

The minimum size of the moving object The objects that are further away from the scanner are represented as smaller objects (in the number of laser data

points) than the objects closer to the laser. Also the noise levels increase with range (property of the laser range finder). Therefore a fixed threshold is selected for the minimum number of laser points that is needed in a segment to label it as a valid segment (not noisy).

The separation of a moving object in the world frame between two laser scans is directly related to the magnitude of the relative velocity between the object and the robot. Based on the above parameters, the minimum detectable relative velocity of an object will be:

$$V_{\min} = \frac{\Delta d_c}{\Delta t}. \tag{4.5}$$

4.3.4 Moving Object Detection

After achieving the final segmentation, the next objective is to accurately and completely identify the moving object. Generally, the segments in M_C^s represent moving objects. However, there are instances where M_C^s either represents only a part of the moving object or not represent any moving objects ($M_C^s = \phi$), when actually there are moving objects present in the laser scans. To facilitate a development of a systematic algorithm to completely recover the moving object, the following possible case scenarios are enumerated along with their properties.

1. Case 1: (Object is perfectly separate in two scans)

   Fig. 4.2 provides an example of this case. The complete object is represented by M_C^s, and as such no further processing is required.

2. Case 2: (Object is only partially separated in two scans)

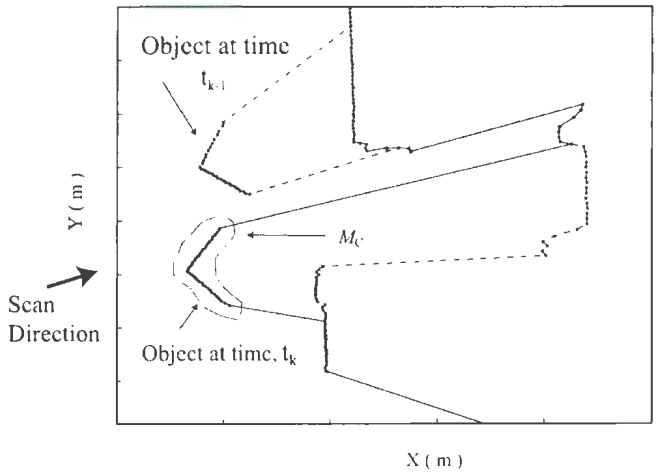


Figure 4.2: Perfectly separated object positions.

Fig. 4.3 provides an example of this case. Only part of the object is represented by M_C^s. Also in this particular case it is observed that one continuous segment in O_C^s belongs to the moving object. This is a common observation when scans are taken with a higher sampling time or when the object itself is moving slowly.

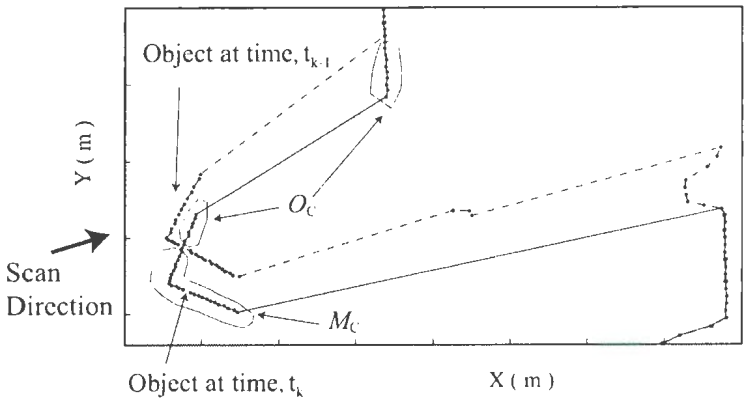


Figure 4.3: Partially separated object positions.

3. Case 3: (Object moving away from scanner)

An example of an object moving away from the scanner is provided in Fig. 4.4.

As can be seen in the figure, the moving object will be completely missing in the M_C^s but will be represented by a segment in O_C^s. Therefore it can be concluded that, if the set $M_C^s = \phi$, with $O_C^s \neq \phi$, then a segment in the O_C^s will correspond to the actual moving object. However, when $M_C^s \neq \phi$, it cannot be concluded that a moving object is completely missing from M_C^s; for example, when there is more than one moving object and only one of them moves away from the scanner. In such a case $M_C^s \neq \phi$, but there will be one missing moving object in M_C^s.

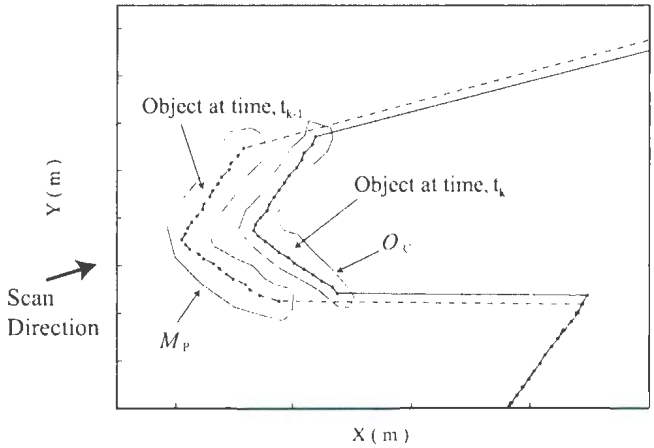


Figure 4.4: Object moving away from the scanner.

4. Case 4: (Object moving towards the scanner)

   This is the opposite of case 3 and M_C^s will represent the complete moving object. Thus, this is similar to case 1 and no further processing is required.

5. Case 5: (Lateral movement with minimum or no radial movement) In this case, M_C^s only has a partial representation of the moving object. The missing part of the moving object will belong to the continuous segment set, A_C^s.

From the above five cases it is clear that in some cases straightforward segmentation would not yield the complete moving object. In cases 2, 3 and 5 further

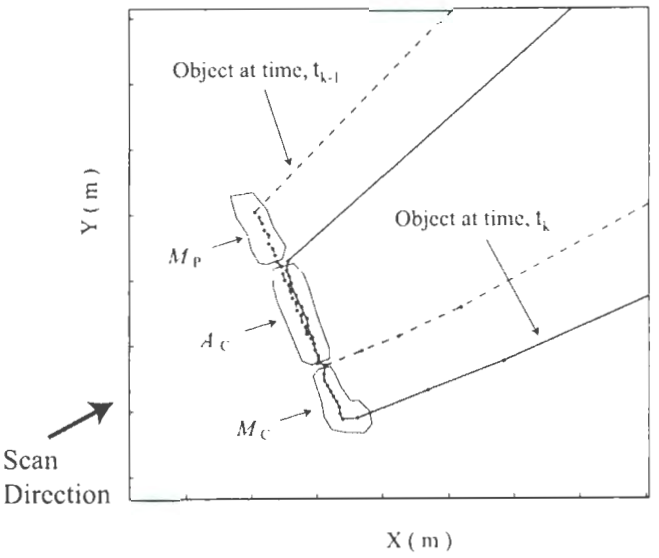


Figure 4.5: Partial laterally separated object positions.

processing is necessary to recover the complete object. It should be noted that the issues relating to false positives are relevant to all five cases. Of all the cases, the 3rd case is the most difficult to resolve, especially in the presence of false positives and/or multiple moving objects. In order to resolve the 2nd and 5th cases a set join operation is defined.

Definition: (Join of two continuous segment sets, Join(A,B)) When either end of a continuous segment of set A is adequately close to either end of a continuous segment of set B, they are joined and placed in the set A, replacing the contributing element of set A. The joined segment is deleted from the second set in order to avoid repeated join of the same segment in set B with multiple segments in set A.

The above operation can be iteratively applied until there is no reduction in the number of segments in set B. Generally, one pass could properly reconnect most of the disconnected segments. Algorithm 3 is applied to recover the complete moving objects that belong to cases 2 and 5.

Algorithm 3 Recover the complete M_C^s

1: **if** $M_C^s \neq \phi$ **then**
2: $M_C^s \leftarrow \text{Join}(M_C^s, A_C^s)$
3: $M_C^s \leftarrow \text{Join}(M_C^s, O_C^s)$
4: **end if**

The first statement connects the segments in M_C^s with the segments in A_C^s and the second connects from O_C^s. A_C^s is joined first, since in the second case there could be segments in A_C^s that represent the moving objects in crossing points between the scans of the moving objects.

Algorithm 4 can be used to recover the moving object when M_C^s is empty (in some instances of case 3). It should be noted that this method is susceptible to introducing false positives from the segments in O_C^s that correspond to stationary objects. As a rule for implementation, this algorithm should be used when only one moving object is present in the environment. This single moving object condition can be detected from the number of segments in O_C^s.

Algorithm 4 Replace M_C^s

1: **if** $M_C^s = \emptyset$ and $O_C^s \neq \emptyset$ **then**
2: $M_C^s \leftarrow O_C^s$
3: **end if**

4.3.5 Experimental Results

This section provides the results of the object detection algorithm described in this section. In each of these experiments 50 scans are acquired in approximately 10 seconds. Each laser scan is taken with a field of view of 180° at a resolution of 0.5°. The laser remained stationary during all the experiments. When laser readings are closer than 10cm to each other, they are assumed to correspond to the same object. Fig. 4.6 and 4.7 show the final results of the segmentation algorithm. As can be seen,

the algorithm shows acceptable results in recovering the complete object scenarios relevant to cases 2 and 5.

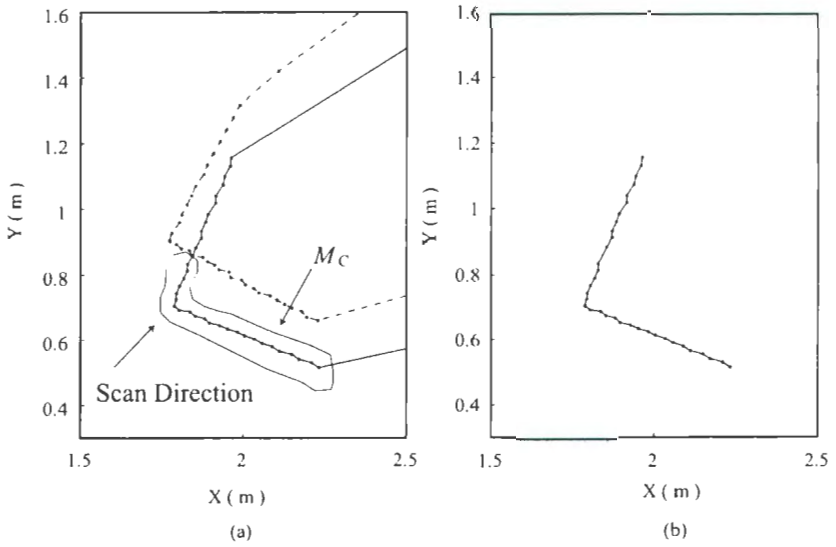


Figure 4.6: The detection of a moving object similar to case 2. (a) Two laser scans. (b) Detected moving object.

4.4 Moving Object Detection and Position Calculation

Once the laser segments are identified they have to be labeled according to the object that they represent, either moving or stationary. When the moving objects are isolated from the laser segments, the object positions (centroid of the foot print of the object) have to be calculated for the purposes such as velocity estimation.

4.4.1 Moving Object Position Calculation

The position of moving objects is estimated from all the recovered information that is available in the form of scan segments. In order to support any higher level functions

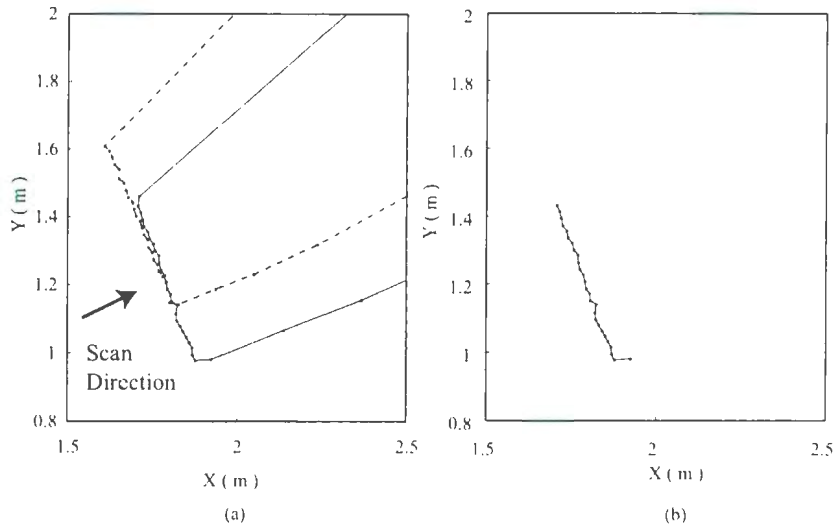


Figure 4.7: The detection of a moving object similar to case 5. (a) Two laser scans. (b) Detected moving object.

related to moving objects their position has to be accurately calculated. The most common method for object position calculation is to estimate the centroid of the footprint of the object based on the laser data, where the object position can be calculated directly using the current data corresponding to the object. As the laser range finder always observes only one side of the object at any given time, this method will only yield an approximate position estimate. If the object is observed over a long period of time or the object is actively observed, the complete object can be reconstructed using the data from scanning multiple directions.

In this work the object position is recovered by constructing the simple convex hull of the laser readings in each segment in M_C^s. Also, M_C^s might contain false positives that may appear as very short segments compared to the actual objects. Thus, the segments that are below a predefined size threshold are ignored. Threshold value must be selected with careful consideration to the nature of the moving objects in terms of their size and their distance to the scanner. Once the convex hulls of the

selected scans segments are constructed, the actual object position can be considered to be at the centroid of the convex hull. Accuracy of the object position will depend on the shape and size of the moving object. Therefore it is very difficult to quantify the absolute uncertainty of the object position from the observed data. Fig. 4.8 shows an example of a segmented object, its convex hull, and the estimated position, along with a view of the real object from the scanner.

