

# Navigation and Control for Assistive Robotics

by

Lakshitha Dantanarayana

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

at the

Centre for Autonomous Systems  
Faculty of Engineering and Information Technology  
**University of Technology, Sydney**

October 2016

# Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed:

---

Date:

---

# Navigation and Control for Assistive Robotics

by

Lakshitha Dantanarayana

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

## *Abstract*

In this thesis, we address the problem of navigation and control for assistive robots. Autonomously creating a suitable representation of the environment (a map) and having the ability to localise a robot in that environment are considered to be the cornerstones of autonomous robot navigation. We present a distance function based framework to represent the occupancy of two-dimensional environments and a chamfer distance based sensor model to relate measurements captured from a sensor mounted on the robot to the environment representation. Employing the proposed representation and the sensor model, we propose two novel strategies to localise the robot on the map using an extended Kalman filter and an optimisation based method. These methods are computationally more efficient and are free of environment dependent tuning, which are necessities for assistive robots to operate in different environments ranging from small households to large shopping centres. We also demonstrate an adaptation of the popular particle filter based localisation algorithm using the distance function representation.

A mapping algorithm that utilises the proposed distance function based framework, which can be used to create maps of considerably large scale crowded indoor environments with low error accumulation is also presented. Although we do not consider the effect of sensor uncertainties, we demonstrate that the algorithm can efficiently build high-quality maps that can be used in practical scenarios of importance associated with assistive robots.

---

We present experiments conducted using simulations, public domain datasets, and experimental datasets we collected in real environments to evaluate and compare these algorithms.

The control strategy used in an assistive robot needs to be specifically designed to suit the task that the robot is expected to perform. Using standard user centred design methods often result in complicated, unintuitive control interfaces for assistive robots, which are difficult to be integrated into the daily activities of the end users.

We demonstrate that design approaches based on the principles of cooperative design can be used to alleviate the complexities in the design process. We propose and develop a control system based on admittance control for a robotic hoist, and evaluate it using user studies to experimentally illustrate that this design framework could be used for developing controllers for assistive robots in general.

The analysis of electromyographic measurements and forces exerted by the end users while using the robotic hoist confirm that the robot has the potential to reduce musculoskeletal injuries amongst care workers in the aged and disabled care sector, by providing assistance during the patient transfer process. As a result of the cooperative design process, the control interface became simple, intuitive, and easy to use, which made the robot readily incorporable to the work-flow of care facility.

# *Acknowledgements*

I take this opportunity to thank many others around me, without whose support and encouragement, this work would not have been possible.

Especially to my primary supervisor, Professor Gamini Dissanayake for the opportunity to work on this topic and the guidance provided throughout. His direction, motivation, and expertise allowed me to reach this significant milestone in my life.

I'm also grateful to my alternate supervisor Doctor Ravindra Ranasinghe for his continuous support. It was an immense help, and he contributed significantly to the completion of this work.

I also wish to thank my co-supervisor, Associate Professor Shoudong Huang, for always being willing to render assistance whenever it was requested, even at short notice.

My gratitude also goes to Professor Tomonari Furukawa for his helpful advice and for providing me with the valuable opportunity to be a part of the Terrestrial Robotics, Engineering, & Controls Lab (TREC) at Virginia Tech, USA and the DARPA Robotics Challenge 2015 team. I gained an immense amount of valuable experience throughout my visit. I'm grateful to everyone in Team VALOR for making my stay at TREC and in Blacksburg, VA, memorable.

I'm appreciative to all my colleagues at Centre for Autonomous Systems (CAS), I find it to have been a privilege to work with such motivated and intelligent group of people. Thanks for making me a part of the CAS family and sharing all the laughs and good times.

Last, but not least, I'm forever in debt to my parents, because without the sacrifices they've made on my behalf, I wouldn't be where I am today.

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>List of Figures</b>	<b>viii</b>
<b>Acronyms &amp; Abbreviations</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Background . . . . .	1
1.2 Thesis Outline . . . . .	5
1.3 Contributions . . . . .	6
1.4 Publications Related to this Thesis . . . . .	7
<b>2 Environment Representation for Robotics</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Distance Functions . . . . .	13
2.3 Sensing and Sensor Models . . . . .	17
2.3.1 A Sensor Model for Distance Function Representations . . . . .	19
2.3.2 Projection of a Laser Scan on a Distance Function . . . . .	20
2.4 Chamfer Distance . . . . .	22
2.4.1 Chamfer Distance based Sensor Model for Distance Function based Environment Representations . . . . .	23
2.4.2 Chamfer Distance Based Sensor Model for a Camera . . . . .	24
2.4.3 Image Template Matching Using Chamfer Distance . . . . .	25
2.4.4 Chamfer Distance Based Sensor Model for a Laser Range Finder . .	28
2.5 Conclusion . . . . .	31
<b>3 Distance Function based Algorithms for Mobile Robot Localisation</b>	<b>32</b>
3.1 Introduction . . . . .	32

3.2	Related Work . . . . .	34
3.3	An Extended Kalman Filter for Localisation in an Occupancy Grid Map . .	38
3.3.1	Chamfer Distance Based Observation Model . . . . .	39
3.3.2	Formulation of the Extended Kalman Filter . . . . .	39
3.3.2.1	Prediction . . . . .	39
3.3.2.2	Observation . . . . .	40
3.3.2.3	Update . . . . .	40
3.3.3	Improving the Robustness of the Algorithm . . . . .	41
3.4	Optimisation Based Localisation in Occupancy Grid Maps . . . . .	42
3.4.1	Optimisation Algorithm . . . . .	43
3.5	Particle Filter based Localisation on Occupancy Grid Maps . . . . .	44
3.5.1	Monte-Carlo Localisation (MCL) . . . . .	46
3.5.2	Estimation of Covariance and Measurement Likelihood . . . . .	48
3.6	Experimental Results . . . . .	50
3.6.1	Experimental Set-up . . . . .	51
3.6.2	Pose Accuracy and Uncertainty . . . . .	53
3.6.3	Performance with Dynamic Objects . . . . .	58
3.6.4	Execution times . . . . .	63
3.6.5	Localisation Using Ceiling Data from an Upward Looking Camera .	66
3.7	Conclusion . . . . .	68
<b>4</b>	<b>A Distance Function based Algorithm for Robotic Mapping</b>	<b>70</b>
4.1	Introduction . . . . .	70
4.2	Related Work . . . . .	71
4.3	Scan Matching Preliminaries . . . . .	76
4.4	CD Mapping: Chamfer Distance based Scan-to-Map Matching . . . . .	77
4.5	Grid-based ND-to-ND Scan-to-Map Matching . . . . .	79
4.6	Experimental Results . . . . .	82
4.6.1	Pose accuracy . . . . .	83
4.6.2	Mapping Real Environments . . . . .	87
4.7	Conclusion . . . . .	88
<b>5</b>	<b>Control of Assistive Robotic Robots</b>	<b>92</b>
5.1	Introduction . . . . .	92
5.2	Background . . . . .	93
5.3	User Intention Recognition . . . . .	96
5.3.1	Admittance Controller . . . . .	97
5.3.2	Classifying the Main Modes of Operation . . . . .	98
5.4	Experiments and Trials . . . . .	101
5.4.1	Comparing the Operation of the Smart Hoist vs. the Standard Hoist	102
5.4.2	User Trials . . . . .	104
5.5	Discussion . . . . .	109
5.6	Conclusion . . . . .	111
<b>6</b>	<b>Conclusions</b>	<b>113</b>

6.1	Summary of Contributions . . . . .	114
6.1.1	Distance Function based Environment Representation and Chamfer Distance based Sensor Model . . . . .	114
6.1.2	Distance transform based Mobile Robot Localisation . . . . .	114
6.1.3	Distance Transform based Robotic Mapping . . . . .	115
6.1.4	Control for Assistive Robots . . . . .	115
6.2	Discussion of Limitations and Future Work . . . . .	116

**Appendices** **118**

**A System Overview of the Assistive Robotic Smart Hoist** **118**

A.1	Hardware Structure . . . . .	118
A.1.1	Sensors . . . . .	118
A.1.1.1	Handles . . . . .	118
A.1.1.2	Cameras . . . . .	119
A.1.1.3	Boom Sensors . . . . .	119
A.1.2	Actuators . . . . .	120
A.1.3	The User Interface . . . . .	120
A.1.4	Hoist Computer . . . . .	122
A.1.5	Peripherals . . . . .	122
A.1.5.1	Hoist Controller . . . . .	122
A.1.5.2	Motor Controller . . . . .	122
A.1.6	Batteries and Uninterruptible Power Supply(UPS) . . . . .	122
A.2	System Assembly and Enclosure Design . . . . .	123
A.3	Measures Taken for Occupational Health and Safety . . . . .	124
A.3.1	Foot-guards . . . . .	124
A.3.2	Maximum Run Speed . . . . .	124
A.3.3	Braking . . . . .	125
A.3.4	Emergency Stop (EStop) . . . . .	125

**Bibliography** **126**



# List of Figures

1.1	The robots we used to collect real-time data. . . . .	4
2.1	Excerpt from the topological map of train transit network Sydney, Australia	10
2.2	Different metrics for generating distance functions in 3D space. . . . .	15
2.3	Shows (a) an example shape, (b) unsigned Euclidean distance variation and (d) the signed distance variation along the dotted line in (a). . . . .	15
2.4	Shows (a) a binary image and (b) its DF. . . . .	16
2.5	Shows a section from the binary OGM of the Intel research lab its distance function variation in grey levels. . . . .	16
2.6	Projection of the laser scan from an estimated robot pose . . . . .	20
2.7	Pinhole camera model showing the projection of a 2D shape on the image plane. . . . .	26
2.8	(a) shows an example template image, where (b) shows the template at an initial location on the distance function of the reference image. Chamfer distance value is 2.8 . . . . .	26
2.9	(a) is the colour map of variation of chamfer distance when the template is placed across every location on the grid, while (b) shows the template on the location with lowest chamfer distance. . . . .	27
2.10	A graffiti wall, with a camera observing a section of it. . . . .	27
2.11	Shows chamfer distance variation obtained by placing the template in Figure 2.10d across all the locations of the distance function obtained using the Figure 2.10b. . . . .	28
2.12	Chamfer distance variation at two different locations of the Intel research labs dataset . . . . .	29
2.13	Chamfer distance variation against robot orientation, in the vicinity of the true robot pose. . . . .	29
3.1	Trajectory of the robot in the Dataset 1 . . . . .	54
3.2	Error for each component of robot pose against ground-truth using the proposed EKF based algorithm for Dataset 1. . . . .	55
3.3	Error for each component of robot pose against ground-truth using the proposed optimisation based algorithm for Dataset 1. . . . .	56
3.4	Error for each component of robot pose against ground-truth using the proposed particle filter based algorithm for Dataset 1. . . . .	57
3.5	Trajectory of the robot in the Dataset 1. Results obtained by using standard AMCL, with beam based likelihood model. . . . .	57

3.6	Error for each component of robot pose against ground-truth using standard AMCL, with beam based likelihood model for Dataset 1. . . . .	58
3.7	Trajectory of the robot in the Dataset 2, with the EKF algorithm. . . . .	59
3.8	Trajectory of the robot in the Dataset 2, with the optimisation algorithm. . . . .	60
3.9	Trajectory of the robot in the Dataset 2, with the particle filter algorithm . . . . .	61
3.10	Trajectory of the robot in the Dataset 2, with standard AMCL, using beam based likelihood model . . . . .	62
3.11	Trajectory of the robot in the Dataset 3 . . . . .	63
3.12	A sparse illustration of crowd movement during collection of Dataset 3 over 29.54 minutes. . . . .	64
3.13	Shows the RMS errors of the proposed EKF based and optimisation based algorithms when input sensor measurements are artificially corrupted. . . . .	65
3.14	Per scan execution time for localisation algorithms. . . . .	65
3.15	Planar ceiling at the Centre for Autonomous Systems(CAS), University of Technology, Sydney(UTS) . . . . .	67
3.16	Shows the trajectory the robot takes with results obtained from ceiling image based localisation algorithm. . . . .	67
4.1	Map of the Intel Research Laboratories . . . . .	73
4.2	Flow chart for the CD mapping algorithm. . . . .	78
4.3	Two-dimensional grid space overlapped with the environment (left) and the normal distribution based on scan points in the $j^{\text{th}}$ cell (right) . . . . .	80
4.4	The grid map is represented by multiple normal distributions (right) and new scans to be matched to the grid map (left) . . . . .	80
4.5	Maps generated using the proposed CD mapping algorithm . . . . .	84
4.6	Presents results of three algorithms from the simulation run on Intel research labs map . . . . .	86
4.7	Errors for all poses for simulated Intel dataset . . . . .	87
4.8	Ground floor at Roselands Shopping Centre . . . . .	89
4.9	Long exposure time-lapse image showing movement of crowd during collection of data. . . . .	90
5.1	Standard Joey patient hoist from Haycomp Pty. Ltd. . . . .	95
5.2	UTS-IRT Smart Hoist. . . . .	96
5.3	A typical floor plan of a resident's room at the IRT Woonona care facility. . . . .	100
5.4	Mapping the direction of force $\alpha$ to the direction of motion. . . . .	101
5.5	Block diagram of the Controller. . . . .	101
5.6	Comparison of the forces required to do (a) Push, (b) Pull and (c) In-place turn on the standard and Smart Hoists. . . . .	103
5.7	Muscle groups assisting basic manoeuvres with the hoist . . . . .	105
5.8	Comparison of muscle action based on EMG signals during a push action. . . . .	106
5.9	Comparison of muscle action based on EMG signals during a pull action. . . . .	106
5.10	Comparison of muscle action based on EMG signals during an in place turn to the left. . . . .	107
5.11	User trials conducted at IRT Woonona, Australia. . . . .	107

---

5.12	Administering forces on the handles in the shown directions makes the hoist rotate around an instantaneous centre of rotation. . . . .	110
A.1	Modified handle bars of the Smart Hoist . . . . .	119
A.2	Strain gauge location on the boom for weight measurement. . . . .	120
A.3	Revolution 2 <sup>TM</sup> omni-directional crab drive motors from 221 Robotic Systems, and how they're retrofitted to the Smart Hoist. . . . .	121
A.4	User Interface of the Smart Hoist, the main screen. . . . .	121
A.5	Control pad/ joystick interface to control the Smart Hoist . . . . .	121
A.6	Motor controller (left) and the Hoist PC (right) assembly. . . . .	123
A.7	Batteries used in the Smart Hoist. . . . .	124

# Acronyms & Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
AMCL	Adaptive Monte-Carlo localisation
AR	Assistive robotics
BMI	Body Mass Index
C-LOG	A Chamfer Distance Based Method for Localisation in Occupancy Grid-maps
CAS	Centre for Autonomous Systems
CD	Chamfer distance
CPU	Central Processing Unit
DARPA	Defense Advanced Research Projects Agency
DF	Distance function
DT	Distance transform
EKF	Extended Kalman filter
EMG	Electromyography
EP	End Point
FEA	Finite Element Analysis

---

GP	Gaussian process
GPS	Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ICL	Iterative Closest Line
ICP	Iterative Closest Point
ICS	Iterative Closest Surface
IF	Information Filter
IRT	Illawarra Retirement Trust
KL	Kullback-Leibler
LiDAR	Light detection and ranging
LRF	Laser range finder
MCL	Monte-Carlo Localisation
ML	Maximum Likelihood
MVC	Maximum Voluntary Contraction
NEES	Normalised estimation error squared
NDT	Normal distribution transform
ND	Normal distribution
OG	Occupancy Grid
OGM	Occupancy Grid Map
PF	Particle filter
RBPF	Rao-Blackwellized particle filter

---

RGB	Red, Green, and Blue
RGBD	Red, Green, Blue, and Depth
RMS	Root-mean-square
ROS	Robot Operating System
SDF	Signed Distance Function
SLAM	Simultaneous Localisation And Mapping
SLS	Selective Laser Sintering
SVD	Singular Value Decomposition
TREC	Terrestrial Robotics and Control Laboratory
TSDF	Truncated Signed Distance Function
UCD	User Centred Design
UI	User Interface
UPS	Uninterruptible Power Supply
UTS	University of Technology, Sydney
VSDF	Volumetric Signed Distance Function
WSN	Wireless sensor networks

# Nomenclature

## General Notations

$t$	Time (continuous)
$k$	Time (discrete step)
$\square$	Denotes any variable
$\alpha$	Direction of linear force.
$C$	Coefficient of damping, linear motion.
$C_\omega$	Coefficient of damping, angular motion.
$C(\cdot, \cdot)$	A cost function.
$DF$	A distance function matrix obtained by transforming a binary image or an OGM
$d_{DF}$	Abbreviation for $DF(\mathbf{x}_o)$
$d_{CD}$	Abbreviation for $d_{CD}(\mathbf{z} \mid \mathbf{x}, DF)$
$\delta_{ij}$	Spatial relation between the pose at $i^{\text{th}}$ and $j^{\text{th}}$ locations.
$\epsilon_k$	NEES metric.
$\bar{\epsilon}_k$	Average-NEES metric.
$\varepsilon(\cdot)$	Error between pose relations (SLAM benchmarking metric).
$\bar{\varepsilon}(\cdot)$	Mean error between pose relations (SLAM benchmarking metric).
$\eta$	Normalisation constant.
$\eta_r$	Noise of a range measurement.
$F$	Linear force.
$F(\cdot)$	Control function.
$f_{ex}$	Feature extraction function.

---

$\gamma$	Threshold for association of NDs.
$h(\cdot, \cdot)$	Observation function.
$I$	Moment of inertia
$J_{\square}$	Jacobian matrix with respect to $\square$ .
$K$	Kalman gain
$m$	Mass.
$\mathbf{m}$	Map of the environment.
$\nabla F_{\square}$	Jacobian of the control function with respect to $\square$ .
$\nabla h_{\square}$	Jacobian of the observation function with respect to $\square$ .
$\mathcal{O}(\cdot)$	Big O notation.
$\omega$	Angular velocity.
$P_{\square \square}$	Covariance matrix of the state vector.
$\begin{matrix} \{B\} \\ \{A\} \end{matrix} \mathbf{P}$	Relative transform between two coordinate frames.
$\phi$	Orientation of the robot.
$Q_k$	Covariance matrix of control noise at time $k$ .
$r_i$	The $i^{\text{th}}$ range reading.
$S$	The space of all possible sensor readings.
$\Sigma_{\square}$	Covariance matrix.
$\sigma_r$	Standard deviation of range noise.
$\mathbf{s}$	$\mathbf{s} \in S$ , a single observation at a certain state of the robot, $\mathbf{x}$
$\tau$	Torque.
$\theta_i$	The bearing of the $i^{\text{th}}$ sensor reading.
$u_k$	Control command at time $k$ .
$v$	Linear velocity.
$\mathbf{x}$	Robot Pose vector in 2D space. Consists of the position components $x$ , $y$ and the orientation component $\phi$ .
$\mathbf{x}_o$	Observation coordinates. In the case of a laser range finder sensor, consists of $n$ readings of range $r_i$ and bearing $\theta_i$ . Can be translated into global Cartesian coordinates $x_{o_i}$ and $y_{o_i}$ using (2.8).
$Z$	Space of all possible observations.
$\mathbf{z}$	Observation vector, $\mathbf{z} \in Z$ .



**Coordinate Frames & Transforms**

$\{C\}$	Camera coordinate frame.
$\{G\}$	Global coordinate frame.
$\{I\}$	Image coordinate frame.
$\{R\}$	Robot coordinate frame.
$\begin{matrix} \{A\} \\ \{B\} \end{matrix} R$	Rotation matrix between the coordinate system $A$ to coordinate system $B$ .
$\begin{matrix} \{A\} \\ \{B\} \end{matrix} T$	Homogeneous transform matrix from the coordinate system $A$ to coordinate system $B$ .
$\begin{matrix} \{A\} \\ \{B\} \end{matrix} t$	Translation matrix between the coordinate system $A$ to coordinate system $B$ .

**Distributions**

$\mathcal{F}$	Folded normal distribution.
$\mathcal{N}$	Normal distribution.
$\mathcal{U}$	Uniform distribution.

**Operations**

$\dot{\square}$	The first derivative of $\square$ .
$\oplus$	The standard motion composition operator.
$\ominus$	Inverse of the standard motion composition operator.
$CD(\cdot, \cdot)$	Chamfer distance between a template and a reference.
$DF(\mathbf{x}_o)$	Value of the distance function at the observation coordinate $\mathbf{x}_o$ .
$R(\theta)$	2D rotation matrix for an orientation change of $\theta$ .
$\prod(\cdot)$	Product.
$\sum(\cdot)$	Summation.

**State Transitions**

- $\square_{k-1|k-1}$  Previous state.
- $\square_{k|k-1}$  Predicted current state.
- $\square_{k|k}$  Updated current state.

# Chapter 1

## Introduction

### 1.1 Motivation and Background

*Assistive robotics* is an emerging field with the vision of developing fully autonomous robots that are able to work collaboratively with a range of human users; as assistants, as tools, and as companions. These machines need to perceive and understand human behaviour and needs, communicate with users in a human-centred manner, and respond safely and efficiently to directions. Aged and disabled care sector is where such robots are expected to make an important impact.

Global ageing, the rapid increase in the global population over the age of 65, in developed and developing countries is one of the greatest social and economic challenges for our society [1, 2]. In 2010, an estimated 524 million people (eight percent of the world's population) were aged 65 or older. By 2050, this number is expected to nearly triple to 1.5 billion, which represents 16% of the world's population. In Australia, where this research was conducted, five years since June 2009, the number of people aged 65 years and over has increased by 20% to reach 3.5 million people in June 2014. This accounts for 15% of the total Australian population [3].

As people age, their capability in performing daily living activities can lessen due to cognitive and physical impairments. Demographic trends in Australia indicate a continued

decline in the relative availability of informal carers, coinciding with an increased demand for aged care services [4, 5]. Increases in the public costs of care [1, 2] for seniors is inevitable, given their greater longevity, the reduced number of informal carers, and increased community expectations.

The advances in robotics has given rise to increasing opportunities to implement assistive robotic devices to assist senior citizens in their daily living activities and carers in safely performing their duties without being overly exhausted [6]. These technologies are becoming crucial given the ageing population and the decrease in the number of working-age caregivers [7]. In general assistive robotic devices are studied in different categories such as wheelchairs and walkers [8, 9], devices for formal carers such as the humanoid patient transfer helper [10], and devices for families of the elderly people such as tele-presence and health monitoring robots [11].

Assistive robots that are designed for aged and disabled care sector are untethered and equipped with lightweight computers to minimise battery requirements and low-cost sensors to make them affordable. They navigate in a wide variety of environments, for example in private residences, hospitals, offices, and shopping malls, and sometimes even work in multiple environments on the same day. Furthermore, these environments are usually crowded. Therefore, the algorithms used for navigation of assistive robots should be robust and should not have environment dependent tuning parameters. With these constraints in computational resources and environment conditions in mind, the first part of this thesis addresses the problem of developing strategies for navigation for assistive robots.

Maintaining an internal representation of the environment, localising on that environment representation using the available sensors, and creating, updating, or expanding the available environment maps are considered the primary requirements for autonomous robot navigation. Methods that are commonly used for localisation or mapping has evolved over the years but were initially seeded in an era where the sensors mounted on robots that perceive the environment were highly limited in terms of resolution and field of view and produced highly uncertain readings, with low accuracy.

Particle filters, which are the go-to mechanism for localising a robot in environments represented using an Occupancy Grid Map (OGM), can estimate the pose of the robot without

resorting to extraction and association of features from the map and the environment. However, these filters sometimes need thousands of particles, each representing a robot pose hypothesis, which requires more computing power. Furthermore, existing algorithms usually contain environment dependant tuning parameters that need to be adjusted according to the map and the type of environment. Moreover, having dynamic objects such as people walking about can cause these algorithms to fail. As Assistive robots work closely with people in crowded environments, the localisation algorithm is expected to be robust to dynamic objects and have less to no tuning parameters.

With the aim of solving the navigation problem for assistive robots, we present a suite of algorithms that leverage the strengths of distance function representations, which can solve the three key requirements of, (i) efficiently representing the environment, (ii) localising the robot on that environment representation, and (iii) autonomously creating or updating the map efficiently and accurately, within a sound mathematical framework. The algorithms we present are robust to dynamic objects and can work out of the box without requiring any environment specific tuning parameters.

We use simulations, datasets from the public domain, and experimental datasets collected by us under real conditions using the robots shown in Figure 1.1 throughout this thesis in order to evaluate and verify the localisation and mapping algorithms we present. Even though we focused on developing computationally efficient algorithms as solutions to the problems encountered in assistive robots, these algorithms are generic and can be used with any autonomous robot as well.

While localisation and mapping are key requirements for helping assistive robots navigate autonomously, they must also be capable of interacting with human operators intuitively. Because the assistive robots may interact with people who are less technically savvy or differently capable, the control mechanism for such robots should be simple and easily adaptable. Furthermore, as the assistive robots replace a variety of other existing assistive devices, the mechanism of controlling is unique to the particular class of robots, and is difficult to generalise.

In the second part of this thesis, we present important steps to develop operator control methodologies for assistive robots. We demonstrate how the design process can be made



FIGURE 1.1: The robots we used to collect real-time data for experiments presented in this thesis. In sequential order: Turtlebot - modified, Walker revision 1, Walker revision 2, Motorised Walker, Smart Hoist

simple, productive, and realistic by using a Co-Design concept which includes the end user an integral member of the design team.

We use this approach to experimentally design and develop the Smart Hoist assistive robot to aid carers of a retirement facility to more easily transfer non-ambulatory residents. We show how our design of the robotic hoist and its control method was continually shaped by the end user feedback which made it intuitive and easy to use and can be readily integrated to the work-flow of a care facility. The findings from a series of user studies conducted to evaluate the control strategy and user interaction for the Smart Hoist is also reported. All of these experiments were conducted at an aged care facility with the assistance of professional carers. While demonstrating the usefulness of the hoist, this

example suggests the Co-Design framework as a solution to some ideologies surrounding the design of control interfaces for assistive robotics in general.

## 1.2 Thesis Outline

This thesis is organised as follows:

The second chapter presents a novel method for representing the environment that a robot operates in, together with a sensor model that can relate measurements from a sensor mounted on the robot to that environment representation. This representation is lightweight, and can be generated directly by observing the environment or converted using any existing occupancy grid map. The sensor model does not require defining explicit correspondence pairing of sensor readings and therefore is attractive for many applications.

In Chapter 3, we use the environment representation method and the sensor model presented in Chapter 2 to localise a robot using three different approaches, namely, an extended Kalman filter (EKF), an optimisation based technique, and a particle filter. While the particle filter approach is a mere enhancement to the sensor model in the popular adaptive Monte-Carlo localisation (AMCL) algorithm, the other two algorithms are novel, computationally efficient, and more accurate and have far less tuning parameters. The algorithms are proven to be effective by using multiple simulated and real time datasets both available in the public domain and collected by us in real environments.

Chapter 4 presents a method to generate high quality maps with the use of a scan-to-map matching approach, using the distance function based environment representation and the chamfer distance based sensor model we propose in Chapter 2. Although, the maps produced by the algorithm may not be statistically consistent as provided by Simultaneous Localisation And Mapping (SLAM) algorithms widely used in the community, it is able to generate accurate maps of considerably large crowded environments in real-time. As with the localisation algorithm, this mapping strategy only uses one tuning parameter and performs well even in clutter as demonstrated using simulated, public domain, and data collected in crowded environments.

Though maintaining a representation of the environment, localising and mapping are important necessities for any autonomous robot, understanding the user intention is a major requirement for assistive robots. Assistive robots replace a variety of other existing assistive devices. For each class of robots, the mechanism of human user interaction is unique and generalising the control strategy is therefore difficult. Using a case study as an example, Chapter 5 presents a framework based on Co-Design principles which can be used to design control strategies and user interfaces for assistive robots. We use the proposed framework to experimentally design and develop an assistive robotic hoist to aid carers in the aged and disabled care sector. User experience during trials, surveys, analysis of exerted force and Electromyography data involving carers at an aged care facility are used to evaluate the viability of the robotic hoist, which intern experimentally demonstrates the usefulness of the proposed design framework for designing control strategies for assistive robots.

Finally, we will present a summary of conclusions which would include a discussion of limitations of the work presented in this thesis, together with future work in Chapter 6.

### 1.3 Contributions

- A sensor model with theoretically sound uncertainty calculation for distance function based maps.
- An extended Kalman-filter based localisation algorithm for a mobile robot on a 2D occupancy grid map without the use of point or line feature extraction.
- An optimisation based computationally efficient algorithm for localisation of a mobile robot on a 2D occupancy grid map that does not require environment based tuning.
- A theoretically sound *likelihood model* using the distance function based environment representation to make laser range finder version of Adaptive Monte-Carlo localisation algorithm fast more theoretically sound.
- An efficient scan-to-map matching based mapping algorithm with very low error accumulation, that can be used in practical scenarios of importance to assistive robot navigation.



- A framework for designing and developing control strategies and user interfaces for assistive robots.

## 1.4 Publications Related to this Thesis

- Lakshitha Dantanarayana, Ravindra Ranasinghe, and Gamini Dissanayake. C-LOG: A chamber distance based method for localisation in occupancy grid-maps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 376–381. IEEE, 11 2013. doi: 10.1109/IROS.2013.6696379.
- Ravindra Ranasinghe, Lakshitha Dantanarayana, Antony Tran, Stefan Lie, Michael Behrens, and LiYang Liu. Smart hoist: An assistive robot to aid carers. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1285–1291. IEEE, 12 2014. doi: 10.1109/ICARCV.2014.7064501.
- Lakshitha Dantanarayana, Ravindra Ranasinghe, Antony Tran, Dikai Liu, and Gamini Dissanayake. A novel collaboratively designed robot to assist carers. In *The International Conference on Social Robotics*, pages 105–114. Springer, 2014. doi: 10.1007/978-3-319-11973-1.11.
- Tomonari Furukawa, Lakshitha Dantanarayana, Jason Ziglar, Ravindra Ranasinghe, and Gamini Dissanayake. Fast global scan matching for high-speed vehicle navigation. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 37–42. IEEE, 9 2015. doi: 10.1109/MFI.2015.7295742.
- Kunjin Ryu, Lakshitha Dantanarayana, Tomonari Furukawa, and Gamini Dissanayake. Grid-based scan-to-map matching for accurate 2d map building. *Advanced Robotics*, pages 1–10, 3 2016. doi: 10.1080/01691864.2015.1124025.
- Lakshitha Dantanarayana, Gamini Dissanayake, Ravindra Ranasinghe, and Tomonari Furukawa. An extended Kalman filter for localisation in occupancy grid maps. In *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, pages 419–424. IEEE, 12 2015. doi: 10.1109/ICIINFS.2015.7399048.

## Chapter 2

# Environment Representation for Robotics

### 2.1 Introduction

For autonomous robot navigation, a map of the environment together with a sensor model is an essential requirement. The sensor model correlates the observations from a sensor to the map which enables estimation of the location of the robot in the environment. Knowing the pose of the robot in-turn facilitates other functionalities that are required for the autonomous behaviour of the robot such as path-planning and exploration. The method used for representing the map is crucially dependent on the capabilities of the available sensors that observe the environment, as well as the tasks that the robot needs to accomplish. In this chapter, a distance function based method for describing the map of an environment and a strategy for building sensor models that relates the observations from sensors to that map is presented.

A map is essentially a spatial model that represents the physical environment in which a robot operates. Using sensors, the robot acquires information to learn the geometry of an environment. The environment map allows the robots to safely navigate in the environment, identifying objects and people in the environment while intelligently interacting with them. As mentioned earlier, the method used to represent the map depends on various

factors. Most importantly it needs to be tailored for the task which the map is required for. For example, for navigating in an uneven terrain, a three-dimensional (3D) description of the terrain in the vicinity of the robot is required, while for localising a robot in a two-dimensional (2D) environment, a representation that can easily relate measurements from a sensor to the associated map is adequate. The robot either needs to be provided with the map or needs to generate an appropriate map in order to facilitate navigation. The problem of autonomously generating the map of an environment is regarded as another important problem in the pursuit of building truly autonomous mobile robots.

Historically, there have been two major paradigms in building a map to represent the environment: topological and metric. A topological map represents the places and relationships with connectivity between those places. In topological maps, places (0D), paths (1D) and regions (2D) are symbolically described and are linked by relations such as connectivity, order and containment. Sometimes metric relationships such as distance and direction are also associated with topological maps. However, typically, there is no single global frame of reference present in topological maps. Furthermore, these can have different levels of abstraction depending on the usage and the critical information that the map needs to convey. Early work by Kuipers & Byun [12] and Mataric [13] are the most significant in topological mapping, followed by many more work such as [14–22]. Topological maps encode the type of information humans typically use for navigation. Figure 2.1 shows an excerpt from an abstracted topological map of the Sydney-Trains transit network in Sydney, Australia. These maps are more commonly used in applications such as way-finding [23] and navigation [24]. Although topological maps are useful in many situations, robot navigation in environments populated with obstacles and people requires a much more detailed representation that encodes the environment geometry.

Metric maps, captures the geometric properties of the environment. When a sensor that is able to capture properties of specific geometric primitives present in the environment such as points or lines is available, it is more appropriate to represent the map of the environment using the locations of these geometric elements. These are known as feature based maps where a feature is defined to be a geometric entity that can be recognised using sensor observations from different locations within the environment. Leonard et al. [26] presented an extended Kalman filter based algorithm for localising a robot in a



FIGURE 2.1: Excerpt from an abstracted topological map of the transit network in Sydney, Australia [25]

map consisting of point features. While feature based maps are suitable for localisation, when it comes to applications such as path planning, a more comprehensive representation of the environment indicating the free space available for travel is necessary.

The Occupancy Grid Map (OGM) algorithm is one of the earliest and commonly used metric map approaches. Initially proposed by Elfes [27, 28] and Moravec [29], OGMs represent the environment as a discrete state random field with evenly spaced cells (i.e. regular grid of cells). Depending on the probability of occupancy of each cell in the map, the value of each cell encodes its state into one of the three states: occupied, free, or unknown. These maps can be constructed and maintained via a Bayesian framework using information gathered from sensors mounted on a robot. Representations that divide the environment into cells of different sizes based on the extent of occupancy such as quad-tree [30] maps or oct-tree [31] maps can represent environments in a more efficient manner. Use of sets of polyhedra to describe the geometry of environments as proposed by Chatila & Laumond [32], is another efficient alternative to OGMs.

Many existing algorithms that are used for path planning, obstacle avoidance and exploration require maps to be represented as an OGM. Therefore, popular robotics frameworks such as Robot Operating System (ROS) [33] have made OGMs the de facto standard for

environment representation. For this reason, implementations of algorithms that do not use OGMs internally, e.g. [34–36], employ additional strategies to maintain an OGM.

Alternatively, distance function based map representations, where the function value at any location represents the distance to the closest occupied space in an environment, are increasingly becoming utilised to capture geometries of environments [37–41]. The distance function not only encodes the occupied regions of the environment, but also provides a continuous measure of the distance, making it a much richer representation in comparison to an OGM. In KinectFusion, Newcombe et al. [38] extends the representation method proposed by Curless & Levoy [37] that uses Signed Distance Functions (SDFs) to encapsulate the 3D surfaces that are incrementally constructed with the use of range readings from a depth sensor. In contrast to 3D OGMs, which do not have a clear notion of where the surfaces of an environment are or how surfaces can be extracted, the work by Carrillo et al. [40] makes it apparent that there is a clear mathematical strategy for extracting surfaces in environments that are represented by SDFs. Work by Mullen et al. [42] and Chazal et al. [43] uses unsigned distance functions for 3D surface reconstruction and they point out that unsigned distance functions are much more robust to noise and outliers than SDFs.

With a sensor that can acquire the distance to obstacles in its vicinity, it is easy for a robot to build the environmental maps using distance functions. In [44], Thrun, Burgard, & Fox have incorporated distance functions in the *likelihood field model* that they have proposed for robot localisation using a particle filter. However, this approach is rather empirical and does not capture the true characteristics of the distance function. One of the main contributions of this thesis is a systematic strategy for representing the environment using unsigned distance functions and subsequently using them for localisation and mapping.

It is also possible to use Gaussian process classifiers to learn continuous occupancy representation of the environment. The surface generated by the Gaussian process is a single continuous probabilistic spatio-temporal model of the environment which has the capability to overcome some of the limitations with conventional OGMs. Gaussian process based maps have been used in several areas of robotics such as dynamic environment mapping [45] path planning [46, 47], exploration [48], and is the preferred method for

Wi-Fi based mapping and localisation. O’Callaghan et al. [49, 50] adopts a Gaussian process based framework to construct continuous occupancy maps and shows that the maps are more accurate compared to conventional OGMs because of their ability to estimate properties in close-by areas that the sensor has not observed. Moreover, recent work by Kim & Kim [41] uses SDF in a Gaussian process framework they call *GPmaps* to construct surface meshes that are much more smooth and artefact free due to the surface reconstruction qualities built-in to Gaussian processes. Though this representation has multiple advantages and looks highly promising, the cubic computational complexity of Gaussian processes highly limits their real-time applications and ability to be used in small scale systems.

When such a diverse range of map types are available, selection of the appropriate representation predominantly depends on the application. A robot autonomously operating in an environment needs to use maps for a wide range of applications, including but not limited to, localisation, navigation tasks such as path planning, obstacle avoidance, defining goals, and exploration tasks. Furthermore, not all map types can be used in practical applications in real-time due to factors such as computational resource limitations.

In some scenarios, even though one type of map representation is good for localisation, a complex conversion process may be required for that map prior to using it with other applications such as path planning or obstacle avoidance. This may impede the use of such representation. Moreover, representation method may sometimes depend on the environment. For example, a method which depends on prominent features in the environment is bound to fail in a featureless environment. Therefore, the representation method should be chosen by considering the dominant factors such as the environment, the computation platform, and the application.

In this chapter, we propose the use of distance functions as the primary means to represent the environment. Compared to an OGM, distance function based representations encode more information about the environment in a simple data structure that can be easily inferred by a range of other algorithms. With range measurement sensors, distance functions seem to be a more natural way to represent the environment. We further propose a distance function based sensor model which will be the basis for the next two chapters

where we explore the problem of robot localisation and map building in a computationally efficient manner that is suitable for small robots such as assistive robots that have limited computational power.

This chapter is organised as follows: Section 2.2 introduces distance functions, types and variations of distance functions and its properties in detail, with methods of computation and transition from existing OGM based representations. We will formally introduce sensing and sensor models in Section 2.3, and will continue to present a sensor model that formulates sensor observation likelihoods and the associated uncertainties that can be used with distance function maps and 2D range sensors in Section 2.4, which is an original contribution of this thesis. Section 2.5 will conclude the chapter.

## 2.2 Distance Functions

For any given environment populated with objects, a distance transform or distance field is a representation (i.e. a map) of the environment where any point of the distance field holds the shortest distance to the closest boundary. Although naturally the Euclidean distance is used to measure this distance, other simple distance metrics such as City-block distance, chessboard distance, quasi-Euclidean distance or complex metrics such as Wasserstein metric that is used for 3D image reconstruction [43], are used as alternatives depending on the application and computational efficiency. Table 2.1 shows the definition of some simple distance metrics and Figure 2.2 illustrates equi-distant surfaces for each of those metrics.

