UNIVERSITY
OF TECHNOLOGY
SYDNEY

Vector Distance Transform Maps for Autonomous Mobile Robot Navigation

by Janindu Sithumini Arukgoda

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of
Prof. Gamini Dissanayake and Dr. Ravindra Ranasinghe

University of Technology Sydney
Faculty of Engineering and Information Technology

August 2020

Certificate of Original Authorship

I, Janindu Sithumini Arukgoda declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signed: Signature removed prior to publication.

Date: 06 - 08 - 2020

Vector Distance Transform Maps for Autonomous Mobile Robot Navigation

by

Janindu Sithumini Arukgoda

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

Abstract

Robot localization, where the position and the orientation of a mobile robot is estimated based on an *a priori* map, is a fundamental problem in autonomous mobile robot navigation. In this thesis, we present how environments can be represented using the vector distance transform for mobile robot localization, and show that it is a superior representation compared to alternatives such as occupancy grid maps and other distance transform variants such as the unsigned distance transform and the signed distance transform.

We propose an approach based on non-linear least squares optimization for robot localization on environments represented by vector distance transform maps, that also captures the uncertainty of the position estimate. We also propose an approach based on extended Kalman filter for robot localization on environments represented by vector distance transform maps. Using simulations, public domain data and real world experiments, the proposed localization approaches are evaluated and compared with existing localization techniques in multiple robot platforms including both ground and aerial robots navigating in 2D and 3D environments using a series of sensors including LiDARs and cameras.

We propose an information filtering based approach for mobile robot localization where a single endpoint range sensor can be used to accurately localize a mobile robot in motion by rotating it in a direction that will improve its position estimate and the corresponding uncertainty. The proposed approach is evaluated using simulations and real world experiments.

Occupancy grid maps, the most popular type of map for robot localization using a range-bearing sensor, are assumed perfect when used for localization. It is a discrete representation of the environment and the probability of occupancy of each cell is independent from its neighbors. Given a set of robot poses and corresponding range-bearing measurements, both incorporated with uncertainty, a representation of environment based on the vector distance transform using non-linear least squares optimization that is both continuous and uncertain is proposed for robot localization. Simulations are used to demonstrate the accuracy of the map and its uncertainties.

Acknowledgements

I would like to thank those who have helped me accomplish this work.

First and foremost, I would like to express my gratitude towards my supervisors, Professor Gamini Dissanayake and Dr. Ravindra Ranasinghe for taking me in as a doctoral student and for guiding me through every step of the way. None of this work would have been possible without your advice, support and guidance.

I would like to thank Professor Tomonari Furukawa, for giving me the opportunity be a part of Team VICTOR that took part in the MBZIRC 2017 competition and for his insight into approaching research problems.

I appreciate the advice and support I received to formulate my research problems and to conduct experiments from Dr. Lakshitha Dantanarayana, Mr. Asok Perera, Mr. James Unicombe, Mr. Maleen Jayasuriya, Dr. Buddhi Wijerathna, and Mr. Nathanael Gandhi. I am thankful for the staff and the students at the Centre for Autonomous Systems at University of Technology Sydney. I have learnt a lot from their knowledge sharing sessions, talks, seminars, and presentations and it has been a privilege to be a part of such a collective.

I would like to thank the members of Team VICTOR for making the time I spent at the Computational Multiphysics Systems Lab at Virginia Tech memorable.

Dr. Chinthana Wimalasuriya, my supervisor at University of Moratuwa, knowingly or unknowingly inspired me to pursue a career in research, and for that, I am grateful to him.

Finally, I am forever indebted to my parents and my wife for supporting me through the years so that I can focus on my work without having to worry about mundane daily tasks. This thesis is dedicated to them.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
AMCL	Adaptive Monte Carlo Localization
AML	Active Markov Localization
CAS	Centre for Autonomous Systems
EKF	Extended Kalman Filter
GPs	Gaussian Processes
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LRF	Laser Range Finder
NDT	Normal Distribution Transform
OGM	Occupancy Grid Map
PDE	Partial Differential Equation
RTK-GPS	Real-Time Kinematics GPS
SDF	Signed Distance Function
SDT	Signed Distance Transform

SLAM Simultaneous Localization and Mapping

UAV unmanned aerial vehicle

UDF Unsigned Distance Function

UDT Unsigned Distance Transform

UTS University of Technology Sydney

UWB Ultra-wideband

VCD Vector Chamfer Distance

VDF Vector Distance Function

VDT Vector Distance Transform

Nomenclature

	General Notations
t	Time (continuous)
k	Time (discrete step)
DT_u	Unsigned Distance Transform
DT_s	Signed Distance Transform
DT_v	Vector Distance Transform
DT_x	x component of the Vector Distance Transform
DT_y	y component of the Vector Distance Transform
DT_z	z component of the Vector Distance Transform
d_{VCD}	Vector Chamfer Distance
d_{VDT}	Disparity vector between the observation and the Vector Distance Transform
r_i	i^{th} range measurement
σ_r	Standard deviation of range measurement noise
θ_i	i^{th} bearing measurement
Z	Space of all possible observations
z_i	i^{th} observation, $z_i \in Z$
X_o	Observation vector
J	Jacobian matrix
H	Hessian matrix

$\boldsymbol{X_R}$	Robot pose
$\hat{\boldsymbol{X}}_{\boldsymbol{R}}$	Robot pose estimate
P	Robot pose covariance matrix
$g(\cdot,\cdot)$	Motion model
$h(\cdot,\cdot)$	Observation model
$\nabla G_{\square}$	Jacobian of the motion model with respect to $\square$
$\nabla H_{\square}$	Jacobian of the observation model with respect to $\square$
υ	Linear velocity
ω	Angular velocity
ν	Innovation vector
K	Kalman Gain
$\square_{k-1}$	Previous state
$\square_{k\|k-1}$	Predicted current state
$\square_{k}$	Updated current state
$cov(\cdot)$	Covariance matrix
$diag(\cdot)$	Diagonal matrix
$trace(\cdot)$	Trace of a matrix
$\Sigma(\cdot)$	Summation
$\int$	Integral operator
∂	Partial differentiation operator

Chapter 1

Introduction

1.1 Motivation and Background

Robots can perform complex, high precision tasks with great speed, and they can perform them repetitively, which is why robots are predominantly used in assembly lines in automated manufacturing plants. When mobility is added to the list of capabilities of a robot, its potential uses exponentially increase. Locomotive systems ranging from wheels to propellers to bio inspired legs are used on robots to give them the capability of mobility. Such robots are called mobile robots.

When a mobile robot can sense the environment, carry out computations to decide which locomotive action needs to be performed next in order to reach its navigation goal, and then perform it, all without any human intervention, it is called an autonomous mobile robot. Today, we use mobile robots for many applications ranging from simple tasks such as cleaning (Roomba), to more advanced tasks such as autopilot (Tesla models) to extremely complex tasks such as exploring planets (Curiosity rover) with varying degrees of autonomy.

Three questions summarize the problems of navigation, ”Where am I?”, ”Where am I going?” and ”How do I get there?” [1]. For a mobile robot to be autonomous, it needs to be able to answer these problems without any human intervention. To answer all three questions, a robot requires knowledge about the environment in the form of a map.

Maps are mathematical representations of the environment that the robot can understand. Depending on the navigation capabilities of the robot and the types of sensors it is equipped with, different types of maps are used. Most common types of maps used in practice are, feature maps, occupancy grid maps and distance transform maps. These maps may not represent the environment fully accurately, as the process of building the map and/or the data used to build the map may contain errors. It is important that these errors are reflected in the map. How this can be achieved is an important research question.

The first question, "Where am I?" is known as the problem of localization. When the position and the orientation of a robot is known with respect to a map, it is said that the robot is localized. Many different approaches are present in literature to solve this problem based on the type of map and the type of sensors used.

Given a set of points, a distance transform describes the distance from any location in the domain to the closest point. As such, a distance transform encodes more information about the environment than geometric maps commonly used for robot navigation as these typically encode just the points or lines or boundaries between occupied and free space in an environment. Distance transform is a concept that originated in the field of image processing. There are three main variants of the distance transform named as the unsigned distance transform, the signed distance transform, and the vector distance transform. Both the signed distance transform and the unsigned distance transform have been used to represent the environment for autonomous mobile robot navigation.

In this thesis, the vector distance transform is used to represent the environment for autonomous mobile robot navigation. How vector distance transform is more suitable to represent the environment for autonomous mobile robot navigation compared to its counterparts signed and unsigned distance transforms is first discussed. A unified framework for localization of mobile robots on vector distance transform maps is presented and the localization capabilities in 2D and 3D environments using different sensors are explored using experiments. How the calculation of pose uncertainty is made possible in an optimization framework by the use of vector distance transforms is also discussed. It is demonstrated that low cost sensors which produce sparse measurements are adequate to perform the localization task under the proposed unified framework. It is also demonstrated that the error of the map building data can be captured and represented in a vector distance transform map, unifying occupancy grid maps that capture environment geometry using aggregation of point clouds, with maps that are a collection of geometric features such as points and lines.

The proposed frameworks and algorithms are validated using experiments and the results are presented and discussed in each chapter. Both simulation and real world environments are used for experiments where appropriate and the results are compared with existing algorithms where available using benchmark datasets.

1.2 Thesis Outline

This thesis is structured as follows.

Chapter 2 first critically analyses the suitability of unsigned, signed and vector distance transforms for the task of representing the environment for autonomous mobile robot navigation, and presents vector distance transform as the most suitable variant. It then presents a novel sensor model for a laser range finder on vector distance transform maps together with a theoretical framework for capturing measurement error, that requires neither explicit extraction or matching of features from observations, nor computationally expensive operations such as ray tracing.

Localization is one of the fundamental requirements for autonomous mobile robot navigation. Chapter 3 presents two mobile robot localization algorithms for environments represented with vector distance transform maps. The first algorithm is based on non-linear least squares optimization. Taking advantage of the superior properties of vector distance transform over unsigned distance transform, the optimization algorithm can also calculate the pose estimate uncertainty. The second algorithm is based on the extended Kalman filter framework. Both algorithms make use of the sensor model presented in Chapter 2. Extensive experimental evaluations are also presented in Chapter 3 for both mobile robot localization algorithms. For this purpose, data collected from a Turtlebot robot navigating in Two-Dimensional (2D) indoor environments, a personal mobility scooter navigating in unstructured outdoor environments and a customized unmanned aerial vehicle (UAV) navigating in outdoor Three-Dimensional (3D) environments, equipped with different sensors such as LiDARs and monocular cameras, are used.

Chapter 4 presents an information gain based active localization approach for mobile robot localization using a low cost rotating range sensor, based on an extended Kalman filter framework. It is demonstrated how instead of using an expensive LiDAR with a large angular field of view, a cheaper rotating range sensor can be used to localize an autonomous mobile robot by actively rotating the range sensor towards the direction which would result in observations that increase the information about the current robot pose and in turn, reduce its uncertainty. Contrary to some active localization approaches, the proposed approach does not consist of an active navigation component. Therefore the proposed active localization approach can be used when the autonomous mobile robot is navigating to carry out an arbitrary task in the most efficient manner.

Mobile robot localization algorithms that use occupancy grid maps or distance transform maps assume the map to be a perfect representation of the environment as these types of maps are incapable of describing potential errors of their representations to the localization algorithms. In Chapter 5, given a set of noisy robot pose estimates and noisy sensor measurements obtained at these poses, an algorithm for generating the corresponding vector distance transform map of the environment is presented. The proposed algorithm also calculates the uncertainty of the map and represents it efficiently using a vector of cubic spline control points. This is achieved by solving a partial differential equation using a cubic spline surface as an approximation function.

Chapter 6 presents a summary of the contributions in this thesis and discusses the limitations and possible future work as the conclusion.

1.3 Contributions

- Representation of environments using vector distance transform maps for autonomous mobile robot navigation.