Alternatively, the object position can be calculated using the bounding rectangle of the laser segment data. This method usually allows for greater accuracy (through overestimation of the object area) than the convex hull. Therefore in the results shown in the next section bounding rectangles are used to display the position of the object.

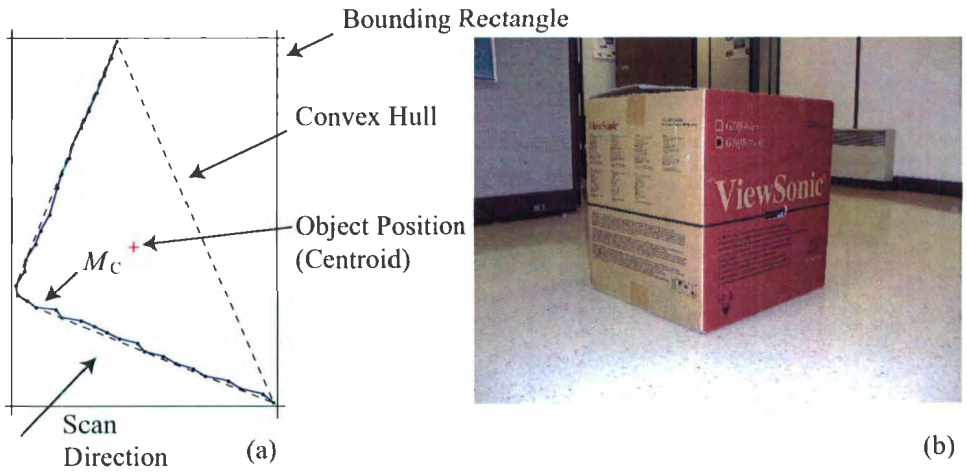


Figure 4.8: (a) The final moving object segment, its centroid of the convex hull (calculated object position) and the bounding rectangle. (b) The actual view of the object.

4.4.2 Experimental Results

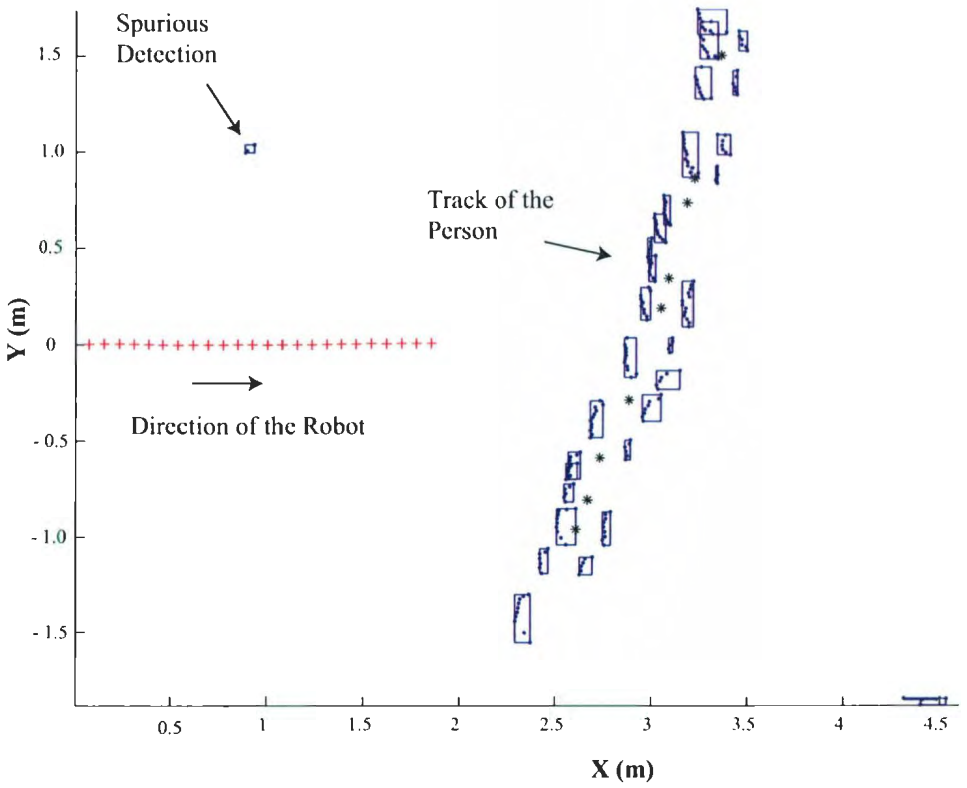
This section shows some examples of the tracking results obtained with people moving in the field-of-view of the moving robot. The laser scanning plane is located about

35cm above ground level. Thus when a person walks across the field of view only the legs are visible as two different moving objects. In this case two results are shown with a person moving across the field of view. In the first result in Figure 4.9 the person is moving close to the robot (1.5m - 2.5m) and as can be seen from the figure, the two legs are visible from time to time as each leg becomes occluded by the other in the walking gait. The data is acquired at 5Hz (Δt = 200ms) and for closest point detection a threshold of 5cm is used. The black stars in Figure 4.9 represent the possible torso position of the person when the scan segments from the two legs are available. The second result (Figure 4.10) is similar to the first result but the person is walking about 5m away from the robot. From both results it is clear that the two legs of the person are not always detected. Apart from the obvious reason of occlusion, the other main reason is that the two legs of a person move at varying velocities during the gait. Therefore, when the velocity is below the minimum detectable, the leg will be undetectable.

4.5 Conclusion

In this chapter a general moving object detection algorithm was presented. The algorithm uses some specific properties of the laser scan data corresponding to moving objects to successfully detect them. The proposed algorithm can be used to detect multiple moving objects from a moving platform in a dynamic environment. Additionally, in comparison to other methods, the proposed algorithm has the ability to recover the complete moving object when the object is moving at a low relative velocity and when the object is moving sideways with respect to the scan direction. Through parameter tuning the detectable minimum relative velocity can be adjusted to suit the application. The results shows that the proposed algorithm can be used

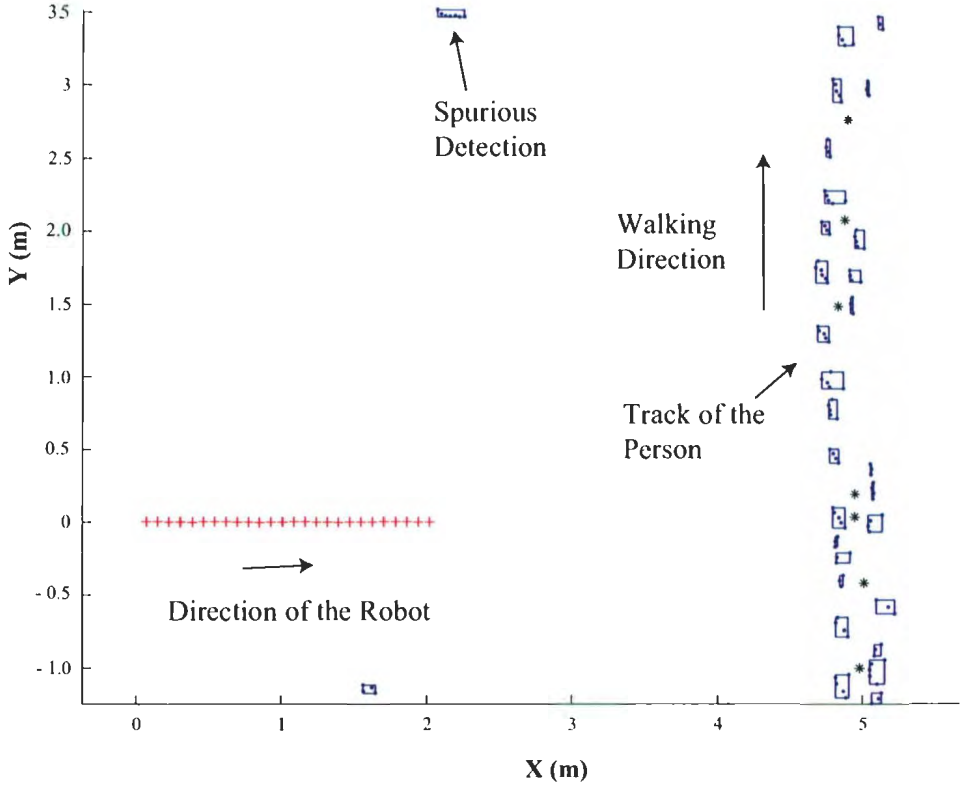
(a)



(b)

Figure 4.9: Track of the walking person at a distance about 1.5m - 2.5m from the robot. The blue bounding boxes represent the detected moving objects while the black stars represent the possible torso position of the person when the scan segments from the two legs are available.

(a)



(b)

Figure 4.10: The track of a person walking approximately 5m from the robot. The blue bounding boxes represent the detected moving objects while the black stars represent the possible torso position of the person when the scan segments from the two legs are available.

to successfully track generic objects such as legs of a human in structured indoor environment. In regular SLAM implementations the environment (or the landmarks) is assumed to be stationary. Therefore, apart from the direct use of moving object detection and tracking, the proposed method can be used as a data preprocessing step in regular SLAM applications to remove the data related to the moving objects from the sensor data. This type of preprocessing will aid in improving the stability of the SLAM filters by preventing any possible moving landmarks from corrupting the data structures.

Chapter 5

Active Control and Robot Parking

Almost all localization and mapping techniques that have been proposed to date have a resulting level of uncertainty for robot pose and map estimation. However, at the end of a navigation task, the robot might need to be precisely positioned to carry out an assigned task, such as pick and place or park. This chapter details a visual servoing control strategy to overcome the limitations in the positioning accuracy considering the nonholonomic nature of the robot and the field of view limitations of the robot.

5.1 Introduction

Automated parking systems have been an important issue in robotic research [114, 115, 116]. Recently automated parking systems have been developed to automate large scale parking garages. Although they are mostly pallet placing for space efficiency, more autonomous parking methods are rapidly a growing concern for many users. Particularly when robots (automated vehicles) are operated autonomously, parking will become an important issue for a variety of reasons: to replenish energy supply (battery charging, refueling), precise alignment against a given target for ma-

terial loading, or parking in idle for a new task assignment. In the majority of cases the robot is required to align precisely with a predefined spatial configuration (target pose).

In the past the automated parking problem has been addressed using a variety of techniques. Much of the initial research was related to design of low level robot controllers, where the control objective was to move the robot autonomously along a preplanned trajectory [117]. Such controllers require the robot to be supplied with precise path information (or a trajectory). Furthermore, the system requires accurate feedback information of the robot position for the controller to estimate its tracking error. Estimation, or acquiring feedback position information, or precise robot localization using odometry data, is a challenging task [118, 8]. Recently research has focused on developing more reactive parking techniques using exteroceptive sensors such as computer vision [28, 29, 30, 31, 32, 26, 27]. These vision based methods use image plane measurements to align the robot with a given reference configuration. The reference configuration is either defined in a world coordinate system relative to the parking station or in an image plane. The robot uses feedback control strategies to achieve the control objectives. Some key requirements of an automated parking system will be: (1) the robot must be able to begin its parking behaviour from any position given its sensors have the capability to recognize the parking station; (2) the parking system must be able to park the robot in a unique pose while providing the required level of accuracy for the application; (3) the robot must maintain the parking station in its field of view during the whole parking process; and (4) the parking system should be able to overcome the nonholonomic nature of the robot.

5.1.1 Related Work

The problem of vision based parking belongs to the general research area of nonholonomic visual servoing [23, 24]. In visual servoing the feedback control loop is driven using the response measurements observed using a camera. The feedback system is based on either using direct image-based (**IBVS**) measurements or position based visual servoing (**PBVS**) [23]. In PBVS, image measurements are used to calculate the position error of the robot pose in a global coordinate system. This method requires transformation of image measurements calculated in pixel units into pose estimation in distance units. The estimated pose error will be used in the feedback control law. Generally, pose estimation is a three-dimensional image interpretation scheme, and errors accumulated in the estimation processes can lead to erroneous pose estimates [23]. In IBVS the error is measured in image plane (in pixels) and is directly used in the error driven control law. Thus, in visual servoing IBVS is preferred over PBVS since it avoids any errors that can be introduced during the position estimation. An inherent drawback in IBVS is that all image features are required to be maintained within the field of view of the camera throughout the control process unless there is redundancy built into the feature set.

Differentially driven mobile robots have nonholonomic constraints [24, 54]; i.e. a robot cannot move sideways. Further, the limited field of view in vision systems generally imposes an additional constraint on the control law. Thus, image based visual servoing of mobile robots is a challenging task under the limited number of degrees of freedom (usually two) and the limited field-of-view available in the vision system. For nonholonomic robots visual servoing can be applied for path following, or it can be used to align the robot with a given pose (parking). Continuous ground curves (a line on the ground or other complex paths like roads) are the most commonly

selected feature type in robot path following [25, 55, 56] where robots use them in the environment to continuously align themselves. In parking techniques, the robot is aligned with a fixed set of features, so that the robot will satisfy a predefined control objective [28, 31]. Robot parking controllers generally belong to either conventional continuous controllers or hybrid controllers.

With conventional controllers [26, 27] the robot is aligned to a set of features seen by the camera using smooth control of robot velocity. The typical issues in the continuous controllers include: the convergence of the solutions when starting from an arbitrary robot pose [26], and inability to obtain a unique final position [27]. These problems arise due to the nonholonomic nature of the robots and the limited field of view of the camera. Some improvements have been reported in vision-based parking controllers using intelligent control techniques such as fuzzy logic and neural networks [33, 34, 35, 36]. Generally, fuzzy logic and neural network based controllers perform satisfactorily, but they do not guarantee convergence. Also, each time there is a change in the appearance of the parking station, the controller has to be manually trained to accommodate the new appearance.