When  $V$  is the set of occupied space in an environment, Euclidean distance function can be expressed by Equation (2.1) at any given point  $\mathbf{x}$  in space.

$$d_{DF} = DF(\mathbf{x}) = \min_{\mathbf{v}_j \in V} \|\mathbf{x} - \mathbf{v}_j\| \quad (2.1)$$

The most commonly used distance metric, the Euclidean distance measures the shortest distance as a scalar value. In this case distances are positive values and therefore the distance function is unsigned. However, the distance function values can be either positive

TABLE 2.1: Different metrics for distance functions in 2D space. These can be directly extended to 3D

Metric	Description
Euclidean	In 2D, the Euclidean distance between $(x_1, y_1)$ and $(x_2, y_2)$ is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
City-block	In 2D, the city-block distance between $(x_1, y_1)$ and $(x_2, y_2)$ is $ x_1 - x_2  +  y_1 - y_2 $
Chessboard	In 2D, the chessboard distance between $(x_1, y_1)$ and $(x_2, y_2)$ is $\max( x_1 - x_2 ,  y_1 - y_2 )$ .
Quasi-Euclidean	In 2D, the quasi-Euclidean distance between $(x_1, y_1)$ and $(x_2, y_2)$ is $ x_1 - x_2  + (\sqrt{2} - 1) y_1 - y_2 $ if $ x_1 - x_2  >  y_1 - y_2 $ , $(\sqrt{2} - 1) x_1 - x_2  +  y_1 - y_2 $ , otherwise

or negative when representing environments with closed shaped objects. In this variant, typically, the distance to a point inside the shape from the boundary is negative. This distance value increases as the query point approaches the boundary of the closed shape from inside. Then it takes positive values outside of the shape. Popular algorithms such as KinectFusion [38] and *GPmaps* [41] use SDF when representing 3D environments and objects. However, Mullen et al. [42] and Chazal et al. [43] demonstrate that unsigned distance functions are more robust to outliers and noise.

Figure 2.3a represents a simple 2D closed shape. Figures 2.3b and 2.3c represent the variation of unsigned and signed distance along the dotted line marked in 2.3a. Even though distance functions are continuous by definition, as seen in the figure, the derivatives of both signed and unsigned distance functions are discontinuous at the cut-loci [? ]; Cut-locus is the boundary at which the object corresponding to the minimum distance changes from one to another. In addition to that, in any distance function, the derivatives are also discontinuous at the surface of the object.

As the distance functions are continuous, computing the distance value at any point ideally should be done on demand. However, it is possible to pre-compute distance function values at discrete intervals in space by quantising the environment to pixels (2D environment) or voxels (3D environment). This is usually implied when using distance function to represent environments already represented using discrete maps such as binary images or binary OGMs.



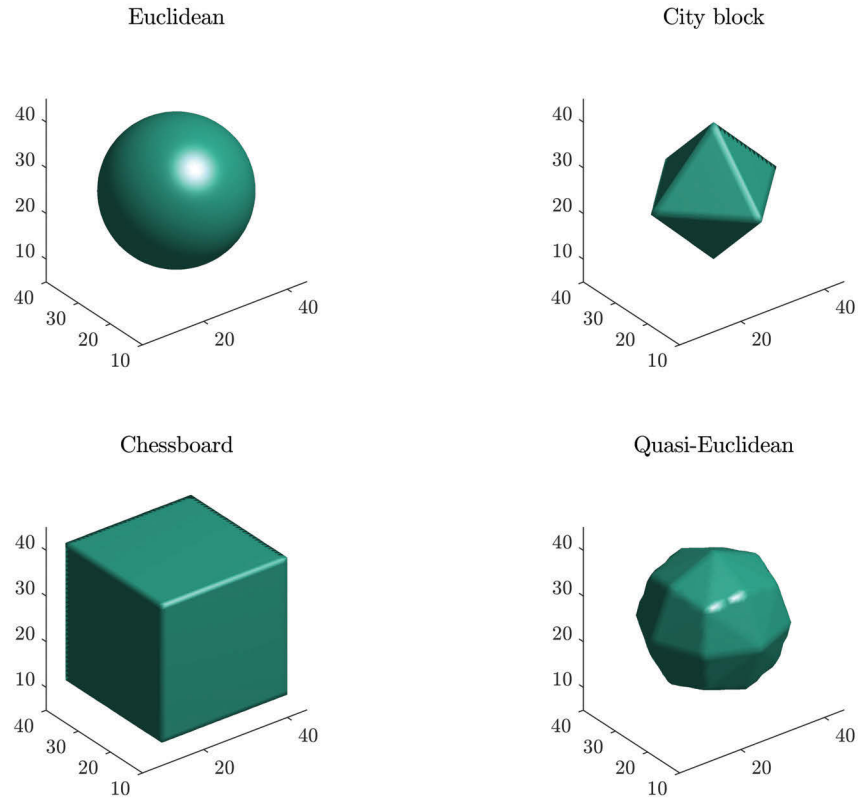


FIGURE 2.2: Different metrics for generating distance functions in 3D space. The figure shows equi-distance surfaces in distance function of a point feature in space. [51]

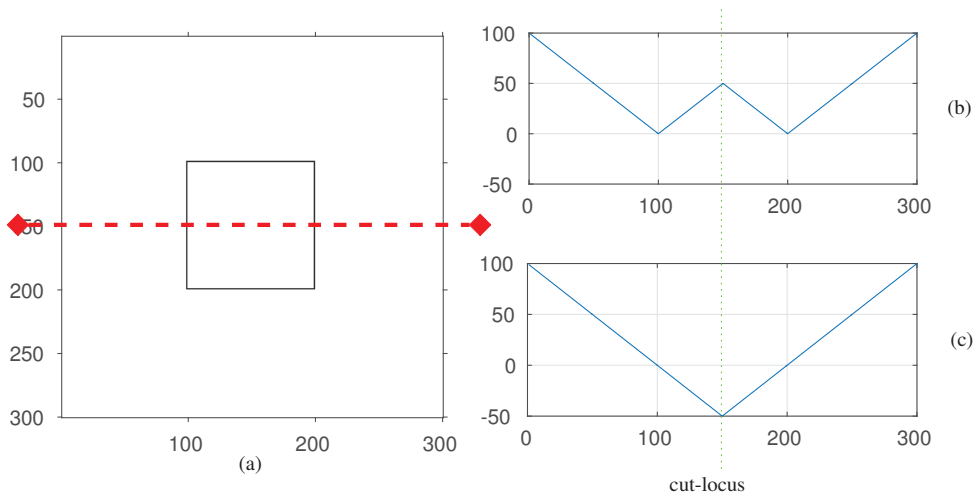


FIGURE 2.3: Shows (a) an example shape, (b) unsigned Euclidean distance variation and (d) the signed distance variation along the dotted line in (a).

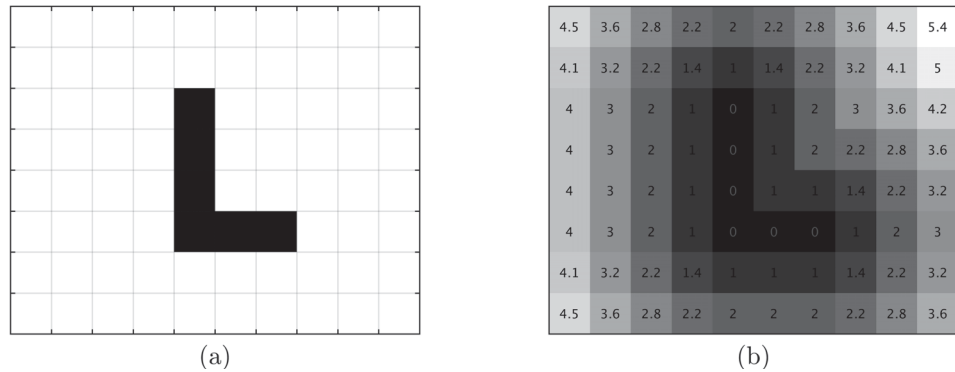


FIGURE 2.4: Shows (a) a binary image and (b) its DF.

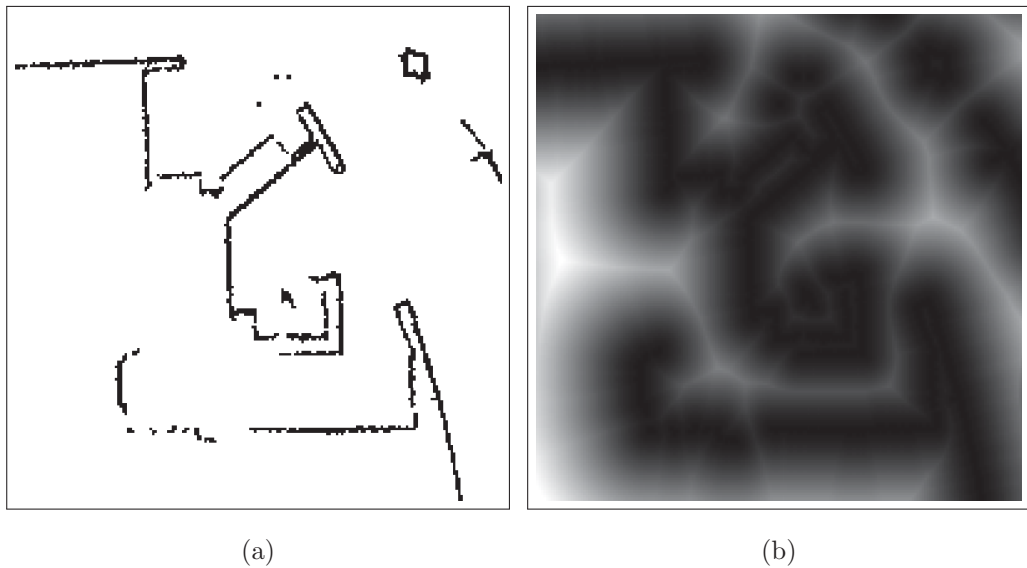


FIGURE 2.5: Shows (a) a section from the binary OGM of the Intel research lab (b) its distance function variation in grey levels.

The operator used to generate discrete distance functions is commonly known as a distance transform. When naïvely implemented, the distance transform process is an exhaustive search which depends on many factors including the resolution of quantisation. However, the algorithm proposed by Rosenfeld & Pfaltz [52] computes the distance transform efficiently by only two passes over any given 2D environment. Furthermore, there are numerous algorithms which are written for both CPU and GPU based systems that can do distance transforms in real-time [53–56]. Figure 2.4 represents a binary image and its distance function, the grey level of the image represents the distance value. Figure 2.5 shows a section from the binary OGM from the Intel research lab dataset [57], and its distance function.

## 2.3 Sensing and Sensor Models

A sensor is the device through which the robot “sees” the world. It measures a physical quantity and converts the measurement into a tractable signal for the robot to use. The sensors enable the robot to be aware of its environment and to perform tasks reliably and accurately. The information (measurements of physical quantities) gathered by the sensors is known as *sensor measurements* or *observations*. Sensors that are commonly used in robots include: (i) contact sensors such as bumpers and wheel encoders, (ii) inertial sensors such as accelerometers, gyroscopes and magnetometers (compasses), (iii) proximity sensors such as infra-red sensors, sonar sensors, radar, and laser range finder (LRF), (iv) visual sensors such as cameras and depth cameras, and (v) absolute sensors such as Global Positioning System (GPS) and visual tracking systems.

The selection of a sensor predominantly depends on the accuracy required by the task, suitability of a sensor for the operating environment of the robot and affordability. For example, even though GPS is suitable for outdoor navigation, it cannot be used in indoor environments where the satellite reception is poor. Even though sensors such as laser range finders or 3D Light detection and ranging (LiDAR) sensors have high accuracy and can be deployed in a wide range of environments, the high cost of these sensors limits their applications. However, it is worthwhile to note that, with the recent growth of robotics applications in the community, laser range finders with acceptable accuracy are now available at affordable prices. In practice, it is quite common to use multiple sensors on a robot as they can be used in a manner to complement each other to improve the overall accuracy.

The observations captured by a sensor are associated with a *sensor model*, which is an abstract representation of a physical sensor together with how the observations captured by a sensor are processed, interpreted and associated with the internal representation of the environment maintained by the robot.

A commonly used probabilistic sensor model in robotics can be described as follows: When  $S$  is the space of all possible sensor readings,  $\mathbf{s}$  ;  $\mathbf{s} \in S$  denotes a single observation from

a certain state of the robot  $\mathbf{x}$ , together they can be used to formulate the conditional density,

$$P(\mathbf{s} \mid \mathbf{x}) \tag{2.2}$$

which can be a formal probabilistic representation of the sensor model<sup>1</sup>.

With some sensors, for example cameras,  $P(\mathbf{s} \mid \mathbf{x})$  would be a high-dimensional density capable of determining the probability of every possible pixel of the camera image, which can make it difficult to compute meaningful estimates even if a full-blown representation of the environment is available.

However, it is possible to overcome this problem by reducing the dimensionality of the sensor observations by the use of appropriate filters,  $f_{ex}$ , to extract just a vector of readings  $\mathbf{z}$ . Then  $f_{ex} : S \rightarrow Z$  [58]. The conditional distribution given in Equation (2.2) can therefore be reduced to  $P(\mathbf{z} \mid \mathbf{x})$ , where  $\mathbf{z} \in Z$ . Furthermore, the state of the robot is usually referred to the robot's pose  $\mathbf{x}$  on a given map  $\mathbf{m}$ . Then this model can be further modified to,

$$P(\mathbf{z} \mid \mathbf{x}, \mathbf{m}) \tag{2.3}$$

The distribution in Equation (2.3) can be interpreted in two ways. First, if the state or the pose of the robot is known to be fixed at the value  $\mathbf{x} = \mathbf{x}$ , then the probability density function of the observation  $\mathbf{z}$  can be expressed as  $P(\mathbf{z} \mid \mathbf{x}, \mathbf{m})$ . Conversely, by making actual observations from a sensor,  $\mathbf{z}$  can be fixed at  $\mathbf{z} = \mathbf{z}$ , after which the state of the robot  $\mathbf{x}$  could be inferred with the distribution  $P(\mathbf{z} \mid \mathbf{x}, \mathbf{m})$ . This distribution is commonly known as a *Likelihood Function* [59].

When the environment is represented using geometric primitives, at an estimated pose  $\mathbf{x} = \mathbf{x}$ , the distribution in the first variant of the sensor model  $P(\mathbf{z} \mid \mathbf{x}, \mathbf{m})$ , can be used to predict the readings from the sensor as it observes the environment. Provided the

---

<sup>1</sup>If the measurement is time variant, the observation  $\mathbf{s}$  and the state  $\mathbf{x}$  is subscripted with time-step,  $k$ .

measurement noise is distributed as zero-mean Gaussian with a variance of  $R$ , the density can be shown as,

$$P(\mathbf{z} \mid \mathbf{x}, \mathbf{m}) \sim \mathcal{N}(\mathbf{z}, R) \quad (2.4)$$

This also forms a measurement model  $h(\cdot)$  in Equation (2.5) which is suitable for Kalman filter based or optimisation based frameworks which will be discussed in Section 3.3 in detail.

$$h(\mathbf{m}, \mathbf{x}) = \mathbf{z} + w \quad ; \text{ where measurement noise, } w \sim \mathcal{N}(0, R) \quad (2.5)$$

When the environment is represented as an OGM, features can be extracted from the map, after which it can be considered as a feature based map, and therefore use a measurement model as above [26]. However, a likelihood function based sensor model can exploit information from OGMs without resorting to feature extraction. The “*beam range finder model*” is a sensor model that is traditionally used with OGMs and range sensors. Ray-casting, which simulates a range reading on a given OGM at a given pose in the direction of a given bearing until an obstacle is reached can be used to obtain the “true” measurement which can be denoted as  $\mathbf{z}^*$  [44]. With the sensor observation at the estimated pose  $\mathbf{x} = \mathbf{x}$  available,  $\mathbf{z} = \mathbf{z}$ , then the likelihood model is,

$$P(\mathbf{z} \mid \mathbf{x}, \mathbf{m}) \sim \mathcal{N}(\mathbf{z}; \mathbf{z}^*, R) \quad (2.6)$$

This distribution can be used in a probabilistic framework such as particle filters (See Section 3.5 for details) to evaluate the fitness of the robot pose hypothesis  $\mathbf{x}$ .

### 2.3.1 A Sensor Model for Distance Function Representations

When the environment is represented using a distance function, the sensor model in Equation (2.3) can be rewritten as,

$$P(\mathbf{z} \mid \mathbf{x}, DF) \quad (2.7)$$

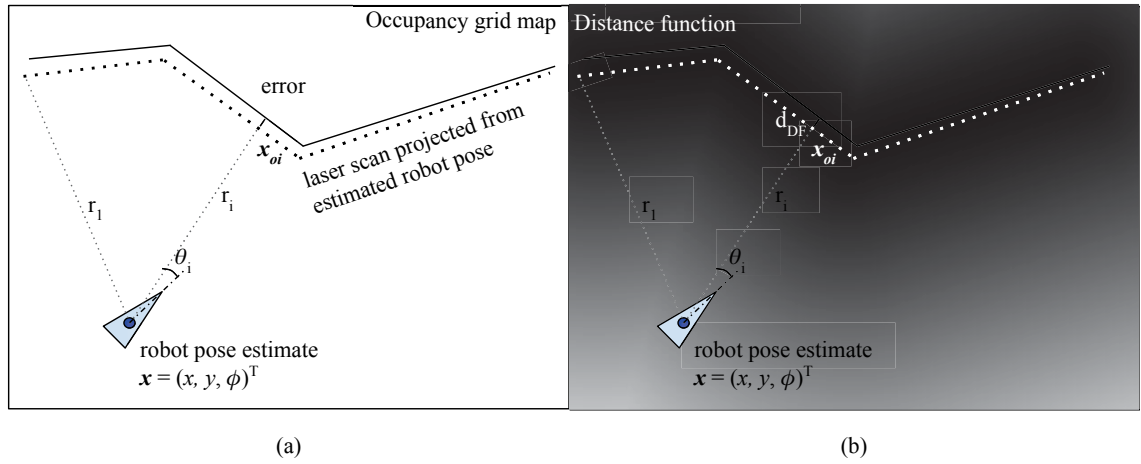


FIGURE 2.6: Projection of the laser scan from an estimated robot pose, (a) on the binary occupancy grid map and (b) on the distance function.

The “*likelihood range finder model*” or the End Point (EP) model proposed by Thrun et al. [44] is a sensor model that uses a distance function based environment representation with a laser range finder sensor. However, this is merely an empirical approximation of beam range finder model which even approximate the uncertainties to the sensor uncertainties by using an empirical mixture model introduced to cope with the high computational expense associated with the ray-casting process.

### 2.3.2 Projection of a Laser Scan on a Distance Function

The observation vector  $\mathbf{z}_i \in Z$ ;  $\mathbf{z}_i = (r_i, \theta_i)$ , of a single laser scan obtained by a robot equipped with a laser range finder, consisting of  $n$  range readings  $r_i$  at given bearings  $\theta_i$  can be projected from the robot pose initial guess  $\mathbf{x} = (x, y, \phi)^\top$ , using Equation (2.8) as shown in Figure 2.6a to obtain the observation vector in Cartesian space  $\mathbf{x}_{o_i}$ .

$$\mathbf{x}_{o_i} = \begin{Bmatrix} x_{o_i} \\ y_{o_i} \end{Bmatrix} = \begin{Bmatrix} x + r_i \cos(\theta_i + \phi) \\ y + r_i \sin(\theta_i + \phi) \end{Bmatrix} \quad (2.8)$$

By projecting the entire laser scan on the distance function, a vector of distance values can be extracted from the distance function at points  $\mathbf{x}_{o_i}$ .

$$\mathbf{d}_{DF} = DF(Z | \mathbf{x}) = \begin{bmatrix} DF(\mathbf{x}_{o_1}) \\ \cdot \\ \cdot \\ DF(\mathbf{x}_{o_i}) \\ \cdot \\ \cdot \\ DF(\mathbf{x}_{o_n}) \end{bmatrix} \quad (2.9)$$

Furthermore, the co-variance of this vector,  $\Sigma_{DF}$  is a diagonal matrix which can be written as,

$$\Sigma_{DF} = \text{diag}(\sigma_{DF, \mathbf{x}_{o_1}}^2, \dots, \sigma_{DF, \mathbf{x}_{o_i}}^2, \dots, \sigma_{DF, \mathbf{x}_{o_n}}^2) \quad (2.10)$$

where  $\sigma_{DF, \mathbf{x}_{o_i}}^2$  can be derived using Equation (2.11).

$$\sigma_{DF, \mathbf{x}_{o_i}}^2 = J_{DF, \mathbf{x}_{o_i}} R J_{DF, \mathbf{x}_{o_i}}^\top \quad (2.11)$$

where  $J_{DF, \mathbf{x}_{o_i}}$  is the Jacobian of the distance function at the query points  $\mathbf{x}_{o_i}$  and the sensor noise co-variance matrix  $R$ .

Moreover,  $J_{DF, \mathbf{x}_{o_i}} = \frac{\partial d_{DF}}{\partial r}$ , provided that the only contributing factor to sensor noise  $R$  is the laser range noise  $\sigma_r^2$ .

$$J_{DF, \mathbf{x}_{o_i}} = \frac{\partial d_{DF}}{\partial r} = \left. \frac{\partial d_{DF}}{\partial x_o} \right|_{\mathbf{x}_{o_i}} \cdot \frac{\partial x_o}{\partial r} + \left. \frac{\partial d_{DF}}{\partial y_o} \right|_{\mathbf{x}_{o_i}} \cdot \frac{\partial y_o}{\partial r} \quad (2.12)$$

Therefore, we can express the distance function based sensor model for laser range finder sensors as,

$$P(\mathbf{z} | \mathbf{x}, DF) \sim \mathcal{F}(d_{DF}, \sigma_{DF}^2) \quad (2.13)$$

When the distance function is unsigned,  $\mathcal{F}$  is a special Gaussian distribution which is known as a folded normal distribution [60, 61], and  $\mathcal{F}$  is a normal distribution for signed

distance functions. The values,  $\frac{\partial d_{DF}}{\partial x_o}$  and  $\frac{\partial d_{DF}}{\partial y_o}$ , can be precomputed by using the distance function of the map to increase efficiency.

The vector  $\mathbf{z}$  in Equation (2.13) is a measure of disparity between a map and a sensor measurement. A scalar measure of disparity has clear computational advantages in the process of computing the measurement likelihood. In the context of image, computer vision literature is abundant with scalar measures of disparity between distance functions and binary images.

Chamfer distance is one of many such distance metrics available that does not require defining explicit corresponding point pairings. Hausdorff distance [62, 63], another popular method that is used in many applications, captures one point which has the worst mismatch from a set of points as opposed to chamfer distance which captures average mismatch of all given points. First introduced by Burrow et al. [64] in 1977, chamfer distance based template matching has gone through many implementations, improvements and value additions over the years which includes making it robust in rotation (i.e. minor orientation changes) [65], scale changes [66], resolution changes, and even robust in high clutter [67].

In the next section, we will formally introduce chamfer distance, and use it to formulate a sensor model that summarises Equation (2.13) into a single measure of disparity.

## 2.4 Chamfer Distance

In computer vision literature chamfer distance is defined and used for template matching with binary images, where a semblance of the binary query shape is located within a larger reference image. Let  $U = \{\mathbf{u}_i\}$  and  $V = \{\mathbf{v}_j\}$  be sets of query and reference images respectively. The chamfer distance between  $U$  and  $V$  is given by the average of distances between each point  $\mathbf{u}_i \in U$ ,  $n(U) = n$  and its nearest edge in  $V$ ,

$$d_{CD} = CD(U, V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j| \quad (2.14)$$

Here  $n$  is the number of points in  $U$ .



With the use of a distance function, it is possible to reduce the cost function (2.14) to (2.15) so that it can be evaluated in linear time,  $\mathcal{O}(n)$  [67].

$$d_{CD} = CD(U, V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} DF(\mathbf{u}_i) \quad (2.15)$$

The chamfer distance is a sum of positive distances and is defined for unsigned distance functions.

In the case of 2D template matching using chamfer distance, the reference image and the template are both binary edge images which can be obtained using an edge filter on the original images. Furthermore, the highest computational complexity lies on the distance transform process to create the distance function from the reference edge image which should be done for every image frame. However, as discussed before, recent high-speed implementations of distance transform enable faster execution and has even made it possible to use chamfer distance for people recognition and tracking on surveillance footage in real-time [56]. It should be noted that when distance function represents a static environment map, its calculation is a one off process.

#### 2.4.1 Chamfer Distance based Sensor Model for Distance Function based Environment Representations

With the above preliminaries, it becomes apparent that chamfer distance serves as a function that can map a high-dimensional sensor observation  $S$  from a camera to a low-dimensional observation  $d_{CD}$ ,  $CD : S \rightarrow d_{CD}$ . With this we can derive the sensor model specified in Equation (2.16). The probability density for chamfer distance, as with the unsigned distance function, follows a special Gaussian distribution known as the folded normal distribution ( $\mathcal{F}$ ).

$$P(\mathbf{s} \mid \mathbf{x}, DF) \sim \mathcal{F}(d_{CD}, \Sigma_{CD}) \quad (2.16)$$

The co-variance of chamfer distance,  $\Sigma_{CD}$  can be obtained by transforming the sensor noise  $R$  as,

$$\Sigma_{CD} = J_{CD} R J_{CD}^{\top} \quad (2.17)$$

The Jacobian,  $J_{CD} = \frac{1}{n} \sum \frac{\partial d_{CD}}{\partial \mathbf{s}} \Big|_{\mathbf{x}_s}$ , can be calculated by linearising the distance function at the appropriate points.

It is also interesting to note that data association, which is a crucial step in the process of obtaining the measurement likelihood, does not need to be explicitly considered in case of chamfer distance. Data association is defined as the process of associating uncertain measurements to known tracks or landmarks [68]. After extracting the feature vector  $\mathbf{z}$  from a sensor measurement, the correspondence of those features to the existing landmarks in the map should be explicitly defined to relate the predictions from the sensor model to the actual measurements. This is a challenging task and therefore is a highly researched problem as any incorrect association could cause failure of the algorithm that uses the measurements. As chamfer distance represents the entire laser scan as a single cluster of readings, its value at any given robot pose  $x$  represents a measure of the placement of the whole scan, not just any individual feature in the laser scan, and therefore data association is implicit.

## 2.4.2 Chamfer Distance Based Sensor Model for a Camera

For camera images the sensor observation matrix  $\mathbf{s}$  is obtained by pre-processing the image to obtain an edge template; further, the appropriate image projective transform would produce the location  $\mathbf{x}_s$ .

Consider the example shown in Figure 2.7. If a simple pinhole camera observes one element of a shape in a 2D scene  $\{^C\}\mathbf{x}_f = (x_f, y_f, z_f)^{\top}$ , its projection on the image plane, in the image coordinate frame<sup>2</sup>  $\{^I\}\mathbf{x}_s = (x_s, y_s)^{\top}$ , has the relationship shown in Equation (2.18), which is commonly known as the projective transform [69]. Here  $f$  is the focal length

<sup>2</sup>Coordinate frames are denoted with  $\{ \cdot \}$ . Global frame is  $\{G\}$ , camera frame is  $\{C\}$ , the image frame is  $\{I\}$ , and the global image frame is  $\{GI\}$

of the camera. It has been further simplified considering that the camera is observing a coplanar scene.

$$\begin{Bmatrix} I \end{Bmatrix} \mathbf{x}_s = \begin{pmatrix} x_s \\ y_s \\ f_s \end{pmatrix} = \frac{f}{z_f} \begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix} \implies \frac{f}{z_f} \begin{pmatrix} x_f \\ y_f \end{pmatrix} = \frac{f}{z_f} \cdot \begin{Bmatrix} C \end{Bmatrix} \mathbf{x}_f \quad (2.18)$$

This can be translated to the image global coordinate frame by defining the transform matrix  $\begin{Bmatrix} GI \\ I \end{Bmatrix} T$  with the use of the projection of the camera location on the 2D scene  $(x, y)^\top$ .

$$\begin{Bmatrix} GI \\ I \end{Bmatrix} T = \frac{f}{z_f} \begin{pmatrix} [R(\theta)] & x \\ & y \\ 0 & 0 & 1 \end{pmatrix} \quad (2.19)$$

where  $R(\theta)$  is the rotation matrix of the camera for a yaw of  $\theta$ , where,

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Then the location of the element in global image coordinate frame is given by,

$$\mathbf{x}_s \text{ or } \begin{Bmatrix} GI \end{Bmatrix} \mathbf{x}_s = \begin{Bmatrix} GI \\ I \end{Bmatrix} T \cdot \begin{pmatrix} \begin{Bmatrix} I \end{Bmatrix} \mathbf{x}_s \\ 1 \end{pmatrix} \quad (2.20)$$

With the help of the transform given in Equation (2.20), the Jacobians in Equation (2.17) can be expressed in terms of the distance function of the reference image,  $\frac{\partial d_{DF}}{\partial \mathbf{s}}$  using the chain rule similar to Equation (2.12).

### 2.4.3 Image Template Matching Using Chamfer Distance

Figure 2.8 shows a simple example of a sample template on the distance function of the reference image shown in Figure 2.4. The lowest chamfer distance is usually identified by

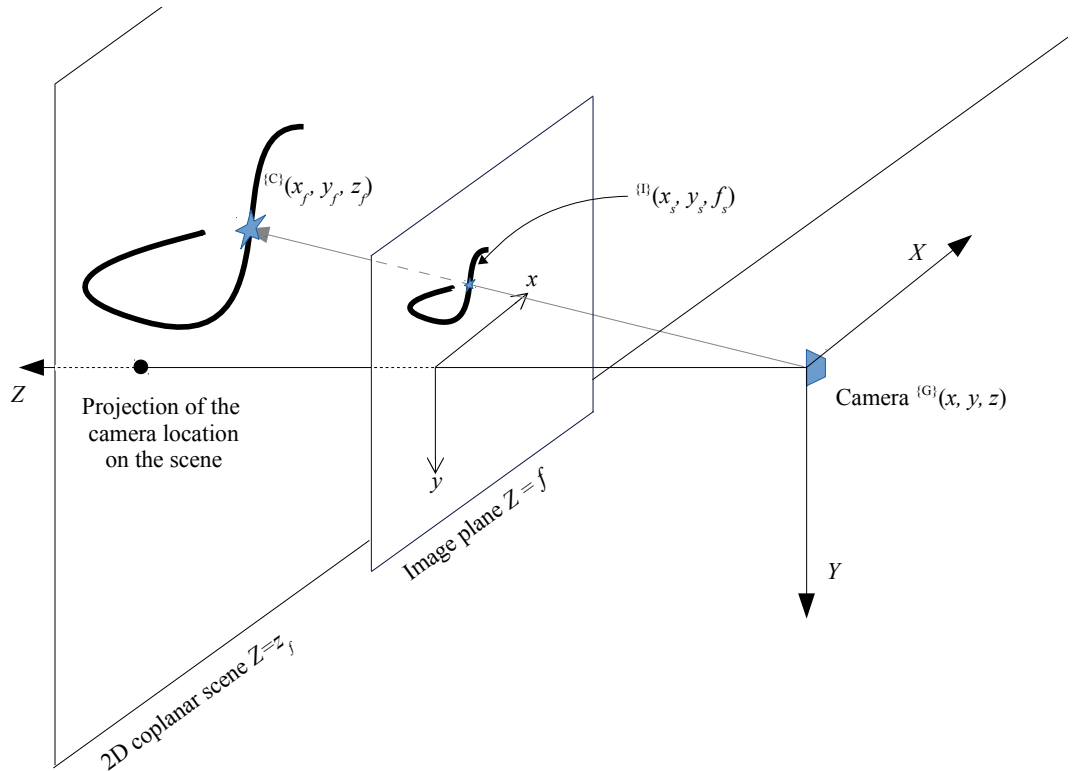


FIGURE 2.7: Pinhole camera model showing the projection of a 2D shape on the image plane.

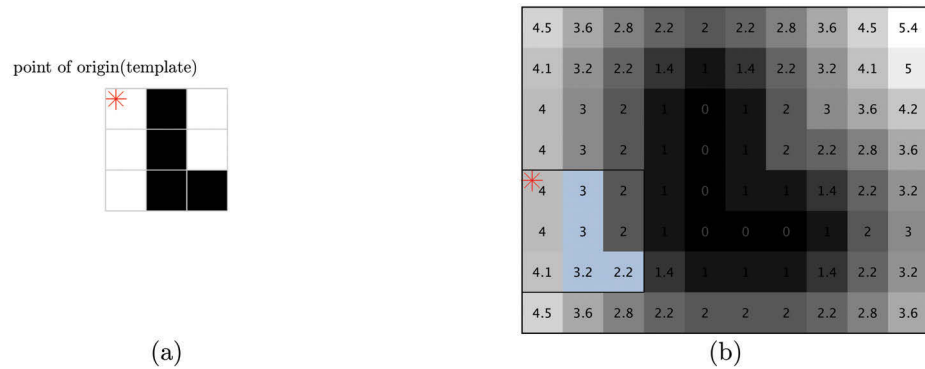


FIGURE 2.8: (a) shows an example template image, where (b) shows the template at an initial location on the distance function of the reference image. Chamfer distance value is 2.8

using a grid search [67].

With Figure 2.9, we can clearly see that the chamfer distance reaches a minimum when the template overlaps with the reference image. In the ideal scenario, the chamfer distance would reach zero at the perfect match.

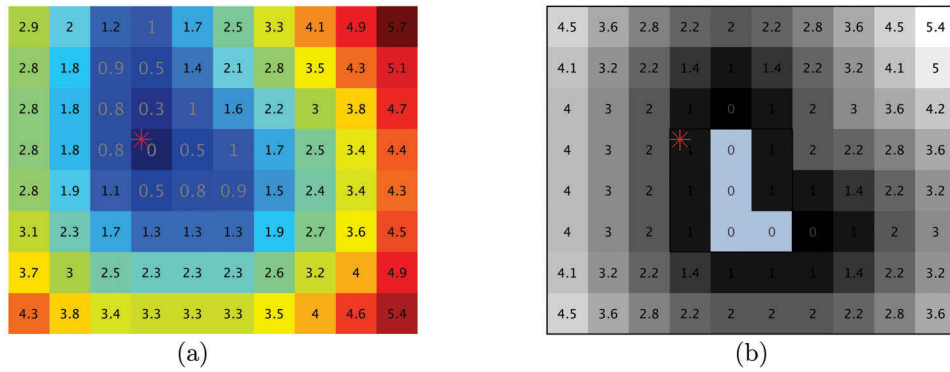


FIGURE 2.9: (a) is the colour map of variation of chamfer distance when the template is placed across every location on the grid, while (b) shows the template on the location with lowest chamfer distance.

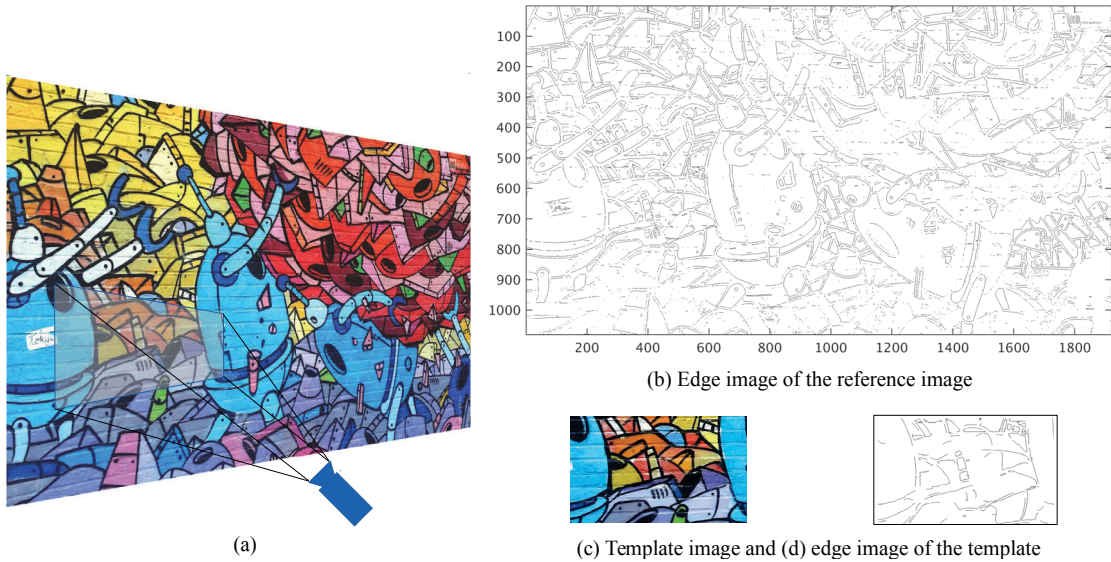


FIGURE 2.10: The camera observes a section of a graffiti wall. Figures (c) and (d) show the template extracted from the camera image to be matched with the total image of the wall, (b). (Image courtesy of paper4pc.com)

Template matching essentially localises a smaller template image on a larger image which is usually obtained using a camera. However, if a camera just observes a part of a larger image as shown in Figure 2.10, provided the edge image of the template has enough unique shapes, and the perspective transformation on the image with the camera is known, the template captured by the camera can be localised in the larger image, effectively localising the camera location provided the motion of the camera is only on a plane parallel to the scene plane and therefore  $f/z_f$  is a constant.

Figure 2.11 shows chamfer distance variation obtained by placing the template in Figure

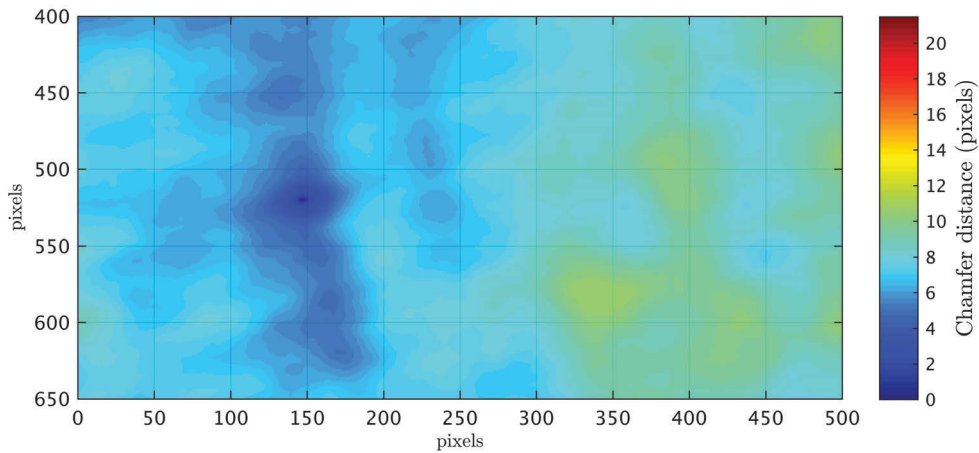


FIGURE 2.11: Shows chamfer distance variation obtained by placing the template in Figure 2.10d across all the locations of the distance function obtained using the Figure 2.10b.

2.10d across locations on the distance function of Figure 2.10b, closer to the vicinity of the true camera pose. We can observe that chamfer distance has a minimum at  $(147, 520)$ , which could be translated to the camera pose.

A similar attempt at localising a robot using an upward looking camera with a mosaic map of the ceiling is discussed in Section 3.6.5. In this case, the larger mosaic of the total ceiling is created using previously collected ceiling images at known poses. The camera only observes a part of the planar ceiling. The chamfer distance based sensor model is used to match the camera observation with the mosaic to obtain the pose of the camera, from which the location of the robot is deduced.

#### 2.4.4 Chamfer Distance Based Sensor Model for a Laser Range Finder

As discussed in Section 2.3.1, when distance function represents a 2D environment and the sensor observing the environment is a laser range finder sensor, the chamfer distance is still a good likelihood measure of disparity between the map and the sensor reading that can be used to evaluate the robot pose hypothesis.

Using Equation (2.15), the chamfer distance for a laser scan can be written as shown in Equation (2.21).