- A sensor model with theoretically sound error calculation for laser range finders on vector distance transform maps.
- An optimization based mobile robot localization algorithm for environments represented by vector distance transform maps, and an accurate pose uncertainty calculation method.
- An extended Kalman filter based mobile robot localization algorithm for environments represented by vector distance transform maps.
- An information gain based active localization algorithm based on an extended Kalman filter for mobile robots equipped with a low cost rotating range sensor.
- A theoretically sound approach to represent the uncertainty of vector distance transform maps

1.4 Publications

1.4.1 Directly Relevant Publications

1.4.1.1 Published

- Arukgoda, J., Ranasinghe, R., Dantanarayana, L., Dissanayake, G. and Furukawa, T., 2017. Vector Distance Function Based Map Representation for Robot Localisation. In *The Australian Conference on Robotics and Automation (ACRA)*, (Vol. 12). ARAA. ISBN: 978-0-9807404-8-6 ISSN: 1448-2053
- Ranasinghe, R., Dissanayake, G., Furukawa, T., Arukgoda, J. and Dantanarayana, L., 2017, December. Environment representation for mobile robot localisation. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, (pp. 1-6). IEEE. doi: 10.1109/ICIINFS.2017.8300384
- Arukgoda, J., Ranasinghe, R. and Dissanayake, G., 2019, July. Robot Localisation in 3D Environments Using Sparse Range Measurements. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, (pp. 551-558). IEEE. doi: 10.1109/AIM.2019.8868466

- Arukgoda, J., Ranasinghe, R. and Dissanayake, G., 2019, August. Representation of Uncertain Occupancy Maps with High Level Feature Vectors. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, (pp. 1035-1041). IEEE. doi: 10.1109/COASE.2019.8842965

1.4.1.2 In Print

- Jayasuriya, M., Arukgoda, J., Ranasinghe, R. and Dissanayake, G., 2020, May. Localising PMDs through CNN Based Perception of Urban Streets. *2020 International Conference on Robotics and Automation (ICRA).*

1.4.2 Partially Relevant Publications

1.4.2.1 Published

- Perera, A., Arukgoda, J., Ranasinghe, R. and Dissanayake, G., 2017, September. Localization System for Carers to Track Elderly People in Visits to a Crowded Shopping Mall. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, (pp. 1-8). IEEE. doi: 10.1109/IPIN.2017.8115936

- Unicomb, J., Dantanarayana, L., Arukgoda, J., Ranasinghe, R., Dissanayake, G. and Furukawa, T., 2017, September. Distance function based 6DOF localization for unmanned aerial vehicles in GPS denied environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5292-5297). IEEE. 10.1109/IROS.2017.8206421

- Hodges, J., Attia, T., Arukgoda, J., Kang, C., Cowden, M., Doan, L., Ranasinghe, R., Abdelatty, K., Dissanayake, G. and Furukawa, T., 2019. Multistage Bayesian Autonomy for Highprecision Operation in a Large Field. *Journal of Field Robotics*, (Vol 36 (1)), (pp.183-203). doi: 10.1002/rob.21829

Chapter 2

Environment Representation Using Vector Distance Transforms

2.1 Introduction

Key tasks of autonomous robot navigation, such as localization and path planning, require the availability of a map and a sensor model that can correlate sensor observations with the map. The map is essentially a machine-understandable model of the environment, and the sensor model associates the sensor measurements with the robot's state and its internal representation of the environment (i.e., the map). For an autonomous mobile robot, the type of map - sensor model pair used depends on the mobility of the robot, the environment it navigates in, and the sensors the robot is equipped with. For an example, a mobile robot navigating in an uneven terrain requires Three-Dimensional (3D) information to be available by the map, and the sensors should also be able to perceive the 3D environment. Conversely, for a mobile robot navigating in a typical indoor environment, a Two-Dimensional (2D) representation of the environment and sensors that can observe the environment in 2D may be sufficient. In this chapter, a Vector Distance Transform (VDT) based environment representation, and a sensor model that can be used in tandem with the VDT map, are presented.

Three strategies commonly used to represent environments can be found in literature namely, feature maps, Occupancy Grid Map (OGM) and distance transforms.

Feature maps describe the environment using geometric primitives that can be observed in the environment such as points, corners, lines and planes. A compact state vector to represent the environment may be obtained from such a set of features using their geometric properties. For an example, in 3D space, the coordinates of points and/or start and end points of a set of lines may be used as a feature vector. When a sensor suitable for observing these features such as a camera or a laser range finder is used, the camera image or the laser scan may be processed to extract such geometric features from the observation and associate them with features present in the map. Observation equations can then be derived to associate the robot pose, sensor observation and the environment to help localize the robot in an estimation theoretic framework such as an Extended Kalman Filter (EKF) [2]. In many practical situations, the associated algorithms being robust to outliers and computationally efficient makes feature maps a suitable candidate for robot localization. However, the process of extracting sparse features from the environment discards a significant portion of information obtained by the sensor. Therefore subsequent tasks such as path planning find sparse feature maps lacking in detail. Alternatively, dense feature maps require a large memory to maintain the map and feature association is computationally more demanding.

OGMs, on the other hand, represent the environment with finite sized cells [3, 4]. Each cell encodes a value that describes the probability of that cell being occupied as *occupied*, *free* and *unknown*. An OGM requires a comparatively larger memory. However, the clear definition of occupied and free space makes OGM the most used type of map for path planning, obstacle avoidance and exploration. OGM is mostly used in tandem with sensors that can measure distances to obstacles. In contrast with feature maps, an analytical expression that associates the robot pose, range observations and the environment cannot be derived. Therefore, ray tracing methods are used together with the particle filter algorithm to localize robots in OGM. The computational burden of ray tracing increases exponentially when a 3D environment is represented by an OGM.

Another limitation of the OGM is that its accuracy is dependent on the resolution. Normal

Distribution Transform (NDT) is a variant of OGM that attempts to overcome this by representing each individual cell in the occupied space using a Gaussian distribution [5, 6]. Furthermore, OGM assumes that the state of each cell is independent from its neighboring cells, which ignores the spatial correlation between cells that represent the occupancy of the same object. Gaussian Processes (GPs) [7] have been used to learn a continuous OGM to overcome these limits [8, 9]. However, learning GPs is computationally expensive with a $O(N^3)$ running time where N is the size of the training data set.

Distance transforms have recently emerged as another environment representation method [10–13]. Distance transforms capture information on the distance to the closest occupied point in an environment. Therefore, the geometry of the environment is not explicitly captured. Instead, it is captured implicitly as the zero level set of the distance transform. Therefore it is a much richer representation of the environment compared to sparse feature maps and OGM that only maintain information about the occupancy. The three main variants of the distance transform are the Unsigned Distance Transform (UDT), the Signed Distance Transform (SDT) and the VDT. The UDT simply represents the distance to the closest occupied point in space. The SDT associates the distance with a sign based on whether the space is inside or outside an enclosed boundary. This makes SDT a richer representation than the UDT. VDT represents the distance using orthonormal vectors, thus giving the richest distance transform representation. This thesis explores the use of a VDT based environment representation for robot localization and mapping.

This chapter is organized as follows. In Section 2.2, an in depth discussion on distance transforms is presented and the use of VDT as a suitable environment representation method for environment representation is justified. Section 2.3 presents a sensor model that complements VDT. Section 2.4 concludes this chapter.

2.2 Distance Functions and Distance Transforms

Distance functions first emerged in the image processing domain [14], for applications such as cluster detection, elongated part detection, regularity detection, skelitonizing and shape factor computation [15, 16]. In a digital image, each pixel is identified by a unique pair of positive integers (i, j). For the set of all possible integer pairs (pixels) A, a function f that maps $A \times A$ into a non negative real number is called

1. Positive definite, $iff f(x, y) \geq 0 \; \forall x, y \in A \; and \; f(x, y) = 0 \iff x = y$
2. Symmetric, $if \; f(x, y) = f(y, x) \; \forall x, y \in A$
3. Triangular, $if \; f(x, y) \leq f(x, z) + f(z, y) \; \forall x, y, z \in A$.

f is a distance function if it is positive definite, symmetric and triangular.

There are several distance metrics used in distance functions, such as city block distance $d_{cityblock}$, chessboard distance $d_{chessboard}$, euclidean distance $d_{euclidean}$ and quasi-euclidean distance $d_{quasi\_euclidean}$. Between two pixels (i, j) and (h, k), these metrics are defined as

$$d_{cityblock}[(i, j), (h, k)] = |i - h| + |j - k| \tag{2.1}$$

$$d_{chessboard}[(i, j), (h, k)] = max(|i - h|, |j - k|) \tag{2.2}$$

$$d_{euclidean}[(i, j), (h, k)] = \sqrt{(i - h)^2 + (j - k)^2} \tag{2.3}$$

$$\begin{aligned} & if \; |i - h| > |j - k| \\ d_{quasi\_euclidean}[(i, j), (h, k)] &= |i - h| + (\sqrt{2} - 1) \cdot |j - k| \\ & otherwise \\ d_{quasi\_euclidean}[(i, j), (h, k)] &= (\sqrt{2} - 1) \cdot |i - h| + |j - k| \end{aligned} \tag{2.4}$$

and are illustrated in Figure 2.1.

For a 2D binary image where pixels belong either to the set S, the set of 1's that belong to objects, or the set $\bar{S}$, the set of 0's that belong to the background, a distance map or a distance transform $L(S)$ [14] is an image defined as

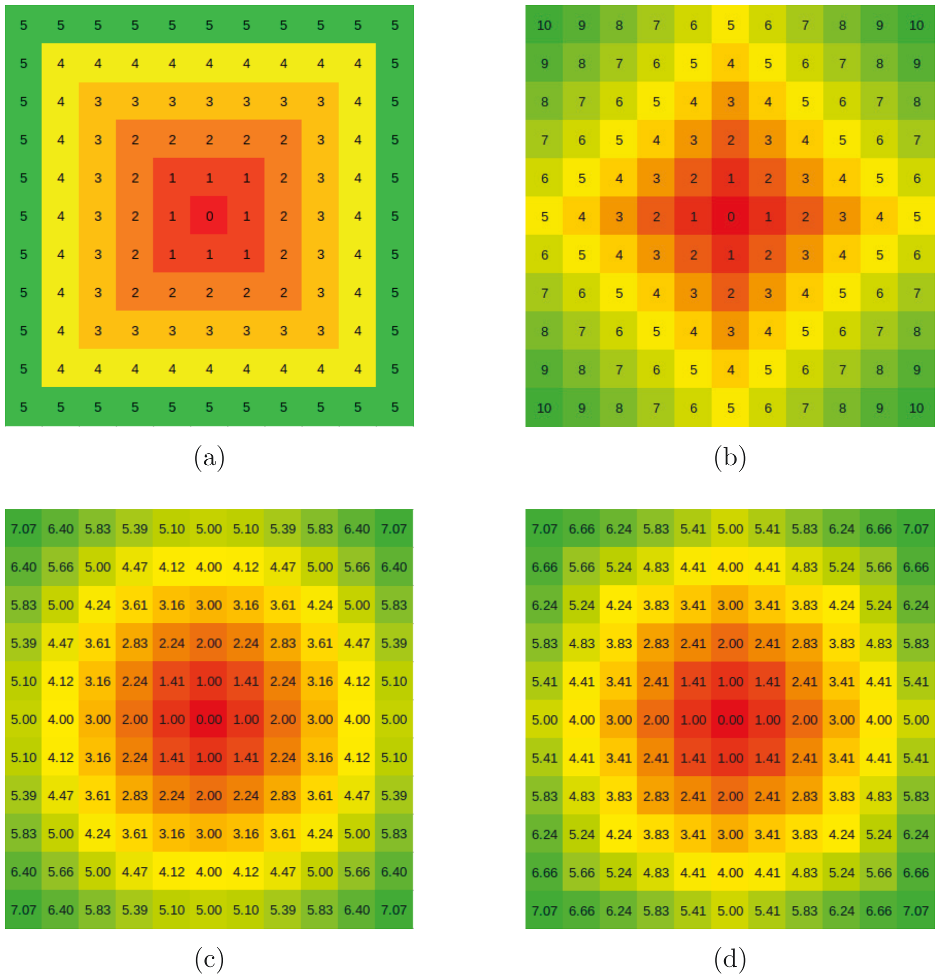


FIGURE 2.1: Distance from pixel (6,6) to all pixels based on (a) Chessboard, (b) City Block, (c) Euclidean and (d) Quasi-Euclidean distance metrics

$$L(x) = min(d[(i,j), \bar{S}]) \;\; \forall x = (i,j) \in S \tag{2.5}$$

That is, all pixels that belong to S have a label assigned in $L(S)$ describing the distance to the closest pixel that belongs to $\bar{S}$. A similar distance map can be derived for the background as well, where $L(\bar{S})$ would assign a label to each background pixel in the image the minimum distance to an object.

There appears to be a confusion in the nomenclature where the term "distance function" is also used in the more recent literature to describe "distance transforms" [17–20]. Observing this, from this point on, the term "distance function" is used synonymously with "distance transform", shifting from its traditional definition.