In contrast, hybrid controllers allow the robot velocities to be controlled in a discontinuous manner. In hybrid controllers, a finite state machine is used to define a set of states to reflect multiple operational contexts in a robotic task. Each state can be equipped with its own control algorithm. Hence, multiple switching control algorithms give rise to discontinuous control of the robot velocities. Most vision based hybrid controllers [28, 29, 30, 31, 32] use this property to overcome nonholonomic and field of view constraints [54] of the system. Therefore in this research hybrid control strategy was chosen in the design of a new parking controller. Lyapunov techniques [57] have been widely adopted in hybrid closed-loop parking controllers [28, 29, 30, 32]. Limitations of the past hybrid methods include: the rapid switching

behaviour around the parking position (zeno behaviour) [28, 31]; partial utilization of the available field of view of the camera [32, 31]; and not explicitly addressing the field of view constraints of the camera [29, 30]. In other related works on vision based robot parking, [119] presents an optical flow based robot docking controller while [120] presents a controller based on the potential fields. The method presented in [119] is only capable of parking perpendicular to a vertical surface and therefore has limited applicability in precision parking applications.

5.1.2 Motivation

With an IBVS based control law, that is dependent only on the image-plane measurements (in pixels), the user will have the ability to provide image-plane templates (reference images) for achieving required parking behavior. As the control strategy employs a finite state automaton, the system can be extended to facilitate many other servoing tasks such as behavior based integrated navigation systems [17].

In order to perform efficient image based parking, a novel vision based, locally convergent control system is developed. The hybrid control solution successfully overcomes the nonholonomic and field of view constraints of the robot and camera, respectively. Additionally, the controller maximally utilizes the available field of view of the camera. Experimental results demonstrate the convergence of the robot to the parking position without any oscillations. Further, the parking controller shows a high accuracy in the parking position as demonstrated by the repeatability tests.

Section 5.2 provides the preliminaries of the parking system and image-plane measurements. In section 5.3 the proposed control strategy is presented. Section 5.4 analyzes the convergence properties of the proposed system. Section 5.5 provides the details of the experimental implementation and the results.

5.2 Visual Servoing System

5.2.1 Parking Station

The objective of the parking system is to move the robot, so that the current view of the parking station accurately aligns with the reference parking station. An image pattern having three distinct features is used as the parking station. A minimum of three features is required to define a unique robot parking position in front of the parking station. As shown in Fig. 5.1(a) the three features are horizontally aligned and equally spaced. The reference image taken at the parking station has features at positions [u_1^r, u_2^r, u_3^r], measured from the left edge of the image, and A^r is the overall width of the parking station in image plane. A centered configuration is chosen to allow for better utilization of the field of view of the camera. Fig. 5.1(b) shows a typical parking station configuration as seen by the camera. The features are extracted using a series of image processing techniques and the error measurements are evaluated by comparing the features against the reference image. It should be noted that A, f_1 and f_2 are scalar quantities and are always positive. The values of e_i's have a positive sign in the direction shown in Fig. 5.1(b). E_A is defined as the difference between the current overall width of the parking station (A) and the required (reference) width (A^r):

$$E_A = A - A^r.$$

Parking condition The robot is considered 'parked' if the following condition holds true.

$$(|E_A| < \delta_A \text{ AND } |e_1| < \delta_u \text{ AND } |e_2| < \delta_u \text{ AND } |e_3| < \delta_u) \tag{5.1}$$

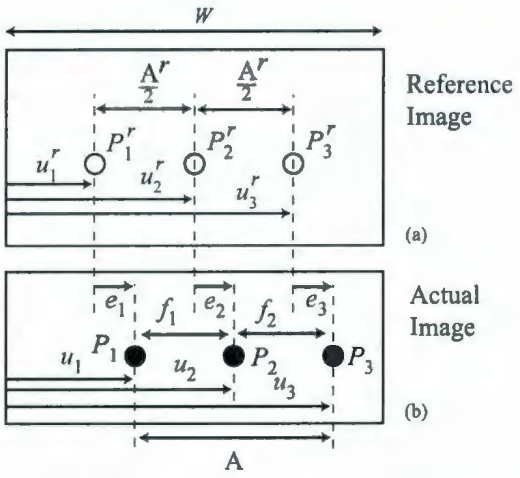


Figure 5.1: Reference and actual image of the parking station

where δ_A and δ_u are threshold values (in pixels) which can be adjusted to achieve an acceptable accuracy of the parking condition. This parking condition corresponds to a region around the desired final robot pose. The area of this region (and hence the accuracy of parking) can be adjusted using δ_u and δ_A to suit the application.

5.2.2 Robot and camera model

The robot pose is defined as $\mathbf{x} = [x,\ y,\ \theta]^T$. Further, a differentially driven robot model is selected with the heading velocity v_k and rotational velocity ω_k as its control inputs (Fig. 5.2).

The robot model can be mathematically expressed as

$$\begin{aligned} \dot{x} &= v\cos(\phi) \\ \dot{y} &= v\sin(\phi) \\ \dot{\phi} &= \omega. \end{aligned} \tag{5.2}$$

Further, the feature positions in the image plane can be derived assuming a pinhole camera model.

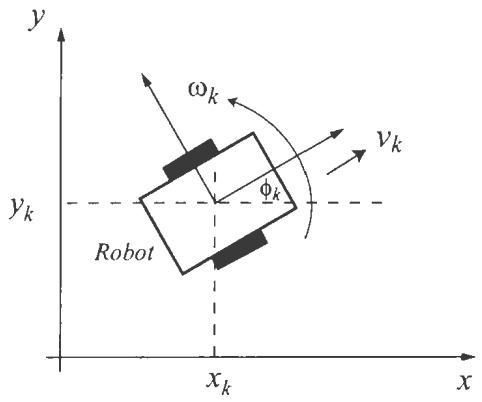


Figure 5.2: Differentially driven robot model.

$$\begin{aligned} u_1 &= \lambda p \tan(\gamma - \beta_2) + \tfrac{W}{2} \\ u_2 &= \lambda p \tan(\gamma) + \tfrac{W}{2} \\ u_3 &= \lambda p \tan(\gamma + \beta_1) + \tfrac{W}{2} \end{aligned} \tag{5.3}$$

where λ is the focal length, p is the number of pixels per meter and W is the width of the image in pixels. Also γ, β_1, and β_2 are as defined in Fig. 5.9. The principal point of the camera is assumed to be at the rotational axis of the robot and the optical axis of the camera is assumed to be parallel to the heading velocity of the robot.

5.2.3 Overall Vision–Based Control System

As shown in Fig. 5.3, the overall objective of the vision based control system is to park the robot at a desired position in front of the parking station. Fig. 5.4 shows the overall block diagram of the closed loop control system. The hybrid controller uses the image plane measurements resulting from image processing to produce the control command of the robot (heading and rotational velocities). The markers on the parking station are identified and segmented using a set of image processing and analyzing steps. Then, the horizontal centroid position of each marker is measured

from the segmented blobs. Finally, the hybrid controller uses the measured image plane values to generate the final robot commands.

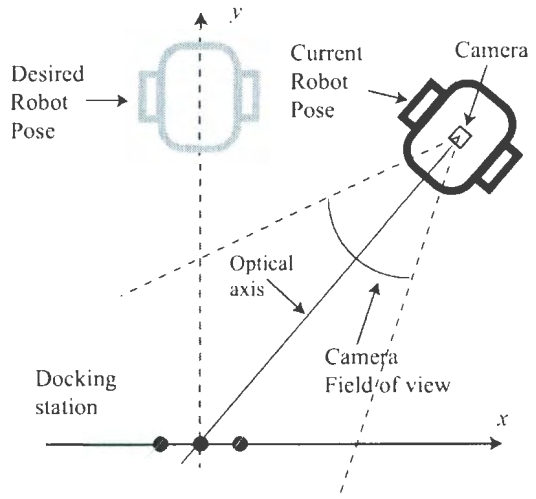


Figure 5.3: The control objective and the introduction of the axis.

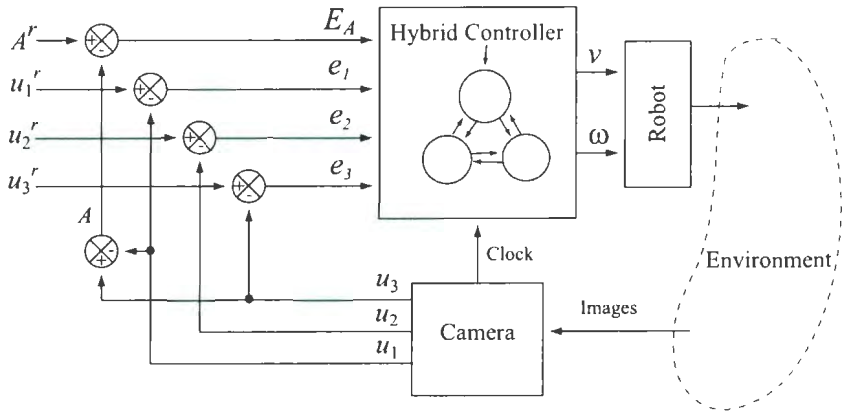


Figure 5.4: Overall block diagram of the control system.

5.2.4 Assumptions

The following assumptions are used to enable the robot to observe the parking station in a compatible manner with the camera model used in the definition of the controller.

It is important to note that these assumptions are general requirements. As such, apart from approximate alignments, no additional effort is taken to satisfy them in the implementation described in Section 5.5.

1. Features are at approximately the same height as the horizontal optical axis of the camera.

2. The robot operates on a flat floor.

3. The camera is fixed and the optical axis is approximately parallel to the heading velocity.

4. At the start, the parking station is within the field-of-view of the camera.

5.3 Control Strategy

The control strategy proposed in this section is mainly motivated by behavior-based robotics [17]. Behavior based robotics provide many biologically inspired intelligent control techniques for mobile robot navigation. Following the principles of behavior-based robotics, the proposed method provides close coupling between sensory information and motor control using simple mathematical relationships. Specifically, a finite state machine (**FSM**) is used at the heart of the parking control system to provide the context (state) of operation (relationship between sensory information and motor speed) based on the current sensory information and the progress of the parking process. Thus, a state in the discrete part of the hybrid system encapsulates a particular continuous control scenario.

5.3.1 Finite State Machine

The FSM proposed for the parking system is based on the intuitive forward and reverse motion that is observed during regular parking operations. The FSM has four states and five transitions. Three states represent active controllers, while the other state represents the termination state. The FSM can be mathematically represented as:

$$\text{FSM} = (X,\ \Sigma\ ,\alpha\ ,x_0\ ,X_m)\,, \tag{5.4}$$

where X denotes the state set; Σ denotes the finite event set; $\alpha : \Sigma \times X \rightarrow X$ denotes the state transition functions; $x_0 \in X$ denotes the initial state; and $X_m \subseteq X$ denotes the terminating (or marked) states set. Fig. 5.5 shows the FSM designed to solve the parking problem and the complete FSM can be represented as follows.

$$\begin{aligned}
X &= \{S_t,\ F,\ R,\ S_p\} \\
\Sigma &= \{\varepsilon_i \mid i = 1 \ldots 5\} \\
\alpha &= \{(S_t, \varepsilon_1) \rightarrow F,\ (S_t, \varepsilon_2) \rightarrow R,\ (F, \varepsilon_4) \rightarrow S_t, \\
&\quad\ (R, \varepsilon_5) \rightarrow F,\ (R, \varepsilon_3) \rightarrow S_p\} \\
&= \{\alpha_i \mid i = 1 \ldots 5\} \\
x_0 &= \{S_t\} \\
X_m &= \{S_p\}.
\end{aligned}$$

When the robot controller enters a particular state it will first execute the initialization functions. During initialization the controller will initialize the data structure related to the current state and will make any required additional measurements.

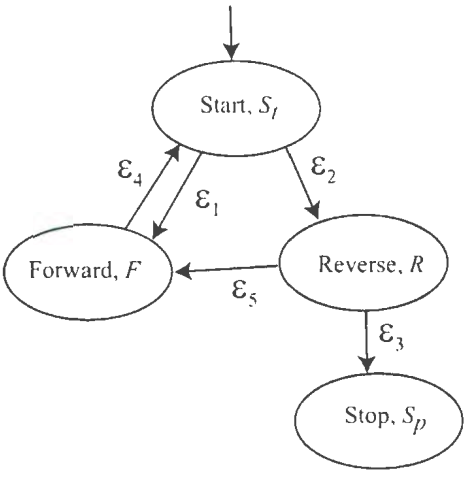


Figure 5.5: Proposed finite state machine

Following initialization, the robot will execute the core functionality of the state in a loop structure which is governed by the motor control algorithm. Once the exit conditions related to the state are satisfied, it will terminate the state by first executing the termination functions, and then flagging the transition event, ε_i. Termination functions usually include the memory cleanup operations and measurement updates to facilitate the proper operation of the next state.

5.3.2 Details of the States

This section provides the operational details of each state. It should be noted that all k_is are suitably chosen control parameters.

Start (S_I) State

In the start state, the robot will be oriented so that the feature in the center (P_2) of the parking station aligns with the center feature in the reference image (P_2^r). Then it will switch the control of the robot to either F or R state depending on the relative size of the parking station in the current image.