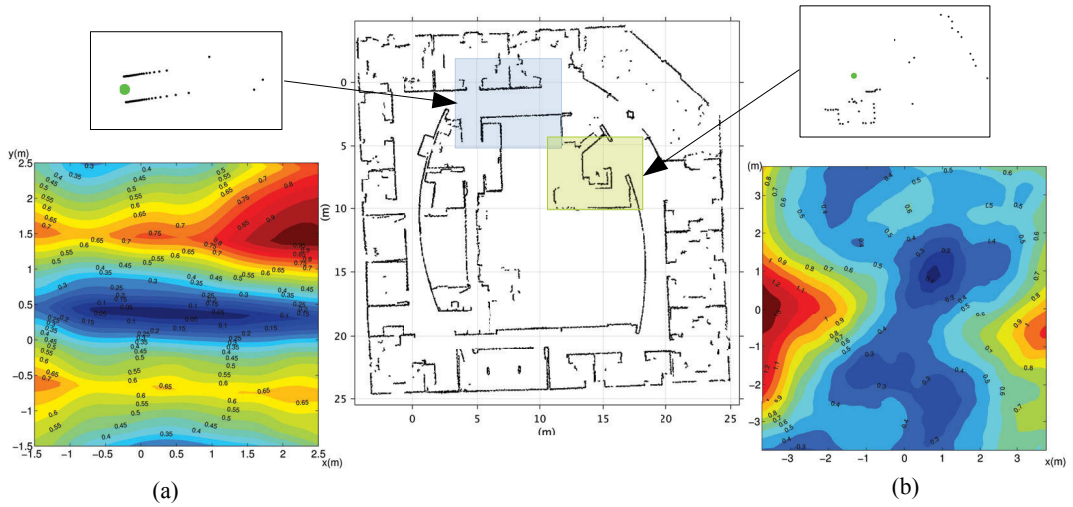


FIGURE 2.12: Shows chamfer distance variation against robot location,  $(x, y)$  at two different locations of the Intel research labs dataset, in the vicinity of the true robot pose. The corresponding laser scan (not to scale) is given above the contour-plots.  $\phi$  is set to its true value.

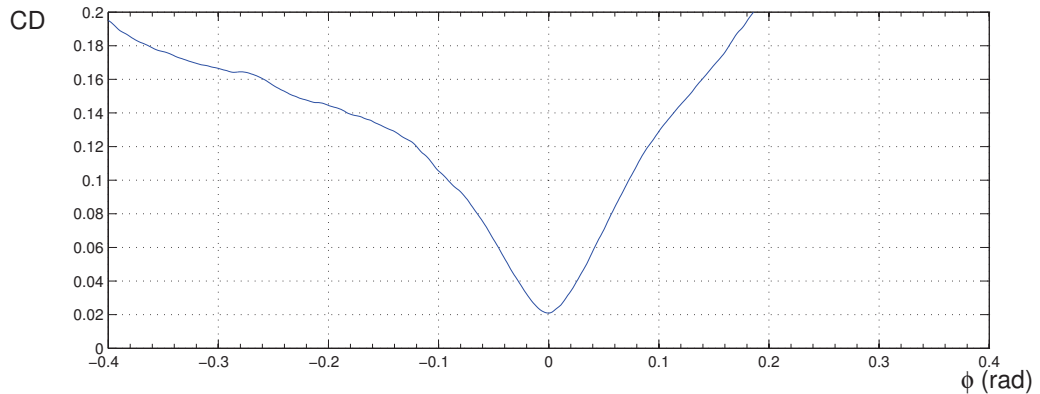


FIGURE 2.13: Chamfer distance variation in the vicinity of the true robot pose  $x$  and  $y$  at their true values and orientation  $\phi$  varied.

$$d_{CD} = CD(\mathbf{z} \mid \mathbf{x}, DF) = \frac{1}{n} \sum_{i=1}^n DF(\mathbf{x}_{o_i}). \quad (2.21)$$

where  $DF$  is the unsigned distance function of the environment map.

The sections highlighted in the contour map Figure 2.12, Figure 2.12a and Figure 2.12b present the variation of chamfer distance relative to a hypothesised robot location  $(x, y)$  that varies in the vicinity of the true pose. The approximate coordinates of the true

pose for Figure 2.12a and Figure 2.12b are (1.1, 1.1)m and (0.45, 0.45)m respectively. Assuming that there is no measurement noise, the minimum chamfer distance, which will be equal to zero, is obtained when the robot is placed at its true pose in the map and the laser scan is perfectly aligned.

Figure 2.13 shows the variation of chamfer distance when  $x$  and  $y$  are kept at their true values and the orientation  $\phi$  is varied between  $\pm 0.4$  radians for the robot's true location used in Figure 2.12b.

Furthermore, the chamfer distance distribution is differentiable with respect to the robot pose  $\mathbf{x}$ , and the partial derivatives of chamfer distance can be deduced with the use of partial derivatives of  $DF$  as shown in Equation (2.22).

$$\nabla d_{CD} = \left( \frac{\partial d_{CD}}{\partial x} \quad \frac{\partial d_{CD}}{\partial y} \quad \frac{\partial d_{CD}}{\partial \phi} \right)^\top$$

$$\begin{pmatrix} \frac{\partial d_{CD}}{\partial x} \\ \frac{\partial d_{CD}}{\partial y} \\ \frac{\partial d_{CD}}{\partial \phi} \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \sum \frac{\partial DF}{\partial x_o} |_{\mathbf{x}_{o_i}} \cdot \frac{\partial x_o}{\partial x} \\ \frac{1}{n} \sum \frac{\partial DF}{\partial y_o} |_{\mathbf{x}_{o_i}} \cdot \frac{\partial y_{o_i}}{\partial y} \\ \frac{1}{n} \left( \sum \frac{\partial DF}{\partial x_o} |_{\mathbf{x}_{o_i}} \cdot \frac{\partial x_o}{\partial \phi} |_{r_{o_i}} + \sum \frac{\partial DF}{\partial y_{o_i}} |_{\mathbf{x}_{o_i}} \cdot \frac{\partial y_o}{\partial \phi} |_{r_{o_i}} \right) \end{pmatrix} \quad (2.22)$$

As with  $DF$ , the partial derivatives  $\frac{\partial DF}{\partial x_o} |_{\mathbf{x}_{o_i}}$  and  $\frac{\partial DF}{\partial y_o} |_{\mathbf{x}_{o_i}}$  can also be pre-computed and stored to be used by other algorithms.

However, as a distance transform is used to obtain the distance function from an OGM, the distance field is no longer continuous. A standard cubic spline can be used to interpolate the distance function values to estimate distances and the derivatives in continuous space. As an added advantage, the spline function would smooth-out the cut-loci and the occupancy boundaries making the derivatives continuous throughout the distance function. Apart from splines, Gaussian processes have also been suggested as smoothing functions for distance function [41], but as previously mentioned they are highly computationally complex.



## 2.5 Conclusion

Distance function based representations are quite versatile and they are popularly used to represent both 2D and 3D environments. In 3D, distance function based representations typically deal with depth based distance measurement sensors, but traditionally, 2D distance functions deal with image sensors such as cameras. In this chapter, we proposed the use of distance functions to represent 2D environments, with a sensor model to relate the observations from a laser range finder sensor to the environment. As the sensor readily measures the distances from the sensor to the objects in the environment, representing the environment as distances to the closest object leads to a simpler sensor model. Important measures such as the sensor uncertainty can be easily translated to the distance function uncertainty.

We also propose to use the chamfer distance as the likelihood measure of disparity between the map and sensor readings (range and bearing measurements) from the laser range finder sensor. Application of chamfer distance drastically reduces the dimensionality in the observation vector which leads to a computationally light-weight implementation of the algorithm that can be used in resource constraint applications such as assistive robots.

The next chapter, demonstrates how the distance function based environment representation method and the sensor model proposed in this section can be used to efficiently localise a robot on an OGM.

## Chapter 3

# Distance Function based Algorithms for Mobile Robot Localisation

### 3.1 Introduction

Determining the position and orientation (pose) of a robot with respect to a given map of the environment is known as localisation. Localisation is one of the prime requirements for a robot to be fully autonomous. Functions that an autonomous robot is expected to perform such as path planning and navigation require the location information. Along with the mapping problem, the robot localisation problem has also been well studied over the last two decades. The methodologies used for robot localisation differ depending on various circumstances, from which, the composition of the environment, sensors that the robot uses to observe the environment, and the type of map available to the robot are the most influential. For example, if the robot is an unmanned aerial vehicle moving through an uneven terrain, the localisation method should be able to track the robot in 3D space with six degrees of freedom. However, if the robot is a wheeled robot that operates in a planar environment, a 2D localisation method that provides the robot position and heading would be adequate.

Sensors mounted on the robot are the primary means for a robot to observe the environment. Application, the environment of operation, the required accuracy, affordability of the sensor, are some factors that profoundly impact on the selection of the appropriate sensors for a robot.

For a robot operating in an indoor Global Positioning System (GPS) denied environment, which does not have access to position tracking systems, 2D laser range finders are one of the go-to sensors for localisation. A laser range finder observes the environment by measuring radial distances to objects at discrete intervals on a single plane. Due to its higher resolution, the larger field of view and most importantly the higher accuracy, it has become the widely accepted sensor for most robotics platforms. For a long time, laser sensors have been unaffordable and therefore sparingly used for mass produced robots such as service robots. However, recent popularity in hobby robots has given rise to much more affordable laser range finders with the same accuracy as the expensive sensors. Other sensors such as cameras and RGB-D sensors are also capable of being used for localisation, but factors such as limitations of field of view and accuracy, and higher complexities associated with pre-processing sensor observations make them less attractive even though they are more affordable compared to laser range finders.

When a sensor makes an observation (also known as a measurement), a sensor model, which is essentially a relationship between the observation, the state of the robot, and the map of that environment, is defined.

Once the sensor model is formulated, a localisation algorithm that complements the characteristics of the sensor model is able to resolve the robot's location on the map using the sensor observations. The back-end solver employed by the algorithm is the primary means of classification for localisation algorithms. Particle filters, extended Kalman filters, or optimisation based approaches are some of the more common approaches used as localisation back-ends. Most of the time these algorithms are specific to the sensor model.

Moreover, depending on the construction of the environment, a single observation from a sensor might not provide enough information to localise the robot accurately. In such scenarios algorithms which have the capability to fuse additional information such as measurements from other sensors, knowledge of the approximate location or integrate

temporal information from the primary sensor might be able to achieve proper localisation. As such, each back-end comes with its own set of advantages and limitations, and should be chosen to match the application.

In this chapter, we use the distance function based environment representation and the associated sensor model presented in Chapter 2 to localise a mobile robot in an indoor environment. We demonstrate the flexibility of the sensor model we propose by applying it to three popular localisation strategies, namely a Kalman filter, a non-linear optimisation based method, and a particle filter based approach to localise a mobile robot equipped with a laser range finder sensor that operates in a 2D planar environment. This chapter also presents the formulation of uncertainties related to the sensor model which is an original contribution of this thesis. This work is the first extended Kalman filter formulation that does not require explicit feature extraction or association to localise a mobile robot in an Occupancy Grid Map (OGM). We use simulations, public domain, and experimental datasets we collected in real crowded environments to evaluate these algorithms.

In order to further highlight the flexibility of the proposed sensor model, we also demonstrate a strategy to localise a robot using observations from an upward looking RGB camera. Here, the chamfer distance based sensor model with a distance function of the ceiling mosaic is used.

The rest of this chapter is organised as follows. Section 3.2 presents related work conducted in the field of robot localisation while Sections 3.3 to 3.5 presents the three strategies of localisation. The Section 3.6 presents experiments conducted using multiple datasets, while a discussion and a conclusion is presented in Section 3.7.

## **3.2 Related Work**

Robot localisation is a well-studied problem with solutions available for many combinations of sensors and map types. When the map is represented using geometric primitives such as points or line segments, extended Kalman filter (EKF) based algorithms are capable of efficiently estimating the robot pose within the map by fusing information gathered from robot odometry and observations associated with geometric beacons in the environment

[26]. EKF based methods are generally computationally efficient but require an initial guess of the prior location of the robot. Therefore, such methods are incapable of solving problems such as the kidnapped robot problem <sup>1</sup>.

When the environment map is available in the form of an OGM, particle filter based approaches have been the preferred method [70] for robot localisation due to their ability to exploit all the measurements available in a range scan. Furthermore, they are relatively easy to implement and are capable of global localisation: the ability to deal with the situation when a suitable initial estimate for the robot pose is unavailable. The widely used adaptive Monte-Carlo localisation (AMCL) [44, 71–75], that is also available as a part of the popular Robot Operating System (ROS) [33] is a particle filter based probabilistic approach for localisation.

The particle filter based approaches use a sensor model and a set of particles representing hypothesised robot locations to estimate the true pose of the robot. A sufficiently large number of particles, adequate to describe the probability density function describing the robot pose needs to be selected in order to generate location estimates with acceptable accuracy. This is the main drawback of this algorithm as the computational efficiency of the particle filter is directly related to the number of particles used for the computation. In AMCL [44], there are many tuning parameters and strategies to dynamically manage the number of particles at an optimum level. Within the particle filter framework, it is not straightforward to identify outliers or dynamic objects. In order to address this problem, AMCL uses a “mixture-model” which categorises the range readings by statistically analysing the probable causes of such outliers and penalising these observations during the particle update step. However, [44] caution that this method would only work in certain limited situations and the categories should be analysed according to the environment.

Apart from estimation and filtering, optimisation based methods have also been used for localisation. However, once again these methods predominantly focus on feature based maps rather than on OGMs. In [76] a genetic optimisation algorithm is used to localise a mobile robot on a geometric beacon based *a-priori* map. Genetic algorithms are also used in [77] for localising on a satellite image geo-map of an outdoor environment using

---

<sup>1</sup>Localisation of a robot when the initial location is unknown is called the kidnapped robot problem.

an laser range finder sensor. Kwok et al. [78] proposes the use of evolutionary computing techniques which include genetic algorithms, particle swarm optimisation, and the ants system for feature based localisation and demonstrates their effectiveness and robustness to noise and dynamic environments.

Localisation of wireless sensor networks is another prominent application which relies heavily on optimisation based methods. Mao et al. [79] explains how different techniques are applied to this unique problem and how optimisation based methods can solve the wireless sensor networks localisation problem.

Scan matching has been popularly used for both mapping and localisation. Techniques like Iterative Closest Point (ICP) [80, 81], Iterative Closest Line (ICL) [82] or Iterative Closest Surface (ICS) and probabilistic likelihood methods [44, 83] are some of the widely used algorithms for matching laser scans obtained from different robot poses. In ICP, each laser endpoint in the query scan is associated with a point, line, or surface in the reference scan (or the map in case of localisation) using a distance metric such as Euclidean distance, after which a rigid body transformation method such as [84] can be applied to compute the alignment. This process is done iteratively for finer alignment of the scans. In probabilistic scan matching methods, sensor error which is the difference between actual sensor measurement and the predicted sensor reading is used to update the likelihood probability of the robot pose hypothesis. The predicted reading is estimated by algorithms such as ray-casting, which are computationally expensive, or likelihood fields [44] for which environment dependent tuning is essential, as it is an approximation to the ray-casting result.

Within the past decade, a variety of algorithms for solving much more complex problem of localising a robot while at the same time building a map of the environment (Simultaneous Localisation And Mapping (SLAM)) have also become available. While the most popular methods for SLAM in the early literature have been based on estimation methods such as EKF [85] and particle filters [34, 86], solving SLAM using optimisation techniques has recently emerged as the preferred solution [87–89] due to their robustness and ease of use. A detailed discussion about SLAM is provided in Chapter 4.

Though optimisation and EKF based back-ends for localisation are easy to use and computationally efficient, they have been so far limited to feature based maps due to the lack of a suitable sensor model for capturing correlations between OGMs and sensor readings. The methods currently available that uses the principles of scan matching either lack a proper co-variance model or have many tuning parameters.

Using upward facing sensors for localisation, either by observing natural features or artificial markers attached to the ceiling is a popular strategy amongst service robots. One of the main reasons for using this method is because the sensors pointed in a lateral direction are highly susceptible to occlusion when operating in crowded environments. The museum tour guide robot Minerva by Thrun et al. [90] uses an upward looking camera to localise the robot within the museum. The light intensities and substantial structure of the museum ceiling image directly above the robot provides one of the main inputs for localisation [91]. However, the authors further elaborate on how the vaulted multi-height nature of the museum ceiling can lead to errors and therefore this technique is only used to complement the localisation results of lateral sensors specially when they are unable to generate a consistent location estimate. Many other work after Minerva, for example, [92–97] completely rely on upward looking sensors. If an upward looking camera is used, the usual method of localisation is to extract features and landmarks from the ceiling [93–95]. These are usually corner features, lamps, ceiling tiles, etc. Work by Hwang, Song, & Kim [93] proposes extraction of compounded features from arbitrary-shaped lines or patterns in the ceiling as landmarks. When the ceiling lacks prominent features, artificial tags that can be easily recognised using a camera can be placed [96] on the ceiling as artificial landmarks. Localisation method for vacuum cleaning robots proposed by Gutmann et al. [92] uses an upward looking optical sensor to detect artificial beacons projected to the ceiling using a system known as “North-Star”. These projections create a pattern known as a vector field, which is initially learned by the robot. However, all of these methods rely on extracting and correctly associating features or landmarks.

### 3.3 An Extended Kalman Filter for Localisation in an Occupancy Grid Map

As discussed earlier in this chapter, while a particle filter is indispensable in localising a mobile robot when its initial position is unknown, it could be argued that once the approximate location is known and if the length of time over which odometry is used to predict robot pose without external sensor data is small (as is typically the case with modern high scan rate laser range finders), the powerful machinery embedded in the particle filter for non-Gaussian estimation is not essential.

In this section we propose a novel <sup>2</sup> EKF based algorithm to localise a mobile robot with a laser range finder in a two-dimensional (2D) OGM when the initial location is approximately known. The key challenge has been to formulate an appropriate measurement equation that can be used to predict the expected observations from a range finder. In the particle filter framework, this is done through techniques such as ray-casting. However, this strategy is not suitable for an EKF as an efficient implementation requires the Jacobians of the observation function in closed form. Therefore, we propose that the observation equation be formulated using the sensor model proposed in Section 2.3, and use the chamfer distance that corresponds to a laser scan within the OGM as a constraint. However, such a constraint relates the robot state and the range readings in an implicit function and as such is not in a form suitable for a standard implementation of an EKF. The strategy proposed by Steffen et al. [99] is, therefore used. In contrast to typical particle filter implementations, the proposed algorithm is easy to tune as only one parameter corresponding to the noise values of the inputs and measurements are required. Furthermore, it is relatively easy to deal with outliers through a probabilistic strategy that only accepts measurements subjected to a desired confidence level.

---

<sup>2</sup>We published the main results of this section in “An extended Kalman filter for localisation in occupancy grid maps.” [98]



### 3.3.1 Chamfer Distance Based Observation Model

After representing the environment using a distance function, the sensor model proposed in Section 2.3 can be used to formulate the observation function with the use of chamfer distance. As shown in Figure 2.6, a robot equipped with a range scanner is placed on the map at the prior pose estimate  $\mathbf{x}_{k|k-1} = (x_{k|k-1}, y_{k|k-1}, \phi_{k|k-1})^\top$ , and the observation vector  $\mathbf{z}$  projected from that pose. When the scan and the map are fully aligned, chamfer distance would approach zero and the robot would be at its true pose. Therefore, setting chamfer distance to zero in Equation (3.1) yields the measurement equation (3.2) that is suitable for robot localisation.

$$h(\mathbf{x}, \mathbf{z}) = \frac{1}{n} \sum_{i=0}^{n-1} DF(\mathbf{x}_{o_i}) = CD \quad (3.1)$$

$$h(\mathbf{x}, \mathbf{z}) = 0 \quad (3.2)$$

In Equation (3.1),  $\mathbf{x}_{o_i}$  corresponds to the endpoints of the observation vector  $\mathbf{z}$ , obtained using Equation (2.8) and  $DF(\mathbf{x}_{o_i})$  are distance function values obtained using a lookup of the interpolated  $DF$  matrix.

Traditional formulation of the EKF requires an observation equation of the form  $\mathbf{z} = h(\mathbf{x})$ . The alternative formulation that is proposed below can directly deal with an implicit form of the measurement equation, is an adaptation of [99].

### 3.3.2 Formulation of the Extended Kalman Filter

#### 3.3.2.1 Prediction

Let the estimate of the robot pose  $\mathbf{x}_{k-1|k-1} = (x_{k-1|k-1}, y_{k-1|k-1}, \phi_{k-1|k-1})^\top$ , be subjected to a control command of  $u_k = (v_k, \omega_k)^\top$ , where  $v_k$  is the linear velocity and  $\omega_k$  is the angular velocity over a period of  $\Delta t$ .

Then the predicted location of the robot is given by Equation (3.3) and its covariance by,

$$\mathbf{x}_{k|k-1} = F(\mathbf{x}_{k-1|k-1}, u_k \cdot \Delta t) \quad (3.3)$$

$$P_{k|k-1} = \nabla F_{\mathbf{x}} P_{k-1|k-1} \nabla F_{\mathbf{x}}^{\top} + \nabla F_u Q_k \nabla F_u^{\top} \quad (3.4)$$

where  $\nabla F_{\mathbf{x}}$  and  $\nabla F_u$  are respectively the Jacobin of the control function  $F$  with respect to  $\mathbf{x}$  and  $u$ , obtained by linearising about the robot pose estimate  $\mathbf{x}_{k-1|k-1}$ , while  $Q_k$  is the control noise covariance matrix.

### 3.3.2.2 Observation

Equation (3.2) provides the observation function,  $h(\mathbf{x}, \mathbf{z}) = 0$ .

Assuming that each range measurement  $r_i$  of the scan  $\mathbf{z}$ , corrupted by noise  $\eta_r$  with  $\mathcal{N}(0, \sigma_r^2)$  and bearing  $\theta_i$  is noise free, the covariance of the measurement vector is given by the diagonal matrix,  $\Sigma_{\mathbf{z}} = \text{diag}(\sigma_r^2)_{n \times n}$ .

### 3.3.2.3 Update

Update equations can be written as follows based on the derivations presented in [99]. The filter gain  $K$  is given by,

$$K = P_{k|k-1} \nabla h_{\mathbf{x}}^{\top} (\nabla h_{\mathbf{x}} P_{k|k-1} \nabla h_{\mathbf{x}}^{\top} + \nabla h_{\mathbf{z}}^{\top} \Sigma_{\mathbf{z}} \nabla h_{\mathbf{z}})^{-1} \quad (3.5)$$

The state update is given by,

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + K(-h(\mathbf{x}_{k|k-1}, \mathbf{z})) \quad (3.6)$$

while the covariance update is,

$$P_{k|k} = (I - K\nabla h_{\mathbf{x}})P_{k|k-1} \quad (3.7)$$

The Jacobians  $\nabla h_{\mathbf{x}}$  and  $\nabla h_{\mathbf{z}}$  at the appropriate linearisation points can be calculated using Equation (3.8).

$$\begin{aligned} \nabla h_{\mathbf{x}} &= \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial \mathbf{x}} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ &= \begin{pmatrix} \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial x} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial y} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial \phi} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \end{pmatrix} = \begin{pmatrix} \left. \frac{\partial d_{CD}}{\partial x} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \left. \frac{\partial d_{CD}}{\partial y} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \left. \frac{\partial d_{CD}}{\partial \phi} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \end{pmatrix} \\ \nabla h_{\mathbf{z}} &= \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial \mathbf{z}} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ &= \begin{pmatrix} \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial r_1} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial r_2} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \cdot \\ \cdot \\ \cdot \\ \left. \frac{\partial h(\mathbf{x}, \mathbf{z})}{\partial r_n} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \end{pmatrix} = \begin{pmatrix} \left. \frac{\partial d_{CD}}{\partial r_1} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \left. \frac{\partial d_{CD}}{\partial r_2} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \\ \cdot \\ \cdot \\ \cdot \\ \left. \frac{\partial d_{CD}}{\partial r_n} \right|_{\mathbf{z}, \mathbf{x}_{k|k-1}} \end{pmatrix} \end{aligned} \quad (3.8)$$

$\frac{\partial d_{CD}}{\partial x}$ ,  $\frac{\partial d_{CD}}{\partial y}$ , and  $\frac{\partial d_{CD}}{\partial \phi}$  in Equation (3.8) can be obtained with the help of Equation (2.22). As previously mentioned, the  $DF$  and its derivatives can be precomputed using the grid map and can be stored to be used during the runtime to calculate the gradient computationally efficiently. The remaining components of the gradient can be analytically derived from Equation (2.8).

### 3.3.3 Improving the Robustness of the Algorithm

A simple innovation gate is used in the algorithm to filter outliers that are related to objects not present in the map, such as people walking around. The individual entries of

$DF(\mathbf{x}_o)$  vector in Equation (3.1) is tested to be within  $2\sigma_{DF}$  bounds and the ones that are beyond these are ignored.

### 3.4 Optimisation Based Localisation in Occupancy Grid Maps

This section describes a method for localising a robot on a 2D map using information gathered using a laser range finder mounted on a robot. It also uses distance function based representation and the chamfer distance based sensor model that we presented in Section 2.3. Therefore, it does not require defining corresponding point parings as required by state-of-the-art optimisation based localisation methods. Unlike the method proposed in Section 3.3 which required the knowledge of the sensor noise parameters, or particle filter based methods which needs environment based tuning, this formulation only has one tuning parameter. Most importantly, we will demonstrate that it is not essential to have odometry measurements for the operation of this method in many practical situations.

The problem of localising a robot can basically be solved if there is a way to locate the laser scan on the given map. If a cost function  $C$ , which is a measure of mismatch between a sensor reading  $\mathbf{z}$  and the map  $\mathbf{m}$  can be defined such that the cost approaches zero when the sensor reading and the map are in alignment, it could be the basis for the optimisation problem as stated in Equation (3.9), with the robot pose as variables.

$$\underset{x \ y \ \phi}{\operatorname{arg\,min}} C(\mathbf{z}, \mathbf{m}) \quad (3.9)$$

When the initial pose is approximately known, the sensor model as described in Section 2.3 essentially defines such a cost function, as in the vicinity of the true pose, chamfer distance between the laser scan and the map shows a clear minimum as shown in Section 2.4.4 c.<sup>3</sup>

---

<sup>3</sup>We published the main results of this section in “*C-LOG: A Chamfer Distance Based Method for Localisation in Occupancy Grid-maps*” [100]

### 3.4.1 Optimisation Algorithm

Robot localisation problem expressed in Equation (3.9) can now be written as,

$$\underset{x \ y \ \phi}{\operatorname{argmin}} d_{CD}(\mathbf{x}_o, DF) \quad (3.10)$$

where DF is the distance function of the OGM  $\mathbf{m}$  and  $\mathbf{x}_o$  is the template generated using the laser scan  $\mathbf{z}$  in Equation (2.8) with the potential robot pose  $\mathbf{x} = (x, y, \phi)^\top$ . Therefore, we propose to use an optimisation algorithm to solve this problem in order to obtain the optimum pose  $\mathbf{x}^*$  that will yield minimum chamfer distance,

Given that the objective function in Equation (3.10) is twice differentiable, this unconstrained non-linear optimisation problem can be solved using a variety of gradient based techniques. In the experiments presented in Section 3.6 the Matlab implementation of the trust-region algorithm was used.

Algorithm 1 details the steps of solving the localisation problem.

---

#### Algorithm 1: Localise Robot

---

**Require:**  $DF, \frac{\partial d_{DF}}{\partial x_o}, \frac{\partial d_{DF}}{\partial y_o}$ , Initial pose estimate  $\mathbf{x} = (x, y, \phi)^\top$

**loop** for every input laser scan =  $\mathbf{z}$

**function**  $\operatorname{argmin}(\mathbf{z}, \mathbf{m})$   
 $\mathbf{x}$

**return**  $\mathbf{x}^*$

**end function**

Report current robot pose =  $\mathbf{x}^*$

next  $\mathbf{x} = \operatorname{CalcNextPoseGuess}(\mathbf{x}^*)$

▷ *Compute the best guess to the pose estimate at the next time step: Update  $\mathbf{x}^*$  using odometry information or use  $\mathbf{x}^*$  if odometry is not available.*

**end loop**

---

The partial derivatives of the objective function with respect to the robot pose  $\mathbf{x}$  are required for solving the optimisation problem described by Equation (3.10), are given in Equation (2.4.4) and Equation (2.22).

A simple gate which only admits only the values that are smaller than a certain error  $\zeta$ , can be used to reject outliers.

$$d_{DT}(\mathbf{x}_{o_i}) \leq \zeta \quad (3.11)$$

We define  $\zeta = \Delta\phi \cdot r_i + \Delta x + \Delta y$ , where  $\Delta x$ ,  $\Delta y$  and  $\Delta\phi$  are the maximum expected errors. In the experiments, 0.15m was used for  $\Delta x$  and  $\Delta y$ , while  $\Delta\phi$  was set to 0.05 rad. This is the only tuning parameter required for this algorithm and clearly it is relatively easy to establish.

### 3.5 Particle Filter based Localisation on Occupancy Grid Maps

As discussed before, particle filters are the preferred localisation algorithm for robots using range-bearing sensors when an OGM is used [70]. Though many implementations have been developed throughout the years, AMCL is considered the de-facto particle filter based localisation method in the community.

An appropriate sensor model is a crucial prerequisite of any probabilistic localisation algorithm. In a particle filter, the sensor model specifies how to capture the likelihood  $P(\mathbf{z} \mid \mathbf{x}, \mathbf{m})$  of a measurement  $\mathbf{z}$  given the map of the environment  $\mathbf{m}$  and a hypothesised robot pose  $\mathbf{x}$ . Commonly used strategy to obtain the likelihood is to first compute the difference,  $e_i = z_i - z_i^*$ , between the actual measurement  $z_i$  and the expected measurement  $z_i^*$  assuming that the robot's true pose is the same as its hypothesised pose.

Clearly, for a hypothesis that is closer to the true robot pose,  $e_i$  is small. Measurement likelihood is then computed assuming that the probability density function of a range measurement from the sensor has a Gaussian distribution with the mean equal to the true range, and the covariance equal to a known fixed value. In addition, a number of authors have proposed mechanisms for incorporating the impact of conditions such as the presence of objects in the environment that are not available in the map, and the possibility of receiving spurious range measurements [44, 101, 102].

The two most common sensor models used to compute likelihoods for particle filter based robot localisation are: (i) beam-based *beam range finder model* [103] which uses ray-casting, a method that measures the length to the next obstacle from the hypothesised robot pose along the path of the laser ray to predict the expected measurement and (ii) the *likelihood range finder model* or the *end point model* [73] which uses the shortest distance from the end of the ray to an occupied cell in the environment map with a small perturbation to approximate the ray-casting results. These two models are selectable within *ROS-AMCL* and the latter compromises accuracy for computational time. The algorithm then combines the likelihood of individual beams as described in Section 3.5.1, with an assumption that the likelihood of errors are independent of each other.

However, as identified by a number of authors [104, 105], although the measurements corresponding to the individual beams within a range scan from a given pose are theoretically independent of each other (by definition), the difference  $e$  between the actual and expected measurements are correlated unless the hypothesised robot pose  $x$  coincides with the true pose. In other words, value for  $e$  for all the beams in the scan is influenced by the difference in pose between the true robot pose and the hypothesised robot pose. Thus, multiplying likelihoods calculated using measurements from individual beams to obtain the likelihood of the entire scan can lead to a highly peaked distribution. Strategies to deal with this issue reported in the literature vary from attempts to minimise the impact of violating the independence assumption (for example, by using a subset of the beams) [44], to sophisticated methods for regularisation and place dependent covariance estimation [104] or learning a sensor model based on Gaussian processes [105].

Interestingly the *ROS-AMCL* implementation uses the likelihood function given in Equation (3.12).

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) = \sum_{i=1}^n P(z_k^i | \mathbf{x}_k, \mathbf{m})^3 \quad (3.12)$$

Although there is no theoretical basis for this formula. An excerpt from the file `amcl_laser.cpp` for ROS jade is shown in Listing 3.1. This formula together with appropriate tuning parameters, has enabled users to obtain good results from *ROS-AMCL* over the years. It

---

```
200 //      p *= pz;
201 // here we have an ad-hoc weighting scheme for combining beam probs
202 // works well, though...
203 p += pz * pz * pz;
```

---

LISTING 3.1: Weighting scheme used in AMCL v1.12.12, `amcl_laser.cpp`, retrieved on 25/06/2016

has also been used to benchmark performance of many other algorithms.

In likelihood field model, even-though distance functions are used for estimating the likelihood, multiple range readings within the range scan are treated as individual measurements and the computation of the associated covariance is done in a somewhat ad-hoc manner [44]. Use of the complete range scan as a single measurement is also a key motivation behind correlation based methods [106].

In this section, a modification to the likelihood field sensor model [73] is proposed, which makes it possible to compute the likelihood of the entire scan without resorting to the independent beam assumption. We use the chamfer distance based sensor model that was proposed in Section 2.3 for this purpose. Another contribution in this section is a rigorously Bayesian approach to compute the covariance of the multiple range readings and the measurement likelihood, while at the same time being more computationally efficient than all currently available sensor models for particle filter based robot localisation. A method to estimate the combined covariance of the chamfer distance in the presence of unknown correlations between the distance functions is also presented.

The next subsection provides an overview of how Monte-Carlo Localisation (MCL) operates.

### 3.5.1 Monte-Carlo Localisation (MCL)

Consider the problem of estimating the pose  $\mathbf{x} = (x, y, \phi)^\top$  of a mobile robot with respect to an *a-priori* map  $\mathbf{m}$  using a particle filter. This is usually done by maintaining a



probability density  $P(\mathbf{x}_k | \mathbf{z}_{1:k}, u_{0:k-1})$  of the pose  $\mathbf{x}_k$  at step  $k$  given all observations ( $\mathbf{z}_{1:k}$ ) up to step  $k$  and all control inputs up to step  $k - 1$  ( $u_{0:k-1}$ ). This probability can be recursively calculated using the Bayesian scheme given by Equation (3.13).

$$P(\mathbf{x}_k | \mathbf{z}_{1:k}, u_{0:k-1}) = \eta \cdot P(\mathbf{z}_k | \mathbf{x}_k) \cdot \int P(\mathbf{x}_k | \mathbf{u}_{k-1}, \mathbf{x}_{k-1}) \cdot P(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1} \quad (3.13)$$

Here,  $\eta$  is the normalisation constant that makes sure all the probabilities sum up to one for all  $\mathbf{x}_k$  and the term  $P(\mathbf{z}_k | \mathbf{x}_k)$  represents the probability of making an observation  $\mathbf{z}_k$  given the robot is at pose  $\mathbf{x}_k$ .

A brief explanation of the MCL algorithm is as follows: First, a set of particles representing the prior probability density function of the robot pose is initialised. Since the previous update, every particle in the cloud is propagated according to the motion model or dynamics of the robot with control inputs  $u_{k-1}$ . This is known as the **Prediction Phase**. Then, the likelihoods of obtaining the observed sensor reading  $P(z_k^i | \mathbf{x}_k, \mathbf{m})$  are determined for all particles on the basis of the observation model  $P(\mathbf{z}_k | \mathbf{x}_k)$  for the current sensor readings  $\mathbf{z}_k$ . The likelihoods of the particles are used to perform “resampling” to obtain a representation of the posterior probability density function. This is known as the **Update Phase**. The mean of the posterior probability density function represents the best estimate of the current robot pose. This process repeats as new information about the robot motion and observations from the sensors are received.

A typical scanning range sensor provides a measurement  $\mathbf{z}_k$  that consists of  $n$  individual readings corresponding to the distance to the closest occupied object in different directions  $\mathbf{z} = (z^1, \dots, z^n)^\top$ ,  $z^i = (r_i, \theta_i)$ . The ray-casting method [103] uses a process which follows the ray projection on the occupancy grid map from the end point of the beam, in the direction of the beam until an occupied cell is reached to obtain the expected measurement from a hypothesised robot pose. Error  $e_i$  for the  $i^{\text{th}}$  beam is then obtained by subtracting the ray-casting result from the actual measurement. In the likelihood field model [73], error for an individual beam,  $e_i$  is the distance to the closest obstacle from the end point obtained

by drawing a ray of length equal to the corresponding measurement from the beam with respect to the hypothesised robot pose. While ray-casting is computationally expensive, the measurement covariance is simply given by the uncertainties associated with the range sensor and the potential errors in the map. Measurement error is typically assumed to be distributed as a Gaussian and the measurement covariance ( $\sigma_r^2$ ) can be obtained through experiments. However, as explained in [44], currently available implementation of the likelihood field technique uses an educated guess for obtaining the measurement covariance. Thus, while the likelihood field method is faster than the beam range finder method, it is not as accurate.

The first contribution of this section is a rigorous method for computing the covariance of the distance function representation from the variance of the sensor reading,  $\sigma_r$ .

In both approaches (i.e. beam range finder and likelihood field models), the likelihood of a particle is calculated as the product of the likelihood of individual beam,  $z_k^i$  as per Equation (3.14).

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) = \prod_{i=1}^n P(z_k^i | \mathbf{x}_k, \mathbf{m}) \quad (3.14)$$

where,

$$P(z_k^i | \mathbf{x}_k, \mathbf{m}) = \eta \cdot e^{-e_i^2 / (2 \cdot \sigma_r^2)}$$

Equation (3.14) implicitly assumes that the  $e_i$ ,  $i = 1, \dots, n$  are uncorrelated.

Another contribution of this thesis is a strategy to compute the likelihood of a hypothesised robot pose without the implicit assumption of independence.

### 3.5.2 Estimation of Covariance and Measurement Likelihood

Covariance of a range measurement from the sensor  $\sigma_r^2$  can be used to compute the covariance of the each end point using Equation (3.15).

$$\Sigma_{\mathbf{x}_{o_i}} = J_{r_i} \sigma_r^2 J_{r_i}^\top \quad (3.15)$$

where  $J_{r_i} = \begin{pmatrix} \frac{\partial x_{o_i}}{\partial r_i} \\ \frac{\partial y_{o_i}}{\partial r_i} \end{pmatrix}$

The covariance of  $d_{DF_i}$  corresponding to the  $i^{th}$  laser beam can now be computed using Equation (3.16).

$$\sigma_{DF_i}^2 = \Sigma_{DF_i} = J_{\mathbf{x}_{o_i}} J_{r_i} \sigma_r^2 J_{r_i}^\top J_{\mathbf{x}_{o_i}}^\top \quad (3.16)$$

where  $J_{\mathbf{x}_{o_i}} = \begin{pmatrix} \frac{\partial d_{DF_i}}{\partial x_{o_i}} & \frac{\partial d_{DF_i}}{\partial y_{o_i}} \end{pmatrix}^\top$ .

Note that  $\frac{\partial d_{DF_i}}{\partial x_{o_i}}$  and  $\frac{\partial d_{DF_i}}{\partial y_{o_i}}$  can be pre-computed numerically from the  $DF$  of the map.

As  $d_{DF_i}$  are related through unknown correlations, the covariance of Chamfer Distance,  $\Sigma_{CD}$  is not equal to  $\frac{1}{n^2} \sum_{i=1}^n \sigma_{DF_i}^2$ . A number of techniques for fusing information with unknown correlations are available in the literature [107, 108]. These methods typically result in the need to solve an optimisation problem. As the chamfer distance is simply the sum of individual distance transforms, a conservative estimate for  $\Sigma_{CD}$  is obtained as per Lemma I.

This result directly extends to  $E = \sum_{i=1}^n e_i$ , leading to Equation (3.17), the measurement covariance.

$$\Sigma_{CD} = \frac{1}{n^2} \left( \sum_{i=1}^n \sigma_{DF_i} \right)^2 \quad (3.17)$$

With the help of the distance function of the OGM, the observation likelihood can now be obtained using Equation (3.18).

$$P(d_{CD,k} | \mathbf{x}, m) = \eta \cdot e^{(-d_{CD,k}^2)/(2 \cdot \Sigma_{CD})} \quad (3.18)$$

We use this observation likelihood to replace the sensor model of the particle filter in AMCL, which can be used to localise the robot.

**Lemma I**

Consider two correlated error measurements  $e_1$  and  $e_2$  with covariance  $\sigma_1^2$  and  $\sigma_2^2$ . If  $E = e_1 + e_2$  then the covariance of  $E$ ,  $\Sigma_E$  is given by Equation (3.19).

$$\Sigma_E = \text{cov}(e_1 + e_2) = J \cdot C \cdot J^T \quad (3.19)$$

where  $C = \begin{pmatrix} \sigma_1^2 & a \\ a & \sigma_2^2 \end{pmatrix}$ .