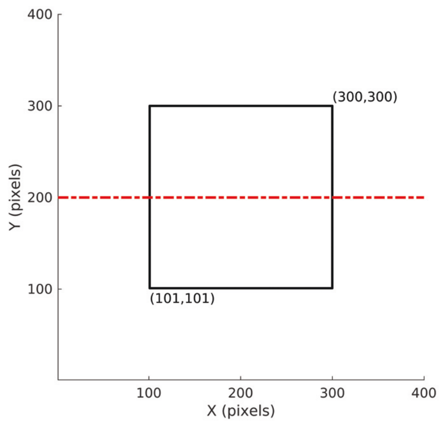


Figure 2.2: Box shaped environment

Distance transforms are used to represent environments in robotics applications, especially for mobile robot localization [20–22]. Euclidean distance is the most natural form of distance metric for environment representation. A distance transform based on euclidean distance metric is inherently non-negative, and is commonly known as an UDT. SDT is used to represent environments where the environment is populated with closed boundaries [23]. The distance transform is assigned to be negative sign for points within the boundary. VDT is another form of distance transform where the distance function at any point in space is described using orthonormal vectors. For a simple box shaped environment shown in Figure 2.2, the respective distance transforms are shown in Figure 2.3 and distance transform values along the dotted red line are plotted in Figure 2.4.

The behavior of the distance transforms at the boundaries can be observed at $x = 101$ and at $x = 300$. The distance transform is continuous in all cases. However, there is a discontinuity in the first order derivative of the UDT at the boundary whereas the first order derivatives of the SDT and VDT are continuous. UDT and SDT at the cut locus, the locus of points that are equidistant from multiple boundaries, are continuous while the first order derivatives at the cut locus are discontinuous in both transforms. In VDT, the $\mathbf{x}$ component of the vector itself is seen to be discontinuous and in general, at least one vector component becomes discontinuous at cut loci.

These discontinuities have to be taken into account when a continuous representation of the distance transform is obtained using a suitable interpolation method. In the case

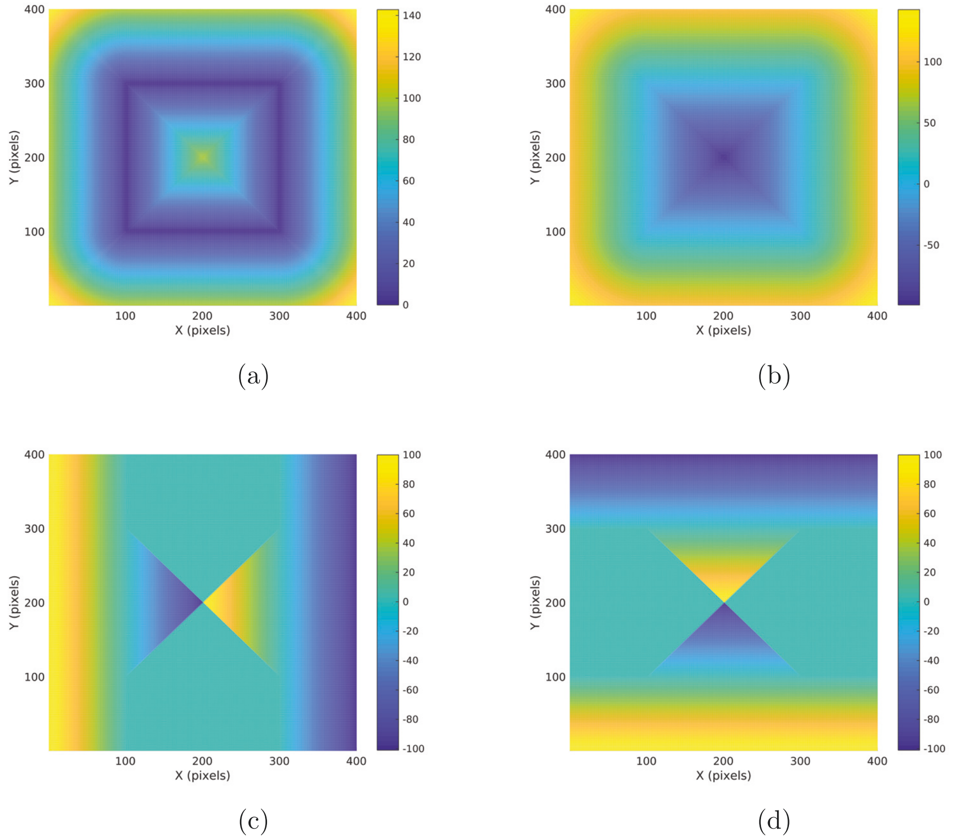


FIGURE 2.3: Shows a comparison between (a) UDT, (b) SDT, (c) **x** component of VDT and (d) **y** component of VDT for the environment shown in Figure 2.2

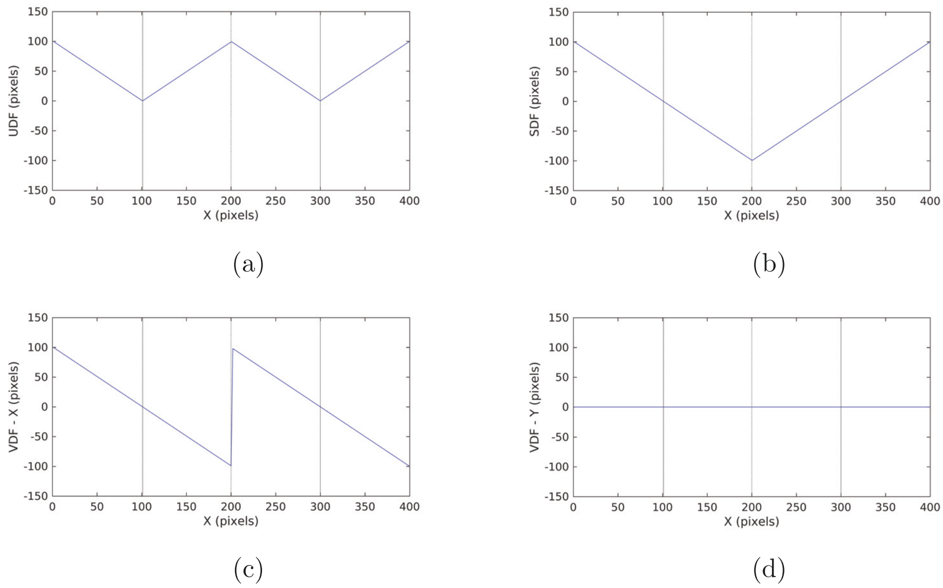
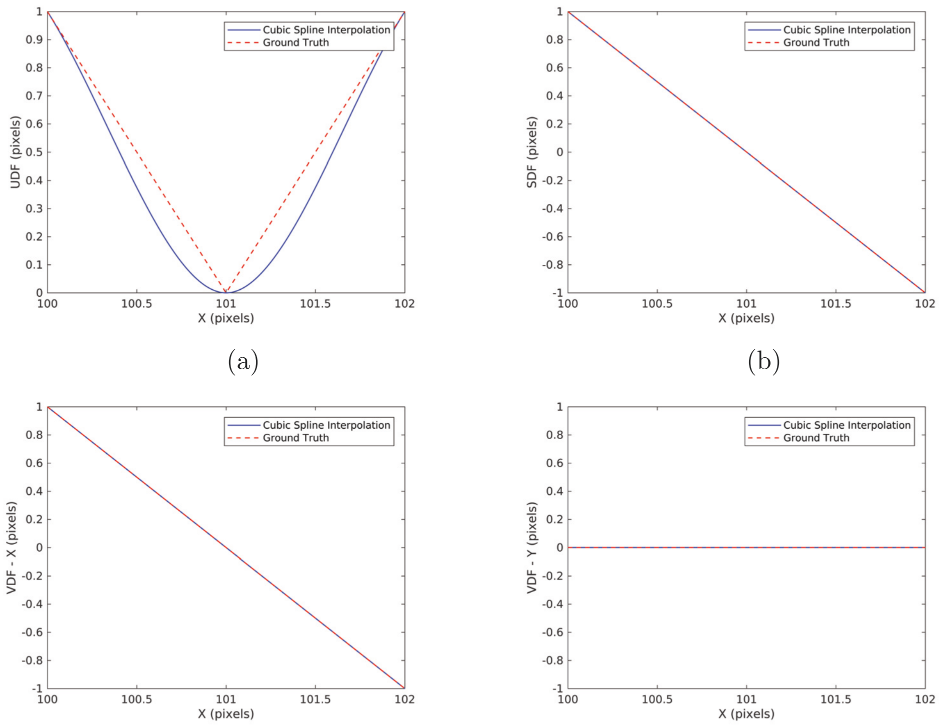


FIGURE 2.4: Shows a comparison between (a) UDT, (b) SDT, (c) **x** component of VDT and (d) **y** component of VDT along the dotted line in Figure 2.2



(a) (b) (c) (d)

FIGURE 2.5: The difference of behaviours of cubic spline interpolations of (a) UDT, (b) SDT, (c) $\mathbf{x}$ component of VDT and (d) $\mathbf{y}$ component of VDT close to the boundary compared with the ground truth

of the environment shown in Figure 2.2, the behaviour of cubic spline interpolations of UDT, SDT and VDT close to the boundary at $x = 101$, $y = 200$ along the $\mathbf{x}$ direction is demonstrated by Figure 2.5. While the interpolations of SDT and VDT correspond with their true values, a significant deviation can be observed in the interpolation of UDT. The corresponding gradients are shown in Figure 2.6, demonstrating how the UDT behaves poorly at the boundary.

The behavior of the cubic spline interpolation of UDT, SDT and VDT close to the cut locus at $x = 200.5$, $y = 200.5$ is demonstrated in Figure 2.7. The interpolations of all three variants are seen to be deviating from the true values at the cut locus. The corresponding cubic spline interpolation of gradient values are shown in Figure 2.8.

When a distance transform is used to represent the environment to localize mobile robots, a continuous representation is essential. The alternative to interpolating a high resolution discrete distance transform is to calculate the function values at query time, which

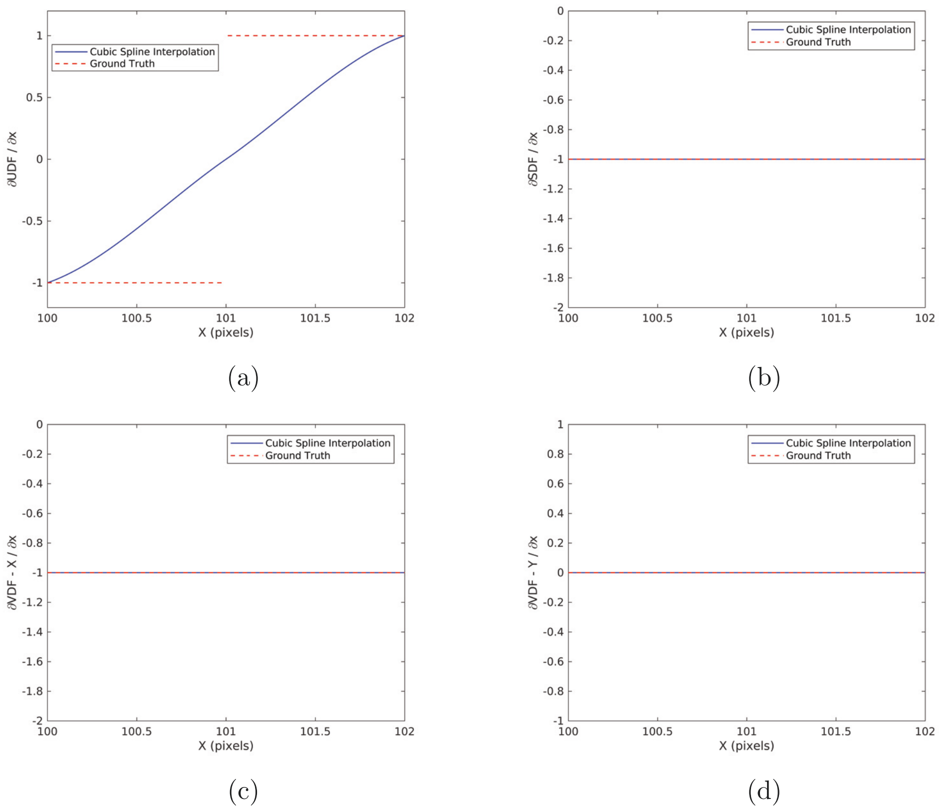


FIGURE 2.6: The difference of behaviours of cubic spline interpolations of the gradients along the **x** direction of (a) UDT, (b) SDT, (c) **x** component of VDT and (d) **y** component of VDT close to the boundary compared with the ground truth

is computationally inefficient. The localization techniques introduced and discussed in Chapter 3 require accurate distance transforms, and their gradients along the **x** and **y** directions close to the boundary. Therefore, when a continuous distance transform is to be obtained using an interpolation method for the purpose of mobile robot localization, SDT and VDT clearly are more suitable than UDT. SDT however has a major limitation where it can only be used to represent environments with closed shapes such that positive and negative signs can be assigned to the function values appropriately. Therefore it cannot be used in a straightforward manner to represent unstructured environments that a typical mobile robot might navigate in. While at cut loci, at least one VDT component is discontinuous and its interpolation produces the largest errors when compared with UDT and SDT, distance function values close to the cut loci are usually disregarded in mobile robot localization techniques, as explained in Chapter 3. Therefore the VDT is proposed as the most suitable distance transform to represent the environments for mobile robot