Algorithm 5 `start()`

1: init: none
2: during:
3: **while** $|e_2| > \delta_u$ **do**
4: $v = 0$;
5: $\omega = -k_1 e_2$; `// proportional controller`
6: **end while**
7: **if** $E_A < 0$ **then**
8: exit event = e_1;
9: goto exit:
10: **else**
11: exit event = e_2;
12: goto exit:
13: **end if**
14: exit:
15: $f_1 = u_2 - u_1$;
16: $f_2 = u_3 - u_2$;

Forward (F) State

When the controller is in this state, depending on the sign of the value $(f_1 - f_2)$, it will align a side feature a distance of c pixels from the edge of the image while moving towards the parking station (Fig. 5.6). The distance c (> 0) defines a virtual edge of the image and it will ensure that the features of the parking station always remain within the field of view of the camera. The objective of the angular velocity control law is to force the error, e, to zero, as shown in Fig. 5.6. The robot will exit this state when the uncontrolled side feature is less than c pixels away from the other edge.

It is always desirable to have a larger turn of the robot (fast transient response) when the robot starts from a position away from the y-axis. However, when the value of c is relatively high the turn angle of the robot is constrained to a lower value. Proof of this property is shown in section 5.4. Under those circumstances the value of c needs to be lower when the robot will have the ability to turn in a large angle while maintaining the parking station in the field of view of the image. As shown

Algorithm 6 `forward()`

```
1: init: none
2: during:
3: if f_1 < f_2 then
4:    // Fig. 5(a)
5:    while W − u_3 > c do
6:       v = v_c;
7:       ω = k_2 e; // proportional controller
8:    end while
9: else
10:   // Fig. 5(b)
11:   while u_1 > c do
12:      v = v_c;
13:      ω = −k_2 e; // proportional controller
14:   end while
15: end if
16: exit event = e_4;
17: exit: none
```

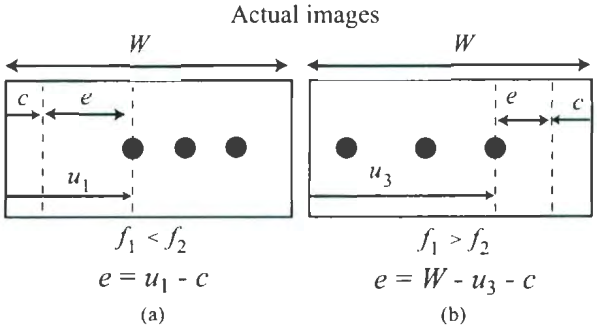


Figure 5.6: The control scenarios in *forward* state. The corresponding control error is labeled as c. (a) If the robot starts with $x > 0$. (b) If the robot starts with $x < 0$. (the global x-y reference frame is defined in Fig. 5.7)

in Fig. 5.7 the robot will take path Q_3 when the value of c is lower and Q_4 when c is higher. Clearly it is advantageous to have a path similar to Q_3 to obtain faster convergence. However, faster turning (convergence) leads to overshoot and oscillation near the y-axis as shown in path Q_1 of Fig. 5.7, which can be avoided by using a larger c value to obtain a path similar to Q_2. In order to preserve both these properties the parameter c is adaptively changed as shown in Fig. 5.8.

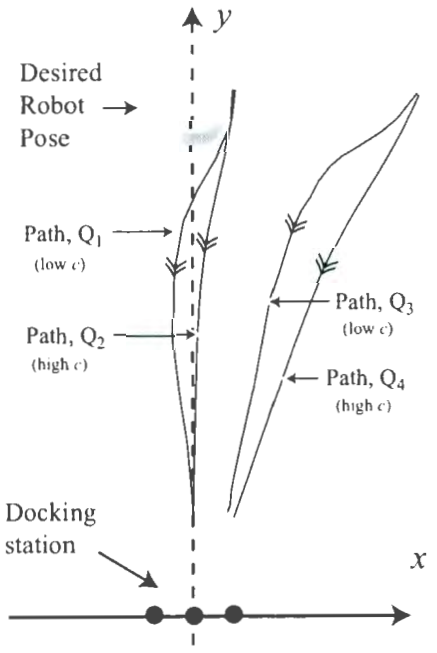


Figure 5.7: The effect of oversteering in the forward state on the final position. Q_2 and Q_3 are desirable robot paths while Q_1 and Q_4 are undesirable.

When the robot is on the y-axis with the center feature aligned, $f_1 - f_2 = 0$. When the robot starting position moves away from the y-axis $|f_1 - f_2|$ will increase. This observation can be used to adaptively change the value of c. In order to calculate the value of c at the start of each forward motion, the proportional relationship $f_{12} = k|f_1 - f_2|$ is defined. In Fig. 5.8 the maximum for f_{12} can be observed by placing the robot at extreme angles that are possible in a given scenario. The maximum for c,

c_{max}, is set using the maximum allowed value of $(W - A^r)/2$. Further, the c_{max} value can be tuned by trial runs of the forward move so that the robot efficiently converges to the parking position without any overshoot from starting positions that are closer to the y-axis. The minimum for c, c_{min} has to be adjusted so that the robot does not bump into the parking station from starting positions away from the y-axis.

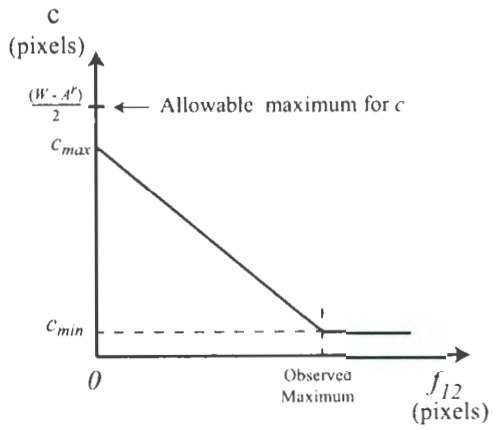


Figure 5.8: The relationship between c and $|f_{12}|$. The allowable maximum for c is defined assuming that the parking station appears symmetric in the images when the robot is at the exact parking condition ($\delta_u = 0$).

Reverse (R) State

During the reverse state the robot will move away from the parking station while aligning the center feature (P_2) with the corresponding feature in the reference frame (P_2^r). The reverse state is essential to move the parking station sufficiently inside the image to facilitate the next forward move. Additionally, while the robot is in the reverse state, it will monitor the parking condition. The robot will exit the reverse state when the robot is parked or when the overall size of the parking configuration is less than a predefined value, A_{min}.

Algorithm 7 `reverse()`

1: init: none
2: during:
3: **while** (**NOT** $A < A_{min}$) **OR** `parked` **do**
4: $v = -v_r$;
5: $\omega = -k_1 e_2$; `// proportional controller`
6: **end while**
7: **if** $A < A_{min}$ **AND** (**NOT** `parked`) **then**
8: exit event = ϵ_5;
9: **else if** `parked` **then**
10: exit event = ϵ_3;
11: **end if**
12: exit: none

Stop (S_p) State

The robot will come to a halt after a successful parking at the stop state.

Algorithm 8 `stop()`

1: init: none
2: during: none
3: exit:
4: $v = 0$
5: $\omega = 0$

5.4 Analysis of Convergence

In this section the convergence properties of the controller are evaluated. The proof of convergence is provided in two stages. In the first step it is shown that the robot converges to a region (R_t in Fig. 5.9) about the final position. At the beginning of this process, if the robot starts outside the region, it will move into the region and if it starts within the region it will remain in the same region. In the second stage it is shown that through the modulation of the controller parameter c, the region R_t is minimized until the final thresholds are satisfied.

5.4.1 Proof of convergence

This section establishes the convergence of the controller. Before analyzing the convergence of the controller, the general operational characteristics and some auxiliary properties are presented. Depending on the initial error in overall size (E_A) of the parking station, the robot either starts moving forward or reverses. Initially, the start state will align the robot with the center feature ($e_2 = 0$) in the parking station. Then during the exit from the start state it will measure the f_1 and f_2 values, which are then used to adjust the c value. If $E_A < 0$ the robot will start to move forward, with the value of c calculated at the exit of the start state. When the robot exits from the forward state it will switch to the start state and align the center feature. After measuring the new values of f_1 and f_2 and calculating the new c, the robot will switch into reverse state ($E_A > 0$). During the reverse motion the controller will attempt to keep the center feature of the parking station properly aligned ($e_2 = 0$) while traveling in a radial straight line. When the parking station appears to be sufficiently small the robot will switch to the forward state. At the forward state the robot will move forward with the latest c value. Additionally, during the reverse motion the robot will evaluate the parking condition. This cycle would continue until the robot converges to the defined position.

As described in section 5.3.2, if the robot starts to move forward from a starting position very close to y-axis with a low c value, it would steer into the other side of y-axis. In order to capture this behavior of the robot with a low c value, the following definition is introduced.

Definition (Region R_t, *see* Fig. 5.9) If the robot (with a fixed c ($< c_{\max}$) value) starts to move forward from a point in the region R_t, it will terminate the forward move at a point inside the region R_t and the subsequent reverse motion will also

terminate at a point inside the region R_t. The region R_t is defined by the value $d\theta$.

While the robot is maneuvering through the forward and reverse cycles, it should stop at the extreme ends to switch between states. The following two properties establish the fact that conditions set in the forward and reverse states guarantee the state switches.

Property 1 The robot would not reverse beyond the curve C_1. Let r be the straight line distance from the center feature (P_2) to the camera when the center feature is properly aligned ($e_2 = 0$). Then assuming the pinhole camera model, at any given position

$$r = \frac{A^r r^r}{A}$$

where A^r and r^r are the reference values of A and r respectively. Also, the apparent size of the parking station is always greater than a predefined minimum A_{min}.

$$A > A_{min}$$

also when $A = A_{min}$, r gains its maximum value, r_{max}. Thus,

$$r_{max} > r \text{which defines} C_1. \tag{5.5}$$

Property 2 The robot would not travel over the boundary C_2 as the apparent size of the parking station would become larger than allowed in a given forward move. The Same arguments as in the case of Property 1 can be used with the

fact $A < W - 2c$ to show that r has a bounded minimum, r_{min} which defines C_2.

The region of attraction of the controller is defined by four boundaries, C_1, C_2, C_3 and C_4. Property 1 and 2 establish boundaries C_1 and C_2. The parking station will appear to be smaller at the extreme values of θ. Therefore, the boundaries C_3 and C_4 (symmetrically placed about the y-axis) are defined by the ability of the vision system to robustly identify the parking station.

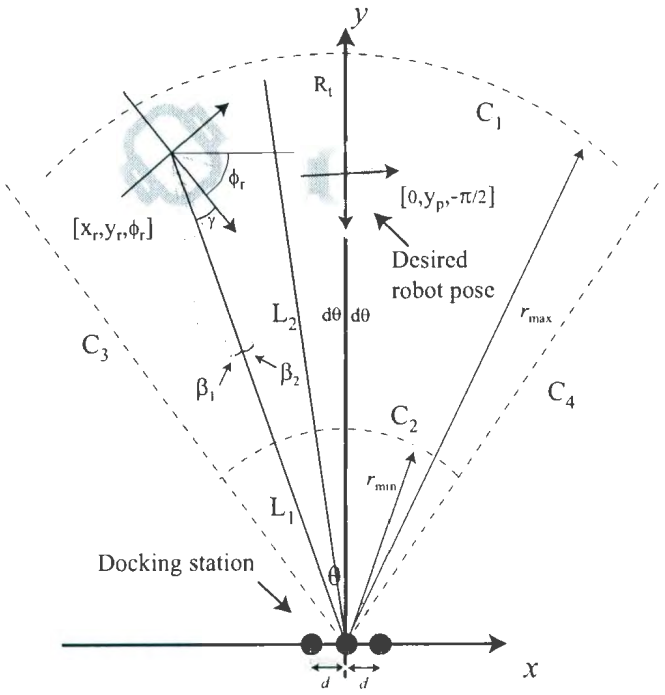


Figure 5.9: Typical robot position and oscillation envelope about the y-axis.

Theorem 5.4.1 *When the robot is outside the region R_t (in Fig. 5.9), at the end of every forward move, the robot comes onto a radial line representing a lower θ value than at the beginning of the forward move. i.e.* $|\theta(k)| > |\theta(k+1)|$

Proof Let's assume that the robot starts from the left side of the y-axis. At the beginning of the start motion the center feature (P_2), is aligned with the feature P_2^r. Also by definition of the controller, the robot will move forward only if

$$A^r > A.$$

Thus,

$$\frac{W-A}{2} > \underbrace{\frac{W-A^r}{2}}_{=c_{\max}}.$$

Using the fact that by the definition $c_{\max} > c$ at all times,

$$W - 2c > A. \tag{5.6}$$

In this case $f_1 > f_2$ and $A = f_1 + f_2$, therefore $f_2 < \frac{A}{2}$, which can be also written as, $f_2 = \frac{A}{2} + \varsigma$ where $\varsigma > 0$. From Fig. 5.1,

$$u_3 = \frac{W}{2} + f_2.$$

Using $f_2 = \frac{A}{2} + \varsigma$,

$$u_3 = \frac{W+A}{2} - \varsigma.$$

$$A = 2u_3 - W + 2\varsigma. \tag{5.7}$$

Using (5.6) and (5.7) it can be shown that

$$e = W - u_3 - c > \varsigma.$$

From the definition of the forward state, when the robot starts from the left hand

side of the y-axis,

$$\omega = k_2 e.$$

Thus, with a positive k_2, at the start of the forward move $\omega > 0$, i.e. the robot always moves to the right side of line L_1 in Fig. 5.9. To complete the proof one has to show that when the robot starts moving forward from the right side of line L_1, it never moves over the line to the other side before it stops moving forward. While the control is in the forward state, the most right feature P_3 is maintained at a distance c from the right edge.

$$u_3 = W - c$$

$$\lambda \tan(\gamma + \beta_1) + \frac{W}{2} = W - c$$

$$\gamma + \beta_1 = \arctan\left(\frac{W}{2p\lambda} - \frac{c}{p\lambda}\right) = \text{constant}. \tag{5.8}$$

Thus, during the forward motion the value of $\gamma + \beta_1$ is required to be maintained at a constant value. When the controller attains the desired value for u_3 (i.e. $W - c$), γ reaches its maximum value. As the robot approaches the parking station β_1 increases and thus γ decreases to maintain the condition specified in (5.8).