Here  $a$  represents the correlation of the two errors. As  $J = (1 \ 1)$  in this instance,  $\Sigma_E = \sigma_1^2 + \sigma_2^2 + 2a$ .

The matrix  $C$  should be positive semi-definite. Therefore,

$$\sigma_1^2 \cdot \sigma_2^2 - a^2 \geq 0$$

Then a conservative estimate (at full correlation,  $a = \sigma_1\sigma_2$ ) for  $\Sigma_E$  is given by Equation (3.20).

$$\Sigma_E = \sigma_1^2 + \sigma_2^2 + 2\sigma_1\sigma_2 = (\sigma_1 + \sigma_2)^2 \quad (3.20)$$

### 3.6 Experimental Results

This section presents results from the following three datasets which consist of simulations, data published in public domain, and experimental data to evaluate the three proposed localisation algorithms. In-turn it will also validate the applicability of the distance function based map representation and the sensor model presented in Chapter 2.

- **Dataset 1:** This dataset was generated using Player/Stage Robot Simulator for ROS [109], using the *Sample Hospital environment* available in the package. The ground-truth is available for this dataset.
- **Dataset 2:** Intel Research Lab [57] dataset.
- **Dataset 3:** This dataset was collected at the ground floor of Broadway Shopping Centre, Sydney, NSW, Australia.

For the Dataset 1 with the simulated Hospital environment, the robot was equipped with a simulated Hokuyo UTM-30LX laser range finder with a maximum range of 30m and 1080 beams with  $0.25^\circ$  angle increment, therefore a  $270^\circ$  field of view. The odometry was simulated with the noise parameters given in table 3.1 and the ground-truth of robot location was available.

The Dataset 2 is a public domain dataset in which a robot has taken three loops on a floor with multiple rooms. Map of the environment was generated with the GMapping algorithm [34] only using the information gathered during loop 3. The experiments were carried out using the remainder of the data to ensure the integrity of the evaluation. This dataset contains robot odometry, however, ground-truth is not available.

Dataset 3 was collected at a shopping centre in Sydney, Australia using a Hokuyo UTM-30LX laser range finder, which has a maximum range of 30m and a  $270^\circ$  field of view, with an angular resolution of  $1/4^\circ$ , which produces 1080 range readings per scan. As it was collected at normal operation hours, a substantial amount of obstruction made by people moving about the shopping centre can be observed. Due to these dynamic objects and the lack of odometry, the mapping method described in Section 4.4 was used to generate the map instead of GMapping.

### 3.6.1 Experimental Set-up

- **Extended Kalman Filter Algorithm**

TABLE 3.1: Simulation properties and noise parameters for Dataset 1

Sensor	Properties
Laser range finder	simulated Hokuyo UTM-30LX
Maximum range	30m
Minimum range	0.1m
Field of view	270°
Angular resolution	1/4°
No. of readings per scan	1080
Parameter	Noise Value
Laser range finder measurement noise	$\mathcal{N}(0, 0.02^2 m^2)$
Linear velocity noise	$\mathcal{N}(0, 0.04^2 m^2 s^{-2})$
Angular velocity noise	$\mathcal{N}(0, 0.01^2 rad^2 s^{-2})$

The proposed algorithm detailed in Section 3.3 was implemented in the MathWorks MATLAB<sup>®</sup> R2016a environment. The datasets were imported to MATLAB prior to the experiments.

- **Optimisation Based Algorithm**

The optimisation based algorithm was implemented for both MathWorks MATLAB<sup>®</sup> R2016a environment and ROS Indigo Igloo in C++ using OpenCV 2.8 and Eigen 3.2 libraries. However, we used the MATLAB variant for experiments due to ease of analysis and presentation of results. It is important to mention that, even the MATLAB variant has the capability to work with ROS in real time, as well as process recorded ROS bag files off-line, with MATLAB Robotics Systems Toolbox.

- **Chamfer Distance Based Particle Filter**

We added our chamfer distance based sensor model presented in Section 3.5 to the ROS Indigo Igloo implementation of AMCL [110]. On arrival of each laser scan, the measurement likelihood was computed through this new model. All the other parts of the ROS AMCL algorithm were left unchanged.

All experiments were conducted on a computer equipped with an Intel<sup>®</sup> Core<sup>™</sup> i5-2400 CPU @ 3.10GHz with 4GB of RAM.

TABLE 3.2: Error values

Algorithm	Mean error(abs)		Mean error(squares)	
	Position (m)	Orientation (deg)	Position ( $\text{m}^2 \times 10^{-3}$ )	Orientation ( $\text{deg}^2$ )
EKF(proposed)	$0.0227 \pm 0.0251$	$0.8999 \pm 1.2109$	$1.146 \pm 4.492$	$4.7226 \pm 12.0253$
Optimisation based(proposed)	$0.0099 \pm 0.0141$	$0.2387 \pm 0.9590$	$0.299 \pm 2.167$	$0.9766 \pm 5.4322$
Particle Filter(proposed)	$0.0290 \pm 0.0248$	$1.0762 \pm 1.9899$	$1.453 \pm 2.255$	$5.1161 \pm 15.7643$
Particle Filter(AMCL-Beam)	$0.0296 \pm 0.0243$	$1.4420 \pm 2.4679$	$1.464 \pm 2.078$	$8.1669 \pm 23.4030$

TABLE 3.3: Estimation credibility

Algorithm	Average-NEES	
	Position	Orientation
EKF(proposed)	$0.8929 \pm 2.5728$	$3.3756 \pm 14.4027$
Particle Filter(proposed)	$0.6241 \pm 0.6997$	$1.6523 \pm 6.9403$
Particle Filter(AMCL-Beam)	$0.1678 \pm 0.1487$	$0.5159 \pm 1.8033$

### 3.6.2 Pose Accuracy and Uncertainty

As the ground-truth is available for the Dataset 1, we can easily compare the results obtained from the three algorithms with the ground-truth. Figure 3.1 shows the trajectory of the robot through the environment. Poses obtained from the three algorithms are sparingly plotted to avoid clutter in the figure. For all three algorithms, 95% uncertainty ellipses are also plotted on the pose.

In Figures 3.2 - 3.4, we compare the pose errors against the ground-truth. The  $2\sigma$  error bounds are also plotted when the algorithm provides uncertainty. The errors for the EKF based algorithm and the particle filter based algorithm are well within the estimated  $2\sigma$  bounds. Furthermore, error values for the optimisation based algorithm is much low compared to other two algorithms. When the sensor measurement cannot provide information to optimise pose in one direction, the error in that direction can suddenly increase as no motion model is used for the optimisation based algorithm as seen in Figure 3.3 (e.g. at time stamps 1035s and 1206s for  $x$  error and time stamps 213s and 1405s at  $y$  error).

Figures 3.5-3.6 shows results from AMCL using the original beam based likelihood model.

Table 3.2 compares the errors between the proposed algorithms. We present mean absolute error and the mean squared errors. It can be seen that all three proposed algorithm has less error values than beam based AMCL algorithm. While the proposed particle filter

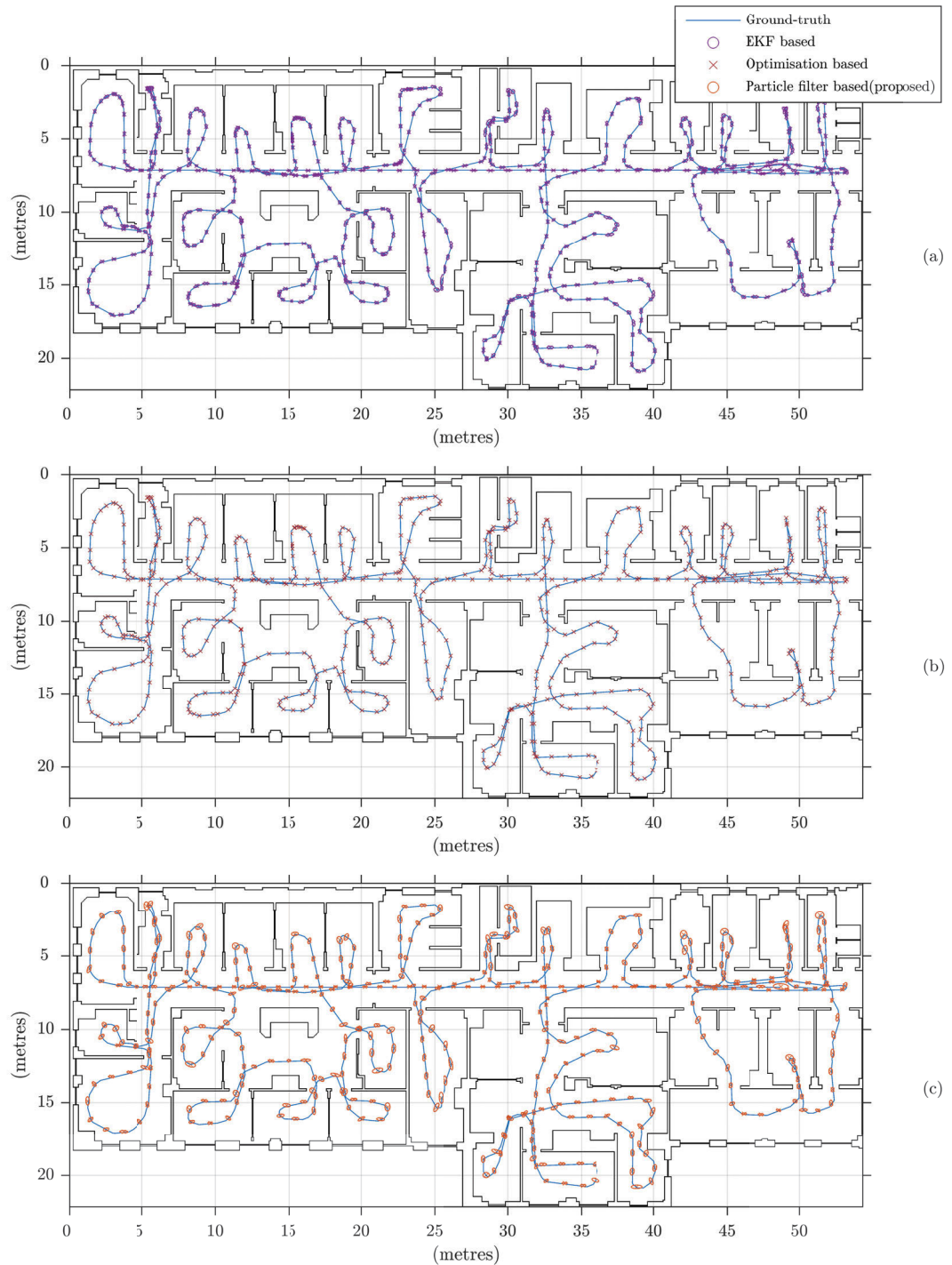


FIGURE 3.1: Trajectory of the robot in the Dataset 1. Ground-truth and results obtained from the three strategies, with uncertainty ellipses. (a) Proposed EKF algorithm. (b) Proposed Optimisation based algorithm. (c) Particle filter algorithm with the proposed sensor model.



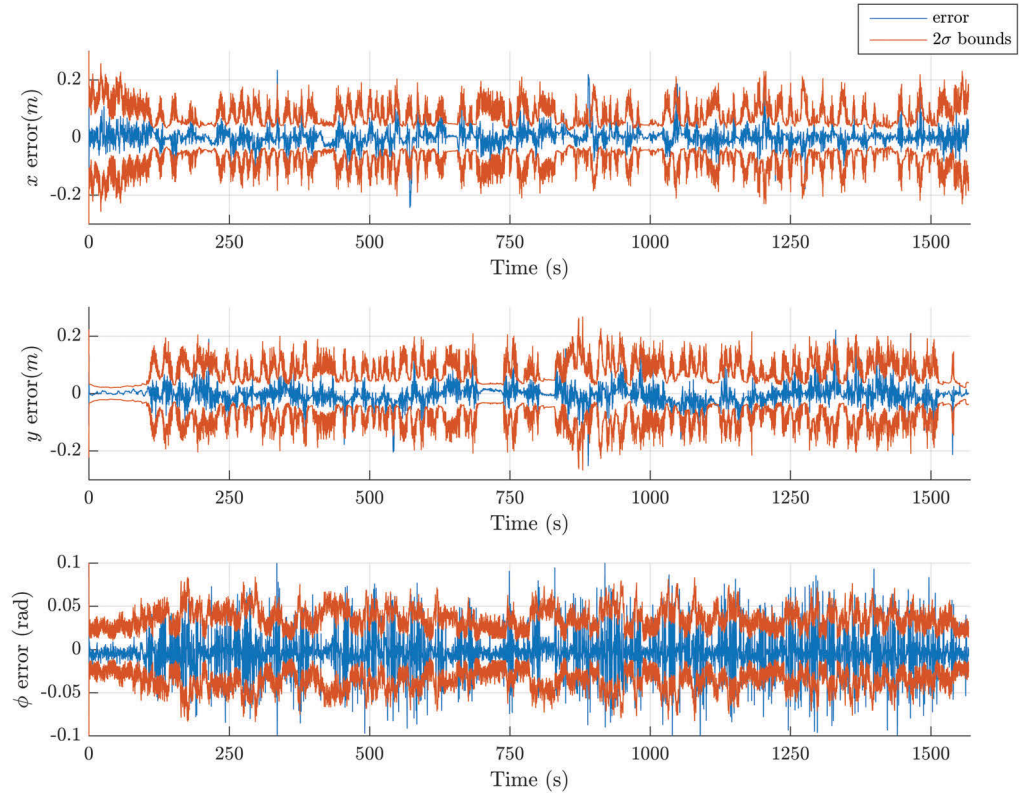


FIGURE 3.2: Error for each component of robot pose against ground-truth using the proposed EKF based algorithm for Dataset 1.

algorithm shows only a slight improvement, the EKF and optimisation algorithms have significantly less mean errors.

Moreover, we used the *average* normalised estimation error squared (NEES) metric to characterise the filter performance [111]. When the ground-truth state is  $\mathbf{x}_k$ , NEES ( $\epsilon_k$ ) is given by,

$$\epsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^\top P_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k) \quad (3.21)$$

where  $\hat{\mathbf{x}}_k$  and  $P_k$  are the estimated pose and its covariance. Then the average NEES can be calculated by,

$$\bar{\epsilon} = \frac{1}{nM} \sum_{k=1}^M \epsilon_k \quad (3.22)$$

where  $n$  is the number of dimensions in the robot state and  $M$  is the number of poses in the trajectory.

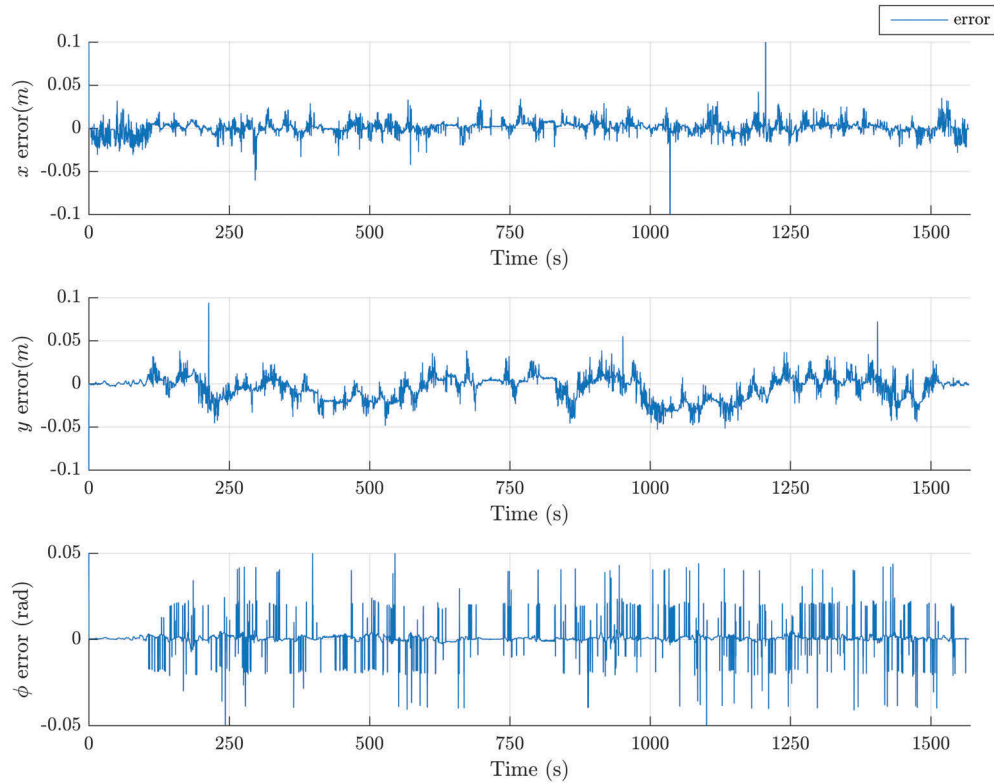


FIGURE 3.3: Error for each component of robot pose against ground-truth using the proposed optimisation based algorithm for Dataset 1.

Table 3.3 compares the average NEES values for the two algorithms 3.5 and 3.3 with the standard beam based AMCL.

The closer the average NEES value is to 1, the better is the credibility of the estimator. If this value is greater than 1 the filter is overconfident and therefore estimates can be inconsistent. The average NEES value considerably below 1 suggests that the filter is underestimating the covariance. This experiment shows that though the error values for all three algorithms are mostly within the 95% confidence interval, the beam based AMCL algorithm underestimate the covariances by a large margin. The proposed EKF algorithm has an average NEES value closer to 1 suggesting the estimation of covariance is consistent.

Figure 3.7-3.9 shows the trajectory of the robot with Dataset 2 . As with the previous dataset, to avoid clutter, not all poses are plotted. Further, to qualitatively analyse the results, we have plotted the laser projection from the estimated pose.

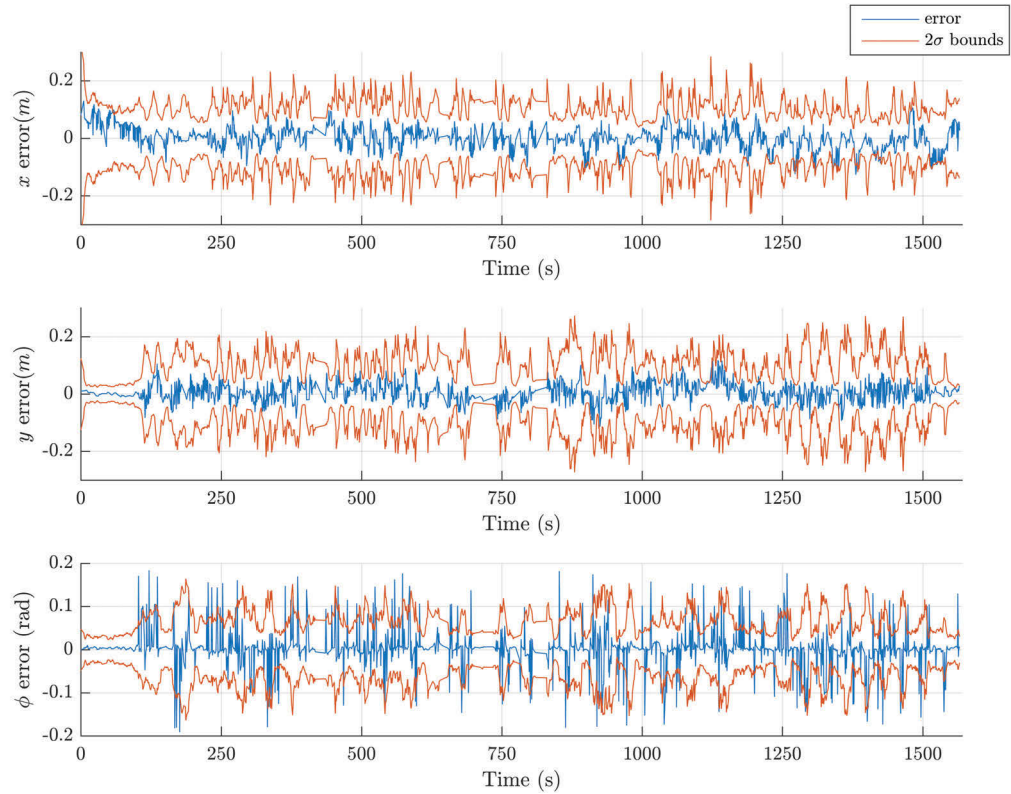


FIGURE 3.4: Error for each component of robot pose against ground-truth using the proposed particle filter based algorithm for Dataset 1.

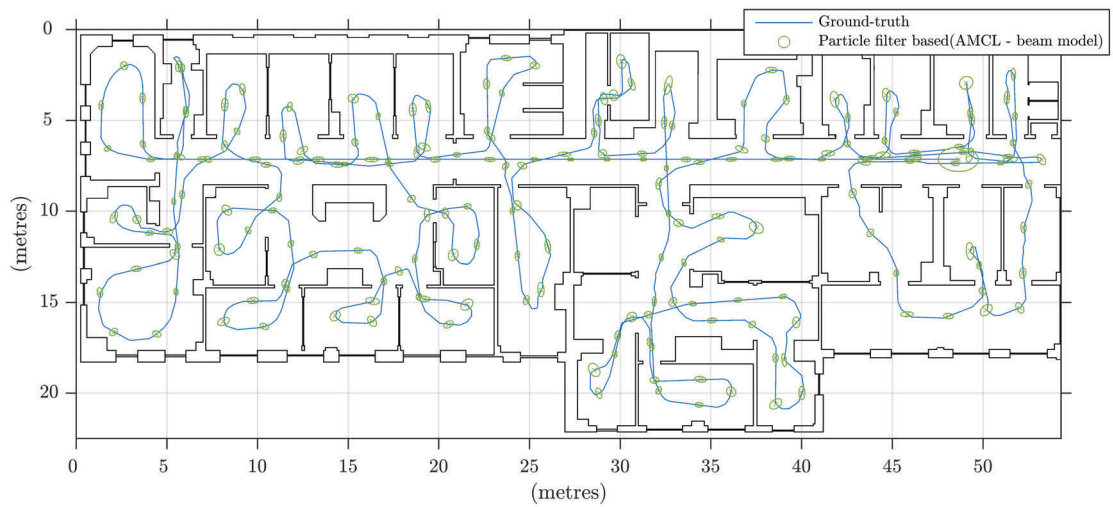


FIGURE 3.5: Trajectory of the robot in the Dataset 1. Results obtained by using standard AMCL, with beam based likelihood model.

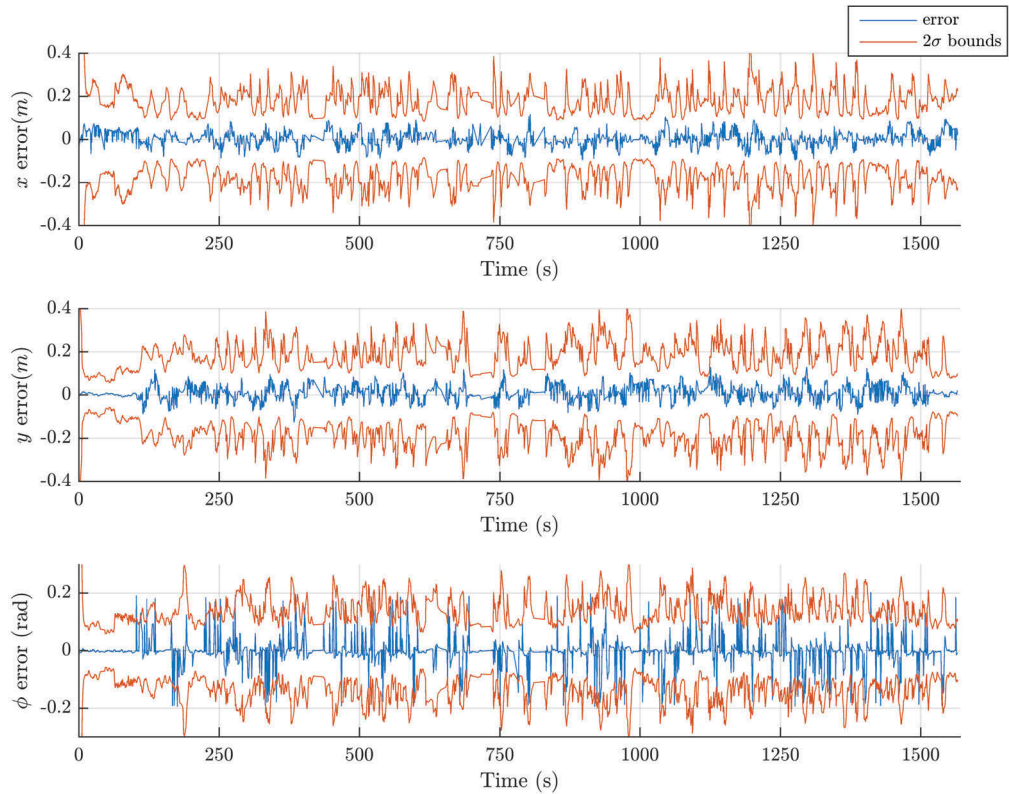


FIGURE 3.6: Error for each component of robot pose against ground-truth using standard AMCL, with beam based likelihood model for Dataset 1.

The Figure 3.11 presents the trajectory plots for the three proposed algorithms with Dataset 3.

All three localisation algorithms perform well in comparison to ground-truth, both with and without odometry. When qualitatively comparing the laser projections as shown in Figures 3.7c-3.9c, the optimisation method has much more accurate alignment compared to the others.

### 3.6.3 Performance with Dynamic Objects

As previously mentioned, the Dataset 3 was collected under natural conditions in a crowded environment. Therefore, the laser observations are mostly corrupted by people. Figure 3.12 is a sparse illustration of the crowd movement during data collection.

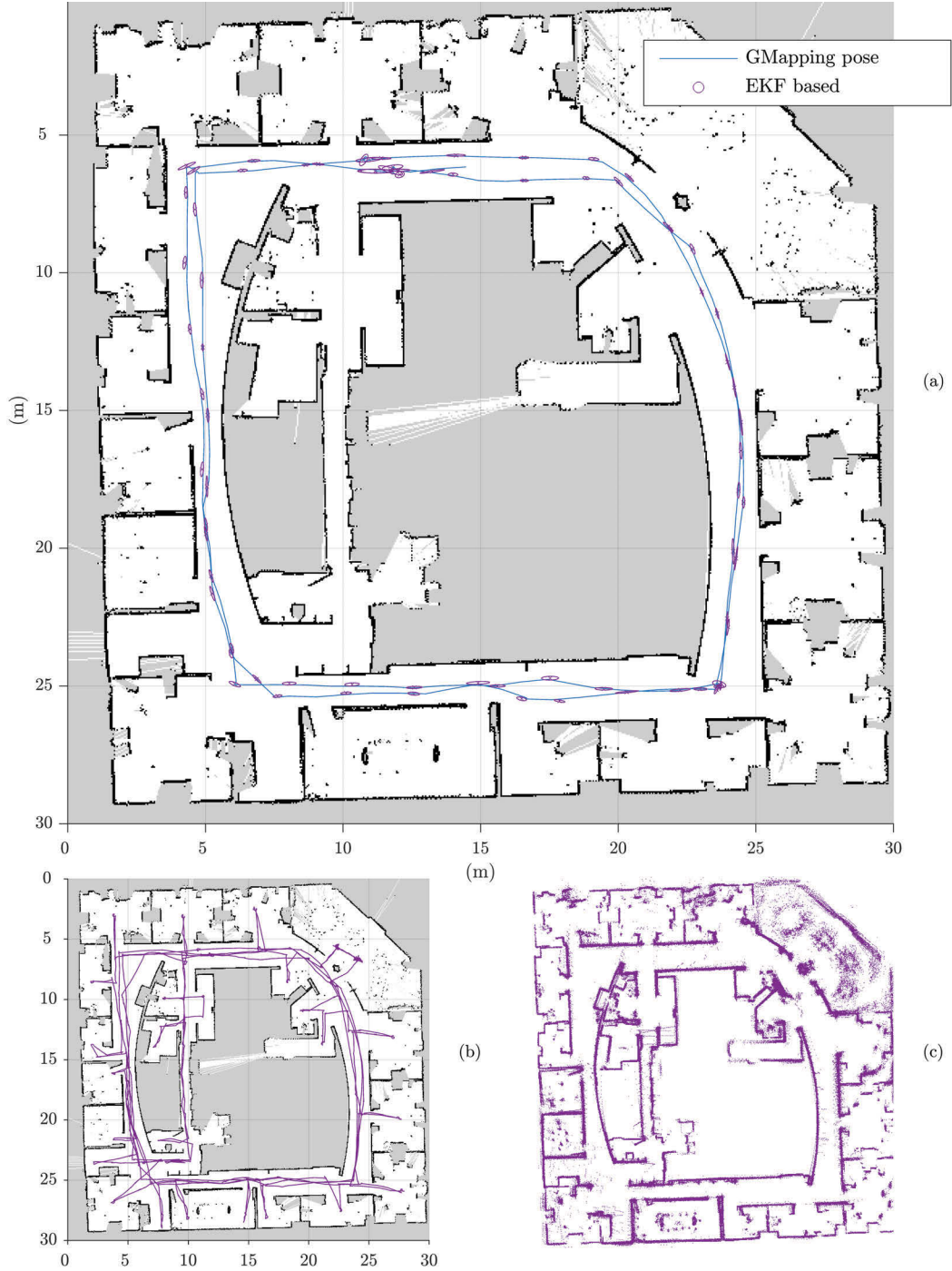


FIGURE 3.7: Trajectory of the robot in the Dataset 2. (a) GMapping poses and results obtained from the EKF based localisation algorithm, with uncertainty ellipses. (b) Complete trajectory. (c) Projection of the laser scan from the estimated pose.



FIGURE 3.8: Trajectory of the robot in the Dataset 2. (a) GMapping poses and results obtained from the optimisation based localisation strategy. (b) Complete trajectory. (c) Projection of the laser scan from the estimated pose.



FIGURE 3.9: Trajectory of the robot in the Dataset 2. (a) GMapping poses and results obtained from the particle filter localisation algorithm, with uncertainty ellipses. (b) Complete trajectory. (c) Laser projection from the localised pose.





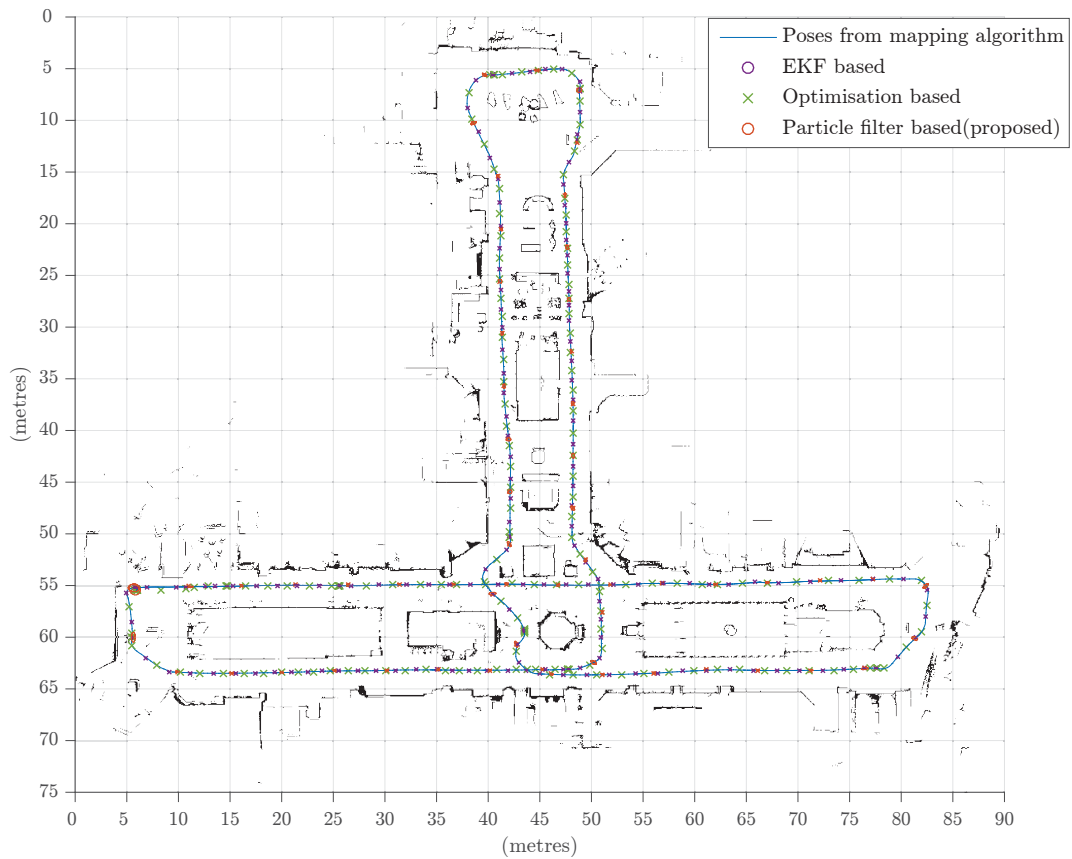


FIGURE 3.11: Trajectory of the robot in the Dataset 3. Poses from the mapping algorithm and results obtained from the three localisation strategies, with uncertainty ellipses.

To further evaluate the performance of the novel EKF based and optimisation based algorithms under dynamic objects, we performed a simulation experiment using Dataset 1. In this experiment we artificially corrupted a percentage of input laser scans with a uniform random distribution of  $\mathcal{U}(0, r_i)$ . Figure 3.13 shows the root-mean-square (RMS) error at different degrees of corruption. Both algorithms can continue to localise without losing track even with up to 60% of the input sensor measurements corrupted by dynamic objects. After certain corruption level, the algorithms still work most of the trajectory, however fails to keep track until the end of the robot trajectory.

### 3.6.4 Execution times

Figure 3.14 compares the time taken to process one laser scan by the three proposed algorithms. For particle filter based algorithms, the number of particles was set to be

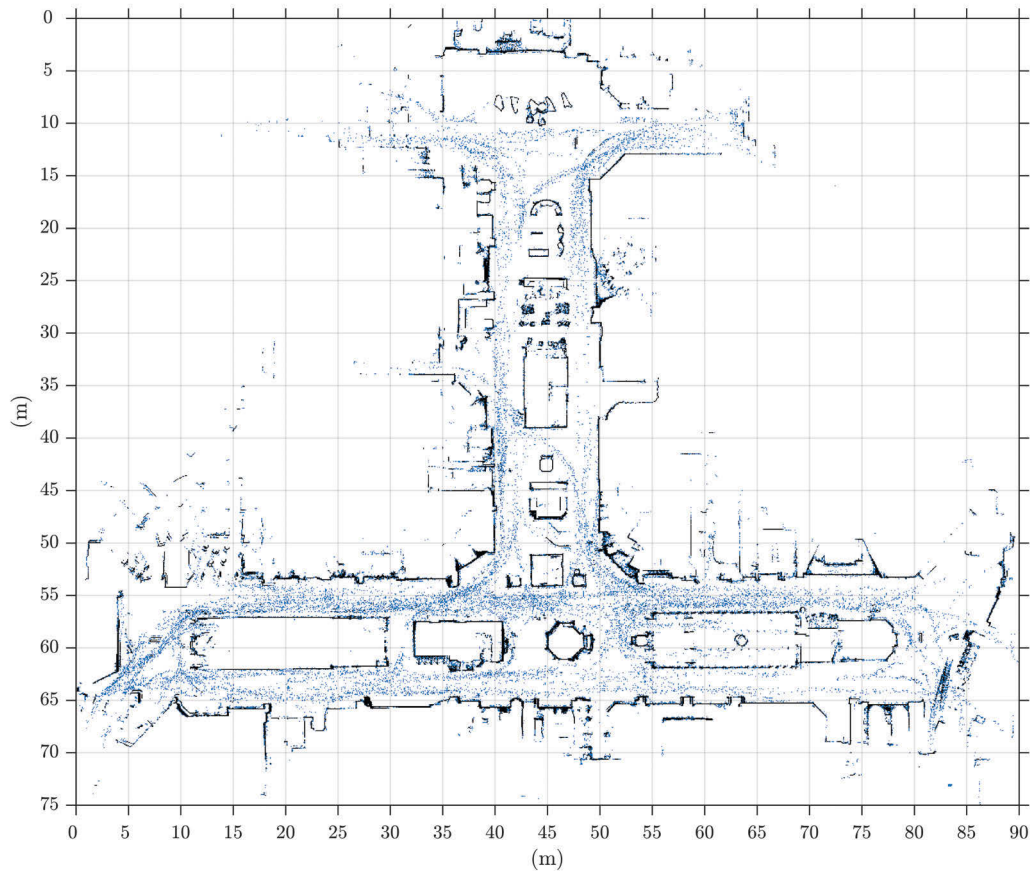


FIGURE 3.12: A sparse illustration of crowd movement during collection of Dataset 3 over 29.54 minutes.

fixed at 5000 particles. The experiments on Dataset 1 which are used for measuring the times and average time to process a laser scan are presented. The laser range finder (LRF) sensor in Dataset 1 produces 1081 laser readings per scan. For comparison purposes, this figure also show the execution times for the standard beam-based and likelihood-field sensor models. However, it is noted that the above algorithms are the ones distributed with ROS, which could not be the most efficient implementation of AMCL.

Although, all algorithms have linear time complexity, as expected, the beam based likelihood function takes the longest due to the complexity of the ray-casting process. Particle filter based methods have the highest computational times due high processing to manage a large number of particles in each iteration.

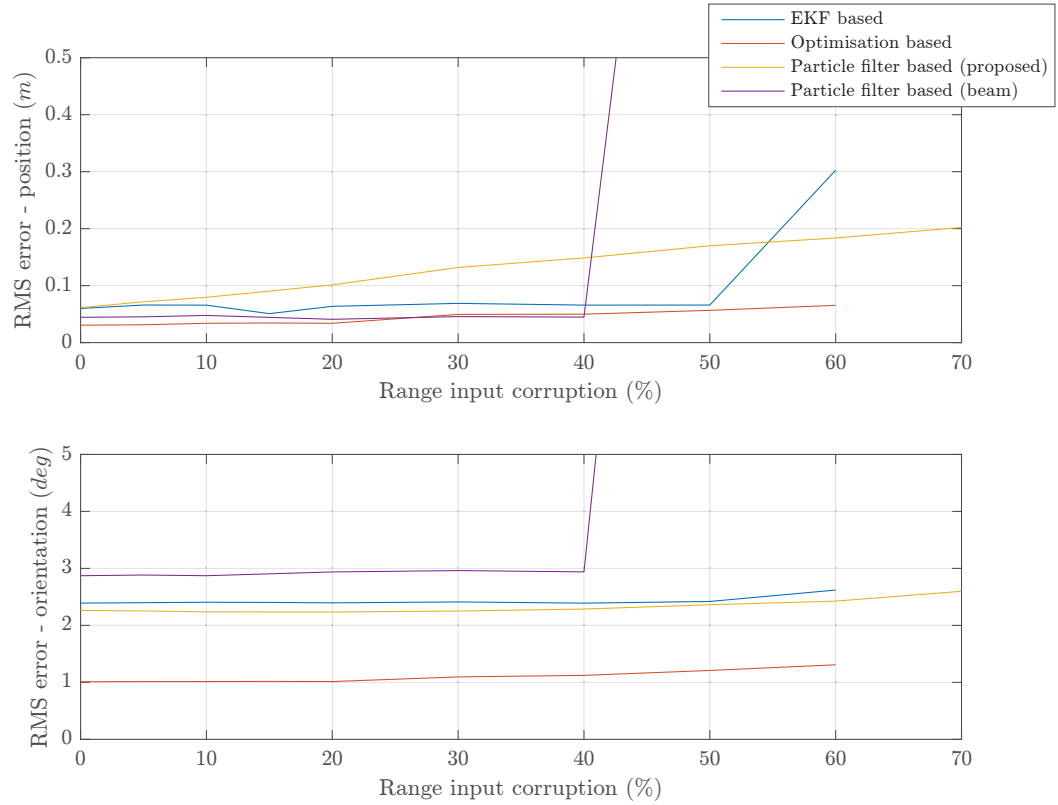


FIGURE 3.13: Shows the RMS errors of the proposed EKF based and optimisation based algorithms when input sensor measurements are artificially corrupted.