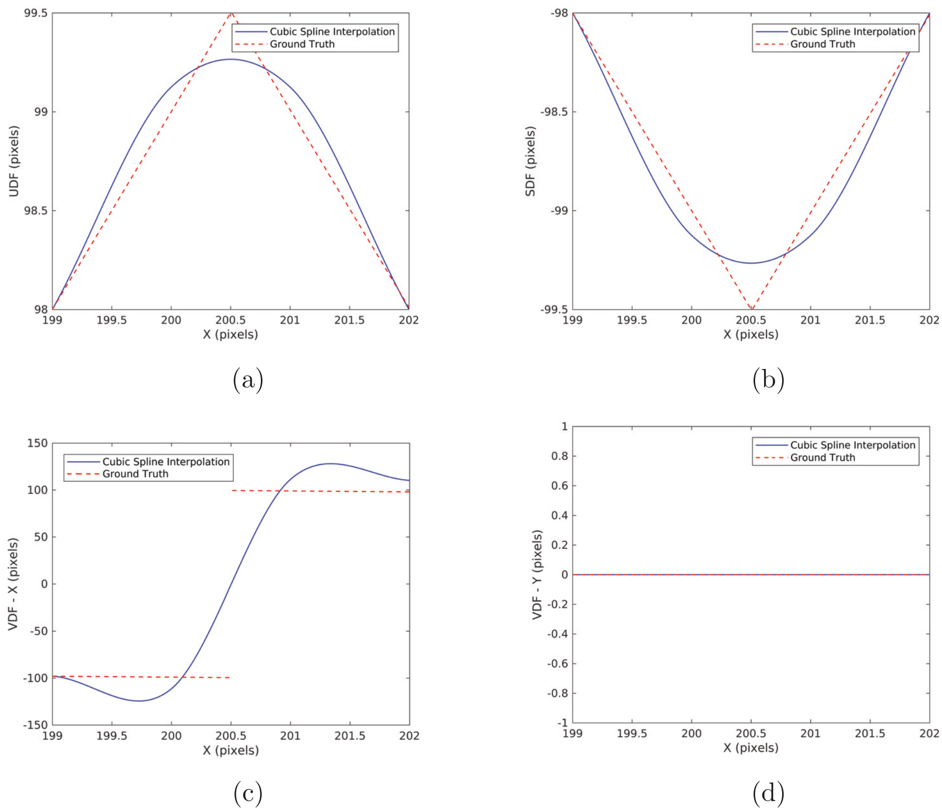


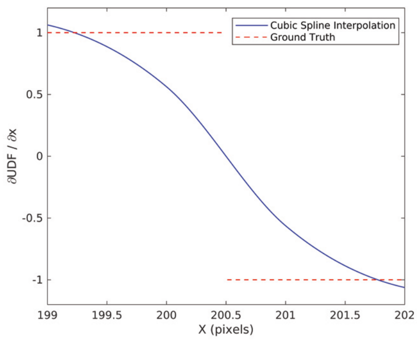
FIGURE 2.7: The difference of behaviours of cubic spline interpolations of (a) UDT, (b) SDT, (c) **x** component of VDT and (d) **y** component of VDT close to the cut locus compared with the ground truth

localization applications.

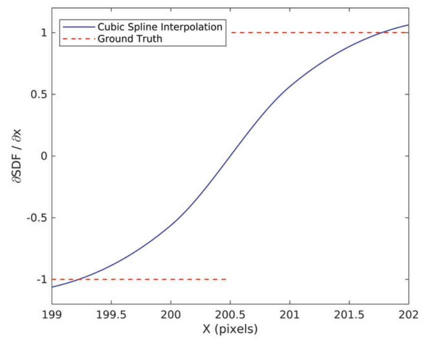
Another advantage of VDT over UDT and SDT is that the direction to the closest occupied point in space is also captured. When a UDT or an SDT is queried at a point P and a distance value d (which is the distance to the closest occupied point) is returned, the closest occupied point in space lies in the circumference of a circle origined at P with a radius $|d|$. When a VDT is queried at point P however, the exact location of the closest occupied point can be easily derived.

2.3 Sensor Models

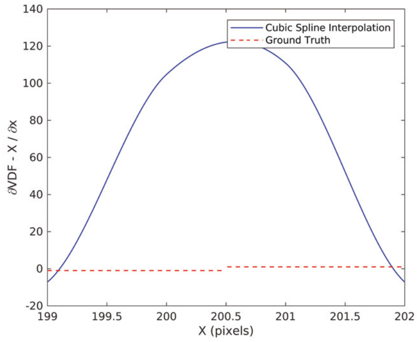
A sensor can be defined as a device that can measure some physical quantity and convert it into an electronic signal that is processed by a computer. Autonomous robots are usually



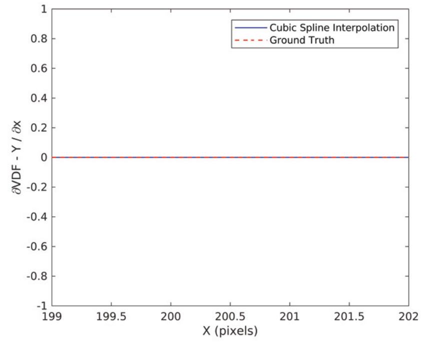
(a)



(b)



(c)



(d)

FIGURE 2.8: The difference of behaviours of cubic spline interpolations of the gradients along the $\mathbf{x}$ direction of (a) UDT, (b) SDT, (c) $\mathbf{x}$ component of VDT and (d) $\mathbf{y}$ component of VDT close to the cut locus compared with the ground truth

equipped with multiple sensors that allow it to estimate its own state and the state of the environment it is in. At a very high level, sensors can be classified as proprioceptive sensors and exteroceptive sensors. Proprioceptive sensors are sensors that measure physical quantities of the robot itself. Wheel encoders and Inertial Measurement Unit (IMU) are typical examples of proprioceptive senors. Exteroceptive sensors, conversely, are sensors that measure physical quantities of the environment that are not a part of the robot, such as cameras, sonars and laser range finders.

A sensor model associates the sensor measurements with the internal representation of the environment based on the robot state. In the most generic form, it may describe the likelihood of an observation $\mathbf{z} \in Z$ given the state of the robot $\mathbf{x}$ as the conditional probability

$$P(\mathbf{z} \mid \mathbf{x},\ m) \tag{2.6}$$

where Z is the set of all possible observations and m is the internal representation of the environment or the map. The construction of the sensor model can be done by fixing the robot state $\mathbf{x} = x$ and then examining the resulting probability density function $P(\mathbf{z} \mid x,\ m)$. Once the sensor model is constructed, for the observation $\mathbf{z} = z$, the robot state $\mathbf{x}$ can be inferred by the probability distribution of $P(z \mid \mathbf{x},\ m)$ [24].

In an environment represented by features, when the robot is equipped with a suitable sensor that can observe these features, the distribution of the sensor measurements at an estimated robot pose $\mathbf{x} = x$ can be modeled as

$$P(\mathbf{z} \mid x,\ m) \sim \mathcal{N}(z, \sigma_r^2) \tag{2.7}$$

where the measurement noise is a zero mean Gaussian distribution with variance σ_r^2 and z is the expected sensor measurement. The calculation of the expected sensor measurement based on a robot pose estimate and a feature at a known location is straightforward. This allows a measurement model $h(.)$ to be defined as

$$h(x\ , m) = \mathbf{z} + \omega_k \tag{2.8}$$

where $\omega_k \sim \mathcal{N}(0, \sigma_r^2)$ is the measurement noise. The error between the expected sensor measurement obtained using this sensor model and the actual sensor measurement is used to model the mobile robot localization problem, as discussed in Chapter 3.

When the environment is represented by an OGM and the robot is equipped with a laser range finder, an expected measurement z can be calculated using ray casting, also known as ray tracing. For a laser range finder with a range limited to an interval $[r_{min}, r_{max}]$, the distribution of the sensor measurement can be modeled as

$$P(\mathbf{z} \mid x,\ m) = \begin{cases} \eta\, \mathcal{N}(z, \sigma_r^2), & if\ r_{min} \leq \mathbf{z} \leq r_{max} \\ 0, & otherwise \end{cases} \tag{2.9}$$

where the normalizer η can be expressed as follows

$$\eta = \left(\int_{r_{min}}^{r_{max}} \mathcal{N}(z, \sigma_r^2)\, d\mathbf{z} \right)^{-1} \tag{2.10}$$

for OGMs [25].

A sensor model for a robot equipped with a laser range finder navigating in an environment represented by a VDT map is presented next.

2.3.1 A Sensor Model For Laser Range Finder on Vector Distance Transform

The end points of a laser scan $z_i \in Z;\ z_i = \{r_i, \theta_i\}$ can be projected into the 2D space based on a robot pose estimate $\boldsymbol{X_R} = (x_R, y_R, \phi_R)^\top$ as

$$X_o = \begin{Bmatrix} x_{o_i} \\ y_{o_i} \end{Bmatrix} = \begin{Bmatrix} x_R + r_i \sin(\theta + \phi_i) \\ y_R + r_i \cos(\theta + \phi_i) \end{Bmatrix} \tag{2.11}$$

to obtain an observation vector X_o in the Cartesian space. Projecting this observation vector over the distance transform used to describe the environment produces a vector of distance values, each value explaining the distance between the corresponding laser end point and the closest occupied point in the space, based on the type of distance transform used to represent the map. 2.12 shows the vector of distance values d_{VDT} obtained when a VDT is used to represent the environment.

$$d_{VDT} = VDT(X_o) = \begin{bmatrix} DT_x(X_{o_1}) \\ DT_y(X_{o_1}) \\ . \\ . \\ DT_x(X_{o_i}) \\ DT_y(X_{o_i}) \\ . \\ DT_x(X_{o_n}) \\ DT_y(X_{o_n}) \end{bmatrix} \tag{2.12}$$

Without loss of generality, two assumptions may be taken about laser range finders.

1. Each range measurement is independent within a laser scan
2. Range measurement noise σ_r is the only contributing factor to the sensor noise

These two assumptions allow the sensitivity of d_{VDT} due to the noise in sensor measurement to be described by a diagonal covariance matrix as

$$\Sigma_{d_{VDT}} = diag(\sigma^2_{DT_x,r_1}, \sigma^2_{DT_y,r_1}, ..., ..., \sigma^2_{DT_x,ri}, \sigma^2_{DT_y,ri}, ..., \sigma^2_{DT_x,rn}, \sigma^2_{DT_y,rn}) \tag{2.13}$$

σ^2_{DTx,r_i} and σ^2_{DTy,r_i} can be calculated as

$$\begin{aligned} \sigma^2_{DTx,r_i} &= J_{DT_x,r_i} \cdot \sigma^2_r \cdot J^\top_{DT_x,r_i} \\ \sigma^2_{DTy,r_i} &= J_{DT_y,r_i} \cdot \sigma^2_r \cdot J^\top_{DT_y,r_i} \end{aligned} \tag{2.14}$$

where the Jacobians J_{DT_x,r_i} and J_{DT_y,r_i} are derived as

$$\begin{aligned} J_{DT_x,r_i} &= \left.\frac{\partial d_{DTx}}{\partial r}\right|_{r_i} = \left.\frac{\partial DTx}{\partial x_{o_i}}\right|_{x_{o_i}} \cdot \left.\frac{\partial x_{o_i}}{\partial r}\right|_{r_i} + \left.\frac{\partial DTx}{\partial y_{o_i}}\right|_{y_{o_i}} \cdot \left.\frac{\partial x_{o_i}}{\partial r}\right|_{r_i} \\ J_{DT_y,r_i} &= \left.\frac{\partial d_{DTy}}{\partial r}\right|_{r_i} = \left.\frac{\partial DTy}{\partial x_{o_i}}\right|_{x_{o_i}} \cdot \left.\frac{\partial x_{o_i}}{\partial r}\right|_{r_i} + \left.\frac{\partial DTy}{\partial y_{o_i}}\right|_{y_{o_i}} \cdot \left.\frac{\partial x_{o_i}}{\partial r}\right|_{r_i} \end{aligned} \tag{2.15}$$

where $\frac{\partial DTx}{\partial x_{o_i}}$, $\frac{\partial DTx}{\partial y_{o_i}}$, $\frac{\partial DTy}{\partial x_{o_i}}$ and $\frac{\partial DTy}{\partial y_{o_i}}$ can be precomputed at the time of map generation. A sensor model similar to those presented in (2.7) and (2.9) for a laser range finder on a VDT can now be written as

$$P(\mathbf{z} \mid x, m) \sim f(d_{VDT}, \Sigma_{d_{VDT}}) \tag{2.16}$$

For a VDT, f is a normal distribution. A similar sensor model can be derived based on SDT and UDT as well. However, while f is a normal distribution in the case of SDT, f is a folded normal distribution in UDT [13, 26].

d_{VDT} vector in 2.12 describes the error or the disparity between the expected and actual sensor measurements at robot pose $\mathbf{x} = x$. This vector is different from its counterparts derived for SDT and UDT in that each element pair in the vector also describes the direction towards the closest occupied point. How this disparity vector can also be used to localize mobile robots in an EKF framework is explained in Chapter 3.