If the robot is to cross the line L_1 it should have a negative γ value and the robot should be able to attain this value early in the forward travel. However, from (5.8) and the preceding explanation this is clearly not true. Hence, at the end of each forward move the robot decreases the value of θ. ∎

As shown in Theorem 4.1, when the robot reaches the end of the forward move it will be at a position with a lower θ value than at the end of the previous forward move. At the end of the forward travel, the robot will switch back to start state, align

the center feature, and measure the values f_1 and f_2 in order to calculate c using (as defined in section 5.3.2),

$$c = -k_s k|f_1 - f_2| + c_{\max}. \tag{5.9}$$

Using the pinhole camera model, (5.9) can be expanded to

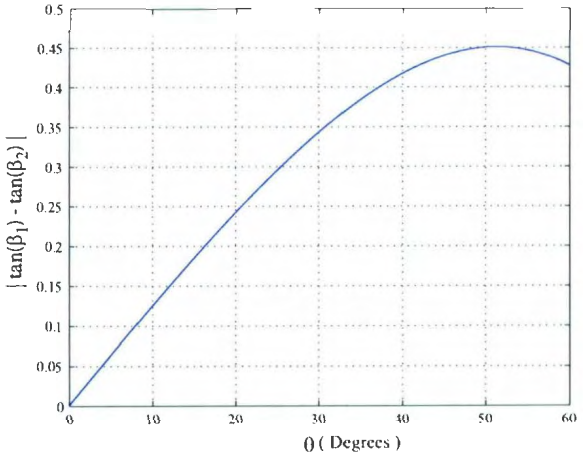
$$c = -k_s k\lambda|\tan(\beta_1) - \tan(\beta_2)| + c_{\max}.$$

At the end of the forward motion the two side features are approximately at the same distance from the edges of the image. Hence it can be assumed that the curve C_2 is an approximate semicircle with the middle feature as its center and with a radius of $r_{\min}$. When the center feature is aligned ($e_2 = 0$), it can be easily shown that

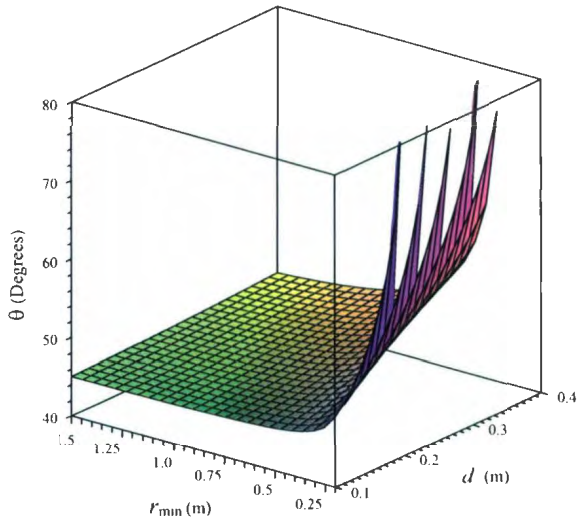
$$|\tan(\beta_1) - \tan(\beta_2)| = \frac{2d^2 \sin(\theta)\cos(\theta)}{d^2 \sin^2(\theta) - r_{\min}^2} \tag{5.10}$$

where d is the separation of the features in the parking station in meters and $r_{\min}$ is the minimum approach distance as shown in Fig. 5.9.

The typical behavior of the $|\tan(\beta_1) - \tan(\beta_2)|$ around $0 \leq \theta < 60^o$ is shown in Fig. 5.10(a) $d = 0.12$m and $r_{\min} = 0.2$m. As it is shown in Fig. 5.10(a), function $|\tan(\beta_1) - \tan(\beta_2)|$ behaves as a monotonically increasing function until about 45^o. Additionally, Fig. 5.10(b) shows the variation of the maximum of the $|\tan(\beta_1) - \tan(\beta_2)|$ with respect to d and $r_{\min}$ and it can be seen that for all practical values of d and $r_{\min}$, the maximum of θ has a minimum of around 45^o. Usually the region R_t is confined to a maximum θ of approximately 15^o. Hence, it can be concluded that the function behaves as a monotonically increasing function within practical limits d and $r_{\min}$.



(a)



(b)

Figure 5.10: Properties of $|\tan(\beta_1) - \tan(\beta_2)|$. (a) The behavior of $|\tan(\beta_1) - \tan(\beta_2)|$ with $d = 0.12$m and $r_{\min} = 0.2$m and (b) the plot of the maximum of θ for the practical limits of d and $r_{\min}$.

Using (5.10), (5.9) can be rewritten in the form

$$c = -k_s k\lambda \left| \frac{2d^2 \sin(\theta)\cos(\theta)}{d^2 \sin^2(\theta) - r_{\min}^2} \right| + c_{\max}. \tag{5.11}$$

Using the fact that the first term in the right hand side of (5.11) is a monotonically increasing function in all practical limits, it can be shown that

$$\lim_{\theta \to 0} k_s k\lambda \left| \frac{2d^2 \sin(\theta)\cos(\theta)}{d^2 \sin^2(\theta) - r_{\min}^2} \right| = 0$$

Therefore,

$$\lim_{\theta \to 0} c = c_{\max}.$$

Theorem 5.4.2 *As the value of c increases from its initial value to the maximum, the area of the region R_t will shrink to zero.*

$$\lim_{c \to c_{max}} d\theta = 0$$

Proof Let's assume that the robot starts from the right side of the y-axis. From Fig. 5.8,

$$c_{\max} - c = k_s f_{12}.$$

Also from the error calculation in the control law (Fig. 5.6),

$$e = k_1(u_1 - c).$$

$$e = -k_1(c_{\max} - u_1) - k_1 k_s f_{12} \tag{5.12}$$

The control objective at the forward move is to align the left feature (P_1), to a position of distance c from the edge (i.e. $u_1 \to c$). Thus, when the control objective is satisfied

(5.12) will become

$$e = -k_1(c_{\max} - c) - k_1 k_s f_{12}. \tag{5.13}$$

Also from the pin hole camera model,

$$f_1 - f_2 = \lambda(\tan(\beta_1) - \tan(\beta_2)).$$

$$f_{12} = k\lambda|\tan(\beta_1) - \tan(\beta_2)|$$

In the region R_t (i.e. close to y axis), the difference in the values of β_1 and β_2 will become very small. Therefore, for all practical purposes f_{12} can be assumed to be zero when the robot is in the region R_t. Thus, (5.13) will become

$$e \approx -k_1(c_{\max} - c). \tag{5.14}$$

At the start of the forward motion the steering angle is proportional to the the error, e. Thus, the steering angle of the robot in the forward motion is proportional to $c - c_{\max}$. From the definition of the region R_t, the area of the region R_t $(d\theta)$ is proportional to the steering angle. Thus,

$$d\theta \propto (c_{\max} - c)$$

and

$$\lim_{c \to c_{\max}} d\theta = 0.$$

∎

Finally, using the Theorem 5.4.1 and the definition of R_t it can be shown that the robot will converge into the region R_t and in Theorem 5.4.2 it is shown that the region

R_t shrinks (decreasing $d\theta$) with the increasing c value. Hence, the robot converges to the final parking position if it starts with the parking station in the field of view of the camera. As previously discussed in this section the robot looses its ability to turn towards the y-axis when c is set to $c_{\max}$. Therefore in practice it is necessary to set the maximum value of the c just below the theoretical maximum of $c_{\max}$. This will facilitate a small steering angle when the robot is starting from a position close to the y-axis. When $d\theta$ is sufficiently small, during the reverse motion the robot will find its parking position, given sufficiently large values for δu_i and δA.

5.5 Experiments and Results

The proposed method was implemented using a Pioneer 3AT mobile robot. Image processing routines and the finite state machine were implemented in the onboard computer (Pentium III, 860MHz). The parking system uses a Basler A102fc camera that is fixed to the robot frame. The camera acquires images at approximately 19 frames per second.

5.5.1 Image processing

Images captured from the onboard camera are used to detect the parking station. Intel OpenCV Image processing libraries are used for image processing and analysis[1]. The acquired images are thresholded to isolate the features of the parking station that appear darker than the background. Then all the small spurious image blobs are filtered out, using a threshold value based on the area of each blob. The center area of the remaining features (of the parking station) is calculated and used in the parking controller as the feature position in the image plane.

[1] http://www.intel.com/technology/computing/opencv/index.htm

5.5.2 Tuning

The desired accuracies and parking times can be achieved through the tuning of the system parameters. The tunable system parameters are listed in Table 5.1 along with their selection criteria.

5.5.3 Results

Three experiments were carried out to validate the parking controller, and an additional set of parking tasks was carried out to quantify the repeatability of the parking process. In the first experiment (I) the robot parks from a position closer to the y-axis. In the second (II) and third (III) experiments the robot starts parking from positions on the positive and negative sides of the x-axis, respectively. Table 5.2 lists the values of the parameters used to obtain the results described below. Table 5.3 provides the summary of the experiments and Fig. 5.12, 5.13 and 5.14 show the results of the experiments I, II and III, respectively. The robot path in each of the experiments was obtained using gyro corrected odometry information. In Fig 5.12 it is clear that when the robot starts from closer to y-axis it parks by taking only a one forward reverse cycle. When the robot starts at positions away from the y-axis it only takes a few forward reverse cycles to park, as shown in Fig. 5.13 and Fig. 5.14. Additionally, Fig. 5.15 shows the effect of the adaptation of the value of c on the parking process. From Fig. 5.15(b) it is apparent that although the robot successfully completes the parking, it takes a higher number of cycles (thus, a longer time) to park. When the value of c is very low due to oscillations, the robot takes a much longer time to park, as shown in Fig. 5.15(c).

From the heading velocity curve (V) in each plot (d) of Fig 5.12, 5.13 and 5.14, it is apparent that reverse speed drops to a lower value when the robot gets closer

Table 5.1: System parameters and their selection criteria.

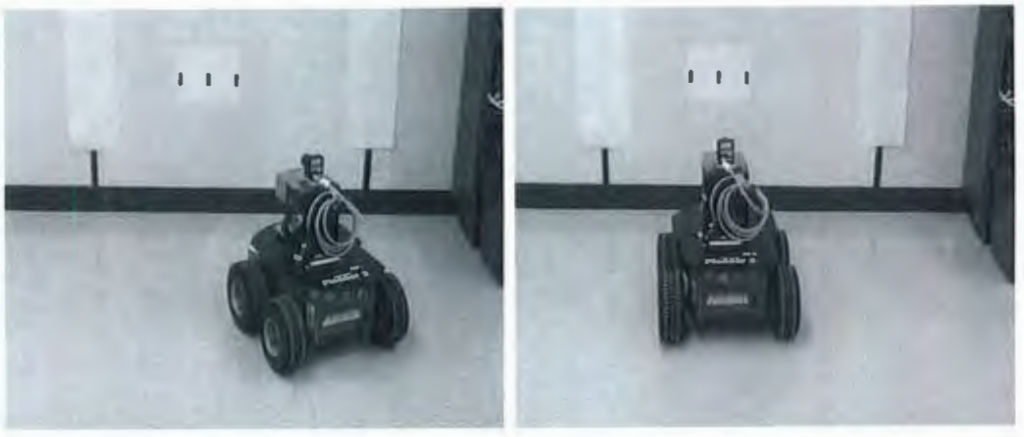
Parameter	Description
$[k_1(\text{start/reverse})\ k_2(\text{forward})]$	Gain values of the proportional controllers for rotational velocity control. Should be adjusted so that the robot has a fast velocity response without oscillations.
c_{min}	Minimum allowable distance to the virtual edge from real image edge. Should be adjusted so that the robot does not bump into the parking station.
c_{max}	Maximum allowable distance to the virtual edge from real image edge and it has to be less than the allowable maximum of $\frac{W-A'}{2}$. The value should be adjusted just below the allowable maximum so that the robot has enough steering angle to get on the y-axis when it is starting from a position closer to the y-axis (also *see* the proof of Theorem 5.4.2)
$[\ \delta_u\ \delta_A\ ]$	Defines the accuracy of the final parking position. Lower values define a tighter parking condition.
$[\ v_c\ v_r\ ]$	Forward and reverse heading velocities. Should be adjusted to the maximum values based on the frequency of the image acquisition and processing cycle.
$A_{\min}$	The minimum value of A that defines the outer boundary of the region of attraction. Should be selected so that when the parking station appears at $A_{\min}$, the camera should be able to capture enough information to robustly identify it.

Table 5.2: Selected parameter values in parking system.

Parameter	Value
k_p(reverse) k_p(forward)	0.25 0.1
c_{min}	255 pixels
$B/2$	106 mm
$[\ A^r\ u_1^r\ u_2^r\ u_3^r\ ]$	[432 478 695 910] pixels
$[\ v_c\ v_r\ ]$	[140 100] mm/sec
$[\ \delta_u\ \delta_A\ ]$	[1.5 1.5] pixels

Table 5.3: Summary of the set of experiments.

Exp.	Parking Time (s)	Forward–Reverse Cycles
I	10.9	1
II	37.8	3
III	47.3	3



(a) (b)

Figure 5.11: The P3AT robot at (a) a typical starting position and (b)the parked position of the robot.