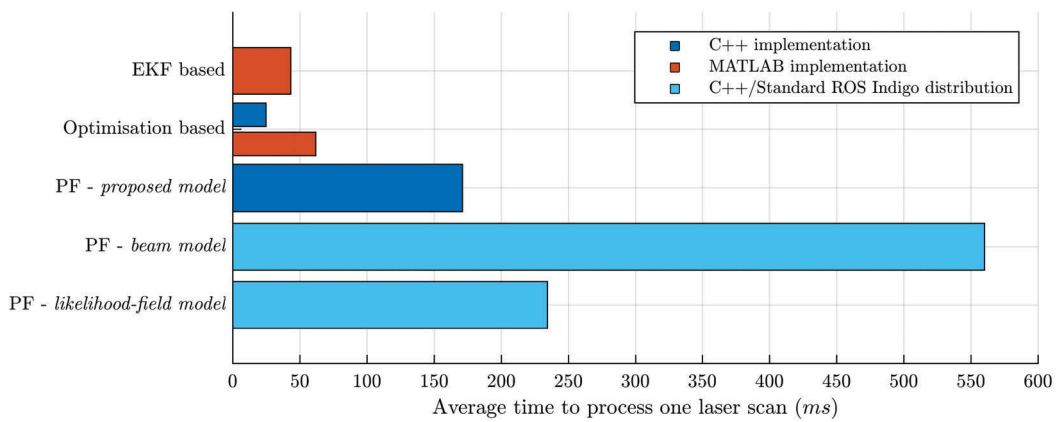


FIGURE 3.14: Per scan execution time for localisation algorithms.

### 3.6.5 Localisation Using Ceiling Data from an Upward Looking Camera

Localising a mobile robot with an upward looking camera is popular with service robots that operate in dynamic environments as it is typical for commonly used lateral sensors to generate inferior sensor measurements due to occlusions.

In order to evaluate the merits of the proposed chamfer distance based sensor model (Section 2.4.1), we performed some preliminary experiments using an upward looking camera as the sensor, together with a distance function map of the ceiling mosaic.

For these experiments, we used a modified version of the Willow Garage Turtlebot®. The upward looking camera that was fitted with the robot was a low cost Microsot LifeCam VX-2000 webcam which has a resolution of  $640 \times 480$  and a diagonal field of view of  $55^\circ$ . The robot was also equipped with a Hokuyo UTM-30LX laser range finder sensor. The laser range finder based localisation results were used as the approximated ground-truth for this experiment. The robot was tele-operated at the office environment in the Centre for Autonomous Systems(CAS), University of Technology, Sydney(UTS).

The ceiling of this environment is planar with lights and air-ducts inset onto the tiled ceiling as shown in Figure 3.15. Camera observations from the initial run was used to create a mosaic of the ceiling, which was transformed to a distance function. The observations are images captured from the upward looking camera mounted on the robot. The robot has wheel-encoders, and therefore reports odometry which was used in the ceiling localisation process.

Figure 3.16 shows the trajectory when the robot makes two loops.

It can be clearly seen in the figure that when the robot travels under the textured area of the ceiling the algorithm effectively localises the robot. However, when the robot goes through the area where the ceiling does not have any extractable texture, or when the camera is saturated by bright lights this algorithm fails to perform. In these occasions the odometry is used to update the robot pose.



FIGURE 3.15: Planar ceiling at the Centre for Autonomous Systems(CAS), University of Technology, Sydney(UTS)

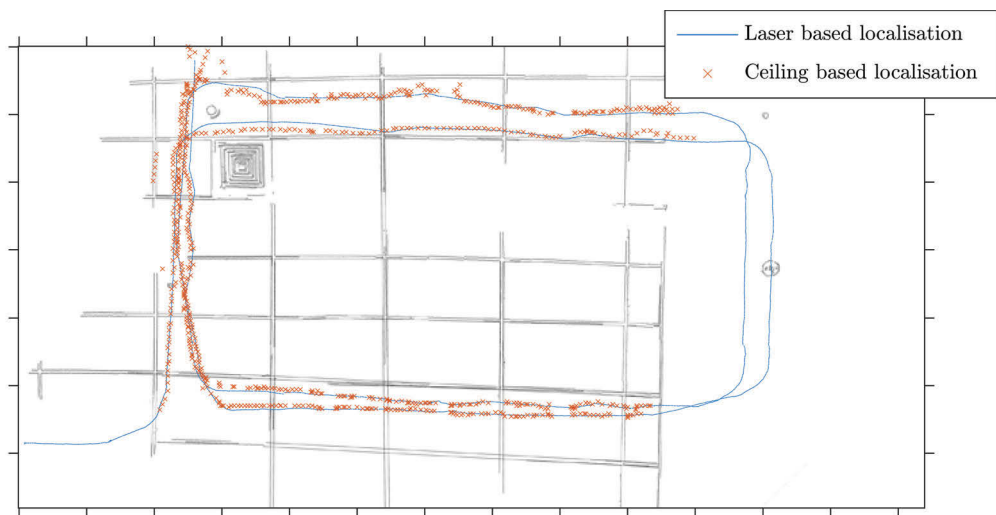


FIGURE 3.16: Shows the trajectory the robot takes with results obtained from ceiling image based localisation algorithm.

### 3.7 Conclusion

Chapter 2 of this thesis presented a distance function based environment representation method together with a chamfer distance based sensor model to relate observations from a laser range finder sensor mounted on a mobile robot to distance function based map. This chapter demonstrates the use of the proposed chamfer distance based sensor model with three different localisation algorithms namely (i) an extended Kalman filter, (ii) a particle filter, and (iii) an optimisation based technique to estimate the pose of a robot in a given 2D map.

Localising a robot using an EKF framework is another popular technique that had been used by the robotics community for a long time. Though it is highly efficient and performs well when the environment is represented using features or geometric primitives, so far EKF has not been able to use an OGM based environment representation without a pre-processing step that converts the said Occupancy Grid to a geometric primitive based map and defines explicit correspondence pairing between such features of the sensor reading and the map. The Section 3.3 describes the formulation of an EKF that uses the map representation method and the sensor model proposed in Chapter 2. Unlike the standard EKF based localisation methods, the algorithm we propose does not rely on extracting features from the map or defining feature correspondence. Moreover, the sensor and the control noise parameters are the only tuning parameters required by this algorithm. These can be easily acquired through experiments.

The Section 3.4 introduces an optimisation based algorithm to localise a mobile robot on an OGM. Similar to the previous algorithm, this method also uses the sensor model and the environment representation proposed in the previous chapter, and therefore, does not require extraction of geometric primitive features from the map or the environment as needed by state-of-the-art optimisation based localisation algorithms. Though the use of optimisation is not popular in solving the localisation problem, it is the most recommended method for solving the much more complex SLAM problem (see Section 3.2). Furthermore, when the initial location is approximately known, the proposed method can operate with a modern day laser range finder without the use of odometry, making it an ideal solution

for a standalone localisation package for a mobile robot. Having just one parameter that can be tuned intuitively is a key benefit of this algorithm.

As the particle filter is the go-to localisation method used by the robotics community to localise a robot in an OGM, the Section 3.5 evaluates the use of the proposed sensor model with particle filters. The widely used particle filter based localisation algorithm, AMCL, uses a sensor model which relies on errors from individual beams being independent. However, in reality, the beams are not entirely independent as pointed out by a number of authors and therefore AMCL uses empirical methods to deal with the issues that may potentially arise. The proposed sensor model does not explicitly rely on this assumption and estimates the associated sensor covariance in a theoretically sound manner rather than using another heuristic mixture model as in AMCL. Although particle filter based solutions can be used to localise robots in many challenging environments, having dozens of tunable parameters and the fact that it requires a large number particles and hence more computation power to maintain acceptable accuracy, makes it less attractive to be used in robots with limited computing power.

Moreover, this chapter presents results from a number of experiments and evaluates the proposed algorithms. These results also verify that the environment representation framework and the sensor model proposed in Chapter 2 works with both range-bearing sensors and monocular cameras.

The next chapter further extends the use of the proposed sensor model to generate distance function maps from raw sensor readings obtained by a robot equipped with a laser range finder. With a laser range finder, the algorithm can even operate without odometry and generate accurate maps under considerable clutter and dynamic objects.

## Chapter 4

# A Distance Function based Algorithm for Robotic Mapping

### 4.1 Introduction

The ability to create and update maps of unknown environments without any human intervention is regarded as another one of the most important requirements in the pursuit of building truly autonomous mobile robots. Simultaneous Localisation And Mapping (SLAM) is the most prominent way of addressing this. Despite significant progress in this area over the last two decades, SLAM still poses great challenges. At present, there are many robust methods for mapping environments that are static, structured, and of limited size. Mapping unstructured, dynamic, or large-scale environments remains largely an open research problem [112].

Though SLAM is generally computationally expensive, it is an essential requirement for building a statistically consistent map. The main complexities in SLAM deal with how it handles uncertainties of the sensor observations. However, when the sensor accuracy is much higher than the desired resolution of the map, and the map is of small to moderate size, e.g. house, office or hospital building, shopping complex, the complex mechanisms embedded in SLAM may no longer be necessary to create accurate maps of the environment.



Scan matching is a method commonly used to create such maps with the use of 2D or 3D Light detection and ranging (LiDAR) sensors that produce a depth scan of the environment as the observation. There are numerous methods of scan matching present in the literature, but in essence, the strategy involves matching every new observation to one or more previous observations to estimate the relative displacement between the two frames. The resulting trajectory can then be used to generate the map of the environment.

In this chapter we leverage the efficiency and simplicity of the distance function based environment representation and the associated sensor model we introduced in Chapter 2 to present a novel mapping mechanism that could be used with a lightweight computing platform. This method is able to generate accurate maps of indoor environments using laser range finders, with and without the use of robot odometry. We use simulation, public domain, and datasets we collected on real environments to assess the validity and the merits of the algorithm. Comparisons with other state-of-the-art scan-to-map matching and scan-matching based mapping algorithms are also presented. It is important to note that the algorithm we propose does not consider the impact of sensor uncertainty. However, we demonstrate that high quality maps can be generated to be used in practical scenarios of importance to assistive robots.

This chapter is organised as follows: The next section presents related work in the field. Section 4.3 discusses preliminaries of scan-to-scan and scan-to-map matching. Section 4.4 formulates the proposed chamfer distance based mapping algorithm. In Section 4.5 we present normal distribution transform (NDT) based scan-to-map matching algorithm, an extension to one of the state-of-the-art scan based mapping techniques which we used to compare the proposed mapping method. Section 4.6 presents experimental results and comparisons, while Section 4.7 concludes the chapter.

## 4.2 Related Work

Probabilistic methods have dominated the robotic mapping field since early 90's after Smith, Self, & Cheeseman [113, 114] published a series of seminal papers introducing a powerful statistical framework for mapping, and introduced the problem of simultaneously

solving the mapping problem while localising the robot relative to the growing map. Since then many algorithms have emerged as solutions to this problem which became popular as Simultaneous Localisation And Mapping or SLAM algorithms [115].

One family of solutions to SLAM use Kalman filters to estimate the map and the robot location [85, 116–118]. The resulting maps usually describe the location of landmarks, or significant features in the environment. Similarly, another branch of SLAM solutions, which uses particle filters [34, 86, 119] as the solving mechanism where the map is constructed from as a tree of particle trajectories with each node representing a possible realisation of the map. The node with the least error is chosen as the optimum estimate. An OGM of Intel Research Laboratories Seattle, USA, obtained by the use of *GMapping* [34] which is the most commonly used laser range finder based SLAM algorithm in the community is shown in Figure 4.1. GMapping uses a Rao-Blackwellized particle filter formulation [120] to derive the trajectory of the robot.

It is also possible to formulate SLAM as a parameter estimation problem with all the robot poses where observations are taken from, and all observed landmarks or features are treated as a set of unknown parameters that are to be estimated. An objective function that is based on Maximum Likelihood (ML) of those parameter estimates can be formulated by using robot odometry measurements and robot-to-feature observation information to relate the unknown parameters [87–89, 121]. Due to the availability of large memory spaces, and the fact that the Jacobians of the objective function to be optimised are sparse, current algorithms that use the optimisation framework can even efficiently solve SLAM problems consisting of a few thousand of robot poses and a few million robot-to-feature observations [122]. The use of such optimisation techniques has become the preferred solution to SLAM, due to their robustness and ease of use.

Even-though feature based SLAM has become extremely popular in the field and has many applications, in environments where it is not straight forward to extract features, or when an OGM is required from sensors such as laser range finders, scan based SLAM methods [34, 123–125] are commonly employed. These methods directly utilise the unprocessed scan readings as observations which are generally sets of points obtained from sensors such as laser range finders. One of the fundamental issues of the scan-based SLAM



FIGURE 4.1: An occupancy grid map of the Intel Research Laboratories, Seattle, USA generated using GMapping.

has thus been the association or matching of one scan to another, namely the scan-to-scan matching. When the correct correspondence and a cost function that is known to align the scans is available, a traditional SLAM back-end can be used to solve the problem. In the *Scan-SLAM* framework proposed by Nieto, Bailey, & Nebot [123] and in the work by Diosi & Kleeman [124], the authors propose the use of an EKF to estimate the robot poses and the map after a scan matching process. More recent work on scan matching based SLAM proposes a graph based optimisation solution to process the scan-to-scan matching output [125]. In the method proposed by Lv et al. [126], incremental scan matching is performed to construct *sparse point sub-maps* which are realigned to minimise error when a loop is detected and a loop closure step is performed, which can be used to produce more accurate maps.

Newer optimisation based algorithms, commonly known as pose-graph SLAM [122, 127, 128], completely marginalise the features or landmark locations and only maintain robot poses and relationships between poses as its internal representation, without using an explicit geometric representation of the environment. Such relative pose estimates can be obtained using a scan matching technique for laser range finders or an image registration method for cameras, provided that at different consecutive poses the sensors can observe and recognise the same region of the environment. When the robot visits previously observed regions after a long traversal, a loop-closure detection technique is used to create further relationships[129]. Once the optimum robot pose estimates are available, a map of the environment can be constructed if required. One main advantage in this framework is that any sensor can be used to construct the pose relationship, which make the algorithms

independent from the sensor selection.

However, with the advent of highly accurate laser range finder sensors, real-time map generation for practical purposes has shifted from the use of SLAM which is usually computationally expensive, to other techniques that solely rely on scan matching [35, 36, 83, 130–132]. These methods range from using ICP scan-to-scan matching algorithm to grid based algorithms that uses normal distribution transform methods. These methods do not consider the uncertainties associated with the sensor measurement. Therefore, although they are not guaranteed to be statistically consistent, maps generated may be adequate in practical situations faced by assistive robots.

The most common scan-to-scan matching approach is based on the ICP technique [80, 133, 134] which allows the point-to-point matching between two scans by minimising the total distance between them. Despite the popularity of the technique, the point-to-point matching may yield inappropriate data association since two corresponding points are not actually on the same position in the environment. Weiss & Puttkamer [135] proposed a technique that avoids the point-to-point matching problem by calculating an angle between two neighbouring scan points and using the angles to match the two scans. ICP has further evolved to include point-to-line and point-to-plane to effectively deal with association problems, but they are still environment dependant.

In order to overcome point and plane association issues, Biber & Straßer [131] represented the environment as a subdivided grid space and collectively describing a scan within each grid cell by a normal distribution (ND). This grid-based normal distribution transform (NDT) technique spatially associates every point of the new scan inside a grid cell to the corresponding normal distribution in the cell. NDT methods do not suffer from the point-to-point correspondence issues and does not depend on the environment structure such as point-to-plane matching does, though matching performance varies according to the grid size and the accuracy of the initial guess.

Furthermore, a number of approaches have been proposed on integration of multiple scan matching techniques to improve the accuracy. Early efforts include the work of Lu and Milios [136], which performed the matching of the new scan to the previous scan and further matched it to all the scans by storing the past scans. Thrun et al. [137] used an

expectation maximisation (EM) algorithm that finds the best matching scan in the past to the new scan from all the past scans. The matching of the new scan to all the past scans is then achieved by the scan-to-scan matching of the new scan to this best past scan. Although they have demonstrated capabilities in accurate matching, the approaches could still see error accumulation without a loop closure as neither implement a powerful scan-to-scan matching nor utilise all the past scans. In [131], the authors use a point-to-ND technique to match the new scans to multiple past scans sequentially to reduce error accumulation. Gutmann et al. [130] presents a similar method that matches the new scan with a limited collection of scans using a sliding-window technique which improves the accuracy of the map immensely, while keeping the algorithm computationally tractable. In the ideal scenario a new scan should be matched with a combined scan of all past scans, however this can be highly computationally expensive. Bosse et al. [138] introduced a subspace-to-map matching technique where the new scan is matched to all the past scans of a subspace of concern with any scan-to-scan matching technique and the subspaces are subsequently associated to each other for global mapping. This technique achieves the matching of the new scan to all the past scans, but the accuracy could still decrease as the new scan points not in the subspace are not matched to the past scans. Ryu et al. [139] extends work presented by Straßer and Biber [131] and proposes an ND-to-ND based scan-to-map matching technique which can build further accurate maps with less error accumulation due to the proposed multi-ND matching mechanism. Another advantage of most of these techniques for scan matching is that they can directly be extended to 3D mapping with minimal effort [132].

KinectFusion by Nicombe et al. [38] and the method proposed by Tomono [140] constructs a Truncated Signed Distance Function (TSDF) based three-dimensional (3D) global model by doing global transform of all previously acquired scans, and uses signatures from the global model for tracking the next robot pose, which is essentially a scan-to-map matching technique. However, these methods can be computationally expensive as the metric that is used for matching needs to be extracted from the global model every iteration by using an expensive ray-casting strategy.

### 4.3 Scan Matching Preliminaries

Scan matching is the process of estimating the rigid body transform between two 2D or 3D scans taken from different robot poses, so that the scans are best aligned.

When scans are captured by a range sensor on a moving robot, they are sequentially obtained with respect to different robot coordinate systems. Let  $\{R^-\}_{\mathbf{z}_{k-1}} = \{\{R^-\}_{z_{k-1}^i} | \forall i \in \{1, \dots, m\}\}$  be the previous scan in the previous robot coordinate system, and  $\{R\}_{\mathbf{z}_k} = \{\{R\}_{z_k^i} | \forall i \in \{1, \dots, n\}\}$  be the new scan in the new robot coordinate system, where  $k$  is the time step,  $n$  is the number of points in the scan.  $\{R^-\}$  and  $\{R\}$  denote the robot coordinate systems before and after correction, respectively. Given the two scans, a scan-to-scan matching technique iteratively finds relative transformation parameters,  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}_k = [\mathbf{t}_k, \phi_k]^\top$ , composed of a translation,  $\mathbf{t}_k = [t_k^x, t_k^y]^\top$ , and a rotation,  $\phi_k$ , between the two coordinate systems by locally matching the two scans. The first step in this process is to transform the new scan in the new robot coordinate system to that in the previous coordinate system using an initial guess,  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}'_k$ , of the transformation parameters using Equation (4.1). This initial guess can be estimated from readings of other sensors such as odometry, or set to zero assuming that the two scans are close enough.

$$\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{z}_k^i (\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}'_k) = \mathbf{R}(\phi'_k) \begin{Bmatrix} R \\ R \end{Bmatrix} \mathbf{z}_k^i + \mathbf{t}'_k = \begin{bmatrix} \cos(\phi_k) & -\sin(\phi_k) \\ \sin(\phi_k) & \cos(\phi_k) \end{bmatrix} \begin{bmatrix} \{R\}_{z_k^{x_i}} \\ \{R\}_{z_k^{y_i}} \end{bmatrix} + \begin{bmatrix} t_k^{x'} \\ t_k^{y'} \end{bmatrix} \quad (4.1)$$

where  $\mathbf{t}'_k = [t_k^{x'}, t_k^{y'}]^\top$ , and  $\begin{Bmatrix} R \\ R \end{Bmatrix} \mathbf{z}_k^i = [\{R\}_{z_k^{x_i}}, \{R\}_{z_k^{y_i}}]^\top$ .

Thereafter, any of the scan-to-scan techniques mentioned in Section 4.2 can be used to compute the transformation  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}_k$  between points of the new scan  $\{R^-\}_{\mathbf{z}_k}$  and the previous scan  $\{R^-\}_{\mathbf{z}_{k-1}}$ . In incremental scan-to-map matching, instead of matching the new scan to the previous scan, it is matched to the already formed map.

When a new scan is obtained, it is first transformed to the previous robot coordinate using  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}'_k$ . Some of the latest work on scan-to-map matching [36, 139, 141] uses scan-to-scan matching methods such as ICP, as an initial conditioning step to improve the initial guess.

Once the  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}'_k$  transform is applied, the scan can be further transformed to the global coordinate system  $\{G\}$  using,

$$\begin{Bmatrix} G \end{Bmatrix} \mathbf{z}_k^i = \mathbf{R}(\begin{Bmatrix} G \end{Bmatrix} \theta_{k-1}) \begin{Bmatrix} R^- \end{Bmatrix} \mathbf{z}_k^i + \begin{Bmatrix} G \end{Bmatrix} \mathbf{x}_{k-1} \quad (4.2)$$

where  $\begin{Bmatrix} G \end{Bmatrix} \mathbf{x}_{k-1} = (\begin{Bmatrix} G \end{Bmatrix} x_{k-1}, \begin{Bmatrix} G \end{Bmatrix} y_{k-1}, \begin{Bmatrix} G \end{Bmatrix} \theta_{k-1})^\top$  is the robot pose in the global coordinate system estimated at time step  $(k - 1)$ .

However, at this point the scan is not yet perfectly aligned with the map due to errors in the initial guess, and a scan-to-map matching algorithm is used to estimate the final transform  $\begin{Bmatrix} G^+ \\ G \end{Bmatrix} \mathbf{p}_k$ , in the corrected global coordinate frame  $\{G^+\}$ .

## 4.4 CD Mapping: Chamfer Distance based Scan-to-Map Matching

In this section we present a scan-to-map matching mapping method which uses the distance function based environment representation method and chamfer distance based sensor model presented in Chapter 2.

Modern day laser range finders that are widely used in robotics applications have higher data-rates compared to the motion speed of mobile robots. Therefore, when a new area is being explored, a significant part of the new laser scan overlaps with the existing map. This overlapping data is used to localise the robot on the map and the new component of the scan is then merged to extend the map.

A more detailed description of the method is as follows:

When a new laser scan is obtained at time-step  $k$ , the scan is transformed from  $\{R^-\}$  coordinate frame to  $\{R\}$  using odometry information with the use of Equation (4.1). If robot odometry is unavailable, the relative transform  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}_k$  can be set to zero. Contrary to other popular algorithms, the use of ICP to improve the initial guess can be seen as unnecessary additional overhead as the typical convergence basin of the chamfer

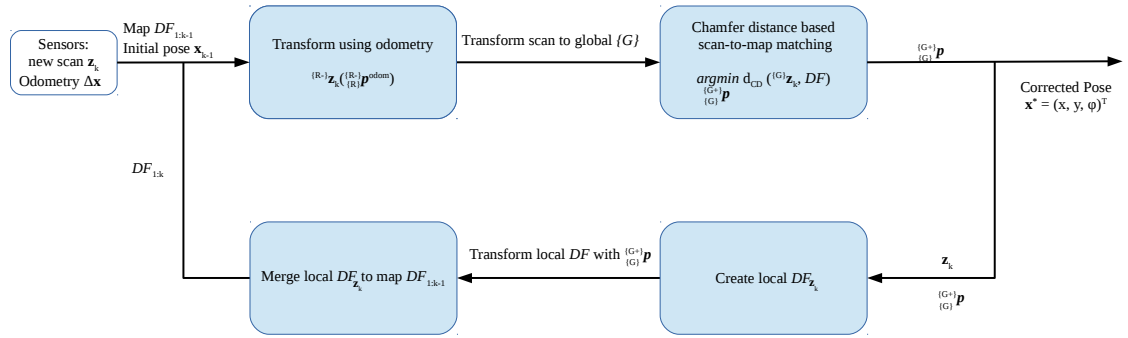


FIGURE 4.2: Flow chart for the CD mapping algorithm.

distance based objective function is approximately  $\pm 3\text{m}$  for translations of  $(x, y)$ , and approximately  $\pm 30^\circ$  for rotations of  $\phi$  as detailed in Section 2.4.4.

After the transform, it is necessary to convert the updated initial robot pose and the scan to the global coordinate frame  $\{G\}$  using Equation (4.2), to perform scan-to-map matching. At this stage, if the laser scans contain points that are out of the current map boundaries, those points can be safely excluded from the matching phase.

After which, we use Equation (4.3) to perform matching of the new scan to the existing distance function.

$$\arg \min_{\substack{\{G^+\} \\ \{G\}} p_k} d_{CD}(\{G\} z_k, DF) \quad (4.3)$$

The objective function above forms an unconstrained non-linear optimisation problem that can be solved using a variety of gradient based techniques similar to Section 3.4. We used the Matlab implementation of the trust-region algorithm for the experiments presented in Section 4.6.

The matching result would be the corrected transformation  $\{G^+\} p_k$ , which can be used to transform the scan to the corrected global coordinate frame  $\{G^+\}$ . Figure 4.2 depicts a flowchart for the mapping process.

Once the corrected robot pose is obtained, the new scan can be located in the global frame. Thereafter the new scan can be merged to the existing distance function in order



to expand the map. The update only happens to the area overlapped by the new scan. First, the local distance function of the new scan,  $DF_{Z_k}$  is created. Then the updated map can be obtained by fusing the distance function of the new scan to the existing map,  $DF_{1:k-1}$ , simply by Equation (4.4), without the need for recreating the entire distance function from scratch.

$$d_k^{ij} = \min \left( d_{k-1}^{ij}, d_{Z_k}^{ij} \right) \quad (4.4)$$

where  $d_k^{ij} \in DF_k$  [overlapping region],  $d_{k-1}^{ij} \in DF_{k-1}$  [overlapping region], and  $d_{Z_k}^{ij} \in DF_{Z_k}$ .

## 4.5 An Extension to Grid-based ND-to-ND Scan-to-Map Matching<sup>1</sup>

In this section we propose an extension to ND-to-ND based scan-to-map matching based mapping algorithm proposed by Ryu et al. [139]. The resulting algorithm produces maps that are more accurate than the existing NDT based mapping methods. This work was aimed at comparing the accuracy of the proposed mapping algorithm discussed before with a state-of-the-art scan-to-map matching based mapping algorithm.

This algorithm follows the generic scan-to-map matching methodology. A brief overview of the algorithm is as follows:

In NDT based techniques, the environment map is represented using a set of grids with each cell containing a normal distribution which represents all scan points in that grid cell. Figure 4.3 shows an example map and the grid structure. For some of the cells, in addition to the main normal distribution, multiple smaller map normal distributions are initiated to make the algorithm viewpoint invariant.

Initially robot odometry is used to determine the transform between  $\{R\}$  and  $\{R^-\}$ ,  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}'_k$ . However, as an additional refinement step, ICP is performed between the new

<sup>1</sup>We originally published this work in “Grid-based scan-to-map matching for accurate 2D map building” [36]

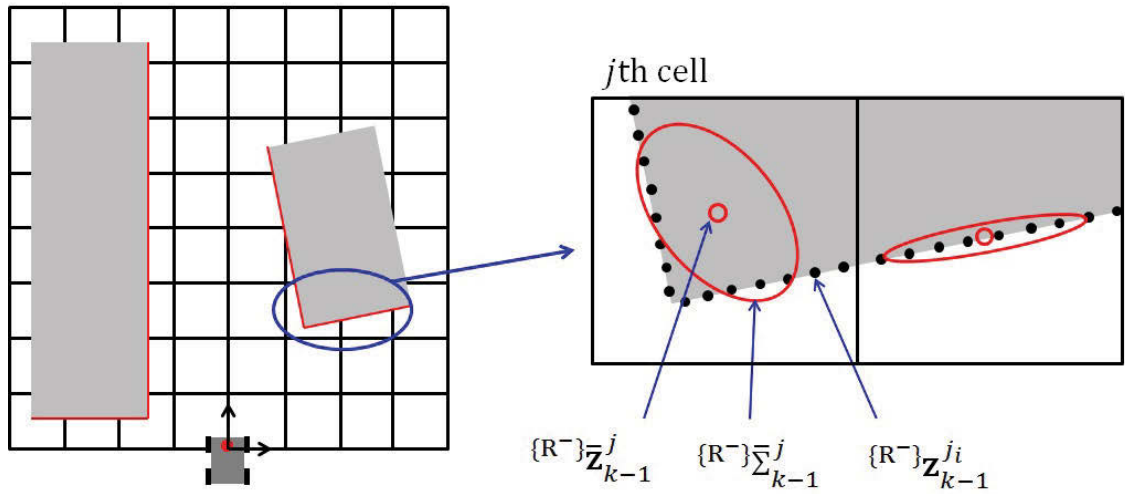


FIGURE 4.3: Two-dimensional grid space overlapped with the environment (left) and the normal distribution based on scan points in the  $j^{\text{th}}$  cell (right)

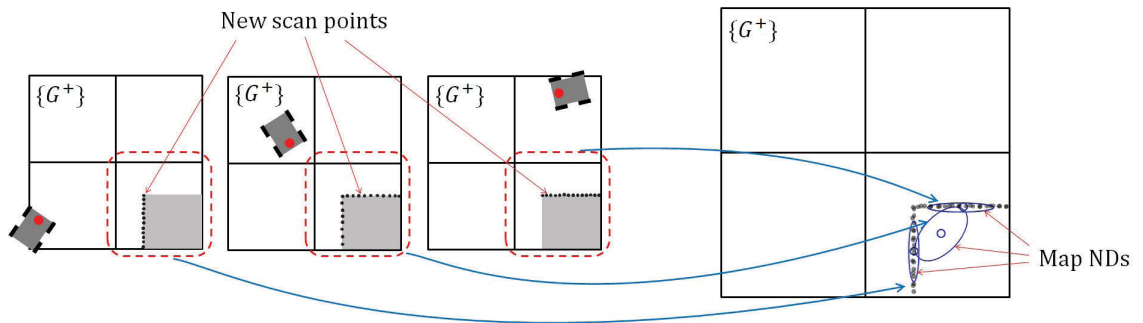


FIGURE 4.4: The grid map is represented by multiple normal distributions (right) and new scans to be matched to the grid map (left)

scan and the previous scan and the transformation is updated to  $\begin{Bmatrix} R^- \\ R \end{Bmatrix} \mathbf{p}_k^{\text{ICP}}$ . Once the points are transformed to the  $\{G\}$  frame using Equation (4.2), a ND-to-ND based global correction is performed.

Figure 4.4 illustrates the grid map with multiple map normal distributions in each cell together with possible new scans that would correspond to that map normal distribution. As shown in the figure, the new scan of an object can be significantly different depending on where the scan is taken, therefore one normal distribution would not be able to capture every viewpoint. The grid map with multiple normal distributions allows the matching of the new scan to a map normal distribution irrespective of the robot viewpoint.

With the new scan in the  $\{G\}$  frame, it can be split into the same grid cells and a new set of normal distributions are created for this scan.

The selection of a matching map normal distribution that corresponds to the scan normal distribution starts with quantifying the similarity of the scan normal distribution to each map normal distribution in the same cell. The similarity score can be computed by Kullback-Leibler (KL) divergence,  $D_{KL}$ , which is a mathematically solid method for measuring the distance between two probability distributions:

$$\begin{aligned}
S\left(\mathcal{N}(\{G\}\bar{\mathbf{z}}_k^j, \{G\}\bar{\Sigma}_k^j), \mathcal{N}(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l})\right) \\
&= -D_{KL}\left(\mathcal{N}(\{G\}\bar{\mathbf{z}}_k^j, \{G\}\bar{\Sigma}_k^j) \parallel \mathcal{N}(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l})\right) \\
&= -\frac{1}{2} \left\{ \text{tr}\left((\bar{\Sigma}_{1:k-1}^{j_l})^{-1}\{G\}\bar{\Sigma}_k^j\right) + (\bar{\mathbf{z}}_{1:k-1}^{j_l} - \{G\}\bar{\mathbf{z}}_k^j)^\top (\bar{\Sigma}_{1:k-1}^{j_l})^{-1} \right. \\
&\quad \left. (\bar{\mathbf{z}}_{1:k-1}^{j_l} - \{G\}\bar{\mathbf{z}}_k^j) - \ln \frac{\det(\{G\}\bar{\Sigma}_k^j)}{\det(\bar{\Sigma}_{1:k-1}^{j_l})} - \lambda \right\} \quad (4.5)
\end{aligned}$$

where  $l \in \{1, \dots, n_{k-1}^j\}$ ,  $\lambda$  is the dimension of the normal distributions, and  $\mathcal{N}(\{G\}\bar{\mathbf{z}}_k^j, \{G\}\bar{\Sigma}_k^j)$  and  $\mathcal{N}(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l})$  are the scan normal distribution and the  $l^{\text{th}}$  map normal distribution, respectively.

Out of several map normal distributions, the most similar one to the scan normal distribution is the one with the highest similarity value. For the  $j^{\text{th}}$  cell, this can be expressed as:

$$\min \left\{ S\left(\mathcal{N}(\{G\}\bar{\mathbf{z}}_k^j, \{G\}\bar{\Sigma}_k^j), \mathcal{N}(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l})\right) \mid \forall l \in \{1, \dots, n_{k-1}^j\} \right\} \quad (4.6)$$

The optimum distribution from Equation (4.6),  $l^{\text{th}}$  map normal distribution, is regarded as the matching map normal distribution for the scan normal distribution if the similarity is greater than the specified threshold parameter,  $\gamma$ :

$$S\left(\mathcal{N}(\{G\}\bar{\mathbf{z}}_k^j, \{G\}\bar{\Sigma}_k^j), \mathcal{N}(\bar{\mathbf{z}}_{1:k-1}^{j_{l^*}}, \bar{\Sigma}_{1:k-1}^{j_{l^*}})\right) > \gamma \quad (4.7)$$

which can be tuned experimentally.

After the matching map normal distribution is identified for each scan normal distribution, it is possible to estimate  $\begin{Bmatrix} G^+ \\ G \end{Bmatrix} \mathbf{p}_k$  by matching all the scan normal distributions to the corresponding matching map normal distributions with the use of the objective function given in Equation (4.8) which is the sum of similarity  $S$  between the scan normal distribution and map normal distribution.

$$\underset{\begin{Bmatrix} G^+ \\ G \end{Bmatrix} \mathbf{p}_k}{\operatorname{arg\,max}} \sum_j S\left(\mathcal{N}(\bar{\mathbf{z}}_k^j, \bar{\Sigma}_k^j), \mathcal{N}(\hat{\mathbf{z}}_k^j, \hat{\Sigma}_k^j)\right) \quad (4.8)$$

After estimating the corrected transform  $\begin{Bmatrix} G^+ \\ G \end{Bmatrix} \mathbf{p}_k$ , the scan is transformed to the new coordinate frame  $\{G^+\}$  using Equation (4.1). Now the new scan can be merged to the map using a weighted mean as given in Equation (4.9).

$$\bar{\mathbf{z}}_{1:k}^{j*} = \frac{m_{1:k-1}^j \bar{\mathbf{z}}_{1:k-1}^{j*} + m_k^j \bar{\mathbf{z}}_k^j}{m_{1:k-1}^j + m_k^j} \quad (4.9)$$

$$\bar{\Sigma}_{1:k}^{j*} = \frac{m_{1:k-1}^j \bar{\Sigma}_{1:k-1}^{j*} + m_k^j \bar{\Sigma}_k^j}{m_{1:k-1}^j + m_k^j} \quad (4.10)$$

where  $m^j$  is the number of scan points in the  $j^{\text{th}}$  cell, which is also updated to include the new number of points after the merger.

## 4.6 Experimental Results

In this section we compare the distance function based mapping algorithm with other state-of-the-art algorithms. For this, we used several datasets which included public domain, simulation and datasets we collected in real crowded environments. We present results for qualitative comparison as well as quantitative comparison using established error metrics.

The following algorithms were used for comparing the performance of the mapping technique proposed in Section 4.4.

1. Grid-based ND-to-ND Scan-to-Map Matching from Section 4.5.
2. Incremental ICP scan matching [142, 143].
3. GMapping [34]

#### 4.6.1 Pose accuracy

First, we adopt the relative relations comparison proposed by Kümmerle et al. [144] in our evaluation. This metric compares the relative relations of two poses from the trajectory with near ground-truth relations. When  $\delta_{i,j} = \mathbf{x}_j \ominus \mathbf{x}_i$  at two instances  $i$  and  $j$  in the robot trajectory, the error metric  $\varepsilon(\delta_{i,j})$  is given by,

$$\varepsilon(\delta_{i,j}) = \delta_{i,j} \ominus \delta_{ij}^* \quad (4.11)$$

Here,  $\delta_{ij}$  are the relations estimated by the mapping algorithm, while  $\delta_{ij}^*$  are near ground-truth reference relations obtained by the authors of the datasets by manually matching scans. The operator  $\ominus$  is the inverse of the standard motion composition operator as defined by Lu & Milios [136], which in the 2D case is given by Equation (4.12).

$$\delta_{a,b} = \mathbf{x}_b \ominus \mathbf{x}_a = \begin{pmatrix} [R(\phi_b)^\top] & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_b - x_a \\ y_b - y_a \\ \phi_b - \phi_a \end{pmatrix} \quad (4.12)$$

where  $R(\phi_b) = \begin{pmatrix} \cos \phi_b & -\sin \phi_b \\ \sin \phi_b & \cos \phi_b \end{pmatrix}$  is the 2D rotation matrix for yaw  $\phi_b$  around the  $z$  axis.

We can present translational ( $trans(\cdot)$ ) and rotational ( $rot(\cdot)$ ) components of the collective error as mean of absolute errors and RMS error forms with the use of Equation (4.13) and Equation (4.14).

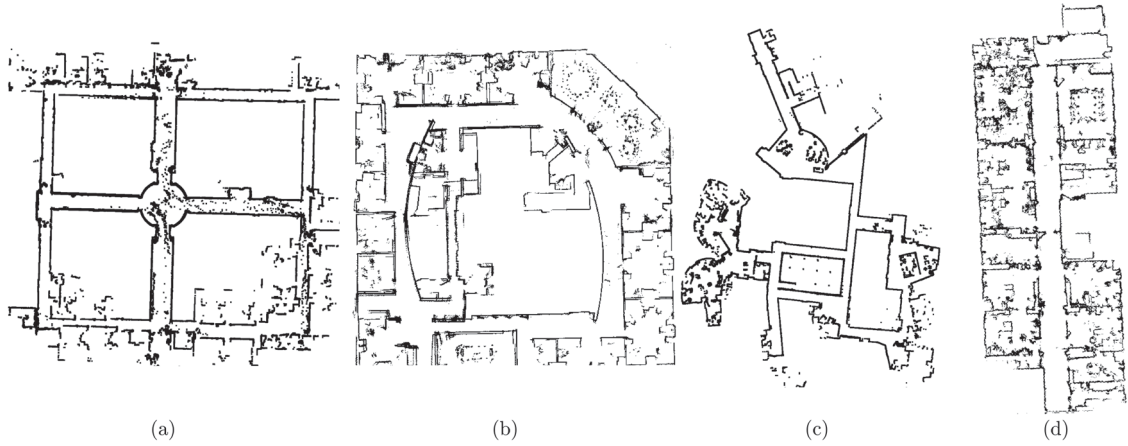


FIGURE 4.5: Maps generated using the proposed CD mapping algorithm for four public domain datasets. (From left to right: ACES Building at the University of Texas, Intel Research Lab Seattle, MIT CSAIL Building Boston, and building 079 at University of Freiburg.)

$$\begin{aligned}\bar{\varepsilon}(|\delta_{i,j}|) &= \frac{1}{N} \sum_{i,j} |\varepsilon(\delta_{i,j})| \\ &= \frac{1}{N} \sum_{i,j} |\text{trans}(\varepsilon(\delta_{i,j}))| + \frac{1}{N} \sum_{i,j} |\text{rot}(\varepsilon(\delta_{i,j}))| \quad (4.13)\end{aligned}$$

$$\begin{aligned}\text{RMS}(\varepsilon(\delta_{i,j})) &= \sqrt{\frac{1}{N} \sum_{i,j} \varepsilon(\delta_{i,j})} \\ &= \sqrt{\frac{1}{N} \sum_{i,j} \text{trans}(\varepsilon(\delta_{i,j}))} + \sqrt{\frac{1}{N} \sum_{i,j} \text{rot}(\varepsilon(\delta_{i,j}))} \quad (4.14)\end{aligned}$$

The following four public domain datasets were used for this evaluation: ACES Building at the University of Texas, Intel Research Lab Seattle, MIT CSAIL Building Boston, and building 079 at University of Freiburg. These datasets were selected because the near ground-truth relations,  $\delta_{i,j}^*$  for the datasets, obtained using manual scan matching, are also available in the public domain [145]. The results of other algorithms we use for comparison are quoted from Kümmerle et al. [144] and Burgard et al. [146]. Table 4.1 presents our results together with two other algorithms: Incremental scan matching using a variant of ICP based on work by Censi [147] and GMapping based on a RBPF for learning grid maps by Grisetti, Stachniss, & Burgard [34] available on OpenSLAM.org [148].