A scalar disparity measure can also be derived based on d_{VDT}. Hausdorff distance [12] and Chamfer distance [27] are such scalar disparity measures that have been used in mobile robot localization [20–22]. The Hausdorff distance captures the largest error while the Chamfer distance captures the average error within the vector. Given that d_{VDT} values are not constrained to the domain of positive real numbers, using Chamfer distance leads to a loss of information. Therefore, a scalar error metric inspired by Chamfer distance called Vector Chamfer Distance (VCD) is introduced as

$$d_{VCD} = \sum_{i=1}^{2n} d_{VDT}(i)^2 = \sum_{i=1}^{n} \Big(DTx(X_{o_i})^2 + DTy(X_{o_i})^2 \Big) \tag{2.17}$$

How this scalar disparity measure can be used to localize mobile robots in a non-linear optimization framework is also explained in Chapter 3.

2.4 Discussion and Conclusions

Distance transforms can be used to represent environments for robot navigation applications. Being a continuous representation that captures the geometry of the environment without the requirement of extracting features from observations and explicitly associating them to the map, it is proposed that distance transform can be preferred over feature maps and OGMs for environment representation on autonomous robots. Unsigned and signed variants of the distance transform have been used to represent environments for robot localization applications. In this chapter, the vector distance transform is proposed as a superior alternative to its signed and unsigned counterparts as it is well behaved at the boundary region compared to the unsigned distance transform, it can be used to represent complex environments unlike signed distance transform, and it contains more information about the environment than both signed and unsigned distance transforms.

Next, a sensor model that complements the vector distance transform is introduced for laser range finders. The observation vector and its sensitivity to noise in sensor measurements are analytically derived. A vector of disparity between the expected and actual sensor measurement d_{VDT}, and a scalar measure of disparity between the expected and actual sensor measurement d_{VCD} inspired by Chamfer distance, are proposed for the use of robot localization.

In the next chapter, Chapter 3, mobile robot localization on environments represented by vector distance transforms using the proposed sensor model is presented, and discussed using experimental results.

Chapter 3

Vector Distance Transform Based Robot Localization

3.1 Introduction

“Where am I?”, “where am I going?” and “how should I get there” are the three questions that need to be answered in navigation [1]. Discovering the answer to the first question is known as localization. The answer lies in working out where you are in a given environment, based on what you observe and what prior information you have. In mobile robot navigation, localization can be formally defined as estimating the robot pose with respect to an *a priori* map based on sensor measurements and prior information. An accurate pose estimate can simplify many tasks in autonomous navigation such as motion planning. In this chapter, two strategies for robot localization when the environment is represented using a Vector Distance Transform (VDT) are presented and discussed.

Autonomous mobile robots operating in a wide range of environments use various strategies to estimate their pose in the environment. When the environment is known and a map of the environment is available, the pose estimation is a localization problem. Alternatively, when a robot explores uncharted territories and an *a priori* map is not available, it has to resort to Simultaneous Localization and Mapping (SLAM) where the environment and the robot pose within it are both estimated simultaneously [28]. While SLAM is a powerful technique that can be applied in many situations, building a map first and navigating in that known environment using localization is more desirable if the situation allows it. This is mainly due to the fact that, given a state estimation framework, SLAM is inherently a more computationally expensive task, as it has to estimate a larger state vector. Also, it is counter-intuitive to continuously estimate a known, static environment. Furthermore, when an *a priori* map is available, path planning can be done efficiently.

3.1.1 Robot Localization in Outdoor Environments

In outdoor environments, Global Navigation Satellite Systems (GNSS) such as Global Positioning System (GPS) [29] and Global Navigation Satellite Systems (GLONASS) are used for estimating the geolocation. GPS is so widely used, in literature the term GPS is used interchangeably with GNSS. However, their accuracy usually ranges from a few meters to over 20m [30]. While approaches such as differential GPS (DGPS) [30], assisted GPS (AGPS) [31] and Real-Time Kinematics GPS (RTK-GPS) [32] can improve the accuracy, their reliability, especially in urban areas, is poor.

Alternatively, on-board sensors such as cameras and LiDARs have also been used to locate robots in outdoor environments, both as an alternative to GPS and in tandem with it.

Using cameras, robots can be localized in outdoor environments by matching the observed images with a database of position-annotated template images [33–35], using feature descriptors such as SIFT [36] and SURF [37]. While this approach does not require maintaining large-scale maps, the localization is coarse. Visual SLAM systems have also been developed for outdoor environments such as Stereo-Vision-SLAM [38], S-PTAM [39], LSD-SLAM [40, 41], ORB-SLAM [42], ORB-SLAM2 [43], and RTAB-MAP [44]. Further study on SLAM is presented in Chapter 5.

One of the most popular applications of localization in outdoor environments is autonomous driving. The self-driving vehicles require additional information about the environment such as the number of lanes in the road, road furniture and local speed limits, from the map for decision making. This requires an information rich map that is also used for localization. Local Dynamic Map (LDM) is a standard method for representing information in multiple layers [45] that self-driving pioneers such as Waymo, Uber and Tesla use. A typical LDM will represent the topological information about the environment in the first layer, the static elements such as buildings and pavements in the second layer and more dynamic information such as traffic light timing and variable speed limits on next layers. Generating these maps and maintaining them are resource heavy.

The dynamic and unstructured nature of outdoor environments makes building and maintaining consistent maps over a period of time difficult. Visual odometry is an alternative approach for pose estimation without the use of an *a priori* map, where sequential camera images are used to estimate the relative motion between images [46] in a form of dead reckoning. Images captured from monocular cameras [47–50], stereo cameras [51–53], and omnidirectional cameras [54, 55] can be used for visual odometry. Traditionally, features are extracted from the image feed and these features are tracked for motion estimation [46, 51–54]. The alternative approach is known as the direct method, where the information is obtained from the entire image without explicit feature extraction [47, 55]. Semi-direct visual odometry uses a hybrid of feature based and direct methods [48, 49]. Similar to odometry measured from actuators such as rotary wheel encoders, visual odometry suffers from drift over time. If information from an Inertial Measurement Unit (IMU) is available, the visual odometry can be augmented with it to reduce this drift and improve the accuracy and it is known as visual inertial odometry [56–61]. VINS-MONO [62] and Kimera-VIO [63] are two popular open source visual inertial odometry implementations.

3.1.2 Robot Localization in Indoor Environments

When the environment is represented using geometric primitives such as points, line segments and corners, a robot equipped with a sensor such as a LiDAR that can acquire range and / or bearing can be localized using an Extended Kalman Filter (EKF) [2, 64]. EKF is a computationally efficient algorithm for estimating the mean and the covariance of a state vector describing a nonlinear system. The algorithm consists of a prediction step for propagating the state mean and covariance over time based on a model and an observation step for correction of the state mean and covariance based on sensor measurements. A drawback of the EKF is that it requires a close initial guess for the state vector. Therefore it cannot be used in scenarios where the initial location of the robot is unknown, as in the kidnapped robot problem. Also it should be noted that when features are extracted from the sensor observation, a significant portion of information obtained is discarded.

Alternatively, when the environment is represented using an Occupancy Grid Map (OGM) particle filters have been used widely for robot localization [25, 65, 66]. The particle filter algorithm uses, as the name suggests, a population of weighted particles to approximate the posterior probability density function that describes the state of a system. In robot localization, each particle hypothesizes the robot pose and at the initialization step they are uniformly weighted and randomly distributed. Using a sensor model, at each observation, the likelihood of each hypothesized pose is updated and then the particle population is resampled by purging less likely particles (lower weighted particles) and replacing them by populating more likely particles. When the robot is equipped with a LiDAR sensor, ray tracing is used for the sensor model, as explained in Section 2.3 in Chapter 2. The advantages of the particle filter are that the measurement and propagation errors need not be gaussian, and the robot can be localized without the need of an initial guess. However, to approximate the posterior probability density function that describes the robot pose, a sufficiently large population of particles is required and this makes the algorithm computationally less efficient than an EKF, specially in 3D. Adaptive Monte Carlo Localization (AMCL) is the most widely used particle filter implementation with 2D OGMs [67]. GMapping [68] and Cartographer [69] are two widely used SLAM algorithms used to build OGMs.

Optimization based methods have also been used to localize mobile robots. Scan matching techniques such as Iterative Closest Point (ICP) [70] are a set of such optimization based methods [71–73]. An optimization based localization method on OGMs using its Unsigned Distance Transform (UDT) is proposed in [20].

Motion capture systems such as VICON [74] and Motive [75] can also be used to estimate the pose of robots with high accuracy in indoor environment. Motion capture systems observe light reflective or light emitting markers attached to the robot using multiple calibrated cameras to estimate the position of each marker.

Ultra-wideband (UWB) signal and Wi-Fi signal based localization strategies are also popular in indoor environment. Using a signal receiver on-board the robot and anchor nodes in the environment transmitting the signal, the robot pose can be estimated based on time of arrival (ToA), time difference of arrival (TDoA), receiver signal strength (RSS) and angle of arrival (AoA). In ToA method, the time taken for the signal to travel between the transmitter and the receiver is used to calculate the distances, and trilateration is then used for position estimation [76, 77]. In TDoA method, the differences between the time it took for the signal to arrive between multiple fixed anchor nodes are used to estimate the robot position [78]. RSS is the measurement of power in the received signal, which deteriorates with distance. RSS fingerprints which consist of readings from multiple sources can be used to estimate the robot pose [79, 80]. In AoA method, the robot position is estimated by measuring the angle of incidence at which the signal arrives and calculating the intersection of lines of bearing [81, 82].

In this chapter localization strategies in a non-linear least squares optimization framework and in an EKF framework based on the VDT representation of environments and the sensor model presented in Chapter 2 are presented. Furthermore, an approach for accurate estimation of localization uncertainty in the non-linear least squares optimization framework is presented. These algorithms are evaluated using simulations, publicly available datasets and real life experiments in indoor and outdoor environments using a series of sensors including RGB cameras and LiDARs.

This chapter is organized as follows. Section 3.2 presents the non-linear least squares optimization based localization strategy and the approach for accurate pose uncertainty estimation. Section 3.3 presents an EKF formulation for robot localization in VDT maps. Section 3.4 presents experiments and Section 3.5 concludes this chapter with a discussion.

3.2 Non-linear Least Squares Optimization For Localization on Vector Distance Transform Based Maps

This section presents a strategy based on non-linear least squares optimization to localize mobile robots in environments represented by VDTs. The optimization based localization strategy uses the sensor model and the scalar disparity measurement introduced in Section 2.3.1. Therefore there is no requirement for explicit feature extraction and association. Furthermore, the uncertainty of the pose estimate due to the noise in sensor measurements is also calculated as a covariance matrix. However, this approach requires an initial estimate of the pose and is unable to perform global localization.

3.2.1 Pose Estimation

When a robot obtains a sensor measurement S of the known environment M, if a function f can evaluate the disparity between the robot's hypothesized pose $\boldsymbol{X_R} = (x_R, y_R, \phi_R)^\top$ and the observation, the robot's pose estimation becomes a minimization problem.

$$\hat{\boldsymbol{X}}_{\boldsymbol{R}} = \underset{x_R, y_R, \phi_R}{\operatorname{argmin}} f(S, M) \tag{3.1}$$

Essentially, this approach attempts to overlap the projection of the sensor measurement (such as a laser scan) with the environment, expecting the best overlap with minimum disparity to obtain the best estimate for the robot pose. The Vector Chamfer Distance (VCD) disparity measurement d_{VCD} introduced in Section 2.3.1 is proposed as a suitable scalar disparity measurement that can be used as the cost function for the minimization problem in (3.1).

Figures 3.1 and 3.2 demonstrate how the value of d_{VCD} behaves close to the true robot pose. d_{VCD} is at its minimum when the hypothesized robot pose has zero errors. That is, d_{VCD} is at its minimum when the hypothesized robot pose is the best estimate for the true robot pose. However, it can be noted that this value will only be zero when the map has no errors and the sensor measurement contains no noise (Chapter 5 presents a novel approach to represent map uncertainty).