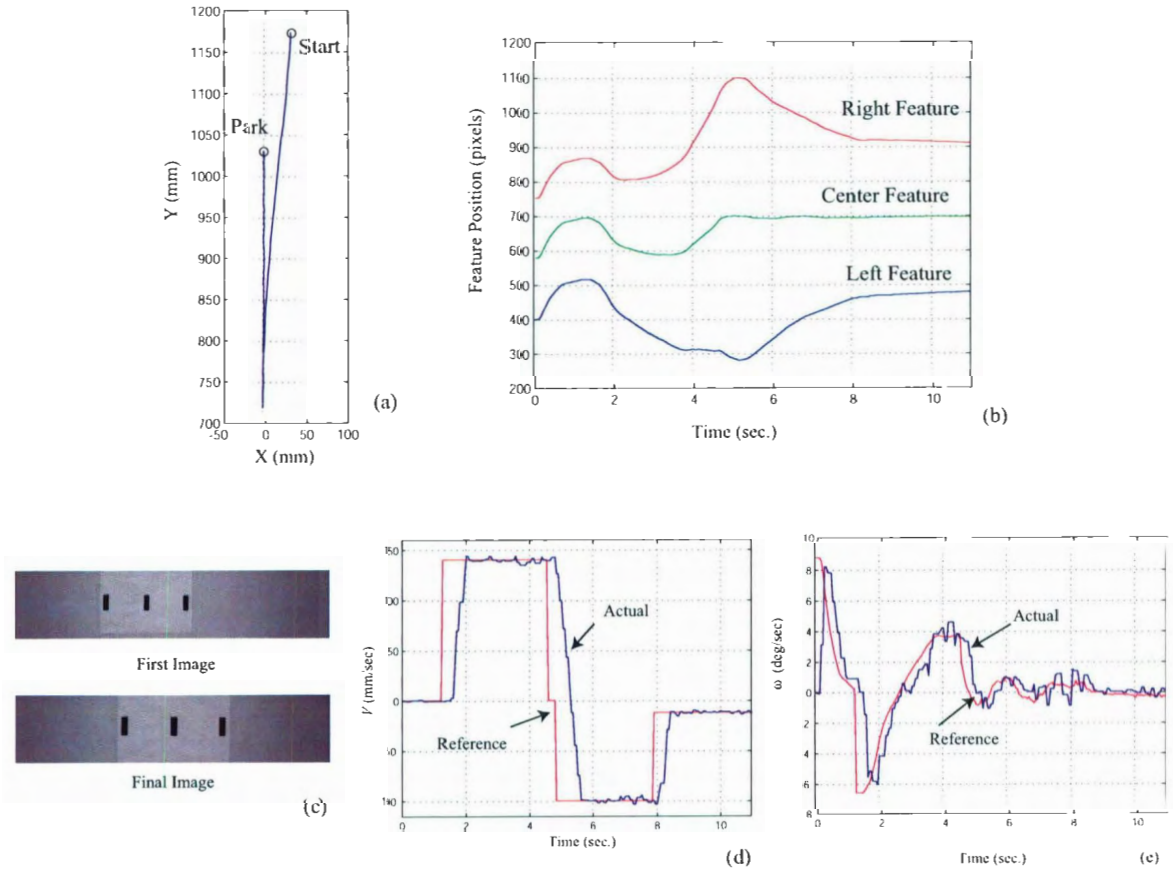


Figure 5.12: Results of the Experiment I. (a) The x-y trajectory of the robot during parking. (b)The trajectory of the feature positions during parking. (c) The initial and final images acquired during the parking routine. (d, e)The commanded and actual heading and rotational velocities of the robot during parking.

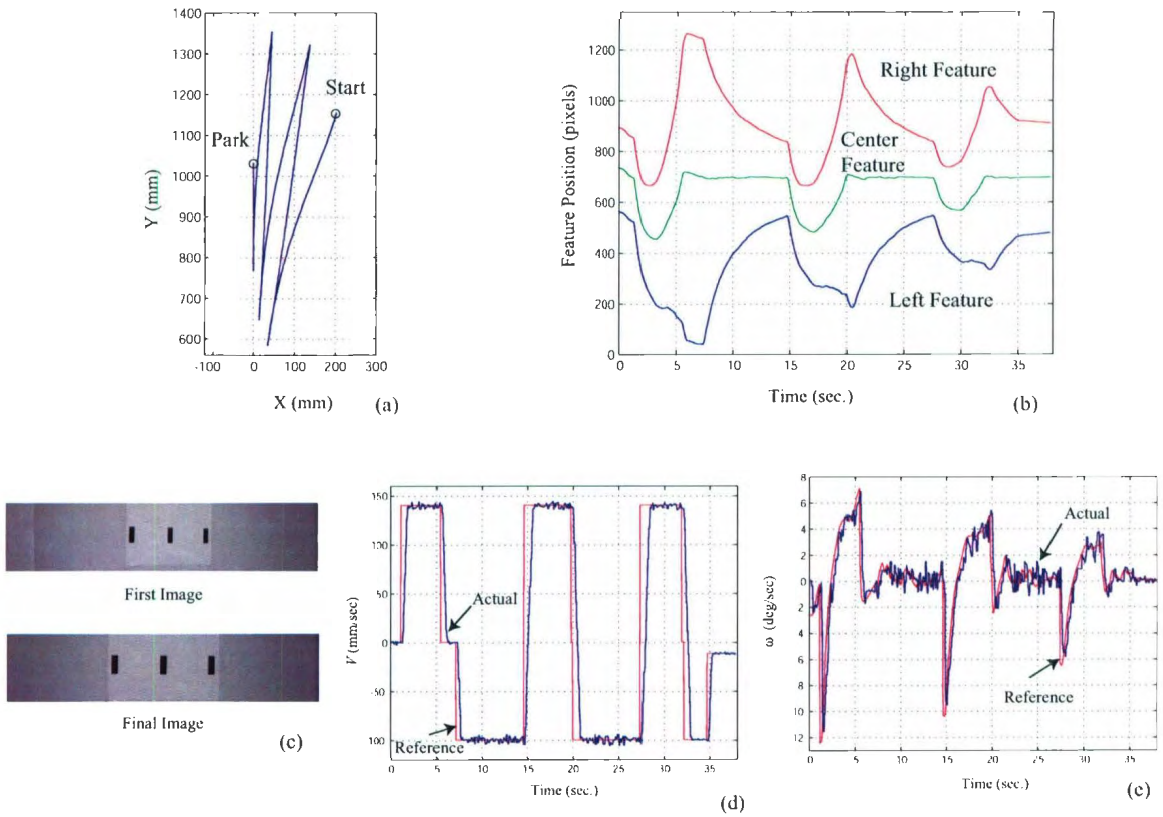


Figure 5.13: Results of the Experiment II. (a) The x-y trajectory of the robot during parking. (b)The trajectory of the feature positions during parking. (c) The initial and final images acquired during the parking routine. (d, e)The commanded and actual heading and rotational velocities of the robot during parking.

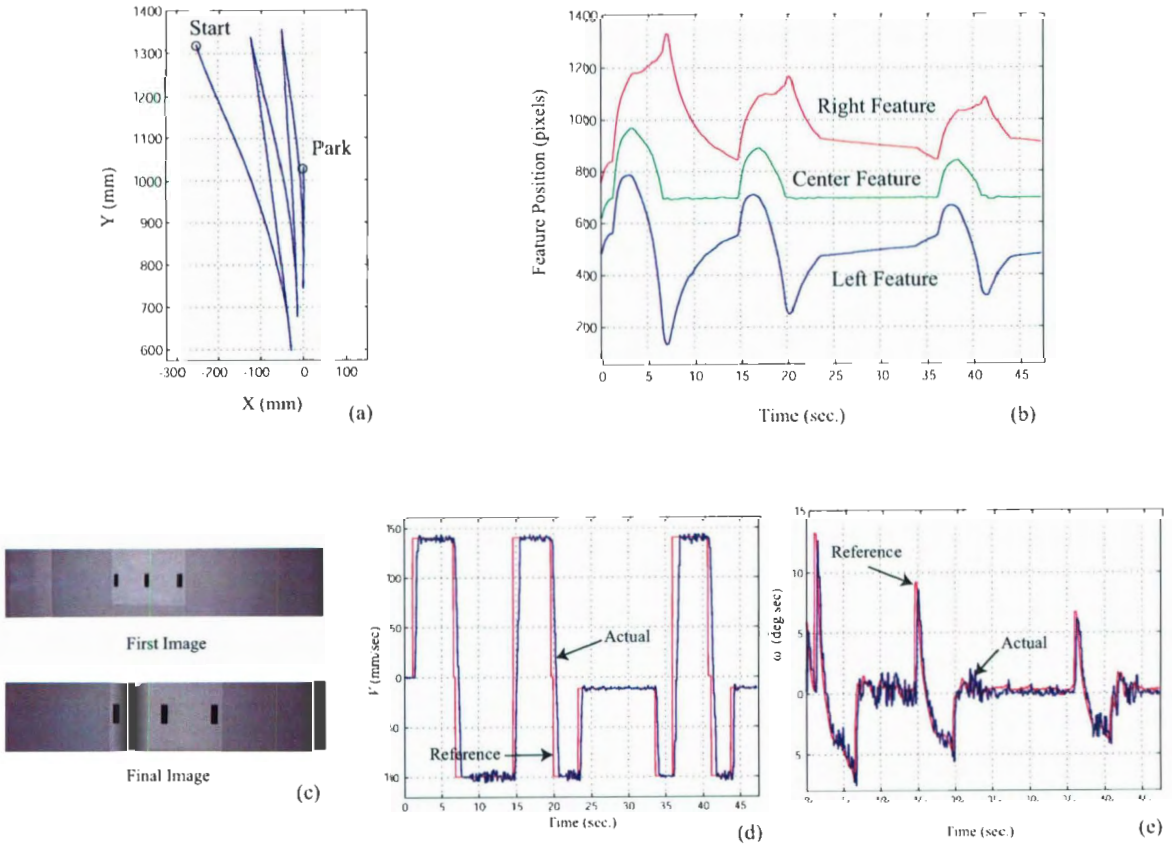


Figure 5.14: Results of the Experiment III. (a) The x-y trajectory of the robot during parking. (b)The trajectory of the feature positions during parking. (c) The initial and final images acquired during the parking routine. (d, e)The commanded and actual heading and rotational velocities of the robot during parking.

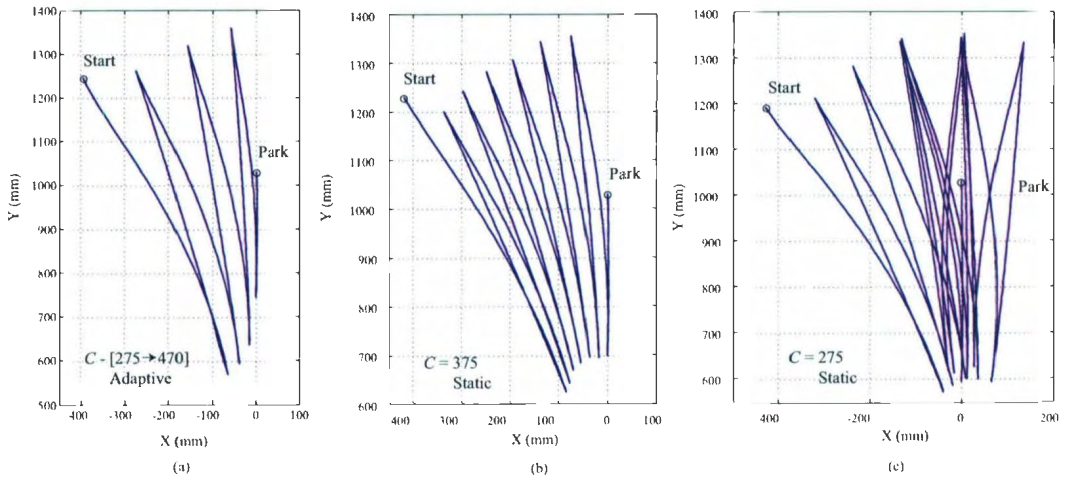


Figure 5.15: Effects of the c-adaptation on the path of the robot. (a) Successful parking with c-adaptation. In this case the allowed maximum of the c value is 478 and c varies from a c_{min} of 275 to a c_{max} of 470. (b) Parking with a higher static value of c (c) Parking attempt with a low static value of c.

to the potential parking position. The low robot velocity around the parking area is necessary to decrease the probability of missing the detection of the parking condition. This is a significant design consideration when low frame rate image processing is used. In this implementation a discrete velocity change is selected when $|E_A| < 25$ and f_{12} is less than a predefined threshold. Another possibility is to change the velocity smoothly based on the value of E_A as opposed to a discrete change.

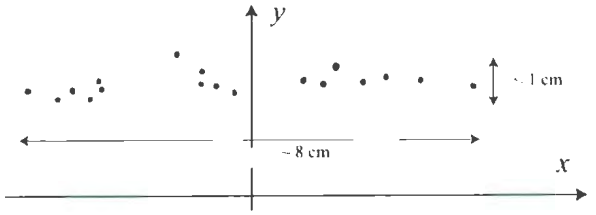


Figure 5.16: Final parking positions for 20 different arbitrary starting positions. Note: Some readings may overlap each other.

Fig. 5.16 shows the final parking positions for 20 different arbitrary starting positions with $\delta_u = \delta_A = 1.5$ pixels. The final robot position was obtained using a marker

attached to the robot frame and by scribing the position of the robot after each parking task. The robot achieves less than 1 cm accuracy in the y-direction, and 8 cm accuracy in the x-direction. It should be noted that a simple thresholding method followed by a blob analysis is used to detect the rectangular features in the parking station. The lower precision in the y-direction is mainly attributed to the number of features in the parking station. Accuracy in the y-direction can be improved by using a parking station with more features. Although addition of more features improves the accuracy in both axis, the improvements will be much more significant in y-direction. Further, the repeatability of the image-plane measurements under low quality fluorescent lighting affects the accuracies in both x and y-directions.