TABLE 4.1: Quantitative Results of Different Approaches/datasets

Trans. error(m)	CD Mapping	Scan Matching*	GMapping (50 particles)*
Aces (abs)	0.024 ± 0.037	0.173 ± 0.614	0.060 ± 0.049
Aces (RMS)	0.050 ± 0.094	0.638 ± 1.651	0.077 ± 0.105
Intel (abs)	0.019 ± 0.042	0.220 ± 0.296	0.070 ± 0.083
Intel (RMS)	0.048 ± 0.102	0.369 ± 0.526	0.105 ± 0.184
CSAIL (abs)	0.019 ± 0.025	0.106 ± 0.325	0.049 ± 0.049 †
CSAIL (RMS)	0.032 ± 0.058	0.342 ± 0.853	0.071 ± 0.114 †
FR 79 (abs)	0.026 ± 0.172	0.030 ± 0.427	0.061 ± 0.044 †
FR 79 (RMS)	0.041 ± 0.391	0.066 ± 0.829	0.078 ± 0.141 †

Rot. error (deg)	CD Mapping	Scan matching*	GMapping (50 particles)*
Aces (abs)	0.3 ± 0.3	1.2 ± 1.5	1.2 ± 1.3
Aces (RMS)	0.4 ± 0.6	1.9 ± 3.3	1.8 ± 2.8
Intel (abs)	0.3 ± 0.4	1.7 ± 4.8	3.0 ± 5.3
Intel (RMS)	0.5 ± 1.0	5.1 ± 13.1	6.1 ± 13.7
CSAIL (abs)	0.4 ± 1.1	1.4 ± 4.5	0.6 ± 1.2 †
CSAIL (RMS)	1.2 ± 4.6	4.7 ± 10.5	1.4 ± 4.2 †
FR 79 (abs)	0.4 ± 0.4	1.7 ± 2.1	0.6 ± 0.6 †
FR 79 (RMS)	0.6 ± 1.1	2.7 ± 3.8	0.8 ± 1.4 †

\* Results quoted from [146]. † Scan matching has been applied as a pre-processing step.

For the second evaluation we use a simulated dataset generated in the Intel Research lab environment as the original Intel research lab dataset does not have ground-truth to evaluate the results against. This dataset has the same noise parameters as stated in Table 3.1.

In Figure 4.6, we present the resultant maps from three algorithms.

Table 4.2 compares the pose output of each algorithm to the ground-truth. For the comparison, we choose the method proposed by Kümmerle et al. [144] as it accounts for rotations in the map. We have randomly chosen 500 pose pairs to compare the relations.

Moreover, as this is a simulated dataset, we can compare errors during the whole mapping process as we have access to ground-truth. Figure 4.7 shows the relevant errors while the Table 4.3 a summary of the errors. The plots clearly show that the errors do not accumulate over time as they do with scan-to-scan incremental ICP mapping methods,

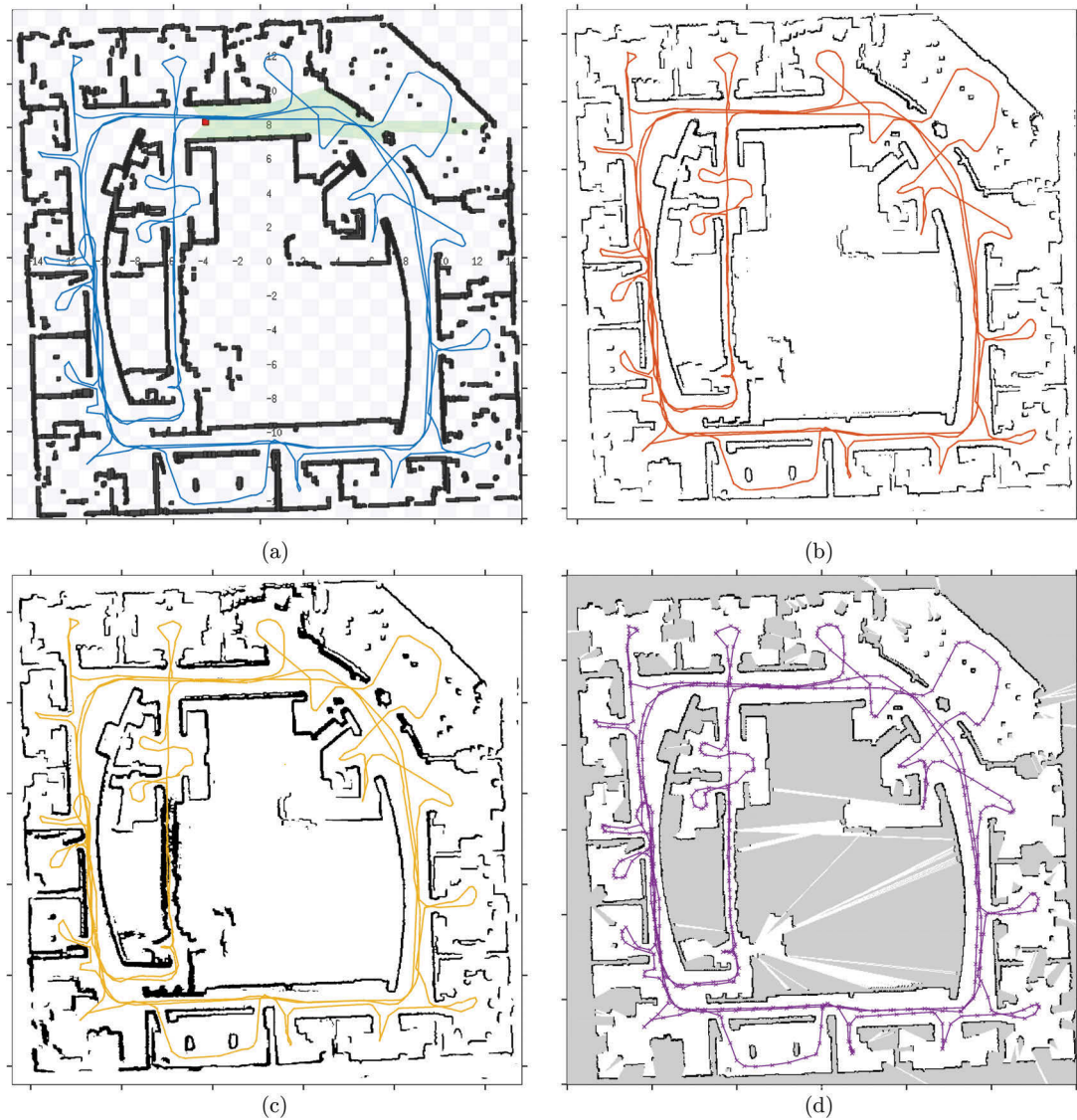


FIGURE 4.6: Presents results of three algorithms from the simulation run on Intel research labs map. (a) ROS Stage simulation environment and ground-truth, (b) resultant map and poses from Distance Function based mapping algorithm, (c) resultant map and poses from ND-to-ND mapping algorithm, and (d) Results from GMapping, for comparison. (GMapping only provides poses for places marked with ‘x’)

TABLE 4.2: Quantitative results of different approaches on simulated Intel dataset, for 500 random relations.

Algorithm	Trans. error(m)		Rot. error(deg)	
	Mean(abs)	RMS	Mean(abs)	RMS
CD mapping	$0.1852 \pm 0.4281$	$0.4282 \pm 0.8480$	$1.9396 \pm 3.2517$	$3.2506 \pm 4.3310$
ND-to-ND	$0.2302 \pm 0.4437$	$0.4436 \pm 0.8422$	$2.2676 \pm 3.4026$	$3.4020 \pm 4.4155$
GMapping	$0.7619 \pm 1.3431$	$1.3427 \pm 2.2948$	$7.8637 \pm 10.4681$	$10.4699 \pm 13.3662$
ICP	$1.0652 \pm 1.4684$	$1.4762 \pm 2.0037$	$7.4289 \pm 9.4157$	$9.4051 \pm 10.9317$



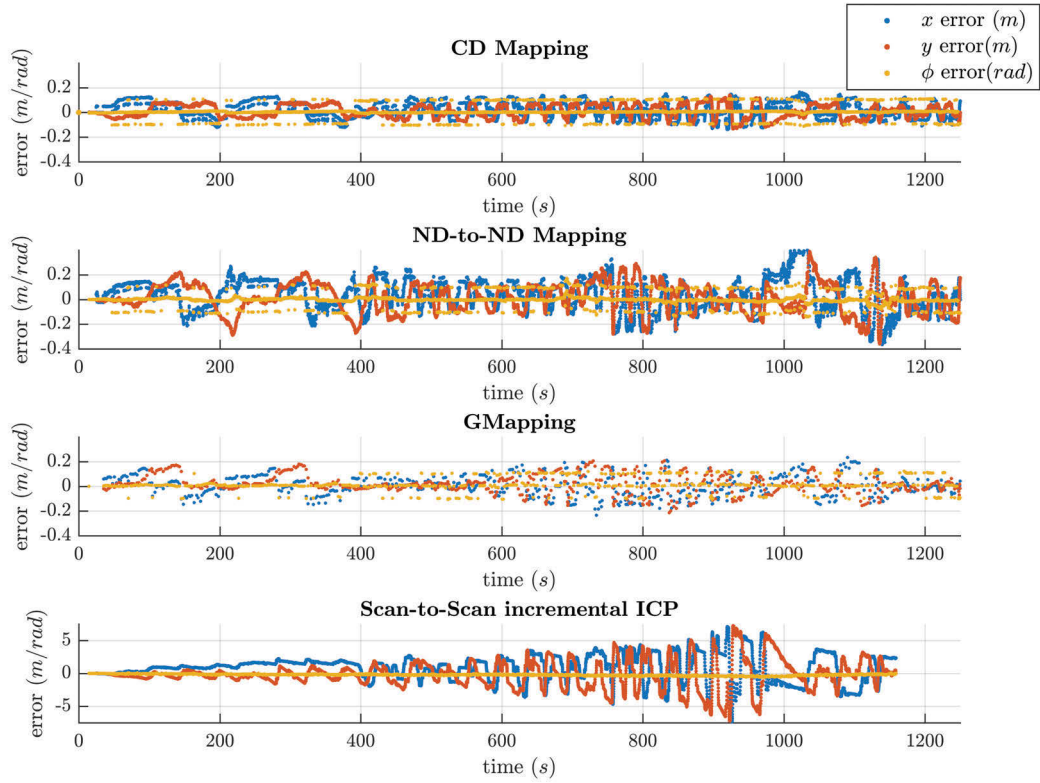


FIGURE 4.7: Errors for all poses for simulated Intel dataset

TABLE 4.3: Quantitative results of different approaches on simulated Intel dataset, for all poses.

Algorithm	Trans. error(m)		Rot. error(deg)	
	Mean(abs)	RMS	Mean(abs)	RMS
CD mapping	$0.0540 \pm 0.0610$	$0.0635 \pm 0.0649$	$1.1360 \pm 2.2849$	$2.3102 \pm 3.4830$
ND-to-ND	$0.1074 \pm 0.1284$	$0.1309 \pm 0.1554$	$1.3483 \pm 2.3777$	$2.3778 \pm 3.5324$
GMapping	$0.0630 \pm 0.0808$	$0.0815 \pm 0.0946$	$2.7425 \pm 3.7186$	$3.8402 \pm 4.1452$
ICP	$1.6646 \pm 2.1226$	$2.1546 \pm 2.6783$	$12.4516 \pm 6.8402$	$14.1977 \pm 13.9609$

which is evidence of implicit loop-closure. However, as seen between times 0 – 200s in the plot, the error keeps on growing until the loop-closure point is reached.

#### 4.6.2 Mapping Real Environments

We used the proposed mapping algorithm to generate maps in several real environments. The data for mapping was collected using the instrumented walker revision 2 shown in

Figure 1.1. It is equipped with a Hokuyo UTM-30LX laser range finder sensor, and the rear wheels are fitted with wheel encoders to log odometry.

The map shown in Figure 4.8 is obtained from the Ground floor of the Roselands Shopping centre, Roselands NSW Australia. The data collection was done in multiple runs starting from different locations on the same floor. The poses are marked in different colours for each individual run. When the robot start pose is at an unknown location, we initially use the optimisation based localisation method introduced in Section 3.4 to localise the robot on the existing part of the map, and then continue to expand the map after the robot successfully localises itself.

Figure 4.8a shows a zoomed in section of the place where the loop is completed after approximately 250m of travel. The path has accumulated approximately 1m of error during this loop. Furthermore, the trajectory amongst the aisles of the lower-left corner does not revisit previous places which causes the error to accumulate slowly, making the map inconsistent. Though the algorithm performs implicit loop-closures for shorter displacements and smaller loops, a large loop which has a prominent displacement like the one shown in Figure 4.8, requires active detection and adjustment of the whole trajectory in order to perform loop closure, which is not performed in the algorithm we propose. This demonstrates a common limitation of scan-to-map matching techniques and the need for SLAM in large scale environments.

Furthermore, the collection of data was done during normal working hours and the shopping centre was crowded during this time. Figure 4.9 shows a sparse long exposure image which highlights the crowd movement, so that the performance of the algorithm in clutter can be qualitatively evaluated.

## 4.7 Conclusion

This chapter introduces a scan-to-map matching based mapping method which combines and enhances the sensor model and the distance function based environment representation proposed in Chapter 2. The algorithm has a number of advantages compared to the existing algorithms, such as the ability to function without other inputs such as odometry,

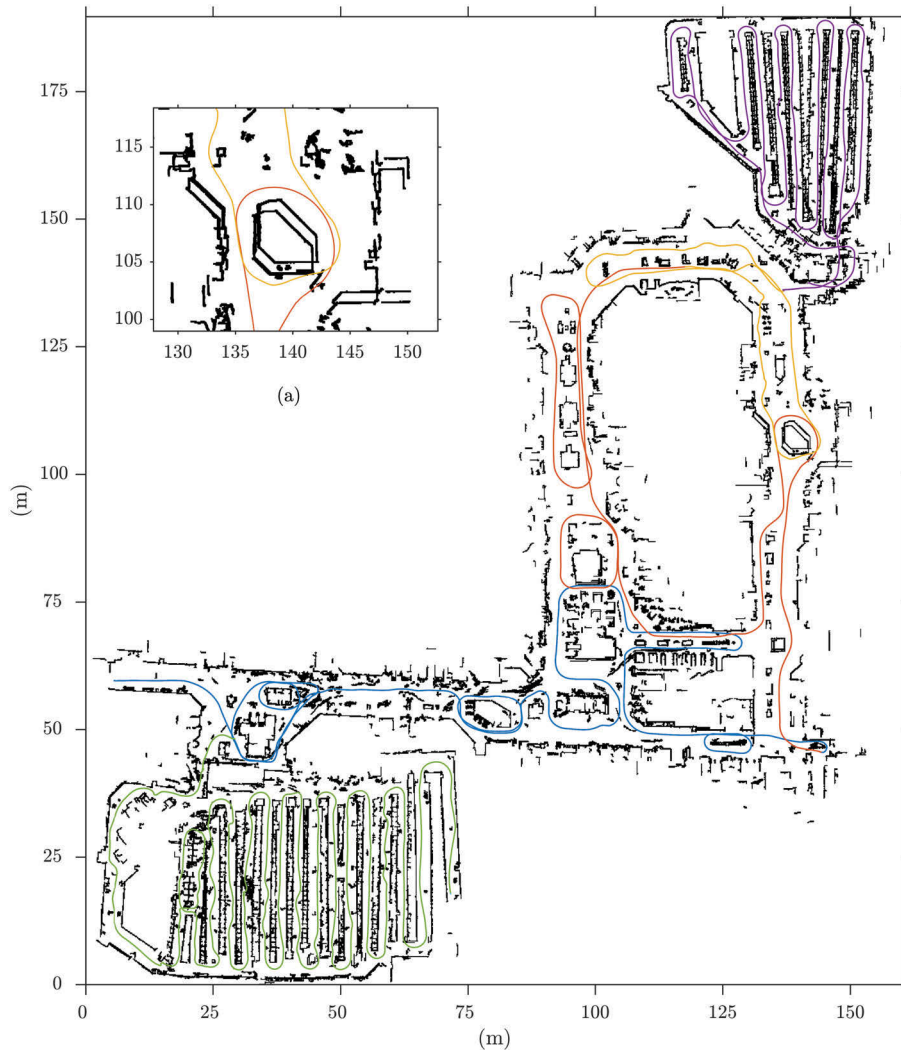


FIGURE 4.8: Ground floor at Roselands Shopping Centre, Roselands, NSW, Australia.  
(a) Zoomed view of the end of the large loop.

light weight computation requirements, and having only one tuning parameter ( $\zeta$ , defined in Equation (3.11)).

We used the *SLAM benchmarking metric* proposed by Kümmerle et al. [144] and Burgard et al. [146] to evaluate the algorithm using a number of public domain and simulated datasets. The results from the algorithm are compared with GMapping, which is an improved RBPF SLAM technique, as well as the extended version of ND-to-ND scan-to-map matching, which we proposed in Section 4.5. This work is an extension of one of

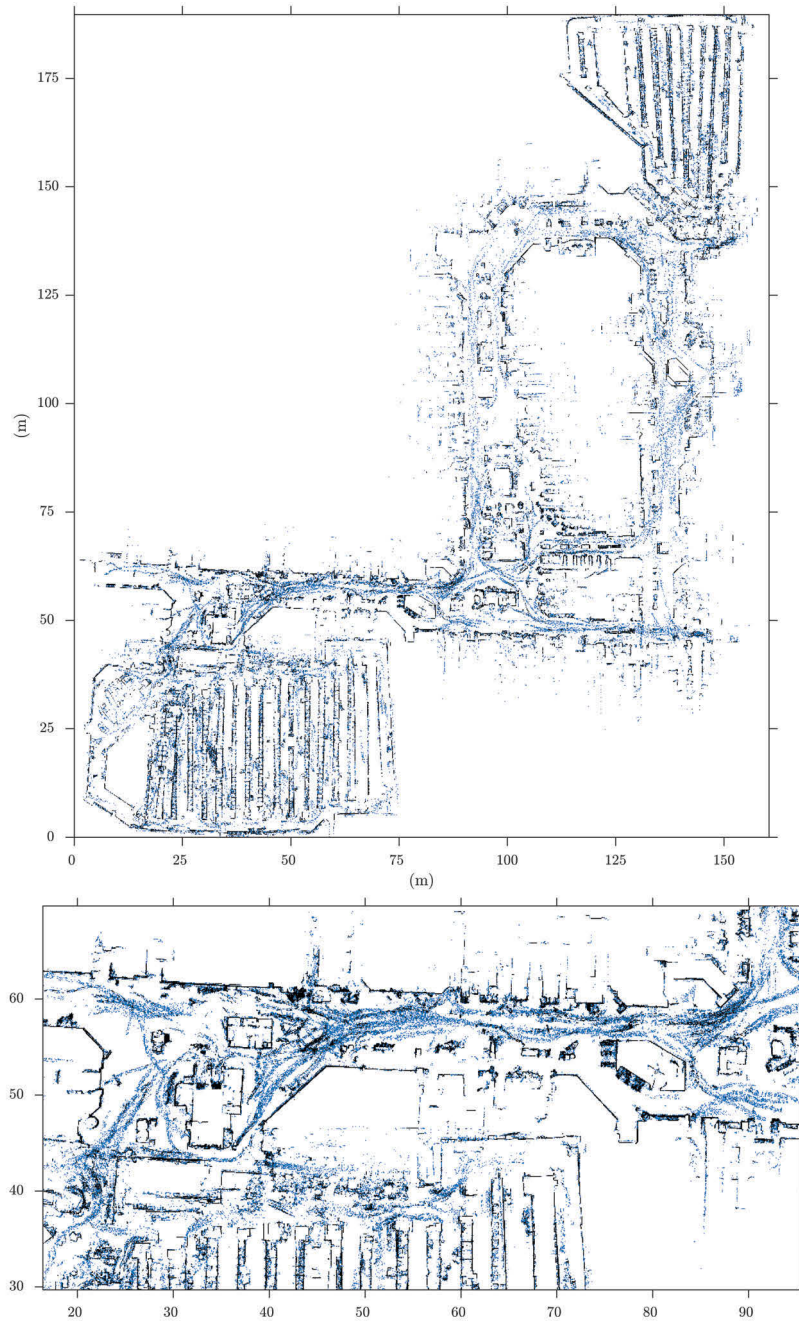


FIGURE 4.9: Long exposure time-lapse image showing movement of crowd during collection of data.

the state-of-the-art normal distribution based scan-to-map matching methods presented by Ryu et al. [139].

We further demonstrate that the proposed mapping technique can be used to generate maps even in highly cluttered environments with very good accuracy, even without odometry. As shown in Section 3.6.1 the distance function maps generated from the algorithm can be readily used for localisation, or can easily be exported to an OGM for other tasks such as path planning.

## Chapter 5

# Control of Assistive Robotic Robots

### 5.1 Introduction

Control of a robot is the process of determining the forces and torques that are required by the actuators in order for the robot to go to a desired position, follow a desired trajectory, perform a task or achieve a low-level goal with desired performance needs [149]. For an autonomous robot, this task becomes complicated as most of this process should happen without human intervention using the information such as the map of the environment, the localisation of the robot, and sensor measurements which the robot may have at its disposal to reach high-level goals. The robot might also be subject to constraints and sub-goals either to optimise the process or to prevent undesired consequences such as accidents or the robot losing the position information (i.e. localisation).

In the case of assistive robots, the complexity of robot control further increases as they need to closely interact with humans in order to assist them do the desired task. Another major concern is the fact that most of the operators of assistive robots may not have a technological background and therefore the interface for interaction with the operator should be simple and intuitive. This not only means that the robot controller should understand the user need using simple cues, but also discern the user intention accurately

and quickly. Furthermore, as the assistive robots replace a variety of other existing assistive devices, the mechanism of human user interaction is highly unique to a given class of applications, and is difficult to generalise.

In this chapter, we present a framework for developing a controller and a user interaction system for assistive robots. We take as an example, the process that we followed to optimally design an assistive robotic hoist (hereafter referred to as the *Smart Hoist*), and also report the results of user studies that accompanied the development process. Throughout the entire design process, the end user of the robot was an integral member of the design team. This ensured that the control interface is intuitive, easy to understand and use, which is a novel strategy for designing assistive robots. Although we focus on developing an assistive robotic hoist, the framework we use is more generic and answers broader questions that might pertain to the development of control strategies for assistive robotic devices.

This chapter is organised as follows: The next section presents the background for the design of the Smart Hoist. Section 5.2 discusses the collaborative design approach we use in building the Smart Hoist, while Section 5.3 gives a brief description of the user intention recognition method used in the Smart Hoist which also explains how we map the user input appropriately to the system output. This section will also demonstrate how the influence of hardware and mechanical limitations were mitigated by understanding the main user operation patterns and requirements. Section 5.4 presents results of experiments conducted to validate the model in practice. We also present a basic evaluation done using Electromyography (EMG) data that suggests assisted manoeuvring actually reduces strain on the musculoskeletal system, and therefore may reduce injuries. In Section 5.5 we present a discussion about the experience of users from a set of trials conducted at an IRT retirement village in Woonona, NSW, Australia. Section 5.6 concludes the chapter.

## 5.2 Background

Due to the increasing demand for aged care services and the continuing decline in the relative availability of informal carers, the scarce trained aged care workforce often find

itself overworked to meet community expectations. This, combined with the high rate of work-related musculoskeletal injuries amongst carers [150–153] and injuries to non-ambulatory residents occurring during transfer (e.g. bed to chair, chair to toilet and bath), gives rise to significant costs and health and safety risks. Therefore, it is equally important to assist carers in order to improve safety and the quality of care services. A report by the Academy of Technological Sciences and Engineering titled “Smart Technology for Healthy Longevity” [154, 155] canvassed various options based on the use of innovative technologies to address this challenge.

With this in mind we initiated a case study to evaluate how assistive robotics can be used to alleviate these issues. We approached Illawarra Retirement Trust (IRT) Research Foundation to discuss avenues for introducing assistive robotics to the work-flow of their age care facilities. A working group was formed which consisted of the care facility management and the operational staff of IRT to put forward suggestions that could potentially help decrease the injury rate amongst carers.

A patient hoist or a patient lifter is the main device that is used to transfer non-ambulatory residents in aged care, disability care and hospital facilities. Shown in Figure 5.1, a standard lifter, is usually equipped with a motorised or a mechanical hoist system. However, moving the hoist is a manual process aided by four swerve drive caster wheels attached to the base of the hoist, which enables holonomic motion. Though manual motion is easy and effortless to operate when the device is not loaded, with a patient on the sling the carers need to exert a considerable amount of force and effort to manoeuvre the hoist, especially when in-place rotations take place.

We suggested to augment the standard hoist by embedding robotic capabilities to assist the carers while they perform the transfer. The final advanced prototype, the Smart Hoist, is a standard hoist integrated with sensors to capture the user intention along with other sensory data about the environment, on-board computing to make decisions, and actuators to provide assistance. Design considerations and system architecture of this Smart Hoist are detailed in Appendix A.

User Centred Design (UCD) and Co-Design [156] are common approaches that solicit end users’ views in design of products. Typically, the design process has four stages: (i)





FIGURE 5.1: Standard Joey patient hoist from Haycomp Pty. Ltd.

research, (ii) concept development, (iii) detailing, and (iv) implementation. In UCD, the designer views the end user as the subject and he/she takes authoritative decisions and controls all four stages, while using the end users as subjects to gather data necessary for the design process. This has been the main approach of solving problems related to robotics systems [157–159].

On the other hand, Co-Design approach consider the end user as a part of the design team and the designers rely on users' expertise on the domain knowledge. During the design process, the designer no longer makes high-level design decisions. That means he/she acts as a mediator while designing cooperatively with end users [160]. Although the Co-Design approach is not widely used in designing robotic systems yet, there are a few examples of its use in human robot interaction [161, 162]. As assistive robots closely interact and associate with human operators, the design process of such robots can greatly benefit by integrating end users in the design team.

The process we followed for designing the Smart Hoist was inspired by the principles of Co-Design, with the end users (both carers and residents) always in the loop for every

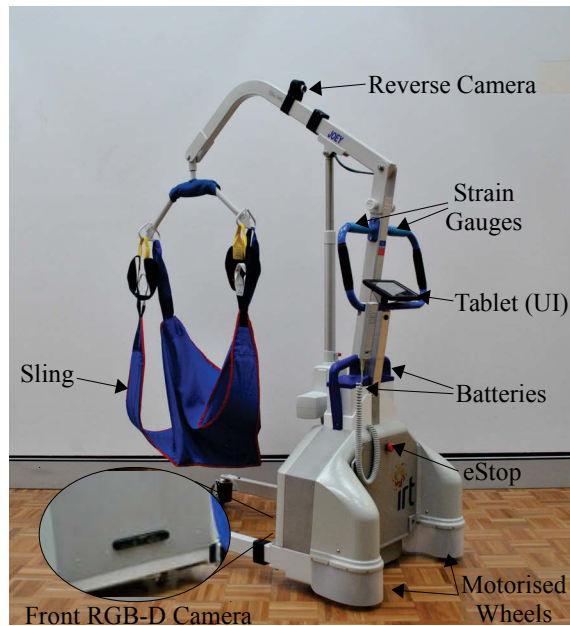


FIGURE 5.2: UTS-IRT Smart Hoist.

design decision. The working group consisted of 10-15 staff personnel of IRT Woonona Facility, which included nurses, carers, physiotherapists, occupational health and work safety coordinators and a few residents. Facility management were involved in a few of these sessions where regulatory matters were being discussed. The nature of the design workshops were more interactive and engaging, and for some components such as the batteries and display, we used mock-ups and cardboard prototypes to narrow-down user requirements.

Figure 5.2 shows a diagram of the Smart Hoist while an overview of the final system can be found in the Appendix A.

### 5.3 User Intention Recognition

Use of a basic control pad/joystick is the most common way of controlling robots. However, at the workshops the users did not favour the use of a control pad mainly due to the nature of the device operation and ethical concerns. Therefore, a major requirement in the design process was to ensure the Smart Hoist mimics the manoeuvring method of a standard hoist as closely as possible. To achieve this, the Smart Hoist must be capable of recognising

operator activity through the handle bars which is the only point of contact between the operator and the Smart Hoist. As needed, the automatic actuation mechanism triggers the motors to provide the required assistance, so that the operator feels they are pushing a less heavy object which is essentially a change to the inertial properties of the system to mimic a lower mass.

Though the retrofitted wheels are equipped with the swerve capability, the response time is not as fast as the caster wheels due to physical constraints. This voids the easy holonomic motion as the standard hoist. So another key consideration was how to address this loss of freedom of motion so that the users have the least impact while fulfilling the essential motions.

### 5.3.1 Admittance Controller

A simple admittance control strategy<sup>1</sup> is used to control the Smart Hoist's linear motion. The admittance controller aims to generate motion commands such that the system has an equivalent mass of  $m$  and an inertia of  $I$  irrespective of the actual inertial properties of the system.

The motion of the system in response to a force  $F$  from the user is governed by Equation (5.1) which can be used to generate a velocity command,  $v$ .

$$F = m \cdot \dot{v} + C \cdot v \quad (5.1)$$

where  $C$  is the desired damping coefficient.

This strategy is only effective if the underlying velocity controller closely follows the input velocity commands under different loads. The motor controller was experimentally verified to uphold this condition.

Similarly, for torque inputs of  $\tau$  around the handles'  $z$  axis, we can use Equation (5.2) to obtain the angular velocity  $\omega$  when the desired angular damping coefficient is  $C_\omega$ .

<sup>1</sup>We initially published this work in "Smart Hoist: An Assistive Robot to Aid Carers" [163]

$$\tau = I \cdot \dot{\omega} + C_{\omega} \cdot \omega \quad (5.2)$$

During the user trials, we experimentally tuned the parameters  $m$ ,  $I$ ,  $C$ , and  $C_{\omega}$ , which constitutes the desired inertial properties of the Smart Hoist, to values the carers were comfortable to use.

### 5.3.2 Classifying the Main Modes of Operation

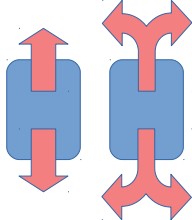
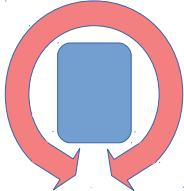
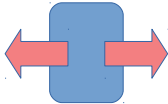
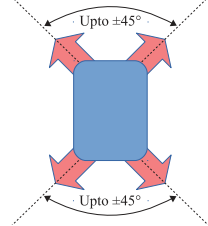
The standard hoist has four swivel caster-wheels, which enables it to have holonomic motion via changes in caster direction. However, the Smart Hoist wheels are not as responsive as the caster wheels. Therefore, we studied main modes of operation of a standard hoist to devise a compromised solution.

We conducted several trials involving IRT carers who regularly do patient transfers using a standard hoist to observe its motion during normal operations. Figure 5.3 shows a typical floor plan of a resident's room at the IRT Woonona care facility. As it can be seen, the rooms and corridors are quite narrow with little or no free space. Even though the standard hoist is holonomic, we could observe a few common modes of motion for the hoist as shown in Table 5.1.

The users of the hoist interact with the device using the two handle bars. As shown in Section A.1.1, we instrumented the handle bars with strain gauges to detect the forces. Essentially we can resolve these forces to lateral linear forces in the  $x$  and  $y$  directions,  $F_x$  and  $F_y$  and torque  $\tau$  around the handles'  $z$  axis.

These linear forces can be combined to a resultant force vector  $F$  in the direction  $\alpha$ . However, though  $F$ ,  $\alpha$ , and  $\tau$  is enough to understand the intended operation mode, the noise in strain signals prevents us from getting a clear estimate of  $\alpha$ . Moreover, the response time of the angle of change in the omni-directional wheels is much slower compared to the changes in direction. Due to these limitations, mapping the intention to the actual motion of the hoist is not straightforward. Therefore, user trials were used

TABLE 5.1: Main modes of motion for a standard hoist

Direction	Usage
 <p data-bbox="419 875 611 974">Forward, backward, and turning</p>	<p data-bbox="692 712 1315 779">Main modes of operation when moving the hoist along corridors and where there is free space.</p>
 <p data-bbox="405 1220 624 1252">In place rotation</p>	<p data-bbox="663 1048 1347 1151">Used while turning in tight spaces. The rooms and corridors in the facility can be small and do not have much space for turning.</p>
 <p data-bbox="438 1496 590 1563">Side-to-side motion</p>	<p data-bbox="716 1339 1294 1406">Used to position the hoist against a bed or a wheelchair.</p>
 <p data-bbox="405 1803 624 1834">Diagonal motion</p>	<p data-bbox="711 1648 1299 1715">Used when going through corridors or narrow doorways.</p>

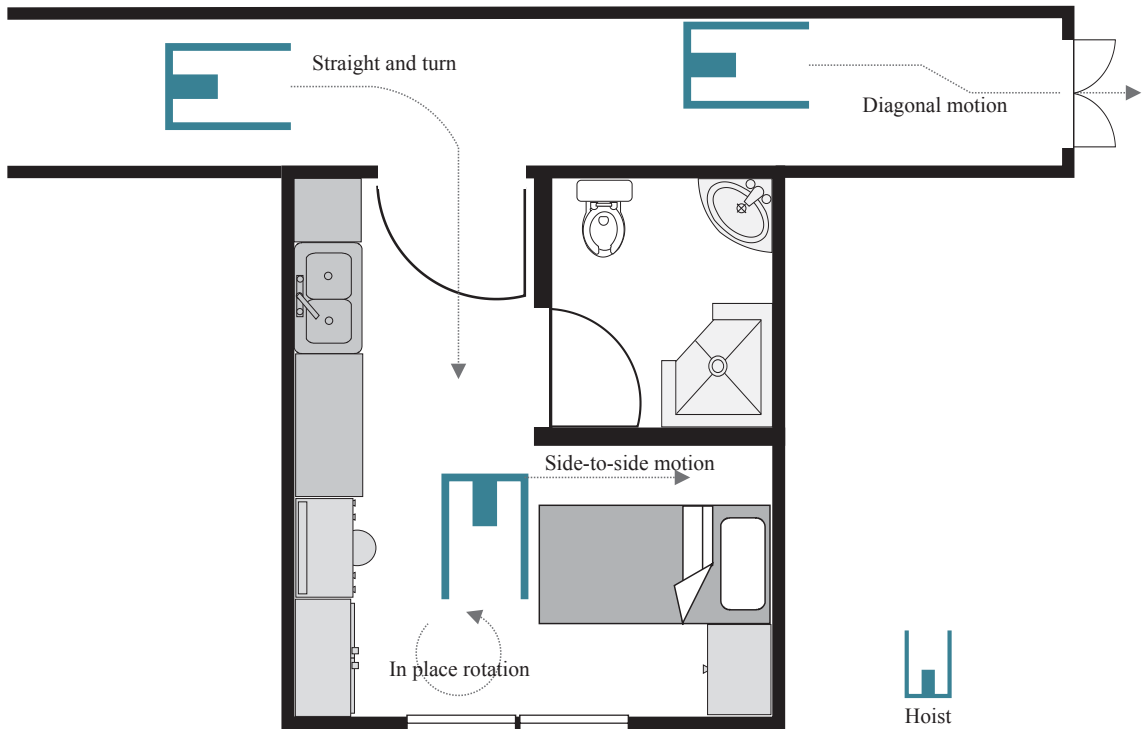


FIGURE 5.3: A typical floor plan of a resident's room at the IRT Woonona care facility.

to experimentally obtain the thresholds illustrated in Figure 5.4 to set the direction of motion.

The forward, reverse, left, and right motions were set to have a  $20^\circ$  dead-band with respect to the input angle, after which the hoist will perform diagonal motion up-to a maximum of  $45^\circ$  from the forward or reverse direction. This is a heuristic classification and the thresholds were experimentally defined during the user trials.

A block diagram of the complete control system is shown in Figure 5.5. The motion logic converts the input motion into the desired drive angles by mapping between the direction of force  $\alpha$  and the desired motion pattern. The navigation assist functionality provides obstacle avoidance capabilities<sup>2</sup>.

<sup>2</sup>We previously published this work in "A Novel Collaboratively Designed Robot to Assist Carers" [164]

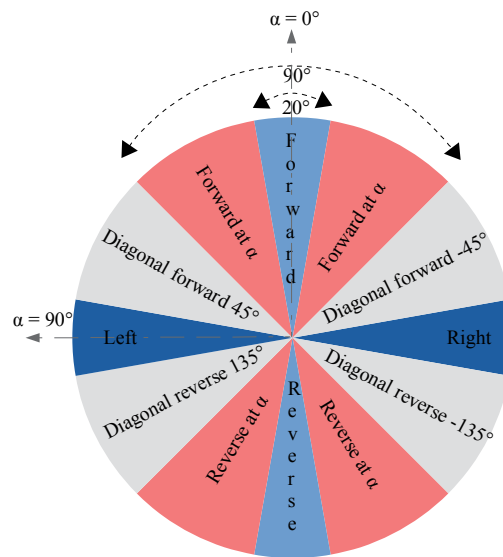
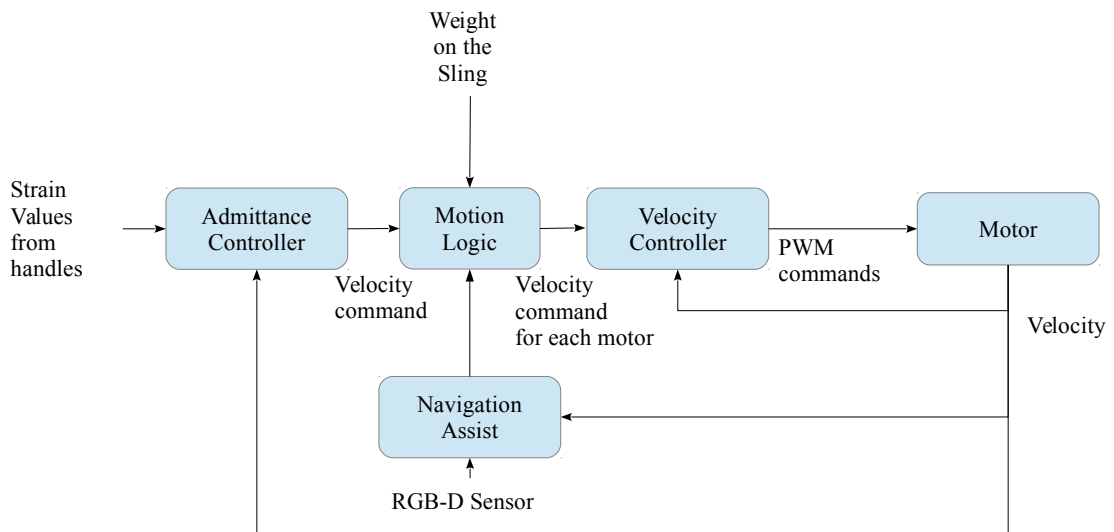
FIGURE 5.4: Mapping the direction of force  $\alpha$  to the direction of motion.