Using d_{VCD}, the localization problem can now be described as

$$\hat{\boldsymbol{X}}_{\boldsymbol{R}} = \underset{x_R, y_R, \phi_R}{\operatorname{argmin}} d_{VCD}(X_o, DT_v) \tag{3.2}$$

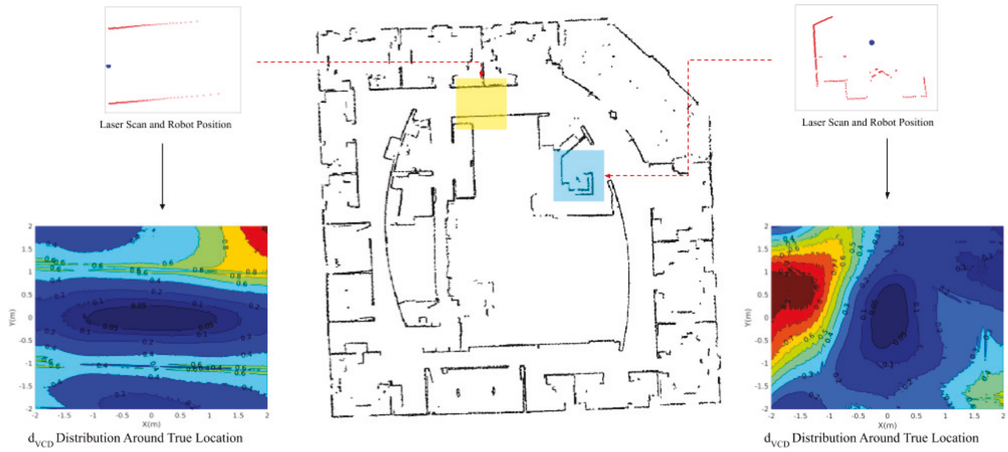


FIGURE 3.1: Behavior of d_{VCD} around the true robot position with the orientation fixed at true value

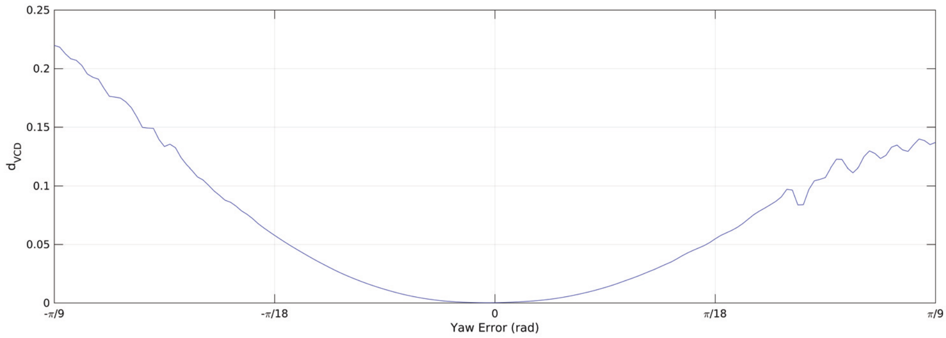


FIGURE 3.2: Behavior of d_{VCD} around the true robot orientation with the position fixed at true value

where the observation vector X_o is obtained using (2.11) on an environment described by a VDT, denoted by DT_v. This is a nonlinear, unconstrained least squares problem and as such, can be solved using many optimization algorithms. In our experiments, we have used Nelder-Mead [83], Trust-Region-Reflective [84] and Levenburg-Marquardt [85, 86] algorithms. The presence of dynamic objects in the environment or the observation of any obstacle not described in the map will add an error to the cost function. Without attempting to associate each observation with the static map, a simple gate function can be used (3.3) to prune the observation vector by removing any observations that do not belong to the map. The value of the gate α is the only tuning parameter required in the proposed localization algorithm described as Algorithm 1.

$$d_{VDT} = VDT(X_o) = \begin{bmatrix} DT_x(X_{o_i}) \\ DT_y(X_{o_i}) \end{bmatrix} \forall_i \ s.t. \ DT_x(X_{o_i}) < \alpha \ \ \& \ \ DT_y(X_{o_i}) < \alpha \tag{3.3}$$

Algorithm 1: Optimization Based Localization

Require: DT_x, DT_y, $\frac{\partial DT_x}{\partial x_{oi}}$, $\frac{\partial DT_x}{\partial y_{oi}}$, $\frac{\partial DT_y}{\partial x_{oi}}$, $\frac{\partial DT_y}{\partial y_{oi}}$, $\frac{\partial^2 DT_x}{\partial x_{oi}^2}$, $\frac{\partial^2 DT_y}{\partial y_{oi}^2}$, Initial guess for pose $\boldsymbol{X_R} = (x_R \ y_R \ \phi_R)^T$, gate parameter α

loop for each input sensor reading $z_i = \{r_i, \theta_i\}$
  $X_o = SENSORMODEL(z_i)$
  $\hat{\boldsymbol{X}}_{\boldsymbol{R}} = OPTIMIZE(X_R)$
  $X_R = UPDATEGUESS(\hat{\boldsymbol{X}}_{\boldsymbol{R}})$
end loop

function SensorModel(z)
  return X_o *observation vector*
end function

function Optimize(X_R)
  return $\hat{\boldsymbol{X}}_{\boldsymbol{R}} = (x \ y \ \phi)^T$ at $\min d_{VCD}$
end function

function UpadteGuess(X_R)
  if *odometry available* **then**
    update X_R with motion model
  end
  return X_R
end function

3.2.2 Calculating the Uncertainty of the Pose Estimate

When a localization algorithm outputs a pose estimate, autonomous systems make decisions based on that information. However, the pose estimate, in almost all practical situations will be at least a little bit erroneous. If the localization system can describe a level of confidence on the output pose estimate, intelligent decisions can be made based on it. Some contributing factors to this error can be listed as

1. Sensor noise

2. Errors in map

3. Localization algorithm failing to converge

In this work, assuming that the map is perfect, a method to calculate the uncertainty of the pose estimate due to the noise in sensor measurement, when the localization algorithm converges, is proposed. Since the estimate for the robot pose $\hat{\boldsymbol{X}}_{\boldsymbol{R}}$ in this algorithm is obtained via a non linear least squares optimization process, an explicit function mapping the sensor measurements to the pose estimate is not available. However, the error of the pose estimate due to the noise in sensor measurement can be calculated using the implicit function theorem [87] as a covariance matrix $cov(\hat{\boldsymbol{X}}_{\boldsymbol{R}})$

$$cov(\hat{\boldsymbol{X}}_{\boldsymbol{R}}) = J \cdot \Sigma_z \cdot J^\top \tag{3.4}$$

where Σ_z is the covariance matrix of the sensor measurement given as $\Sigma_z = diag(\sigma_r^2)$ and J is the Jacobian matrix of the implicit function approximated as

$$J = -H^{-1} * \begin{bmatrix} \frac{\partial^2 d_{VDT}}{\partial x_r \partial r} \\ \frac{\partial^2 d_{VDT}}{\partial y_r \partial r} \\ \frac{\partial^2 d_{VDT}}{\partial \phi_r \partial r} \end{bmatrix} \tag{3.5}$$

using the hessian matrix H defined as

$$H = \begin{bmatrix} \frac{\partial^2 d_{VDT}}{\partial x_r^2} & \frac{\partial^2 d_{VDT}}{\partial x_r \partial y_r} & \frac{\partial^2 d_{VDT}}{\partial x_r \partial \phi_r} \\ \frac{\partial^2 d_{VDT}}{\partial x_r \partial y_r} & \frac{\partial^2 d_{VDT}}{\partial y_r^2} & \frac{\partial^2 d_{VDT}}{\partial y_r \partial \phi_r} \\ \frac{\partial^2 d_{VDT}}{\partial x_r \partial \phi_r} & \frac{\partial^2 d_{VDT}}{\partial y_r \partial \phi_r} & \frac{\partial^2 d_{VDT}}{\partial \phi_r^2} \end{bmatrix} \tag{3.6}$$

The fact that the covariance matrix $cov(\hat{\boldsymbol{X}}_{\boldsymbol{R}})$ can be calculated using the implicit function theorem (as 3.4) is an advantage of VDT over UDT as the derivatives of UDT are not differentiable on the boundaries.

3.3 Extended Kalman Filter For Localization on Vector Distance Transform Based Maps

When a good initial estimate of the robot pose is available, an Extended Kalman Filter (EKF) can be used to localize mobile robots as well. An EKF is a recursive two step algorithm for estimating the mean and covariance of a state vector based on a prediction model and an observation model. It is a widely used framework to localize mobile robots when the environment is represented by features and the mobile robot is equipped with a sensor that can observe these features [2].

3.3.1 Prediction Step

In the prediction step, the EKF framework predicts the mean values of the state vector $\boldsymbol{\hat{X}_{R_{k|k-1}}}$ and the covariance matrix $P_{k|k-1}$ based on the state estimate $\boldsymbol{\hat{X}_{R_{k-1}}}$ and its covariance matrix P_{k-1} at the previous time step based on a motion model g, which is independent of the type of map used. When an odometry measurement is available in the form of linear and angular velocities $U_k = [\upsilon_k, \omega_k]$, the prediction step is formulated as

$$\begin{gathered}\boldsymbol{\hat{X}_{k|k-1}} = g(\boldsymbol{\hat{X}_{k-1}}, U_k) \\ \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \\ \hat{\phi}_{k|k-1} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k-1} + \Delta T \cdot \upsilon_k \cdot cos(\hat{\phi}_{k-1}) \\ \hat{y}_{k-1} + \Delta T \cdot \upsilon_k \cdot sin(\hat{\phi}_{k-1}) \\ \hat{\phi}_{k-1} + \Delta T \cdot \omega_k \end{bmatrix}\end{gathered} \tag{3.7}$$

where ΔT is the time between steps k and $k-1$. The pose covariance $P_{k|k-1}$ is calculated as

$$P_{k|k-1} = \nabla G_x \cdot P_{k-1} \cdot \nabla G_x^\top + \nabla G_u \cdot Q \cdot \nabla G_u^\top \tag{3.8}$$

where the Jacobians ∇G_x and ∇G_u are derived as

$$\nabla G_x = \frac{\partial g}{\partial X}\Big|_{\hat{X}_{k-1},U_k} = \begin{bmatrix} 1 & 0 & -\Delta T \cdot \upsilon_k \cdot sin(\phi_{k-1}) \\ 0 & 1 & \Delta T \cdot \upsilon_k \cdot cos(\phi_{k-1}) \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

and

$$\nabla G_u = \frac{\partial g}{\partial U}\Big|_{\hat{X}_{k-1},U_k} = \begin{bmatrix} \Delta T \cdot cos(\phi_{k-1}) & 0 \\ \Delta T \cdot sin(\phi_{k-1}) & 0 \\ 0 & \Delta T \end{bmatrix} \tag{3.10}$$

and Q is a diagonal matrix that represents the linear and angular velocity measurement noise values along the diagonal.

3.3.2 Observation Step

The basis of the observation step in an EKF is to obtain a disparity measure between the state estimated in the prediction step and the actual observation such that the state estimate can be corrected by incorporating the information gained by the observation. This is achieved by generating an expected observation based on the current state estimate and comparing it with the actual observation. However, with a VDT representation of the environment, this disparity is captured implicitly as a vector of distance values derived as (2.12) by projecting the observation at the k^{th} step z_k using the predicted pose $\hat{\boldsymbol{X}}_{\boldsymbol{R_{k|k-1}}}$. Aiming to minimize this disparity, the observation model can be formulated as

$$h(\hat{\boldsymbol{X}}_{\boldsymbol{R_{k|k-1}}}, z_k) = d_{VDT}(X_{o_k}) = \vec{0} \tag{3.11}$$

where X_{o_k} is the observation vector at the k^{th} step obtained using (2.11).