5.6 Conclusion

This chapter described a novel hybrid controller for parking robots autonomously against a set of features seen by a regular camera. The control law is based on image based visual servoing. Thus the parking strategy does not require any trajectory generations or odometry robot position feedback information to achieve accurate parking conditions. A comprehensive analysis is provided to prove the guaranteed convergence of the hybrid controller. Experimental results are shown to validate the system performance. The field-of-view constraints are adjusted to achieve robust and faster convergence. In comparison to other reported parking systems [28, 31] the proposed method requires fewer number of iterations to achieve the parking condition. As an example, the Lyapunov based hybrid control strategy in [28] requires a significant number of iterations even for the simple case, similar to Experiment I, whereas the proposed method has demonstrated the same parking process using only one iteration. Similar problems can be observed in the visual servoing based technique presented in

[31]. Comparatively, even in the extreme cases when the robot starts away from the y-axis (Experiments II and III), the proposed system has the ability to converge with fewer iterations (in this case four). In addition to these performance improvements, proposed method has demonstrated the repeatability and accuracy of the method and has proven the robustness of the system. The proposed method has a demonstrated repeatability of ± 4 cm and ± 0.5 cm in the x and y directions, respectively.

Chapter 6

Conclusion

The main goals of this thesis were to investigate the vision and laser range finder based applications in SLAM, moving object detection and precise visual servoing in mobile robotics. These three goals were achieved by devising novel methods, using the important characteristics of the available methods. This chapter provides an overall conclusion of the topics and a summary of the key contributions of this thesis followed by a list of possible future research directions.

6.1 Discussion

The laser and vision sensors can be fused together to exploit the advantages of each sensor while overcoming the disadvantages of the other sensor. The multi sensor SLAM application discussed in this thesis demonstrates that computer vision and laser range scanners can be used to accurately detect and measure visually salient landmarks in the environment. Further, such measurements can be readily integrated into the EKF based SLAM method to build maps of typical indoor environments. From the results it is evident that using a calibrated laser-vision sensor, a higher

number of high quality landmarks can be detected. The major advantage of the laser-camera method is that landmarks can be detected and accurately located during each iteration. In contrast, bearing only methods require multiple frames of sensor data to initialize a landmark with acceptable accuracy, and to some extent stereo vision suffers from the same problem, in addition to its high computational complexity. Additionally, the robustness of detection in the proposed sensor unit can be further improved by incorporating laser based landmark detection methods in addition to vision based landmark detection.

General moving object detection in mobile robotics can support both safe navigation and robust mapping in the presence of moving objects in the environment. In this thesis a general moving object detection algorithm was presented. The algorithm uses some specific properties of the laser scan data corresponding to the moving objects to successfully detect them. The proposed algorithm can be easily used to detect multiple moving objects from a moving platform in a dynamic environment. Additionally, in comparison to other methods, the proposed algorithm has the ability to recover the complete moving object when the object is moving at a low relative velocity and when the object is moving sideways with respect to the scan direction. Through the tuning of the parameters, the detectable minimum relative velocity can be adjusted to suit the application. The results demonstrate that the proposed algorithm can be used to successfully track many different types of moving objects. In regular SLAM implementations the environment (or the landmarks) is assumed to be stationary. Therefore, apart from the direct use of moving object detection and tracking, the proposed method can be used as a data preprocessing step in regular SLAM applications to remove the data related to the moving objects from the sensor data. This type of preprocessing will aid in improving the stability of the SLAM filters by preventing any possible moving landmarks from corrupting the data structures.

Visual servoing a nonholonomic mobile robot is a challenging task when the camera field of view constraints is considered. A novel hybrid controller for parking mobile robots is proposed in this thesis for autonomously parking the robot against a set of features seen by a regular camera. The control law is based on image based visual servoing. Thus the parking strategy does not require any trajectory generations or odometry robot position feedback information to achieve accurate parking conditions. A comprehensive analysis is provided to prove the guaranteed convergence of the hybrid controller. Experimental results are shown to validate the system performance. The field-of-view constraints are adjusted to achieve robust and faster convergence. In comparison to other reported parking systems [28, 31] the proposed method requires fewer number of iterations to achieve the parking condition. As an example, the Lyapunov based hybrid control strategy in [28] requires a significant number of iterations even for the simple case, similar to Experiment I, whereas the proposed method has demonstrated the same parking process using only one iteration. Similar problems can be observed in the visual servoing based technique presented in [31]. Even when the robot starts at extreme poses, the proposed system has the ability to converge with fewer number of iterations. In addition to these performance improvements, this thesis has demonstrated the repeatability and accuracy of the method and has proven the robustness of the system.

6.2 Conclusion

The key contributions of this thesis are:

1. Multisensor landmark localization and detection: The landmarks are first detected and then localized with respect to the robot frame by using a single camera and a laser range finder. The novelty of the proposed method arises

from the fact that each step of the process - detection and localization - is accomplished using the most suitable sensor whereas other similar methods [1] used sensor fusion techniques to fuse similar types of observations.

2. Moving object detection: Moving object detection algorithm uses pre-registered laser data from laser range finder to extract the measurements that correspond to moving objects. In contrast to other similar work, the proposed method systematically addresses all possible cases of moving object by how they appear in laser data. Further minimum detectable relative velocity can be tuned such that the algorithm is flexible enough to detect objects that are moving at a wide range of velocities.

3. Robot parking using visual servoing: A convergent novel parking algorithm was developed and implemented using a single camera fixed to robot frame. The parking algorithm uses a hybrid controller to overcome nonholonomic constraints of the robot and limited field of view constraints of the camera. The key properties of the proposed controller are: (1) the maximum use of the available field of view, (2) avoids the zeno behaviour, (3) global convergent regardless of the initial pose of the robot.

6.3 Publications Resulting from the PhD Program

6.3.1 Journal Papers

1. Dilan Amarasinghe, George K. I. Mann, and Raymond G. Gosine. Landmark Detection and Localization for Mobile Robot Applications; A Multisensor Approach. *ROBOTICA*, 2008 [Submitted for review].

2. Dilan Amarasinghe, George K. I. Mann, and Raymond G. Gosine. Vision based hybrid control scheme for autonomous parking of a mobile robot. *Advanced Robotics*, 21(8):905 930, May 2007.

6.3.2 Refereed Conference Papers

1. Dynamic object identification by a moving robot using laser data. In *International Federation of Automatic Control World Congress (IFAC WC 2008)*, July 2008.

2. Integrated laser-camera sensor for the detection and localization of landmarks for robotic applications. In *International Conference on Robotics and Automation(ICRA2008)*, May 2008.

3. Moving object detection in indoor environments using laser range data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2006)*, pages 802 807, October 2006.

4. Vision-based hybrid control strategy for autonomous docking of a mobile robot. In *IEEE Conference on Control Applications (CCA2005)*, pages 1600 1605, August 2005.

6.4 Future Work

During this research the following areas were identified to have possible future research potential:

- In multisensor mapping and localization the effectiveness of the sensor(s) depends on the characteristics of the environment. For example, if the environ-

ment contains a significant number of direction invariant features in the plan view then it would be best to use the laser range scanner to identify the landmarks, whereas in environments that are void of any such landmarks the computer vision can be used to identify landmarks. Switching of the sensor model from vision based landmark detection to a laser only model can be beneficial, as methods for detection of point features in the laser are computationally less demanding than those in vision. Such reductions in computational requirements for localization and mapping can be utilized to improve the processing of navigation and path planning functions. In a simpler implementation the decision of the switching points can be based purely on the number of features. However, further study is required to better understand the best switching scenarios and their effects on the performance of the SLAM.

- Some visually salient landmarks are present in the form of wide vertical strips, in which two side edges are detected as vertical lines. Thus they are recognized as two landmarks and the SLAM algorithm will attempt to initialize them as such in the map. However, since they are often physically close together, only one of them will be initialized into the map. Further, when the robot is away from visual features as described above, the line detection algorithm will often detect a single line due to the limitation in the resolution of the camera. However, as the robot gets closer to the object, it will appear as two landmarks where the data association algorithm will require further classifications in order to decide the best edge to assign to the feature that is already in the map. This type of difficult decision could be avoided by using better image processing techniques that would identify the vertical strip as a single object, both when the robot is closer to the object and when the robot is away from the object.

- As highlighted in Chapter 4 moving object detection has been a widely studied topic in computer vision. However, vision based moving object detection from moving robots poses a challenging task due to the residual motion of the background. Nevertheless as proposed in this thesis, laser range data can be used to effectively detect the moving objects in the environment. The initial information from the detections can be used to enhance the vision based moving object detection through the localization of the moving areas by using a calibrated laser sensor configuration. This localized search of visual information can be used to robustly detect the complete objects. Laser based solutions can only recover information about the laser scan plane.

- The image based visual servoing application proposed in this thesis adapts its parameters to minimize the servoing duration and the length of travel of the robot. In this case a simple linear adaptation scheme has been used in the proposed solution to improve the performance of the visual servoing task with respect to the unmodulated case. The adaptation scheme could be further improved using more intelligent techniques, where the performance of the robot could be optimized with a nonlinear relation between the parameters of the control system and the current robot pose.

- Finally, in any robot platform the final realization is based on an integrated set of algorithms based on a suitable robot architecture. Although in early robotics a purely sense-plan-act cycle based architecture was dominant, later the reactive architecture, where the robot senses, plan, and acts in parallel became popular. Modern robotics architectures closely follow the traits of the reactive methods while utilizing the important characteristics of the early decision cycle based architectures, which are commonly known as hybrid architectures. The

functionality described in this thesis, along with an intelligent path planning method, can be used to realize a completely autonomous robot implementation. As computer vision and SLAM algorithms are usually computationally demanding, the most suitable architecture and the method of implementation should be selected after a careful study of the currently available methods and available resources.

Bibliography

[1] J. A. Castellanos, J. Neira, and J. D. Tardos. Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 17(6):908 914, December 2001.

[2] John E. Bares and David S. Wettergreen. Dante ii: Technical description, results and lessons learned. *International Journal of Robotics Research*, 18(7):621 649, July 1999.

[3] Sebastian Thrun, Dirk Hahnel, David Ferguson, Michael Montemerlo, Rudolph Triebel, Wolfram Burgard, Christopher Baker, Zachary Omohundro, Scott Thayer, and William Whittakert. A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4270 4275, September 2003.

[4] Dimitrios S. Apostolopoulos, Liam Pedersen, Benjamin N. Shamah, Kimberly Shillcutt, Michael D. Wagner, and William L. Whittaker. Robotic antarctic meteorite search: outcomes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4174 4179, May 2001.

[5] Chris Urmson, Ben Shamah, James P. Teza, Michael D. Wagner, Dimitrios Apostolopoulos, and William Whittaker. A sensor arm for robotic antarctic

meteorite search. In *Proceedings of the International Conference on Field and Service Robotics*, July 2001.

[6] Deepak Bapna, Eric Rollins, John Murphy, Mark Maimone, William Red L. Whittaker, and David Wettergreen. The atacama desert trek: Outcomes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 597 604, May 1998.

[7] M. P. Golombek, R. A. Cook, T. Economou, W. M. Folkner, A. F. C. Haldemann, P. H. Kallemeyn, J. M. Knudsen, R. M. Manning, H. J. Moore, T. J. Parker, R. Rieder, J. T. Schofield, P. H. Smith, and R. M. Vaughan. Overview of the mars pathfinder mission and assessment of landing site predictions. *Science*, 278(5344):1743 1748, December 1997.

[8] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

[9] J. Borenstein, H. R. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd, 1996.

[10] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1442 1447, May 1991.

[11] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. J. Cox and G. T. Wilfon, editors, *Autonomous Robot Vehicles*, pages 167 193. Springler Verlag: New York, 1990.

[12] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3 4):253 271, July 1998.

[13] Feng Lu and Evangelos E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 935 938, June 1994.

[14] Momotaz Begum. Robotic mapping using soft computing methodologies. Master's thesis, Memorial University of Newfoundland, July 2005.

[15] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7 8):693 716, August 2004.

[16] Michael Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, July 2003.

[17] Ronald C. Arkin. *Behavior Based Robotics*. MIT Press, Cambridge, MA, 1998.

[18] Masayuki Yokoyama and Romaso Poggio. A contour-based moving object detection and tracking. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 271 276, October 2005.

[19] Nokos Paragios and Rachid Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266 280, March 2000.

[20] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, G. A. Fink, and G. Sagerer. Person tracking with a mobile robot based on multi-modal anchoring. In *Proceedings of the International Workshop on Robot and Human Interactive Communication*, pages 423 429, September 2002.

[21] M. Lindstrom and J.-O. Eklundh. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1364 1369, November 2001.

[22] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. *The International Journal of Robotics Research*, 22(2):99 116, February 2003.

[23] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651 670, 1996.

[24] Richard M. Murray and S Shankar Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700 716, May 1993.

[25] Yi Ma, Jana Kosecka, and Shankar S. Sastry. Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3):521 536, June 1999.

[26] Hong Zhang and James O. Ostrowski. Visual motion planning for mobile robots. *IEEE Transactions on Robotics and Automation*, 18(2):199 208, April 2002.

[27] Jose R. H. Carvalho, Patrick Rives, Aliton Santa-Barbara, and Samuel S. Bueno. Visual servo control for a class of mobile robot. In *Proceedings of the International Conference on Control Applications*, pages 431 436, September 2000.

[28] Pierpaolo Murrieri, Daniele Fontanelli, and Antonio Bicchi. A hybrid-control approach to parking problem of a wheeled vehicle using limited view angle visual feedback. *The International Journal of Robotics Research*, 23(4 5):437 448, April-May 2004.