FIGURE 5.5: Block diagram of the Controller.

## 5.4 Experiments and Trials

In the first part of this section, we present a set of experiments conducted to investigate whether the Smart Hoist helps to reduce the musculoskeletal injuries that carers of aged care facilities commonly face during non-ambulatory resident transfers.

Thereafter, we analyse the results collected from the surveys and user trials to help validate our design, which is directly applicable to the broader field of designing controllers for

assistive robots.

We used both quantitative and qualitative methodologies for the evaluations. Quantitative methods include a comparison of forces required to operate the hoist, an analysis of EMG data collected while the subjects operated the hoists. Furthermore, we also present results from a survey based on a questionnaire conducted during the user trials. For the qualitative evaluation we will discuss some key user experiences and feedback in detail. All experiments were conducted with professional carers at the IRT care facility in Woonona, NSW, Australia.

#### 5.4.1 Comparing the Operation of the Smart Hoist vs. the Standard Hoist

The first experiment is a comparison between the normal hoist and the Smart Hoist with and without load. We use an identical standard hoist frame to conduct the experiments. The handles of the standard hoist were also populated with strain gauges to measure the forces exerted on the handles. Few tasks were performed using the same load of  $\approx 75\text{kg}$  (i.e. same subject) on both hoists and the forces on the handles were recorded. Both hoists were manoeuvred on the same floor, and the task was completed approximately in the same time duration. Conducted tasks were pushes, pulls, and in-place turns.

The comparisons presented in Figure 5.6a and Figure 5.6b show that the standard hoist requires forces in the range of 50N-80N per handle to perform push and pull operations, while the Smart Hoist only requires a force close to 20N per handle to perform the same action as the mass parameter  $m$  of the Smart Hoist was fixed to be 40kg. In Figure 5.6a, it can be seen that the forces on the handles are considerably asymmetric when the standard hoist is being manoeuvred. This is usually the case when the hoist is carrying a patient, and extra manoeuvres are required to stabilise a straight linear motion. However, the Smart Hoist avoids this problem as it has a clear linear operation mode. Similarly, as Figure 5.6c shows, performing an in-place turn on the standard hoist requires a much higher force confirming the comments received from the IRT carers during the preliminary discussions. It can be seen that the Smart Hoist can perform this in-place rotation task with much less effort.



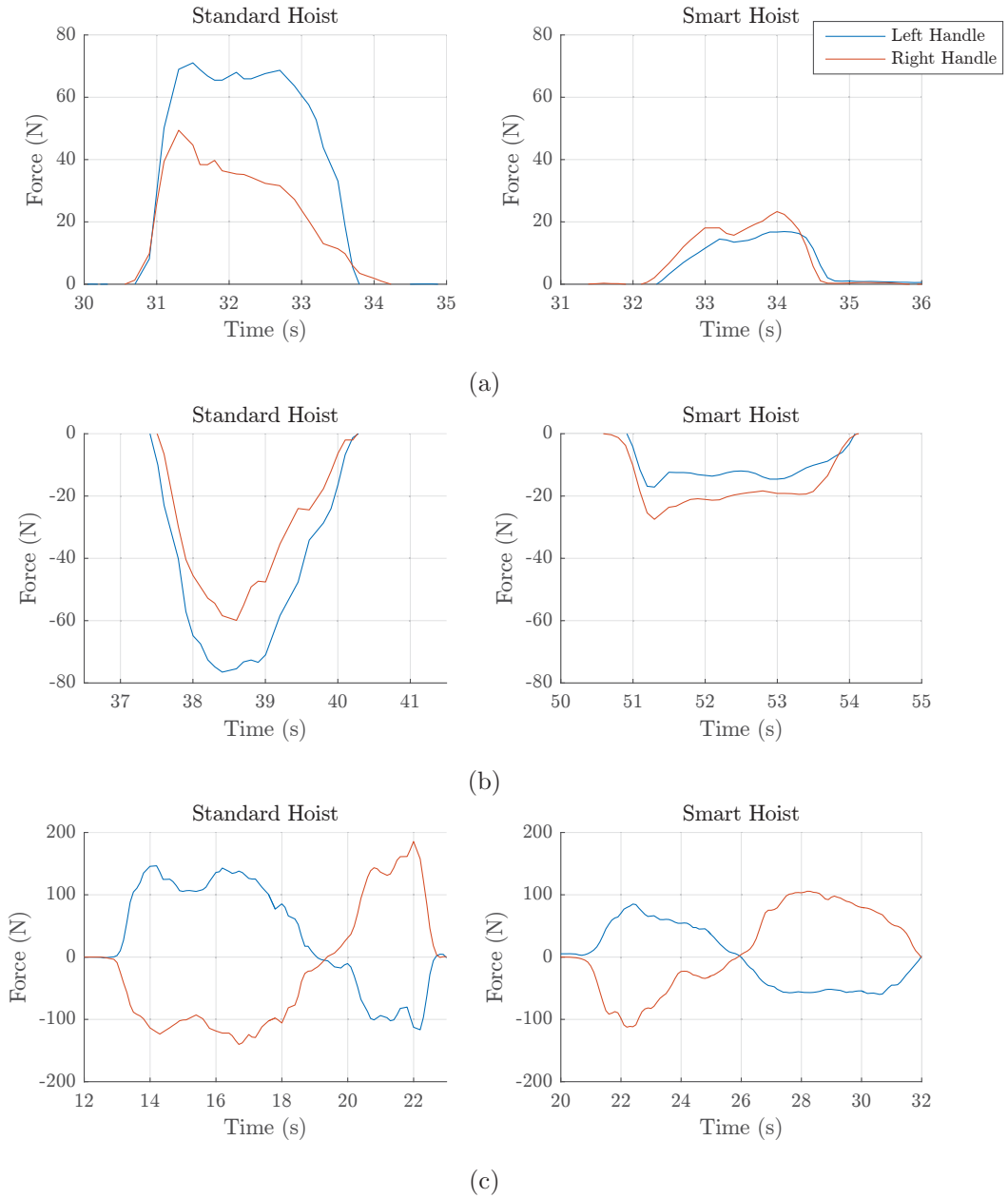


FIGURE 5.6: Comparison of the forces required to do (a) Push, (b) Pull and (c) In-place turn on the standard and Smart Hoists.

Electromyography (EMG) refers to the electrical detection of signals arising from the depolarisation of skeletal muscle. These signals may be detected from skin surface electrodes or from needles placed directly within the muscle [165]. In order to measure the muscle activity while using the Smart Hoist, we used non-invasive EMG data collected using the Delsys<sup>®</sup> EMG system. This system uses wearable electrodes to collect surface EMG signals. The electrodes are placed on the major muscle groups shown in Figure 5.7 that contribute to the above basic actions that are performed while manoeuvring the hoists while they are loaded with a person having a mass of  $\approx 75\text{kg}$ .

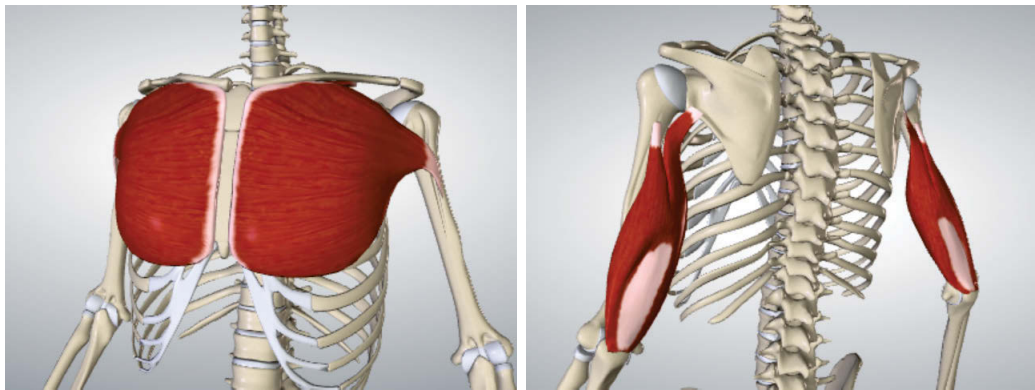
We can observe a clear reduction of muscle activity while using the Smart Hoist while doing both push and pull actions. However, we could not get a clear reading with the twist action mainly due to the fact that non-invasive access to the abdominal external oblique muscles are much limited on people with an average physique, which was the case amongst the carers that volunteered for the experiments. Figure 5.8-5.10 shows a comparison of the results for relevant muscle activity. The EMG signal is normalised to be between  $[-1, 1]$  using values acquired by performing Maximum Voluntary Contraction (MVC) of the relevant muscles immediately prior to the experiments.

#### 5.4.2 User Trials

After we completed the first prototype of the Smart Hoist according to the specifications laid down during the design workshops, we organised a set of user trials in early December 2013 to capture the first impressions of the carers who were part of the design process.

Fifteen volunteers were inducted to use the Smart Hoist, and they were asked to perform basic manoeuvres. The feedback we received from carers during the early design stages of the project and the user trial was extremely valuable in further development of the Smart Hoist. Table 5.2 lists a summary of important comments from that trial. The Smart Hoist underwent many hardware and software changes based on this feedback. Figure 5.11 shows examples of the Smart Hoist in use during these trials.

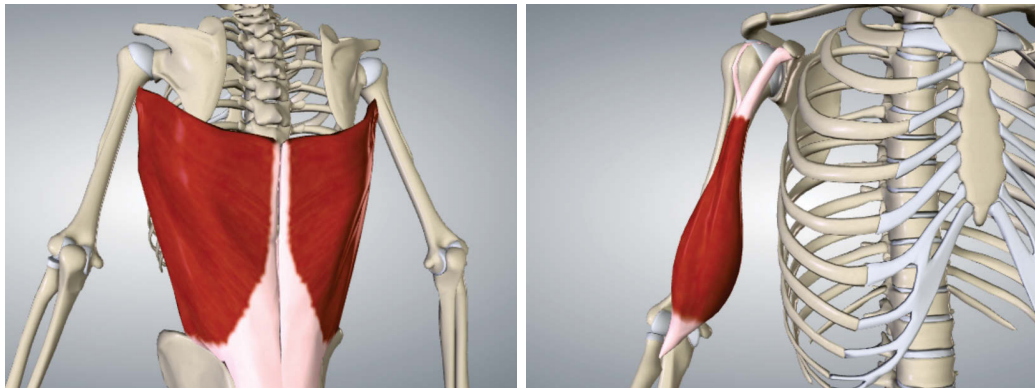
The second trials were conducted late March 2014 with approximately 50 carers. The carers participated in an interactive training workshop, after which they were asked to



(a) Pectoralis major muscles

(b) Triceps brachii muscles

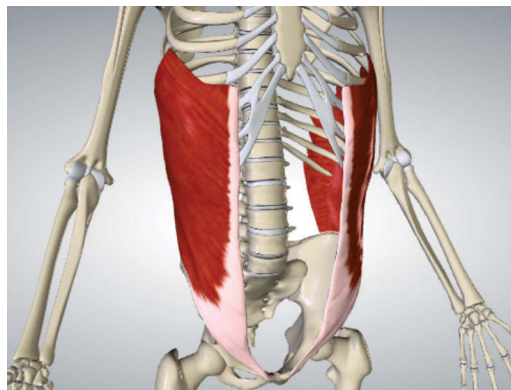
Major muscle groups assisting push action.



(c) Latissimus dorsi muscles

(d) Biceps brachii muscle

Major muscle groups assisting pull action.



(e) Abdominal external oblique muscles

Major muscle groups assisting twist/in place turn action.

FIGURE 5.7: Major muscle groups tested for activity while performing basic manoeuvres with the hoist. (Images courtesy of BioDigital Human project, biodigital.com)

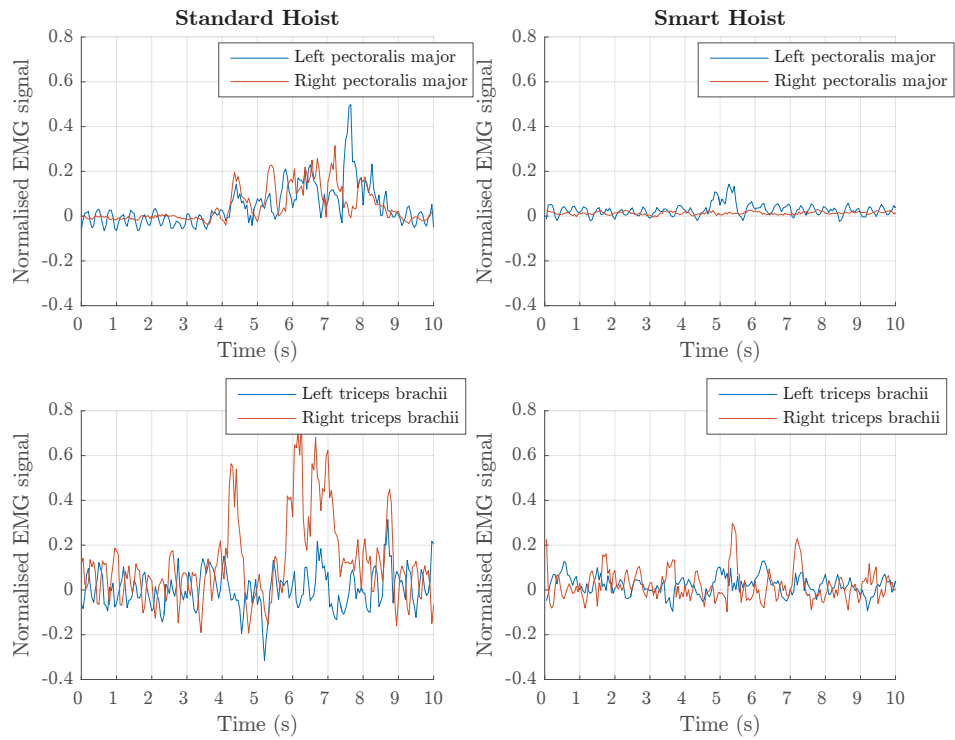


FIGURE 5.8: Comparison of muscle action based on EMG signals during a push action.

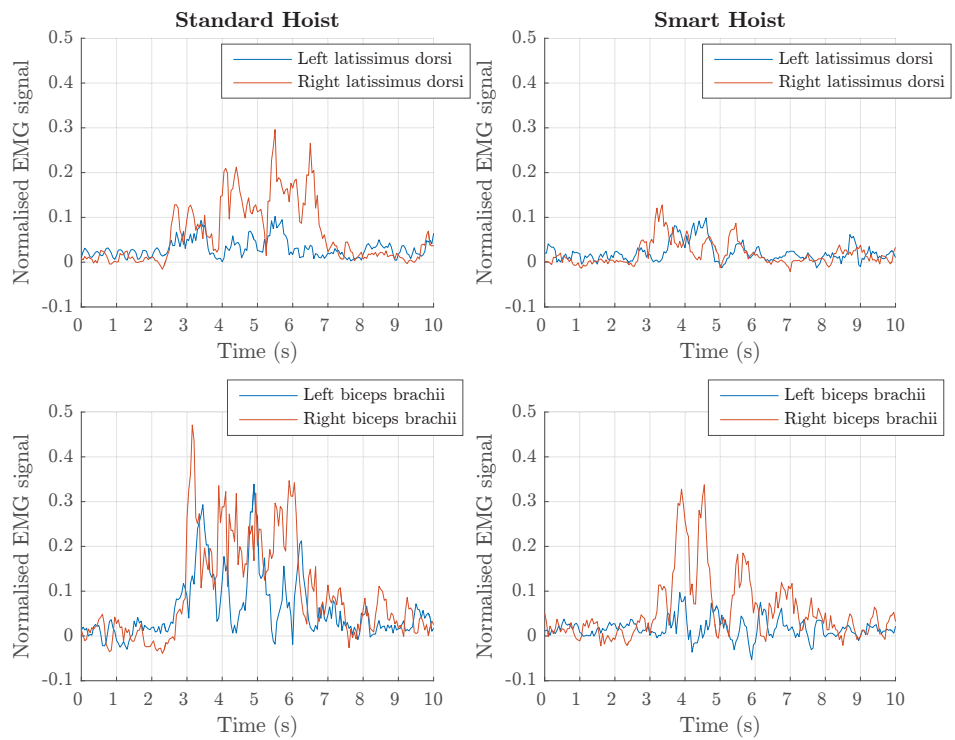


FIGURE 5.9: Comparison of muscle action based on EMG signals during a pull action.

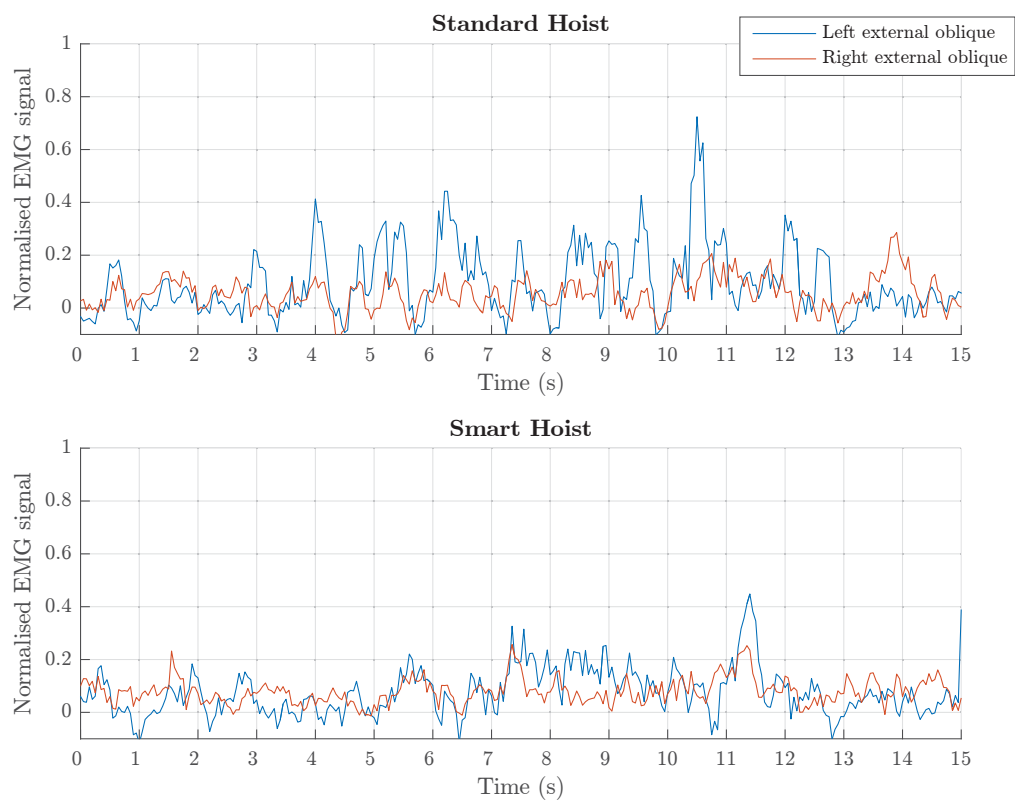


FIGURE 5.10: Comparison of muscle action based on EMG signals during an in place turn to the left.



FIGURE 5.11: User trials conducted at IRT Woonona, Australia.

TABLE 5.2: Summary of carers' comments from the first user trial

<b>Evaluation category</b>	<b>Evaluation Criteria</b>	<b>Score out of 10</b>	<b>Additional Comments</b>
<b>Confidence</b>	Driving & Turning	7	<ul style="list-style-type: none"> <li>• “Need to be slow when moving sideways and turning under load”</li> </ul>
<b>Comfort &amp; Ease of use</b>	Handles & Grips	6	<ul style="list-style-type: none"> <li>• “Too slow to change to sideways mode”</li> <li>• “Needs too much force”</li> </ul>
	Screen	8	<ul style="list-style-type: none"> <li>• “Bigger icons and text”</li> <li>• “High contrast and brightness”</li> <li>• “Include descriptions for the icons”</li> <li>• “Arrow to indicate the wheel direction”</li> </ul>
	Batteries	9	<ul style="list-style-type: none"> <li>• “Long but OK”</li> <li>• “No heavier please”</li> </ul>
	Cameras	9	<ul style="list-style-type: none"> <li>• “Must charge fast”</li> <li>• “Need a higher field of view”</li> <li>• “Can I rely on it? Will it give me a false sense of security?”</li> </ul>
<b>Safety &amp; Efficiency</b>	Overall	10	<ul style="list-style-type: none"> <li>• “I was never worried about my feet”</li> </ul>

perform a complete patient transfer from bed to bathroom in a simulated environment similar to the bedroom shown in Figure 5.3. These included complex manoeuvres such as lifting a patient from a bed, navigating through corridors and around tight corners and lowering the patient into a chair. The aim of this experiment was to assess the intuitiveness and responsiveness of the Smart Hoist in comparison to a standard hoist in a routine transfer. Preliminary outcomes of the second trial upheld the results from the first user trial.

## 5.5 Discussion

Overall, the test results from analysing forces exerted while operating the Smart Hoist confirm that use of the Smart Hoist can reduce the force required to manoeuvre the hoist. Furthermore, the activity of muscles measured by EMG confirms that the muscles were less activated while using the smart hoist, which suggests less strain on the musculoskeletal system. Comments from the trial users confirm this claim as well. They found that they do not need as much effort to use the Smart Hoist as the standard hoist when it is loaded, especially when doing in-place turns on carpeted floors .

At the initial induction, we only introduced the basic manoeuvres to the carers. During the trials, some of them were just explicitly using the motion patterns they were introduced to. However, because of the way the controller is designed, it is possible to perform some natural motion patterns with the Smart Hoist that are not explicitly defined in the controller. For example, with the manoeuvre shown in Figure 5.12, even though not frequently used, it is possible to perform such a motion with both standard hoist and the Smart Hoist, and it could be useful on rare occasions. Though we did not mention such motions during the induction, we were surprised to see the carers instinctively performing such manoeuvres during the trial, making it a very good example of the intuitiveness of the control interface of the Smart Hoist. This helps confirm the fact that the Smart Hoist while reducing the load that the users need to exert for movements, closely mimics the standard hoist.

We also had a joystick interface as an additional functionality on the operator interface tablet. However, the carers' reaction to the interface was not very appealing. Over the 65 trial participants only one person was confident and liked using the joystick and most didn't want to try the interface. Most of the comments were either,

*“Not confident. [I] need more practice.”*

or

*“Need more instructions [about using the joystick]”*

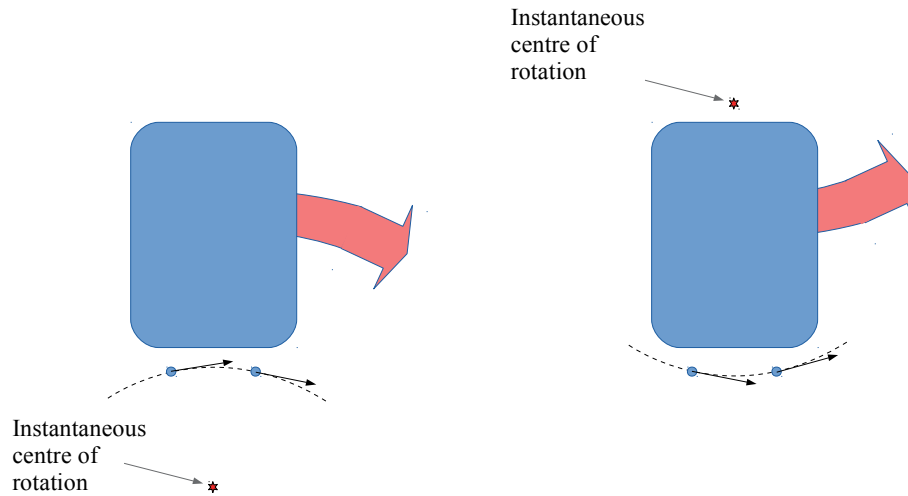


FIGURE 5.12: Administering forces on the handles in the shown directions makes the hoist rotate around an instantaneous centre of rotation.

Based on this feedback, we concluded that users do prefer their assistive robot (i.e. Smart Hoist) to mimic the manoeuvring of the existing standard hoist that they are familiar with. The user trials confirm that the learning curve associated with such a methodology is mostly flat and users can intuitively change over to the robotic device quickly and easily. Introduction of a joystick was not popular amongst users suggesting the unwillingness to change the status-quo. This result can be directly extended to the design of other assistive robotic devices, for example walkers and wheel chairs, which already have a control methodology that users are familiar with.

Moreover, we can also use this exercise to analyse how to integrate an assistive robot efficiently into the work-flow when it has been developed by the actual users.

One example is the design of the battery system for the Smart Hoist, which went through a considerable amount of revisions to achieve the expectations of the users. Factors such as ease of handling, weight, shape, docking mechanism, mounting structure, charging system, and even aesthetics had to be taken into consideration during the development of the batteries. As the Smart Hoist needs to be charged daily, it was important to make sure the process is not cumbersome.

When analysing the results from the survey, another key point was whether the new technological advancements would create a false sense of security. For example, the Smart



Hoist comes with a reverse camera to show collisions while reversing. The Smart Hoist also has an obstacle avoidance system. Both of these features can fail at times: The reverse camera is a 2D camera and therefore has blind spots, while obstacle detection completely fails when the hoist outriggers are under the bed and the camera is occluded. Therefore, these features can actually cause undesired outcomes at times.

However, we observed that the Co-Design aspect of the Smart Hoist actually improved the ability of the system to be smoothly integrated into the work-flow. The users who helped design the device embrace the ownership of the hoist as they have contributed to the design.

Therefore, we can conclude that in general, though there are many hurdles to overcome in fully integrating an assistive robot into a work-flow, an approach such as Co-Design during the design process can immensely improve the acceptance and ensure a smooth transition.

## **5.6 Conclusion**

This chapter describes the outcomes of a series of case studies conducted to design and develop an assistive robotic device, the Smart Hoist, with the aim of reducing workplace injuries that could be sustained by the care workers in aged and disabled care facilities when transferring non-ambulatory residents (e.g. bed to chair, chair to toilet and bath). The device is an extension of the standard patient hoists that are commonly used in such facilities, but with a motorised drive system rather than passive push drive. We use this experiment as an example to validate that a Co-Design inspired framework for designing control strategies for assistive robots can ensure development of intuitive and easy to use robots that can be readily integrated to the daily activity flow of the end users.

Having the end user in the design loop, rather than an iterative approach, is now becoming popular in most other fields, and is valid for the field of robotics. Especially in the design of the drive controller, the interactive Co-design workshops guided us down the path of a more intuitive human-robot interface rather than common control mechanisms such as the joystick drive mode. Similar to a standard hoist, the Smart Hoist is operated by applying forces on its handles. This intuitive control system allows the users to seamlessly migrate

to the Smart Hoist without an added learning curve. Though the mechanical limitations of the Smart Hoist wheels prevents it from operating exactly as a standard hoist, the system we designed through observation and feedback from the end users makes the Smart Hoist seamlessly mimic the standard hoist behaviour so it does not adversely impact the user experience.

The Smart Hoist possesses a set of other features integrated into it which were developed as a result of the Co-Design workshops but they are not discussed in this chapter; However, some details regarding the system design for the Smart Hoist is discussed in Appendix A. Furthermore, the Smart hoist was designed and built as part of a project with an industry partner. Performance specifications of the Smart Hoist did not include autonomous navigation. Therefore localisation and mapping algorithms were not implemented on the Smart Hoist.

## **Special Acknowledgements**

Work in this chapter was supported in part by the IRT Research Foundation, Australia and the Centre for Autonomous Systems, University of Technology, Sydney (UTS), Australia.

We thank the management, the staff and the residents of the IRT Woonona care facility, Australia for their involvement in the design development, implementation, and evaluation of the Smart Hoist. We also thank the rest of the Smart Hoist design team at CAS, UTS for their effort in building the Smart Hoist.

## Chapter 6

# Conclusions

This thesis presented an environment representation framework with a suite of localisation and mapping algorithms designed to operate within this framework that are aimed towards robot navigation in indoor environments. These findings are targeted towards assistive robots, as so far there has not been a comprehensive solution to achieve robotic mapping and localisation in real-time on a compact computational platform suited for rapid deployment of assistive robots. We evaluated these algorithms using a range of different datasets including simulated, public domain, and real-time data collected using the assistive robotic platforms we have developed at the Centre for Autonomous Systems (CAS), University of Technology, Sydney (UTS).

We also presented a framework inspired by the principles of collaborative design, for designing control strategies for assistive robots. We experimentally verified this methodology through an example design and development of an assistive robotic hoist: a robot that can be used to reduce workplace injuries amongst care workers of aged and disabled care sector, during the transfer process of non-ambulatory patients. The outcomes of a series of user trials, surveys, and experiments corroborated that the design process is quite effective in developing a robot that can be easily integrated in to the work-flow of the care facility. Moreover, these results substantiate that the robotic hoist can reduce the muscle strain during operation, hence may help reduce workplace injuries.

## 6.1 Summary of Contributions

### 6.1.1 Distance Function based Environment Representation and Chamfer Distance based Sensor Model

A map of the environment together with a sensor model which correlates the observations from a sensor to the map is an essential requirement for autonomous robot navigation. We proposed and used distance functions to represent the environment. We also presented a method for transforming the sensor uncertainties to the distance metric which forms one of the key contributions of this thesis.

Thereafter, we also defined a chamfer distance based sensor model that makes use of the proposed environment representation. This sensor model can be used to relate an observation from a sensor to the distance function based representation without the need for explicit data association.

### 6.1.2 Distance transform based Mobile Robot Localisation

We used the proposed distance function based environment representation and the associated chamfer distance based sensor model to localise the robot in a 2D, GPS denied indoor environment. We proposed three algorithms of using the sensor model: (i) an EKF based approach, (ii) an optimisation based approach, and (iii) a particle filter based method. Out of these three methods, the first two methods are novel contributions to localise a robot on an OGM that do not rely on the extraction of geometric features or landmarks. The first two methods are highly computationally efficient. While the EKF approach requires sensor and control noise parameters to be experimentally estimated, the optimisation strategy only contain one tuning parameter which can be easily deduced.

The experiment results demonstrated that the optimisation approach provide the most accurate localisation results. However, the EKF approach reports and propagates the robot pose uncertainty throughout the trajectory. Both algorithms require the initial pose to be approximately known and cannot perform global localisation, and hence is unable to solve the kidnapped robot problem. The particle filter algorithm on the other hand,

though it is computationally expensive than other two proposed algorithms, is able to localise the robot without the requirement of an initial pose.

### 6.1.3 Distance Transform based Robotic Mapping

Even though SLAM is the preferred approach for building statistically consistent maps while localising in unknown environments, most of the time it comes at a large computational expense. For robots with limited computing power such as assistive robots, we argue that the accuracy offered by scan matching based mapping would be adequate. We proposed a scan-to-map matching based mapping technique which uses the distance function based environment representation and chamfer distance based sensor model proposed in Chapter 2, that can be used to create high quality maps of 2D environments.

When compared to other algorithms such as ND-to-ND matching, incremental ICP, and GMapping, the proposed algorithm performs well even in crowded environments as illustrated using both qualitatively (by appearance), and quantitatively by using the SLAM benchmarking metric [144]. It is also shown that the proposed method has considerably less error accumulation. We were able to use proposed algorithm in large crowded environments to create maps that can be used to localise a robot, when the other three mapping algorithms failed to produce a coherent map due to dynamic objects in the environment.

However, as a loop closure detection and correction method is not built-in to the proposed mapping algorithm, it is inevitable that it would accumulate errors unless there are frequent implicit loop closures.

### 6.1.4 Control for Assistive Robots

Developing control methods for assistive robots is a very challenging problem that cannot be generalised as it depends on the nature of the human user interaction and the functionality of the assistive robot. In Chapter 5, we presented a framework suitable to design control strategies for an assistive robot by using the design process of an assistive robotic hoist as an example.

Smart Hoist is a patient lifter device that was retrofitted with powered motors to provide assistance to transferring non-ambulatory residents at aged/disability care facilities. Development of this assistive robotic device was in response to the growing number of work-related musculoskeletal injuries amongst the carers of these facilities due to manual manoeuvring of the lifter device. We followed the principals of Co-Design which keeps the end user as an integral member of the design team throughout the conception, design, development, and testing stages.

We further reported results from a number of experiments conducted to prove the effectiveness of the Smart Hoist in reducing musculoskeletal fatigue which may help to reduce related injuries. Feedback from the carers who participated the user trials and surveys were presented to highlight the effectiveness of using such a design approach which is generally applicable to design of any assistive robotic device.

## 6.2 Discussion of Limitations and Future Work

In Chapter 2, we introduced a distance function based representation method that can be used to represent the environment of which the function value provides more information about the obstacles in the environment over the traditional OGM based representations. However, the distance function does not encode the uncertainties that may be present in the process that was used to create the particular map, resulting in a map that is treated to be perfect in the subsequent localisation algorithms. Distance functions used for 3D applications use strategies such as Gaussian processes to represent map uncertainties that can also be used to fill any gaps that may be present in the unexplored areas in the 3D surface [41]. Exploring possibilities of including the map uncertainties in distance functions, while still maintaining the computational efficiency is likely to be fruitful in future work.

Furthermore, in its present form, the environment representation treats the environment as static. During the mapping process most of the dynamic objects present in the environment are statistically filtered out. However, an assistive robot which operates for a long time in a given environment should be able to augment the map to deal with moveable static objects

such as furniture, bins, and pot-plants which are prevalent in indoor environments. In the current mechanism, though we are able to update the map by adding new obstacles, once we have included an obstacle, there is no sound strategy to remove it from the distance function. Addressing the problem by dynamically updating the distance function requires future work.

Another important problem that we faced while creating maps are the errors associated with loop-closures. An active strategy to detect loops and command the robot such the distance between implicit loop-closures are smaller than a predefined value is one possibility. It is also possible to apply a pose-graph based SLAM solver into the mapping mechanism for further correction as suggested by a number of authors as mentioned previously. Investigating the suitability of such methods without sacrificing computational efficiency is an avenue for future work.

We designed and developed a new wheel mechanism to help solve the issues we faced with the previous wheels. These wheels are designed to have better swerve capabilities so that we can closely follow the holonomic action provided by standard hoist caster system. Analysing the impact of these through an extended user study would provide further valuable insights.

Use of the proposed design framework to develop other assistive robotic devices may be help to further regularise the proposed strategy, which is also suggested as future work.

## Appendix A

# System Overview of the Assistive Robotic Smart Hoist

The UTS-IRT Smart Hoist incorporates and builds upon the standard Joey<sup>TM</sup> Lifter from AIS healthcare Pty. Ltd. As part of the transformation, the Joey<sup>TM</sup> Lifter has undergone a series of modifications which were completed with extreme care to avoid compromising its structural integrity, Figures 5.1 and 5.2 shows the standard and Smart Hoist.

### A.1 Hardware Structure

#### A.1.1 Sensors

The Smart Hoist is equipped with a number of sensors to gather information about the environment as well as the user input.

##### A.1.1.1 Handles

The handles of the hoist consists of four strain gauge installations, each in a full-bridge configuration. Each handle contains two bridges, the first pair measures torsion along the mounting axis of the handle as shown in Figure A.1a, the other pair measures the deflection



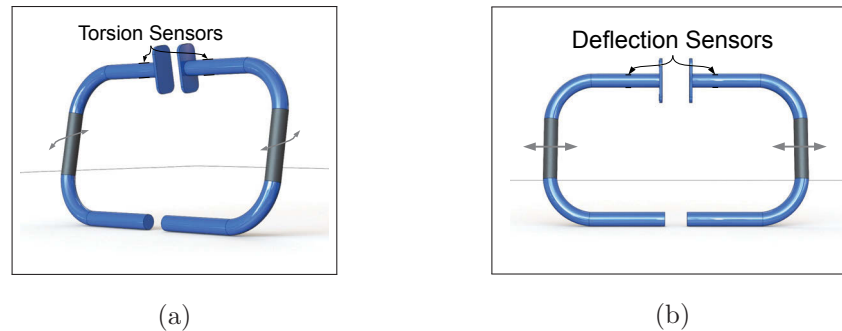


FIGURE A.1: Modified handle bars of the Smart Hoist. (a) Torsion sensor pairs for detection of forward and backward forces. (b) Deflection sensor pairs for detection of sideways forces.

of the handle up and down along the top fixed axis, which is effectively a sideways force on the handle (Figure A.1b). The location of these sensors were selected to obtain an optimal strain reading by performing a Finite Element Analysis (FEA) of the handle structure. In contrast to the standard hoist, the handle is disjointed in the middle to prevent strain transmission from one side to the other.

#### A.1.1.2 Cameras

There are two cameras attached to the system. A High Definition, wide-angle RGB camera at the top of the boom (Figure A.2b) to act as a rear view camera to aid when reversing the hoist, and a RGB-D sensor at the bottom of the Smart Hoist. This camera is used by the navigation assistance system for manoeuvring the hoist in confined spaces and avoiding objects.

#### A.1.1.3 Boom Sensors

Weight measurement and the Body Mass Index (BMI) calculation is an added feature that is generally not present in the standard hoist. As shown in Figure A.2 strain gauges were placed at an optimal location that yielded the highest strain, determined by performing an FEA of the boom structure.



FIGURE A.2: Strain gauge location on the boom for weight measurement.

The strain gauge readings at a given height of the boom can be approximated to have a linear relationship with the weight in the sling. The weight can be estimated at a maximum error of about two percent after proper calibration.

This weight is displayed on the User Interface (UI) and BMI of the patient on the hoist can be easily calculated from the UI. The system controller also uses the weight measurements in its logic for safety actions.

### A.1.2 Actuators

The standard system is factory fitted with two actuators, one controlling the lifting of the boom and the other one controlling the width between the outriggers. These actuators were retrofitted with encoders to monitor their location which is required in weight measurement. In addition to these the rear wheels of the Smart Hoist were replaced with a pair of powered castors shown in Figure A.3. The wheel is 360° steerable and uses the Revolution 2™ assembly from 221 Robotic Systems [166].

### A.1.3 The User Interface

For the purpose of interacting with the carers and displaying critical information of the system, such as battery level, system status, and error messages, a UI has been included with the Smart Hoist.

As Android is readily interoperable with ROS, a Google® Nexus 7 - 2012 Tablet running Android version Jelly bean has been used for the interface. The Tablet is tethered to the Hoist PC using USB to ensure a reliable connection to the host. The kiosk mode UI (Figure A.4) of the Smart Hoist is kept simple and intuitive, allowing users to interact with

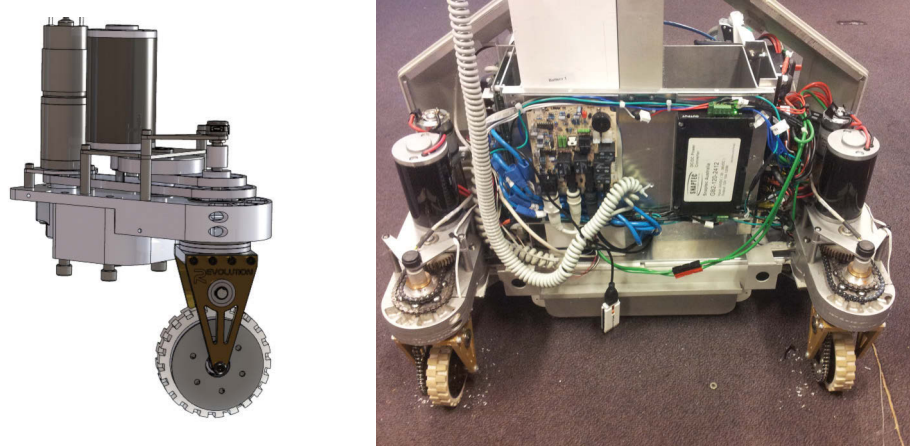


FIGURE A.3: Revolution 2™ omni-directional crab drive motors from 221 Robotic Systems, and how they're retrofitted to the Smart Hoist.