Therefore, the innovation vector is simply

$$\nu_k = h(\hat{\boldsymbol{X}}_{\boldsymbol{R_{k|k-1}}}, z_k) = d_{VDT}(X_{o_k}) \tag{3.12}$$

where the innovation covariance can be calculated as

$$S_k = \nabla H_z \cdot \Sigma_z \cdot \nabla H_z^\top + \nabla H_x \cdot P_{k|k-1} \cdot \nabla H_x^\top \tag{3.13}$$

where Σ_z is the covariance matrix of the measurement similar to that in (3.4) and ∇H_z and ∇H_x are the Jacobians of the observation model derived as

$$\nabla H_z = \frac{\partial h}{\partial z}\Big|_{\hat{X}_{k|k-1},z} = \begin{bmatrix} \frac{\partial DT_x(X_{o_{ki}})}{\partial x} \cdot cos(\theta_i + \phi_{k+1|k}) + \frac{\partial DT_x(X_{o_{ki}})}{\partial y} \cdot sin(\theta_i + \phi_{k+1|k}) \\ \frac{\partial DT_y(X_{o_{ki}})}{\partial x} \cdot cos(\theta_i + \phi_{k+1|k}) + \frac{\partial DT_y(X_{o_{ki}})}{\partial y} \cdot sin(\theta_i + \phi_{k+1|k}) \end{bmatrix} \tag{3.14}$$

and

$$\nabla H_x = \frac{\partial h}{\partial X}\Big|_{\hat{X}_{k|k-1},z} = \begin{bmatrix} \frac{\partial DT_x X_{o_{ki}})}{\partial \hat{x}_{k+1|k}} & \frac{\partial DT_x(X_{o_{ki}})}{\partial \hat{y}_{k+1|k}} & \frac{\partial DT_x(X_{o_{ki}})}{\partial \hat{\phi}_{k+1|k}} \\ \frac{\partial DT_y(X_{o_{ki}})}{\partial \hat{x}_{k+1|k}} & \frac{\partial DT_y(X_{o_{ki}})}{\partial \hat{y}_{k+1|k}} & \frac{\partial DT_y(X_{o_{ki}})}{\partial \hat{\phi}_{k+1|k}} \end{bmatrix} \tag{3.15}$$

where

$$\frac{\partial DT_x(X_{o_{ki}})}{\partial \hat{x}_{k+1|k}} = \frac{\partial DT_x(X_{o_{ki}})}{\partial x} \tag{3.16}$$

$$\frac{\partial DT_x(X_{o_{ki}})}{\partial \hat{y}_{k+1|k}} = \frac{\partial DT_x(X_{o_{ki}})}{\partial y} \tag{3.17}$$

$$\frac{\partial DT_x(X_{o_{ki}})}{\partial \hat{\phi}_{k+1|k}} = -\frac{\partial DT_x(X_{o_{ki}})}{\partial x}\Big(r_i \cdot sin(\theta_i + \phi_{k+1|k})\Big) + \frac{\partial DT_x(X_{o_{ki}})}{\partial y}\Big(r_i \cdot cos(\theta_i + \phi_{k+1|k})\Big) \tag{3.18}$$

and $\frac{\partial DT_y(X_{o_{ki}})}{\partial \hat{x}_{k+1|k}}$, $\frac{\partial DT_y(X_{o_{ki}})}{\partial \hat{y}_{k+1|k}}$ and $\frac{\partial DT_y(X_{o_{ki}})}{\partial \hat{\phi}_{k+1|k}}$ are similarly derived. These calculations are fast as the partial derivatives of the two components of the vector distance transform are pre-computed and stored in memory.

A similar approach has been taken in [88] with a UDT representation of the environment. However, as demonstrated in Section 2.2, the gradients close to the boundaries of a continuous approximation of a UDT deviate from the true values significantly. This leads to incorrect calculation of Jacobians ∇H_x and ∇H_z.

The Kalman gain K_k can now be calculated as

$$K_k = P_{k|k-1} \cdot \nabla H_x \cdot S_k^{-1} \tag{3.19}$$

Following the standard EKF framework, the pose estimate and the covariance matrix can now be updated as

$$\boldsymbol{\hat{X}_k} = \boldsymbol{\hat{X}_{k|k-1}} + K_k \cdot \nu_k \tag{3.20}$$

and

$$P_k = P_{k|k-1} - K_k \cdot S_k \cdot K_k^{\top} \tag{3.21}$$

3.4 Experimental Results

This section presents results from experiments carried out using simulations, publicly available datasets and real life experiments.

3.4.1 Dataset 1 : Simulation in 2D

This simulation dataset was generated using Player / Stage Robot Simulator for ROS [89], using the *Sample Hospital Environment*. The goal of this experiment is to evaluate the accuracy of the proposed localization strategies in a Two-Dimensional (2D) environment that contains ground truth robot pose, and compare the localization error with similar algorithms.

Property	Value
Simulated LiDAR	Hokuyo UTM-30LX
Maximum Range	30m
Minimum Range	0.1m
Field of View	270°
Angular Resolution	0.25°

Table 3.1: Properties of the simulated LiDAR in Dataset 1

Parameter	Noise value
LiDAR measurement noise	$\mathcal{N}(0, 0.02m^2)$
Linear velocity noise	$\mathcal{N}(0, 0.05m^2s^{-2})$
Angular velocity noise	$\mathcal{N}(0, 0.01rad^2s^{-2})$

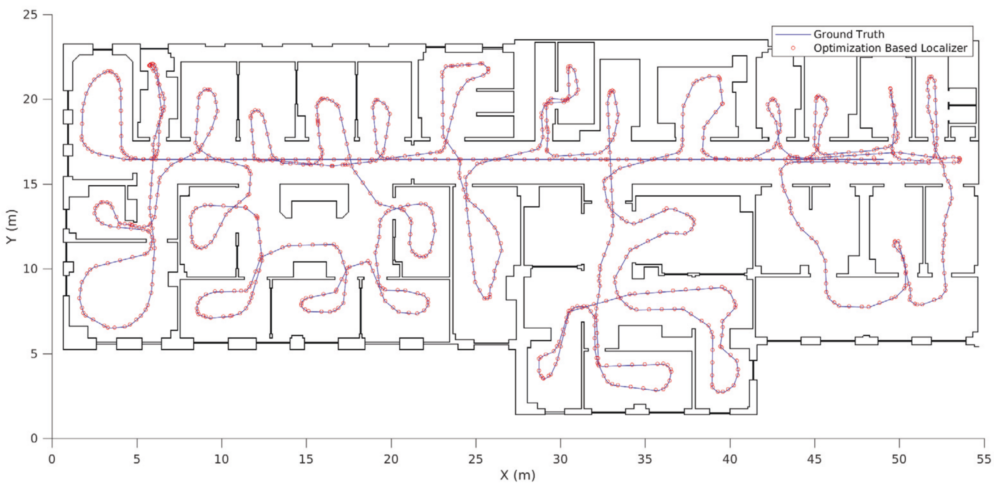
Table 3.2: Noise parameters of the simulated measurements in Dataset 1

The robot is equipped with a simulated LiDAR as per specification given in Table 3.1. Odometry data is simulated and the ground truth robot pose is also available. Noise is introduced to the simulated LiDAR and odometry measurements as described in Table 3.2, consistent with noise levels of real sensors.

3.4.1.1 Localization Accuracy and Uncertainty

The robot is localized using the optimization based localization framework presented in Section 3.2 and the EKF based localization framework presented in Section 3.3. Since the ground truth is available in this simulated dataset, it was used for comparison purposes. Figure 3.3a demonstrates the trajectory estimation from the optimization based localization algorithm compared with the ground truth. Figure 3.3b similarly demonstrates the estimated pose from the EKF based localization algorithm compared with the ground truth. It should be noted that only every 15^{th} pose estimate from the localization algorithm is plotted in each case to avoid the clutter. It can be observed from the two figures that both the optimization and EKF based localization algorithms provided pose estimates that are close to the ground truth trajectory.

Figure 3.4 shows the error of each component of the pose estimate with respect to the ground truth for the optimization based algorithm. Similarly, Figure 3.5 shows the error of each component of the pose estimate with respect to the ground truth for the EKF based algorithm. Since both localization algorithms provide an uncertainty estimate in



(a)

(b)

FIGURE 3.3: Trajectory of the robot in Dataset 1 (a) Proposed optimization based localizer (b) Proposed EKF based localizer compared with ground truth

the form of a covariance matrix, the 2σ boundaries are also plotted. It can be observed that the optimization based algorithm performs marginally better than the EKF based algorithm. This is to be expected as the EKF framework can be thought of as a single step optimizer.

The localization accuracy of the proposed optimization based and EKF based algorithms is also calculated in terms of mean absolute error and mean squared error and presented in Table 3.3. For comparison, corresponding metrics for an optimization based localization algorithm for UDT, an EKF based localization algorithm for UDT and AMCL are obtained

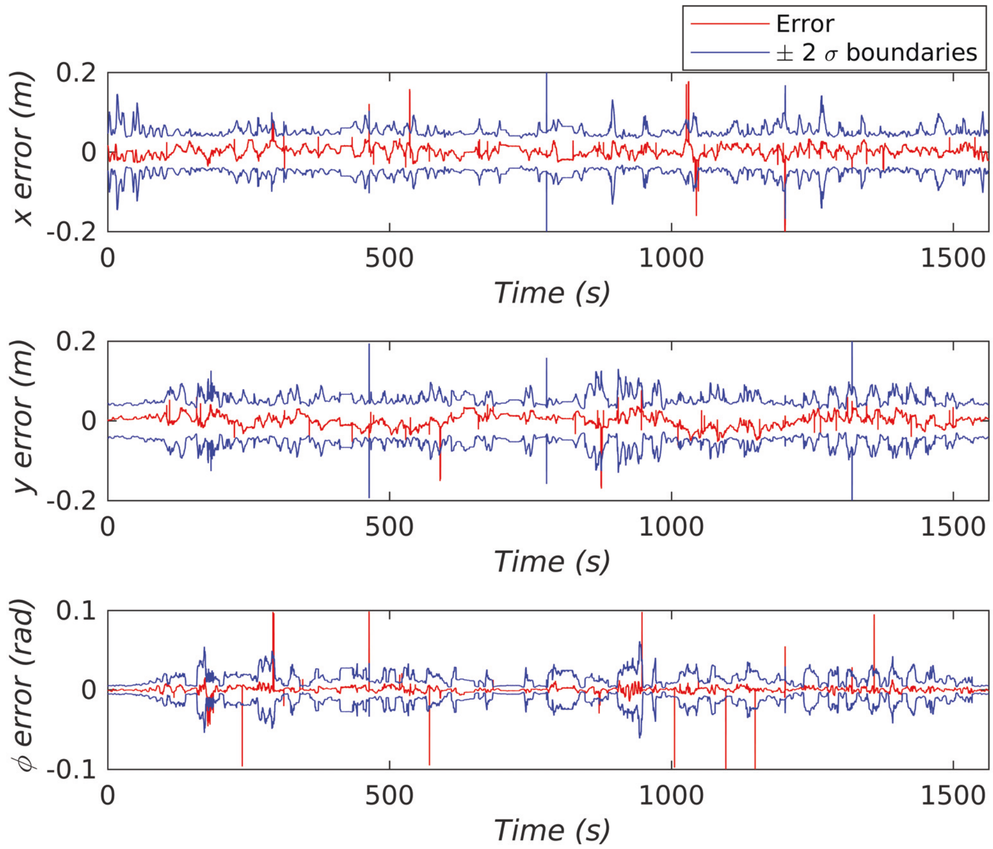


FIGURE 3.4: Error of the robot pose estimate for Dataset 1 with the proposed optimization based localizer compared with ground truth

Algorithm	Mean Absolute Error		Mean Squared Error		P_k
	Position(m)	*Orientation(rad)*	*Position*$(m^2 \times 10^{-3})$	*Orientation*$(rad^2 \times 10^{-3})$	
Optimization (proposed)	0.0181 ± 0.0126	0.0024 ± 0.0037	0.4847 ± 1.700	0.0192 ± 0.2636	✓
EKF (proposed)	0.0235 ± 0.0223	0.0161 ± 0.0280	1.1000 ± 2.6000	1.0000 ± 2.5000	✓
Optimization (C-LOG)	0.0099 ± 0.0141	0.0042 ± 0.0167	0.299 ± 2.167	0.2975 ± 1.7000	✗
EKF (UDT)	0.0227 ± 0.0251	0.0157 ± 0.0211	1.146 ± 4.492	1.4000 ± 3.7000	✓
Particle Filter (AMCL)	0.0296 ± 0.0243	0.0252 ± 0.0431	1.464 ± 2.078	2.5000 ± 7.1000	✓

TABLE 3.3: Error values for Dataset 1 compared with C-LOG, UDT EKF and AMCL. The P_k column denotes whether the corresponding algorithm estimates uncertainty.

from Dantanarayana [13] and presented. The proposed optimization based localization algorithm has an accuracy similar to C-LOG. The advantage of the proposed algorithm over C-LOG is that C-LOG cannot calculate the uncertainty for its pose estimate whereas the proposed optimization based localization algorithm can. The accuracy of the proposed EKF based localization algorithm is also similar to the accuracy of the UDT EKF based localizer.