[29] Fabio Conticelli, Benedetto Allotta, and Pradeep K. Khosla. Image-based visual servoing of nonholonomic mobile robots. In *Proceedings of the International Conference on Decision and Control*, pages 3496 3501, December 1999.

[30] F. Conticelli, D. Prattichizzo, F. Guidi, and A. Bicchi. Vision-based dynamic estimation and set-point stabilization of nonholonomic vehicles. In *Proceedings of the International Conference on Robotics and Automation*, pages 2771 2776, 2000.

[31] Koichi Hashimoto and Toshiro Noritsugu. Visual servoing of nonholonomic cart. In *Proceedings of the International Conference on Robotics and Automation*, pages 1719 1724, April 1997.

[32] Kane Usher, Peter Ridley, and Peter Corke. Visual servoing of a car-like vehicle an application of omnidirectional vision. In *Proceedings of the International Conference on Robotics and Automation*, pages 4288 4293, September 2003.

[33] Soo-Hwan Oh and Se-Young Oh. Image-based visual servoing for mobile robots using neural networks and fuzzy-evolutionary methods. In *Proceedings of International Joint Conference on Neural Networks*, pages 1367 1372, May 2002.

[34] Wolfgang A. Daxwanger and Günther K. Schmidt. Skill based visual parking control using neural and fuzzy networks. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1659 1664, October 1995.

[35] Wolfgang A. Daxwanger and Günther K. Schmidt. Skill based vehicle guidance by use of artificial neural networks. *Mathematics and Computers in Simulation*, 41:263 271, 1996.

[36] W. A. Daxwanger and G. Schmidt. Neural and fuzzy approaches to vision based parking control. *Control Engineering Practice*, 4(11):1607 1614, 1996.

[37] W. Sardha Wijesoma, L. D. L. Perera, and M. D. Adams. Toward multidimensional assignment data association in robot localization and mapping. *IEEE Transactions on Robotics*, 22(2):350 365, April 2006.

[38] M. W. M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229 241, June 2001.

[39] Stefan B. Williams, Hugh Durrant-Whyte, and Gamini Dissanayake. Constrained initialization of simultaneous localization and mappling algorithm. *The International Journal of Robotics*, 22(7 8):541 564, July August 2003.

[40] Andrew J Davison. *Mobile Robot Navigation Using Active Vision.* PhD thesis, University of Oxford, June 1998.

[41] Michael Carsten Bosse. *ATLAS: A Framework for Large Scale Automated Mapping and Localization.* PhD thesis, Massachusetts Institute of Technology, February 2004.

[42] SICK. Laser measurement system, January 2002.

[43] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593 600, June 1994.

[44] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2051 2058, May 2001.

[45] Akio Kosaka and Avinash C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56(3):271 329, November 1992.

[46] P. Saeedi, P. D. Lawrence, and D. G. Lowe. Vision based 3 d trajectory tracking for unknown environments. *IEEE Transactions on Robotics*, 22(1):119 136, February 2006.

[47] Jean-Yves Bouguet and Pietro Perona. Visual navigation using a single camera. In *Proceedings of the International Conference on Computer Vision*, pages 645 652, 1995.

[48] Andrew J Davison and David W Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865 880. 2002.

[49] Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the International Conference on Robotics and Automation*, pages 2918 2924, September 2002.

[50] Chieh-Chih Wang. *Simultaneous Localization, Mapping, and Moving Object Tracking*. PhD thesis, Carnegie Mellon University, 2004.

[51] Abel Mendes, Luis Conde Bento, and Urbano Nunes. Multi-target detection and tracking with a laserscanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 796 801, June 2004.

[52] Michael Montemerlo, Sebastian Thrun, and William Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proceedings of the International Conference on Robotics and Automation*, pages 695 701, May 2002.

[53] Ajo Fod, Andrew Howard, and Maja J Mataric. A laser-based people tracker. In *Proceedings of the International Conference on Robotics and Automation*, pages 3024 3029, May 2002.

[54] Ilya Kolmanovsky and N. Harris McClamroch. Developments in nonholonomic control problems. *IEEE Control Systems Magazine*, 15(6):20 36, December 1995.

[55] Rene Vidal, Omid Shakernia, and Shankar Sastry. Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. In *Proceedings of the International Conference on Robotics and Automation*, pages 584 589, September 2003.

[56] D. Khadraoui, C. Debain, R. Rouveure, P. Martinet, P. Bonton, and J. Grace. Vision based control in driving assistance of agricultural vehicles. *International Journal of Robotics Research*, 17(10):1040 1054, October 1999.

[57] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. *IEEE Robotics and Automation Magazine*, pages 27 35, March 1995.

[58] P. Newman and Kin Ho. Slam-loop closing with visually salient features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

[59] Simon J. Julier and Jeffrey K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4238 4243, May 2001.

[60] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562 3568, October 2006.

[61] Jose A. Castellanos, Jose Neira, and Juan D. Tardos. Map building and slam algorithms. In Shuzhi Sam Ge and Frank L. Lewis, editors, *Autonomous Mobile*

Robots: Sensing, Control, Decision Making and Applications, chapter 9, pages 335 371. CRC Press, May 2006.

[62] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report 95 041, Dept. of Computer Science, University of North Carolina at Chapel Hill, 1995. Updated in 2003.

[63] Stefan Bernard Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD thesis, The University of Sydney, September 2001.

[64] Stephen Se, David Lowe, and James J. Little. Vision based global localization and mapping for mobile robots. *IEEE Transactions in Robotics*, 21(3):364 375, June 2005.

[65] Yaakov Bar-Shalom, X. Rong Li, and Taiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[66] Jose Neira and Juan D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890 897, December 2001.

[67] Juan Nieto, Jose Guivant, Eduardo Nebot, and Sebastian Thrun. Real time data association for fastslam. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 412 418, September 2003.

[68] Jose A. Castellanos, Jose Neira, and Juan D. Tardos. Limits of the consistency of ekf-based slam. In *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, July 2004.

[69] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardós, and J. Neira. Robocentric map joining: Improving the consistency of ekf-slam. *Robotics and Autonomous Systems*, 55(1):21 29, January 2007.

[70] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 1999.

[71] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.

[72] Jose E. Guivant and Mario Nebot. Optimization of simultaneous localization an map-building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242 257, June 2001.

[73] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the International Conference on Robotics and Automation*, pages 842 849, September 2003.

[74] David Ribas, Pere Ridao, Jose Neira, and Juan D. Tardos. Slam using an imaging sonar for partially structured underwater environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5040 5045, October 2006.

[75] Jong-Hyuk Kim and Salah Sukkarieh. Airborne simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 406 411, September 2003.

[76] Angeli Adrien, David Filliat, Stephane Doncieux, and Jean-Arcady Meyer. 2D simultaneous localization and mapping for micro aerial vehicles. In *Proceedings of the European Micro Aerial Vehicles (EMAV 2006) conference*, July 2006.

[77] Juan D. Tardós, José Neira, Paul M. Newman, and John J. Leonard. Robust mapping anf localization in indoor environments uing sonar data. *International Journal of Robotics Research*, 21(4):311 330, April 2002.

[78] Udo Frese. Treemap: An $o(\log(n))$ algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103 122, September 2006.

[79] S. Takezawa, D. C. Herath, and G. Dissanayake. Slam in indoor environments with stereo vision. In *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1866 1871, Septmber October 2004.

[80] John Folkesson, Patric Jensfelt, and Henrik I. Christensen. Vision slam in the measurement subspace. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 30 35, April 2005.

[81] S. Panzieri, F. Pascucci, and G Ulivi. Vision based navigation using kalman approach for slam. In *Proceedings of the International Conference on Advanced Robotics*, July 2003.

[82] Andrew J Davison and David W Murray. Mobile robot localization using active vision. In *Proceedings of the 5th European Conference on Computer Vision*, pages 809 825, 1998.

[83] Andrew J Davison. Real-time simultaneous localization and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 1403 1410, October 2003.

[84] N. M. Kwok and G. Dissanayake. An efficient multiple hypothesis filter for bearing-only slam. In *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 736 741, September October 2004.

[85] N. M. Kwok, G. Dissanayake, and Q. P. Ha. Bearing-only slam using sprt basd gaussian sum filter. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1109 1114, April 2005.

[86] Thomas Lemaire, Simon Lacroix, and Joan Sola. A practical 3d bearing-only slam algorithm. In *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems*, August 2005.

[87] J. M. M. Montiel, Javier Civera, and Andrew J. Davison. Unified inverse depth parametrization for monocular slam. In *Proceedings of the Robotics, Science and Systems*, August 2006.

[88] Fang Fang, Xudong Ma, and Xianzhong Dai. A multi-sensor fusion slam approach for mobile robots. In *Proceedings of the International Conference on Mechatronics and Automation*, pages 1837 1841, July 2005.

[89] Kai O. Arras and Nicola Tomatis. Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision. In *Proceedings of the Third European Workshop on Advanced Mobile Robots*, September 1999.

[90] Kai O. Arras, Nicola Tomatis, and Roland Siegwart. Multisensor on-the-fly localization using laser and vision. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 462 466, 2000.

[91] J. E. Guivant, F. R. Masson, and E. M. Nebot. Simultaneous localization and map building using natural features and absolute information. *Robotics and Autonomous Systems*, 40(2):79 90, 2002.

[92] Woo Yeon Jeong and Kyoung Mu Lee. Visual slam with line and corner features. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2570 2575, October 2006.

[93] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, , and P. Sayd. Monocular vision based slam for mobile robots. In *Proceedings of International Conference on Pattern Recognition*, pages 1027 1031, August 2006.

[94] J. V. Miro, G. Dissanayake, and Weizhen Zhou. Vision-based slam using natural features in indoor environments. In *Proceedings of International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 151 156, December 2005.

[95] Paul E. Rybski, Stergios I. Roumeliotis, Maria Gini, and Nikolaos Papanikolopoulos. Appearance based minimalistic metric slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 194 199, October 2003.

[96] Josep M. Porta and Ben J. A. Krose. Appearance based concurrent map building and localization using a multi hypotheses tracker. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3424 3429, October 2004.

[97] H. M. Gross, A. Koenig, and St. Mueller. Omniview based concurrent map building and localization using adaptive appearance maps. In *Proceedings of*

the IEEE International Conference on Systems, Man and Cybernetics, pages 3510 3515, October 2005.

[98] Josep M. Porta, Jakob J. Verbeek, and Ben Krose. Enhancing appearance based robot localization using sparse disparity maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 980 985, October 2003.

[99] B. J. A. Krose, N. Vlassis, and R. Bunschoten. A probabilistic model for appearance based robot localization. *Image and Vision Computing*, 12(6):381 391, April 2001.

[100] Iwan Ulrich and Illah Nourbakhsh. Appearance based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1023 1029, April 2000.

[101] S. Se and P. Jasiobedzki. Photo realistic 3d model reconstruction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3076 3082, May 2006.

[102] D. Ortin, J. Neira, and J. M. M. Montiel. Relocation using laser and vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1505 1510, April 2005.

[103] D. G. Lowe. Object recognition from local scale invariant features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV97)*, pages 1150 1157, September 1997.

[104] Matthew Brown and David Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59 73, August 2007.

[105] Laura A Clemente, Andrew J Davison, Ian D Reid, Jose Neira, and Juan D Tardos. Mapping large loops with a single hand-held camera. In *Proceedings of the Robotics Science and Systems*, 2007.

[106] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052 1067, 2007.

[107] Ruben Martinez-Cantin, Jose A Castellanos, Juan D Tardos, and J M M Montiel. Adaptive scale robust segmentation for 2d laser scanner. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 796 801, October 2006.

[108] Larry Davis, Vasanth Philomin, and Ramani Duraiswami. Tracking humans from a moving platform. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 171 178, September 2000.

[109] J. Frazier and R. Nevatia. Detection moving objects from a moving platform. In *Proceedings of the International Conference on Robotics and Automation*, pages 1627 1633, May 1992.

[110] Takumi Ebine and Nozomu Hamada. Detection of moving objects using observer motion-based optical flow estimation. *Systems and Computers in Japan*, 6(33):83 92, June 2002.

[111] Sanae Shimizu, Kazuhiko Yamamoto, Caihau Wang, Yutaka Satoh, Hideki Tanahashi, and Yoshinori Niwa. Detection of moving object by mobile stereo omnidirectional system (sos). *Electrical Engineering in Japan*, 3(152):29 38, June 2005.

[112] Jae Hoon Lee, Takashi Tsubouchi, Kenjiro Yamamoto, and Saku Egawa. People tracking using a robot in motion with laser range finder. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2936 2942, October 2006.

[113] Erwin Prassler, Jens Scholz, and Alberto Elfes. Tracking multiple moving objects for real-time robot navigation. *Autonomous Robots*, 8:105 116, 2000.

[114] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):223 240, 1988.

[115] Ronald C. Arkin and Douglas Mackenzie. Temporal coordination of perceptual algorithms for mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 10(3):276 286, June 1994.

[116] Parallel-parking control of autonomous mobile robot. In *International Conference on Industrial Electronics, Control and Instrumentation*, volume 3, pages 1305 1310, November 1997.

[117] Tzuu-Hseng S. Li and Shih-Jie Chang. Autonomous fuzzy parking control of a car like mobile robot. *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*, 33(4):451 465, July 2003.

[118] J Borenstein, H. R. Everett, and L. Feng.

[119] Jose Santos-Victor and Giulio Sandini. Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3):223 238, September 1997.

[120] Brian W. Minten, Robin R. Murphy, Jeff Hyams, and Mark Micire. Low order complexity vision based docking. *IEEE Transactions on Robotics and Automation*, 17(6):922 930, December 2001.