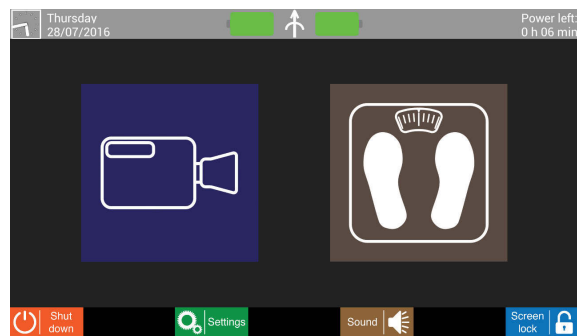


FIGURE A.4: User Interface of the Smart Hoist, the main screen.

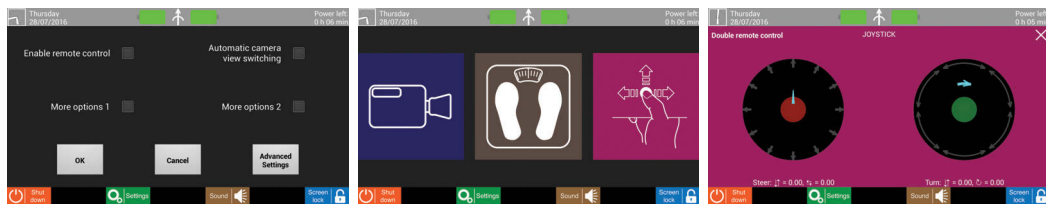


FIGURE A.5: Control pad/ joystick interface to control the Smart Hoist

it easily. Design, development and implementation of the Graphical User Interface (GUI) also adhered to the corporative design approach.

There is a simple control pad/joystick interface to control the Smart Hoist that can be enabled via a hidden settings menu as shown in Figure A.5.

### A.1.4 Hoist Computer

An Intense PC<sup>TM</sup>[167] is included in the Smart Hoist as the main processing platform (Figure A.6). It is equipped with an Intel<sup>®</sup> Core<sup>TM</sup>i7 processor running at 2.8Ghz with 4GB of RAM. The underlying operating system is Ubuntu 12.04 LTE and ROS version Fuerte Turtle.

### A.1.5 Peripherals

#### A.1.5.1 Hoist Controller

A propitiatory embedded controller board was designed and built based on a Microchip PIC24 micro-controller, for low level hardware monitoring, power management, interfacing with other hardware (detecting e-Stop status) and data acquisition from the strain gauges. The Hoist Controller is interfaced to the Hoist PC via a USB interface and it streams data frames at 10Hz to the Hoist PC.

#### A.1.5.2 Motor Controller

The Sasquatch controller from 221 Robotics Systems based on Atmel<sup>®</sup> AtMega 2560 Micro-controller is used for controlling the motors (Figure A.6). It receives information from the Hoist PC at 10Hz through an Ethernet interface. The velocity and steer angle of each individual motor are the input to the Motor Controller.

### A.1.6 Batteries and Uninterruptible Power Supply(UPS)

The Smart Hoist was designed to have two  $LiFePO_4$  Batteries of  $\approx 26V$ , which weigh  $\approx 4.5kg$  (Figure A.7). This chemistry was specifically chosen to increase the life span of the batteries. When both batteries are fully charged the Smart Hoist has a standby time of more than 12 hours and about 5 hours when it is in operation.

A super-capacitor UPS was included to supply power to the Hoist PC and other critical components for approximately three minutes in-between battery changes.



FIGURE A.6: Motor controller (left) and the Hoist PC (right) assembly.

## A.2 System Assembly and Enclosure Design

The enclosure forms part of the Smart Hoist that houses the majority of hardware components including the Hoist PC, the Hoist Controller, the Revolution 2<sup>TM</sup> motor assemblies, the Sasquatch motor controller, the Batteries and the UPS.

Since a standard hoist was used as the basis upon which to build the Smart Hoist, space was limited. This meant the enclosure housing the hardware components needed to be compact and provides its own structural integrity to support all hardware components. Having an Ingress Protection rating of X4 (IPX4) for the enclosure was an additional design consideration.

The majority of the design process was centred around the batteries. After estimating the power budget, it was deduced that 16 cells are required for a usage cycle of  $\approx 5$  hours per charge. As a result, the batteries were split into two packs (Figure A.7a) which were designed to be manageable in weight and easily inserted and removed from the Smart Hoist (Figure A.7b).

The design of the enclosure and placement of hardware components were planned and mapped in Solidworks<sup>TM</sup>. A physical framework was then constructed using Aluminium. All remaining hardware components were mounted to this framework and wired. The external shell of the enclosure was then physically modelled in cardboard and Foamex<sup>TM</sup>



(a)



(b)

FIGURE A.7: Batteries used in the Smart Hoist.

for the initial iteration and the final version of the enclosure was manufactured in Polyamide using an additive manufacturing process called Selective Laser Sintering (SLS).

### A.3 Measures Taken for Occupational Health and Safety

#### A.3.1 Foot-guards

As the Smart Hoist uses powered motors it is extremely important to restrict accidental access to the powered wheels or caster wheels. Though covered foot wear is a mandatory requirement for carers at IRT, for extra safety foot guards were added to all wheels.

#### A.3.2 Maximum Run Speed

The maximum speed of the motors were limited at the Motor Controller firmware level to avoid the propagation of any undesirable velocities. Steps have also been taken to disable the motors in the event of critical failure scenarios (e.g. motor controller loses communications with the Hoist PC) .

### **A.3.3 Braking**

The Revolution 2<sup>TM</sup> motors were modified to have electro-mechanical brakes that would get engaged whenever there are no forces on the handles. This ensures the stability of the Smart Hoist and prevents the hoist from rolling while loading and unloading heavy subjects.

### **A.3.4 Emergency Stop (EStop)**

As with most industrial devices, Smart hoist is equipped with an eStop Button that can be engaged in an emergency. This button immediately disconnects power to the actuator systems. The electro-mechanical brakes are automatically applied in this scenario and can be manually disengaged if necessary.

# Bibliography

- [1] Peter Costello. Intergenerational Report 2007. Technical report, Australian Treasury, Commonwealth of Australia, Canberra, 2007.
- [2] Wayne Swan. Intergenerational Report 2010. Technical report, Australian Treasury, Commonwealth of Australia, Canberra, 2010.
- [3] Population by Age and Sex, Regions of Australia. Technical report, Australian Bureau of Statistics, 2015. URL <http://www.abs.gov.au/>.
- [4] Gary Banks, Angela MacRae, Patricia Scott, Robert Fitzgerald, Louise Sylvan, David Kalisch, Philip Weickhardt, Mike Woods, and Wendy Craik. Annual Report 2009-10. Technical report, Productivity Commission, Canberra, Australia, 2010.
- [5] Mike Woods, Robert Fitzgerald, and Sue Macri. Caring for older Australians. Technical report, Productivity Commission, Canberra, Australia, 2011.
- [6] J Broekens, M Heerink, and H Rosendal. Assistive social robots in elderly care: a review. *Gerontechnology*, 8(2), 4 2009. ISSN 1569-111X. doi: 10.4017/gt.2009.08.02.002.00.
- [7] K Cranswick and D Dosman. Eldercare: What we know today. *Canadian social trends*, 2008.
- [8] ME Pollack, L Brown, and D Colbry. Pearl: A mobile robotic assistant for the elderly. Technical report, AAAI, 2002. URL <http://www.aaai.org/Papers/Workshops/2002/WS-02-02/WS02-02-013.pdf>.
- [9] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. The role of assistive robotics in the lives of



- persons with disability. *American journal of physical medicine & rehabilitation / Association of Academic Physiatrists*, 89(6):509–21, 6 2010. ISSN 1537-7385. doi: 10.1097/PHM.0b013e3181cf569b.
- [10] T Mukai, S Hirano, H Nakashima, Y Kato, Y Sakaida, S Guo, and S Hosoe. Development of a nursing-care assistant robot RIBA that can lift a human in its arms. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5996–6001. IEEE, 10 2010. ISBN 978-1-4244-6674-0. doi: 10.1109/IROS.2010.5651735.
- [11] Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi. Towards evaluation of social robotic telepresence based on measures of social and spatial presence, 2011. URL <http://www.diva-portal.org/smash/record.jsf?pid=diva2:542612>.
- [12] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8(1-2):47–63, 11 1991. ISSN 09218890. doi: 10.1016/0921-8890(91)90014-C.
- [13] Maja J Mataric. *A distributed model for mobile robot environment-learning and navigation*. PhD thesis, MIT, Cambridge, MA, 1990.
- [14] UR Zimmer. Robust world-modelling and navigation in a real world. *Neurocomputing*, 1996.
- [15] B Yamauchi and R Beer. Spatial learning for navigation in dynamic environments. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 26(3):496–505, 1 1996. ISSN 1083-4419. doi: 10.1109/3477.499799.
- [16] Hagit Shatkay and Leslie Pack Kaelbling. Learning topological maps with weak local odometric information. In *IJCAI-97. IJCAI, INC*, pages 920–927. Morgan Kaufmann Publishers Inc., 8 1997. ISBN 1-555860-480-4.
- [17] Hagit Shatkay. *Learning models for robot navigation*. PhD thesis, Brown University Providence, RI, USA, 1 1999. URL <http://dl.acm.org/citation.cfm?id=929685>.
- [18] David Pierce and Benjamin Kuipers. Learning to explore and build maps. In *AAAI'94 Proceedings of the twelfth national conference on Artificial intelligence*,

- pages 1264–1271. American Association for Artificial Intelligence, 10 1994. ISBN 0-262-61102-3.
- [19] David Kortenkamp and Terry Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *AAAI'94 Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pages 979–984. American Association for Artificial Intelligence, 10 1994. ISBN 0-262-61102-3.
- [20] SP Engelson and DV McDermott. Error correction in mobile robot map learning. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2555–2560. IEEE Comput. Soc. Press, 1992. ISBN 0-8186-2720-4. doi: 10.1109/ROBOT.1992.220057.
- [21] H Choset, B. Mirtich, and J Burdick. Sensor based planning for a planar rod robot: incremental construction of the planar rod-HGVG. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 3427–3434. IEEE, 1996. ISBN 0-7803-3612-7. doi: 10.1109/ROBOT.1997.606866.
- [22] Howie Choset. *Sensor based motion planning: the hierarchical generalized Voronoi graph*. PhD thesis, California Institute of Technology ,Pasadena, CA, USA, 10 1996. URL <http://dl.acm.org/citation.cfm?id=238042>.
- [23] Benjamin Kuipers, Dan G. Tecuci, and Brian J. Stankiewicz. The Skeleton In The Cognitive Map: A Computational and Empirical Exploration. *Environment & Behavior*, 35(1):81–106, 1 2003. ISSN 00000000. doi: 10.1177/0013916502238866.
- [24] Georg Gartner and Verena Radoczky. Schematic vs. Topographic Maps in Pedestrian Navigation: How Much Map Detail is Necessary to Support Wayfinding. In *AAAI Spring Symposium: Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance*, pages 41–47, 2005.
- [25] Sydney Trains suburban map, 2016. URL [http://www.sydneytrains.info/stations/network\\_map](http://www.sydneytrains.info/stations/network_map).
- [26] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 6 1991. ISSN 1042296X. doi: 10.1109/70.88147.

- 
- [27] Alberto Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, USA, 1 1989. URL <http://dl.acm.org/citation.cfm?id=916528>.
- [28] A Elfes. Sonar-based real-world mapping and navigation. *Robotics and Automation, IEEE Journal of*, 1987.
- [29] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9 (2):61–74, 7 1988. ISSN 0738-4602.
- [30] Cedric Coaud and Amor Jnifene. Environment mapping using probabilistic quadtree for the guidance and control of autonomous mobile robots. In *2010 International Conference on Autonomous and Intelligent Systems, AIS 2010*, pages 1–6. IEEE, 6 2010. ISBN 978-1-4244-7104-1. doi: 10.1109/AIS.2010.5547019.
- [31] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 4 2013. ISSN 0929-5593. doi: 10.1007/s10514-012-9321-0.
- [32] R Chatila and JP Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138–145. Institute of Electrical and Electronics Engineers, 1985. doi: 10.1109/ROBOT.1985.1087373.
- [33] Open Source Robotics Foundation. Robot Operating System(ROS), 2016. URL <http://wiki.ros.org>.
- [34] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2 2007. ISSN 15523098. doi: 10.1109/TRO.2006.889486.
- [35] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable SLAM system with full 3D motion estimation. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 155–160. IEEE, 11 2011. ISBN 978-1-61284-769-6. doi: 10.1109/SSRR.2011.6106777.

- [36] Kunjin Ryu, Lakshitha Dantanarayana, Tomonari Furukawa, and Gamini Dissanayake. Grid-based Scan-to-Map Matching for Accurate 2D Map Building. *Advanced Robotics*, pages 1–10, 3 2016. ISSN 0169-1864. doi: 10.1080/01691864.2015.1124025.
- [37] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 303–312, New York, New York, USA, 1996. ACM Press. ISBN 0897917464. doi: 10.1145/237170.237269.
- [38] Richard A. Newcombe, Andrew J. Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 10 2011. ISBN 978-1-4577-2185-4. doi: 10.1109/ISMAR.2011.6092378.
- [39] Thomas Whelan, Michael Kaess, and Maurice Fallon. Kintinuous: Spatially Extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012.
- [40] H Carrillo, Y Latif, J Neira, and JA Castellanos. Towards measuring uncertainty in volumetric signed distance function representations for active SLAM. In *Workshop on Multi View Geometry in Robotics (MVGRO) In conjunction with RSS 2014*, 2014.
- [41] Soohwan Kim and Jonghyuk Kim. GPmap: A Unified Framework for Robotic Mapping Based on Sparse Gaussian Processes. In *Field and Service Robotics*, pages 319–332. Springer International Publishing, 2015. doi: 10.1007/978-3-319-07488-7\_22.
- [42] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets. *Computer Graphics Forum*, 29(5):1733–1741, 9 2010. ISSN 01677055. doi: 10.1111/j.1467-8659.2010.01782.x.

- 
- [43] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric Inference for Probability Measures. *Foundations of Computational Mathematics*, 11(6):733–751, 7 2011. ISSN 1615-3375. doi: 10.1007/s10208-011-9098-0.
- [44] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. The MIT Press, 2005. ISBN 9780262201629. URL <http://www.probablistic-robotics.org>.
- [45] Simon T. O’Callaghan and Fabio T. Ramos. Gaussian Process Occupancy Maps for Dynamic Environments. In *Experimental Robotics*, pages 791–805. Springer International Publishing, 2016. doi: 10.1007/978-3-319-23778-7\_52.
- [46] Alen Alempijevic, Robert Fitch, and Nathan Kirchner. Bootstrapping navigation and path planning using human positional traces. In *2013 IEEE International Conference on Robotics and Automation*, pages 1242–1247. IEEE, 5 2013. ISBN 978-1-4673-5643-5. doi: 10.1109/ICRA.2013.6630730.
- [47] Roman Marchant and Fabio Ramos. Bayesian Optimisation for informative continuous path planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143. IEEE, 5 2014. ISBN 978-1-4799-3685-4. doi: 10.1109/ICRA.2014.6907763.
- [48] Maani Ghaffari Jadidi, Jaime Valls Miro, Rafael Valencia, and Juan Andrade-Cetto. Exploration on continuous Gaussian process frontier maps. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6077–6082. IEEE, 5 2014. ISBN 978-1-4799-3685-4. doi: 10.1109/ICRA.2014.6907754.
- [49] S. O’Callaghan, F.T. Ramos, and H. Durrant-Whyte. Contextual occupancy maps using Gaussian processes. In *2009 IEEE International Conference on Robotics and Automation*, pages 1054–1060. IEEE, 5 2009. ISBN 978-1-4244-2788-8. doi: 10.1109/ROBOT.2009.5152754.
- [50] S. T. O’Callaghan and F. T. Ramos. Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62, 1 2012. ISSN 0278-3649. doi: 10.1177/0278364911421039.

- [51] Distance transform of binary image. URL <http://au.mathworks.com/help/images/ref/bwdist.html>.
- [52] A. Rosenfeld and J.L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1(1):33–61, 7 1968. ISSN 00313203. doi: 10.1016/0031-3203(68)90013-7.
- [53] T. Schouten and E. van den Broek. Fast exact Euclidean distance (FEED) transformation. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 594–597. IEEE, 2004. ISBN 0-7695-2128-2. doi: 10.1109/ICPR.2004.1334599.
- [54] Jens Schneider, Martin Kraus, and Rüdiger Westermann. {GPU}-Based Real-Time Discrete Euclidean Distance Transforms With Precise Error Bounds. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 435–442, 2009.
- [55] Jens Schneider, Martin Kraus, and Rüdiger Westermann. GPU-Based Euclidean Distance Transforms and Their Application to Volume Rendering. In AlpeshKumar Ranchordas, Joo Madeiras Pereira, Hlder J. Araújo, and Joo Manuel R. S. Tavares, editors, *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, volume 68 of *Communications in Computer and Information Science*, pages 215–228. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-11839-5. doi: 10.1007/978-3-642-11840-1\_16.
- [56] Michael Rauter and David Schreiber. A GPU accelerated Fast Directional Chamfer Matching algorithm and a detailed comparison with a highly optimized CPU implementation. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 68–75. IEEE, 6 2012. ISBN 978-1-4673-1612-5. doi: 10.1109/CVPRW.2012.6238897.
- [57] Dirk Hähnel. Intel Research Lab (Seattle) Dataset, 2000. URL <http://kaspar.informatik.uni-freiburg.de/~slamEvaluation/datasets.php>.
- [58] Sebastian Thrun. Bayesian Landmark Learning for Mobile Robot Localization. *Machine Learning*, 33(1):41–76, 1998. ISSN 1573-0565. doi: 10.1023/A:1007554531242.

- [59] Hugh Durrant-whyte. Durrant-whyte, Hugh. Technical report, Australian Centre for Field Robotics, 2001. URL <http://www.acfr.usyd.edu.au/pdfs/training/multiSensorDataFusion/dataFusionNotes.pdf>.
- [60] F. C. Leone, L. S. Nelson, and R. B. Nottingham. The Folded Normal Distribution. *Technometrics*, 3(4):543–550, 11 1961. ISSN 0040-1706. doi: 10.1080/00401706.1961.10489974.
- [61] Michail Tsagris, Christina Beneki, and Hossein Hassani. On the folded normal distribution. *Mathematics*, 2(1):12–28, 2 2014. ISSN 2227-7390. doi: 10.3390/math2010012.
- [62] Felix Hausdorff. *Grundzüge der Mengenlehre*. Veit & co., 1914. ISBN 978-3-540-42224-2.
- [63] D.P. P DP Huttenlocher, G.A. A Klanderma, and W.J. J Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. ISSN 01628828. doi: 10.1109/34.232073.
- [64] HG Barrow, JM Tenenbaum, RC Bolles, and HC Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *5th international joint conference on Artificial intelligence - Volume 2*, page 659663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.
- [65] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1270–81, 7 2008. ISSN 0162-8828. doi: 10.1109/T-PAMI.2007.70772.
- [66] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, 1988. ISSN 01628828. doi: 10.1109/34.9107.
- [67] Ming Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, and Rama Chellappa. Fast directional chamfer matching. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1696–1703. IEEE, 6 2010. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5539837.

- [68] Simon Haykin and K. J. Ray Liu, editors. *Handbook on Array Processing and Sensor Networks*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1 2010. ISBN 9780470487068. doi: 10.1002/9780470487068.
- [69] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. ISBN 0521540518.
- [70] C Stachniss and W Burgard. Particle Filters for Robot Navigation. *Foundations and trends in Robotics*, 3(4):211–282, 2014. doi: 10.1561/2300000013.
- [71] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328. IEEE, 1999. ISBN 0-7803-5180-0. doi: 10.1109/ROBOT.1999.772544.
- [72] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 5 2001. ISSN 00043702. doi: 10.1016/S0004-3702(01)00069-8.
- [73] S. Thrun. A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots. *The International Journal of Robotics Research*, 20(5):335–363, 5 2001. ISSN 0278-3649. doi: 10.1177/02783640122067435.
- [74] Dieter Fox. Adapting the Sample Size in Particle Filters Through KLD-Sampling. *The International Journal of Robotics Research*, 22(12):985–1003, 12 2003. doi: 10.1177/0278364903022012001.
- [75] Siddhartha S. Srinivasa, Dave Ferguson, Casey J. Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe. HERB: A home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, 11 2010. ISSN 09295593. doi: 10.1007/s10514-009-9160-9.
- [76] Luis Moreno, Jose M. Armingol, Santiago Garrido, Arturo de la Escalera, and Miguel A. Salichs. A Genetic Algorithm for Mobile robot Localization using Ultrasonic sensors. *Journal of Intelligent and Robotic Systems*, 34(2):135–154, 2002. ISSN 1573-0409. doi: 10.1023/A:1015664517164.



- [77] Can Ulas Dogruer, A. Bugra Koku, and Melik Dolen. Global Urban Localization of an Outdoor Mobile Robot with Genetic Algorithms. In *European Robotics Symposium 2008*, pages 103–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-78317-6\_11.
- [78] N.M. Kwok, D.K. Liu, and G. Dissanayake. Evolutionary computing based mobile robot localization. *Engineering Applications of Artificial Intelligence*, 19(8):857–868, 12 2006. ISSN 09521976. doi: 10.1016/j.engappai.2006.01.020.
- [79] Guoqiang Mao, Bar Fidan, and Brian D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 7 2007. ISSN 13891286. doi: 10.1016/j.comnet.2006.11.018.
- [80] Paul J Besl and Neil D McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992. ISSN 01628828. doi: 10.1109/34.121791.
- [81] Sebastian Thrun, Mark Diel, and Dirk Hähnel. Scan Alignment and 3-D Surface Modeling with a Helicopter Platform. In *The 4th International Conference on Field and Service Robotics*, 2003.
- [82] Andrea Censi. An ICP variant using a point-to-line metric. In *2008 IEEE International Conference on Robotics and Automation*, pages 19–25. IEEE, 5 2008. ISBN 978-1-4244-1646-2. doi: 10.1109/ROBOT.2008.4543181.
- [83] E. B. Olson. Real-time correlative scan matching. In *2009 IEEE International Conference on Robotics and Automation*, pages 4387–4393. IEEE, 5 2009. ISBN 978-1-4244-2788-8. doi: 10.1109/ROBOT.2009.5152375.
- [84] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.
- [85] 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, 6 2001. ISSN 1042296X. doi: 10.1109/70.938381.

- 
- [86] M Montemerlo, S Thrun, D Koller, and B Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Proceedings of the National conference on Artificial Intelligence*, pages 593–598, 2002.
- [87] Hauke Strasdat, J M M Montiel, and Andrew J Davison. Real-time monocular SLAM: Why filter? In *2010 IEEE International Conference on Robotics and Automation*, pages 2657–2664. IEEE, 5 2010. ISBN 978-1-4244-5038-1. doi: 10.1109/ROBOT.2010.5509636.
- [88] Shoudong Huang, Zhan Wang, and G. Dissanayake. Sparse Local Submap Joining Filter for Building Large-Scale Maps. *IEEE Transactions on Robotics*, 24(5):1121–1130, 10 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2003259.
- [89] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 12 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2006706.
- [90] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz. MINERVA: A Tour-Guide Robot that Learns. In W Burgard, T Christaller, and A B Cremers, editors, *Procof the German Conference on Artificial Intelligence KI Germany*, pages 14–26. Springer, 1999. doi: 10.1007/3-540-48238-5.2.
- [91] S. Thrun. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *The International Journal of Robotics Research*, 19(11):972–999, 11 2000. ISSN 0278-3649. doi: 10.1177/02783640022067922.
- [92] Jens-Steffen Gutmann, Gabriel Brisson, Ethan Eade, Philip Fong, and Mario Munich. Vector field SLAM. In *2010 IEEE International Conference on Robotics and Automation*, pages 236–242. IEEE, 5 2010. ISBN 978-1-4244-5038-1. doi: 10.1109/ROBOT.2010.5509509.
- [93] Seo-yeon Hwang, Jae-bok Song, and Mun Sang Kim. Robust Extraction of Arbitrary-Shaped Features in Ceiling for Upward-Looking Camera-Based SLAM. In *International Federation of Automatic Control*, pages 8165–8170, 2011.

- [94] Seo-Yeon Hwang and Jae-Bok Song. Monocular Vision-Based SLAM in Indoor Environment Using Corner, Lamp, and Door Features From Upward-Looking Camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804–4812, 10 2011. ISSN 0278-0046. doi: 10.1109/TIE.2011.2109333.
- [95] Daniel Lecking, Oliver Wulf, and Bernardo Wagner. Localization in a wide range of industrial environments using relative 3D ceiling features. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 333–337. IEEE, 9 2008. ISBN 978-1-4244-1505-2. doi: 10.1109/ETFA.2008.4638415.
- [96] Wen-Tsai Huang, Chun-Lung Tsai, and Huei-Yung Lin. Mobile robot localization using ceiling landmarks and images captured from an RGB-D camera. *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 855–860, 7 2012. doi: 10.1109/AIM.2012.6265979.
- [97] Paulo Alves, Hugo Costelha, and Carlos Neves. Localization and navigation of a mobile robot in an office-like environment. In *2013 13th International Conference on Autonomous Robot Systems*, pages 1–6. IEEE, 4 2013. ISBN 978-1-4799-1247-6. doi: 10.1109/Robotica.2013.6623536.
- [98] Lakshitha Dantanarayana, Gamini Dissanayake, Ravindra Ranasinghe, and Tomonari Furukawa. An extended Kalman filter for localisation in occupancy grid maps. In *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, pages 419–424. IEEE, 12 2015. ISBN 978-1-5090-1741-6. doi: 10.1109/ICIINFS.2015.7399048.
- [99] Richard Steffen. A Robust Iterative Kalman Filter Based On Implicit Measurement Equations. *Photogrammetrie - Fernerkundung - Geoinformation*, 2013(4):323–332, 8 2008. ISSN 14328364. doi: 10.1127/1432-8364/2013/0180.
- [100] Lakshitha Dantanarayana, Ravindra Ranasinghe, and Gamini Dissanayake. C-LOG: A Chamfer Distance based method for localisation in occupancy grid-maps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 376–381. IEEE, 11 2013. ISBN 978-1-4673-6358-7. doi: 10.1109/IROS.2013.6696379.

- 
- [101] Tinne De Laet, Joris De Schutter, and Herman Bruyninckx. A rigorously Bayesian beam model and an adaptive full scan model for range finders in dynamic environments. *Journal of Artificial Intelligence Research*, 33(1):179–222, 9 2008. ISSN 1076-9757.
- [102] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of robot motion: theory, algorithms, and implementations*. MIT Press, 2005. ISBN 9780262033275. URL <http://mitpress.mit.edu/0262033275>.
- [103] D Fox, W Burgard, and S Thrun. Markov Localization for Mobile Robots in Dynamic Environments. *Journal of Artificial Intelligence Research*, 11:391–427, 6 1999. doi: 10.1613/jair.616.
- [104] Patrick Pfaff, Christian Plagemann, and Wolfram Burgard. Improved likelihood models for probabilistic localization based on range scans. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2192–2197. IEEE, 10 2007. ISBN 978-1-4244-0911-2. doi: 10.1109/IROS.2007.4399250.
- [105] Christian Plagemann, Kristian Kersting, Patrick Pfaff, and Wolfram Burgard. Gaussian beam processes: A nonparametric bayesian measurement model for range finders. In *In Proc. of Robotics: Science and Systems (RSS)*, 2007.
- [106] K Konolige and K Chou. Markov localization using correlation. *IJCAI*, 1999.
- [107] U.D. Hanebeck and K. Briechle. New results for stochastic prediction and filtering with unknown correlations. In *Conference Documentation International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI 2001 (Cat. No.01TH8590)*, pages 147–152. VDI/VDE Soc. Meas. & Autom. Control, 2001. ISBN 3-00-008260-3. doi: 10.1109/MFI.2001.1013523.
- [108] S.J. Julier and J.K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, volume 4, pages 2369–2373. American Autom. Control Council, 1997. ISBN 0-7803-3832-4. doi: 10.1109/ACC.1997.609105.

- 
- [109] Richard Vaughan. Player Project, Stage robot simulator for ROS. In *http://wiki.ros.org/stage*, 2009.
- [110] Brian P. Gerkey. Adaptive Monte-Carlo Localization for ROS, 2008. URL <http://wiki.ros.org/amcl>.
- [111] T. Bailey, J. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 424–429. IEEE, 2006. ISBN 0-7803-9505-0. doi: 10.1109/ROBOT.2006.1641748.
- [112] Sebastian Thrun. Robotic Mapping: A Survey. In Gerhard Lakemeyer and Bernhard Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter Robotic Ma, pages 1–35. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003. ISBN 1-55860-811-7.
- [113] R. C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 12 1986. ISSN 0278-3649. doi: 10.1177/027836498600500404.
- [114] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 850–850. Institute of Electrical and Electronics Engineers, 7 1990. ISBN 0-387-97240-4. doi: 10.1109/ROBOT.1987.1087846.
- [115] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1009–1014. IEEE, 2000. ISBN 0-7803-5886-4. doi: 10.1109/ROBOT.2000.844732.
- [116] G. Dissanayake, P. Newman, Hugh F. Durrant-Whyte, Steve Clark, and M. Csorba. An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building. In *The Sixth International Symposium on Experimental Robotics*, pages 265–274. Springer-Verlag, 3 1999. ISBN 1-85233-210-7.

- 
- [117] Jose A. Castellanos and Juan D. Tardos. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Springer, 3 1999. ISBN 978-1-4615-4405-0. URL <http://dl.acm.org/citation.cfm?id=556293>.
- [118] P. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, University of Sydney, 1999. URL <http://www.fieldrobotics.org/~ssingh/pubs/others/pmn.pdf>.
- [119] D. Hahnel, W. Burgard, D. Fox, and S. Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 1, pages 206–211. IEEE, 2003. ISBN 0-7803-7860-1. doi: 10.1109/IROS.2003.1250629.
- [120] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183, Stanford, CA, 2000. Morgan Kaufmann Publishers Inc.
- [121] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 12 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768.
- [122] Rainer Kummerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 5 2011. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5979949.
- [123] Juan Nieto, Tim Bailey, and Eduardo Nebot. Recursive scan-matching SLAM. *Robotics and Autonomous Systems*, 55(1):39–49, 1 2007. ISSN 09218890. doi: 10.1016/j.robot.2006.06.008.

- [124] Albert Diosi and Lindsay Kleeman. Laser scan matching in polar coordinates with application to SLAM. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 1439–1444, Canada, 2005. monton. ISBN 0780389123. doi: 10.1109/IROS.2005.1545181.
- [125] J Yin, L Carlone, S Rosa, ML Anjum, and B Bona. Scan Matching for Graph SLAM in Indoor Dynamic Scenarios. In *International Florida Artificial Intelligence Research Society Conference*, Florida, 2014.
- [126] L V Jixin, Abhijeet Ravankar, Yukinori Kobayashi, Ankit A. Ravankar, and Takanori Emaru. SLAM within indoor loops by using incremental scan registration. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 720–725. IEEE, 12 2015. ISBN 978-1-4673-7242-8. doi: 10.1109/SII.2015.7405068.
- [127] Viorela Ila, Josep M. Porta, and Juan Andrade-Cetto. Information-based compact pose SLAM. *IEEE Transactions on Robotics*, 26(1):78–93, 2 2010. ISSN 15523098. doi: 10.1109/TRO.2009.2034435.
- [128] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona. A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, pages 0278364914523689–, 5 2014. ISSN 0278-3649. doi: 10.1177/0278364914523689.
- [129] Y. Latif, C. Cadena, and J. Neira. Robust loop closing over time for pose graph SLAM. *The International Journal of Robotics Research*, 32(14):1611–1626, 12 2013. ISSN 0278-3649. doi: 10.1177/0278364913498910.
- [130] J. S. Gutmann and K Konolige. Incremental mapping of large cyclic environments. *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA '99 (Cat. No.99EX375)*, pages 318–325, 1999. doi: 10.1109/CIRA.1999.810068.
- [131] Peter Biber and Wolfgang Straßer. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat.*

- No.03CH37453*), volume 3, pages 2743–2748. IEEE, 2003. ISBN 0-7803-7860-1. doi: 10.1109/IROS.2003.1249285.
- [132] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, Juha Ala-Luhtala, and Achim J. Lilienthal. Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping. In Ieee International, editor, *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2233–2238. Conference on Robotics and Automation p, 2013. ISBN 9781467356411. doi: 10.1109/ICRA.2013.6630878.
- [133] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 10 1994. ISSN 0920-5691. doi: 10.1007/BF01427149.
- [134] Yang Chen and Grard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 4 1992. ISSN 02628856. doi: 10.1016/0262-8856(92)90066-C.
- [135] G Weiss and E von Puttkamer. A map based on laserscans without geometric interpretation. In *Proceedings of Intelligent Autonomous Systems 4(IAS-4)*. pages 403.407, 1995.
- [136] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 10 1997. ISSN 09295593. doi: 10.1023/A:1008854305733.
- [137] S Thrun, W. Burgard, and D Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 321–328. IEEE, 2000. ISBN 0-7803-5886-4. doi: 10.1109/ROBOT.2000.844077.
- [138] M Bosse. Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *The International Journal of Robotics Research*, 23(12):1113–1139, 12 2004. ISSN 0278-3649. doi: 10.1177/0278364904049393.



- [139] K. Ryu, T. Furukawa, S. Antol, and G. Dissanayake. Grid Based Scan-to-Map Matching for Accurate Simultaneous Localization and Mapping: Theory and Preliminary Numerical Study. In *ASME Mechanisms and robotics conference*, Portland, OR, 2014. American Society of Mechanical Engineers.
- [140] M. Tomono. Efficient global scan matching using saliency-based scan point resampling. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1856–1861. IEEE, 2005. ISBN 0-7803-8912-3. doi: 10.1109/IROS.2005.1545180.
- [141] Ji Zhang and Sanjiv Singh. LOAM : Lidar Odometry and Mapping in Real-time. *Robotics: Science and Systems*, 2014. ISSN 0929-5593. doi: 10.1007/s10514-016-9548-2.
- [142] Feng Lu Feng Lu, E.E. Milios, Feng Lu, and Milios. Robot pose estimation in unknown environments by matching 2D range scans. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, volume 18, pages 935–938. IEEE Comput. Soc. Press, 1994. ISBN 0-8186-5825-8. doi: 10.1109/CVPR.1994.323928.
- [143] A. Censi. Scan matching in a probabilistic framework. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2291–2296. IEEE, 2006. ISBN 0-7803-9505-0. doi: 10.1109/ROBOT.2006.1642044.
- [144] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, 27(4):387–407, 11 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9155-6.
- [145] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. SLAM benchmarking, 2009. URL <http://kaspar.informatik.uni-freiburg.de/~slamEvaluation>.
- [146] Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti, Bastian Steder, Rainer Kümmerle, Christian Dornhege, Michael Ruhnke, Alexander Kleiner, and Juan D. Tardos. A comparison of SLAM algorithms based on a graph of relations. In *2009 IEEE/RSJ*

- International Conference on Intelligent Robots and Systems*, pages 2089–2095. IEEE, 10 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354691.
- [147] Andrea Censi. An ICP variant using a point-to-line metric. In *2008 IEEE International Conference on Robotics and Automation*, pages 19–25. IEEE, 5 2008. ISBN 978-1-4244-1646-2. doi: 10.1109/ROBOT.2008.4543181.
- [148] Cyrill Stachniss, U Frese, and G Grisetti. OpenSLAM.org - give your algorithm to the community., 2007. URL <http://openslam.org>.
- [149] S G Tzafestas. *Introduction to mobile robot control*. Elsevier, London, London, 2014. ISBN 9780124170490;9780124171039.
- [150] Anna P Dawson, Skye N McLennan, Stefan D Schiller, Gwendolen A Jull, Paul W Hodges, and Simon Stewart. Interventions to prevent back pain and back injury in nurses: a systematic review. *Occupational and environmental medicine*, 64(10): 642–50, 10 2007. ISSN 1470-7926. doi: 10.1136/oem.2006.030643.
- [151] Guy Fragala and Livia Pontani Bailey. Addressing occupational strains and sprains: musculoskeletal injuries in hospitals. *AAOHN journal : official journal of the American Association of Occupational Health Nurses*, 51(6):252–9, 6 2003. ISSN 0891-0162.
- [152] Glynis Collis Pellatt. The safety and dignity of patients and nurses during patient handling. *British journal of nursing (Mark Allen Publishing)*, 14(21):1150–6, 2005. ISSN 0966-0461.
- [153] Paul Reichert. *Patient Handling Ergonomics*. PhD thesis, New Jersey Institute of Technology, 2004.
- [154] W J McG Tegart. *Smart Technology for Healthy Longevity*. Australian Academy of Technological Sciences and Engineering, 2010. ISBN 978 1 921388 11 8. URL <http://www.atse.org.au/Documents/Publications/Reports/Health&Tech/SmartTechforHealthyLongevity-Report.pdf>.
- [155] J Connell and UK Young. Comprehensive scoping study on the use of assistive technology by frail older people living in the community. *Urbis*, 2008.

- [156] Elizabeth B.-N. Sanders and Pieter Jan Stappers. Co-creation and the new landscapes of design. *CoDesign*, 4(1):5–18, 3 2008. ISSN 1571-0882. doi: 10.1080/15710880701875068.
- [157] Marcus Mast, Michael Burmester, Katja Krüger, Sascha Fatikow, Georg Arbeiter, Birgit Graf, Gernot Kronreif, Lucia Pigini, David Facal, and Renxi Qiu. User-Centered Design of a Dynamic-Autonomy Remote Interaction Concept for Manipulation-Capable Robots to Assist Elderly People in the Home. *Journal of Human-Robot Interaction*, 1(1), 2012. ISSN 2163-0364.
- [158] Jennifer L. Burke, Robin R. Murphy, Michael D. Covert, and Dawn L. Riddle. Moonlight in Miami: Field Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise. *HumanComputer Interaction*, 2011.
- [159] J. Casper and R.R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3):367–385, 6 2003. doi: 10.1109/TSMCB.2003.811794.
- [160] Ezio Manzini and Francesca Rizzo. Small projects/large changes: Participatory design as an open participated process. *CoDesign*, 7(3-4):199–215, 9 2011. ISSN 1571-0882. doi: 10.1080/15710882.2011.630472.
- [161] Christian Bogdan, Anders Green, Helge Hüttenrauch, and K Severinson Eklundh. Cooperative Design of a Robotic Shopping Trolley. In *COST-298*, Copenhagen, 2009.
- [162] Stefan Lie, Dikai Liu, and B Bongers. A cooperative approach to the design of an Operator Control Unit for a semi-autonomous grit-blasting robot. In *Australasian Conference on Robotics and Automation(ACRA)*, 2012.
- [163] Ravindra Ranasinghe, Lakshitha Dantanarayana, Antony Tran, Stefan Lie, Michael Behrens, and LiYang Liu. Smart hoist: An assistive robot to aid carers. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*,

- 
- pages 1285–1291. IEEE, 12 2014. ISBN 978-1-4799-5199-4. doi: 10.1109/I-CARCV.2014.7064501.
- [164] Lakshitha Dantanarayana, Ravindra Ranasinghe, Antony Tran, Dikai Liu, and Gamini Dissanayake. A Novel Collaboratively Designed Robot to Assist Carers. In *The International Conference on Social Robotics*, pages 105–114. Springer, Sydney, 2014. doi: 10.1007/978-3-319-11973-1\_11.
- [165] Alexander G. Reeves and Rand S. Swenson. *Disorders of the Nervous System: A Primer*. Dartmouth Medical School, 2004.
- [166] 221 Robotic systems. In <http://www.team221.com/>, 2014.
- [167] <http://www.fit-pc.com/web/products/intense-pc>.