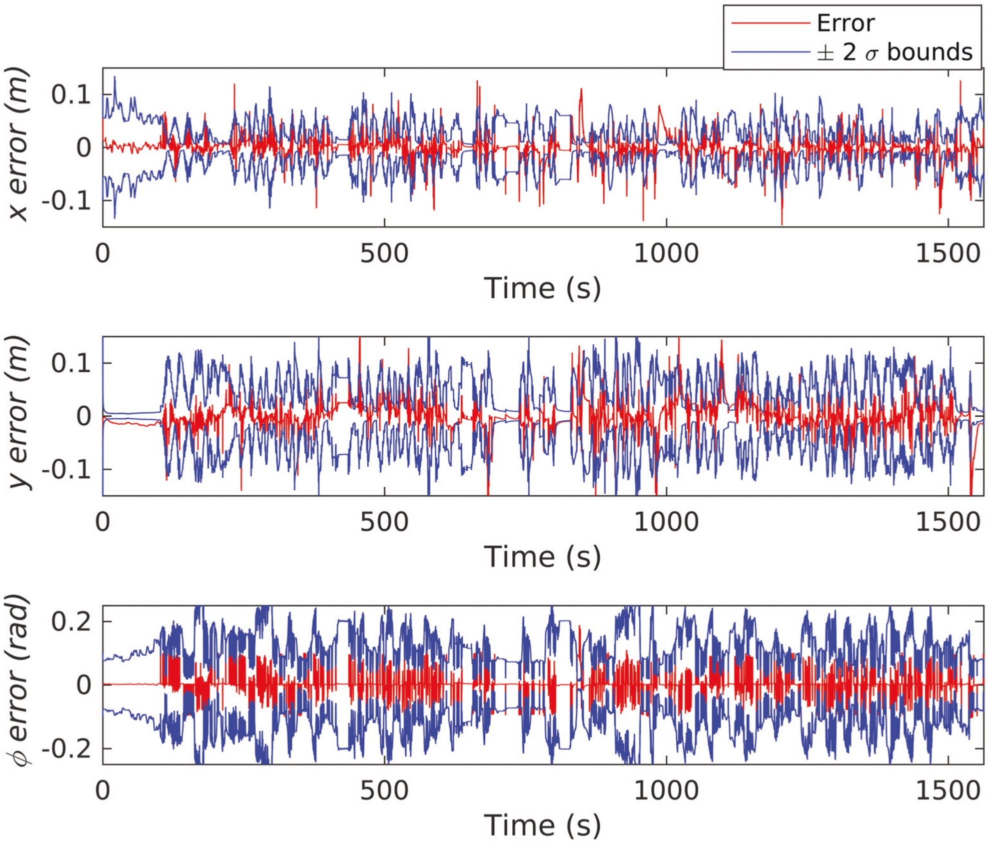


FIGURE 3.5: Error of the robot pose estimate for Dataset 1 with the proposed EKF based localizer compared with ground truth

3.4.2 Dataset 2 : Intel Dataset

This dataset is the publicly available Intel Research Lab (Seattle) SLAM dataset, which consists of LiDAR and odometry measurements. The goal of this experiment is to demonstrate the localization capabilities of the two proposed approaches in a real, 2D, environment represented by a noisy map using noisy sensor measurements.

The robot's trajectory consists of three loops. The map of the environment is built using GMapping with the measurements collected in the third loop. The robot is localized using the proposed algorithms with the measurements collected in the first and second loops. Since ground truth robot poses are not available for this dataset, poses obtained from GMapping, separate from the map building process, using the measurements collected in the first and second loops are used for visual comparison.

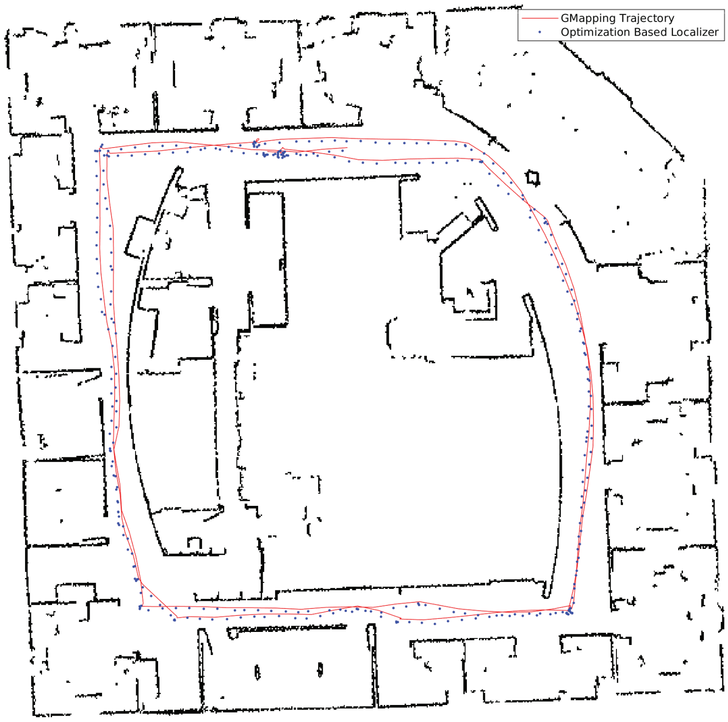


FIGURE 3.6: Trajectory of the robot in Dataset 2 estimated from the optimization based localizer compared with GMapping

Figure 3.6 demonstrates the pose estimates (sparsely for clarity) from the optimization based localization algorithm compared with poses obtained using GMapping. Figure 3.7 demonstrates the complete trajectory estimation from the optimization based localization algorithm. To show that the pose estimates are reasonably accurate, the projection of LiDAR measurements using the estimated poses is demonstrated in Figure 3.8. Figures 3.9, 3.10 and 3.11 demonstrate the corresponding figures for the EKF based localization algorithm.

It can be visually observed that both algorithms estimate the trajectory of the robot well. Since the ground truth values for robot poses are not available, a quantitative measure for pose estimate accuracy cannot be derived. However, the reprojection of the LiDAR measurements from the estimated poses (Figures 3.8 and 3.11) demonstrate that both approaches estimate the robot pose at a high degree of accuracy.

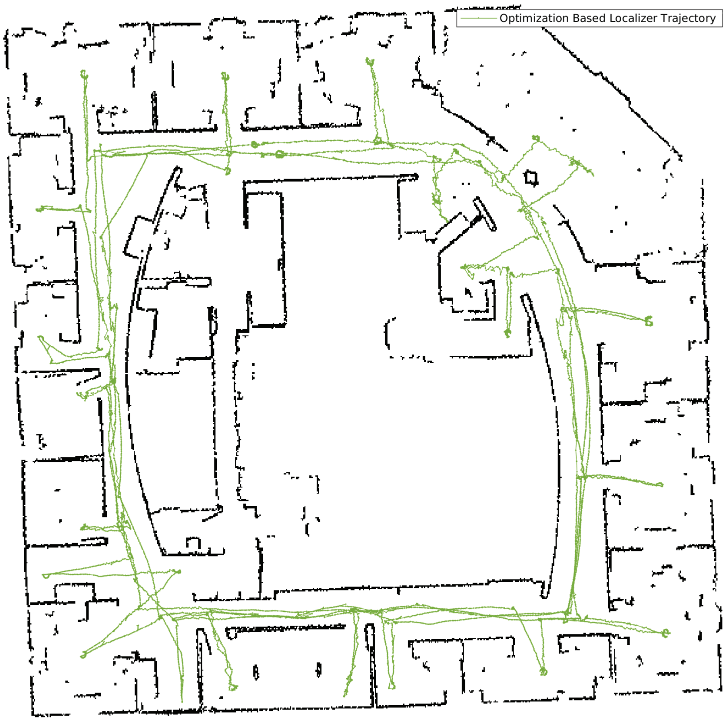


FIGURE 3.7: Complete trajectory of the robot in Dataset 2 estimated from the optimization based localizer

3.4.3 Dataset 3 : Localization of a UAV

The goal of this experiment is to demonstrate the capability of the proposed optimization based localization algorithm on an unmanned aerial vehicle (UAV) platform using a monocular camera for observations in an outdoor environment.

This dataset was collected by Team VICTOR from Virginia Tech during their preparations for Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017, an international robotics competition involving unmanned ground and aerial vehicles, in Kentland Farm, Blacksburg, VA, USA. In tandem with the competition specification, a simulated roadway in the shape of a figure of 8 was marked on the runaway.

The hexarotor UAV setup shown in Fig. 3.12 was custom built for the purpose of the competition. It is equipped with a perspective camera, a fisheye camera and a laser range finder which are rigidly fixed to the UAV body pointing downwards, an RTK-GPS module

FIGURE 3.8: LiDAR measurements of Dataset 2 projected from the poses estimated using the optimization based localizer

and a Pixhawk® flight controller with an IMU. In this environment, the RTK-GPS module provides highly accurate location estimates, which can be used as the ground truth of the UAV pose.

The state estimator in the Pixhawk® flight controller uses its internal IMU to estimate its roll and pitch angles. These estimates are adequately accurate for high frequency UAV control and are available for high level navigation tasks such as path planning and way point following. Furthermore, the high precision laser range finder fixed to the UAV body pointing downwards can then be used to estimate the operating altitude of the UAV. Given these conditions, the localization problem can be reduced from estimating $\hat{\boldsymbol{X}}_{\boldsymbol{R}} = (x_R, y_R, z_R, \phi_R, \theta_R, \psi_R)^\top$ to estimating $\hat{\boldsymbol{X}}_{\boldsymbol{R}} = (x_R, y_R, \phi_R)^\top$.

For localization, the perspective camera is used. Images are captured in synchronization with the current UAV pose estimate $\hat{\boldsymbol{X}}_{\boldsymbol{R}} = (x_R, y_R, \phi_R)^\top$, the pitch and yaw angles from the IMU and the height information from the laser range finder. The images are corrected for lens distortion and the edges are extracted from it. The set of edge pixels (λ_i, μ_i) are then transformed from the image plane to the ground plane using the assumption that

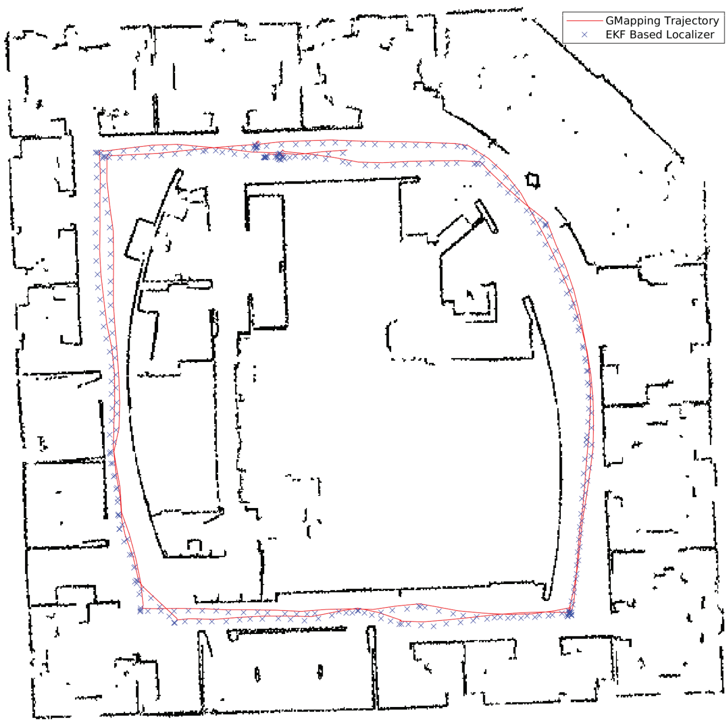


FIGURE 3.9: Trajectory of the robot in Dataset 2 estimated from the EKF based localizer compared with GMapping

the ground is a locally flat terrain using (3.22). R is the rotation matrix explaining the attitude of the UAV and f is the focal length of the camera obtained via calibration.

$$\mathbf{x}_{o_i} = \begin{Bmatrix} x_{o_i} \\ y_{o_i} \end{Bmatrix} = \begin{Bmatrix} x_r + z\frac{\lambda_i R_{1,1} + \mu_i R_{1,2} - f R_{1,3}}{\lambda_i R_{3,1} - \mu_i R_{3,2} + f R_{3,3}} \\ y_r + z\frac{\lambda_i R_{2,1} + \mu_i R_{2,2} - f R_{2,3}}{\lambda_i R_{3,1} - \mu_i R_{3,2} + f R_{3,3}} \end{Bmatrix} \tag{3.22}$$

Figure 3.13 shows a comparison between the poses produced by the RTK-GPS and the proposed localizer. It can be seen that the localizer fails when the UAV is not on top of the runway. This is to be expected because the system fails to detect and extract features that can be used for localization. It can also be observed that the localization algorithm can quickly recover once a sufficient amount of features is detected. The trajectory of the UAV estimated by the optimization based localizer (using VCD disparity measurement) when it detects a sufficient number of features is drawn for visual comparison with the UAV trajectory obtained by RTK-GPS measurements. Figure 3.14 shows shows the x, y

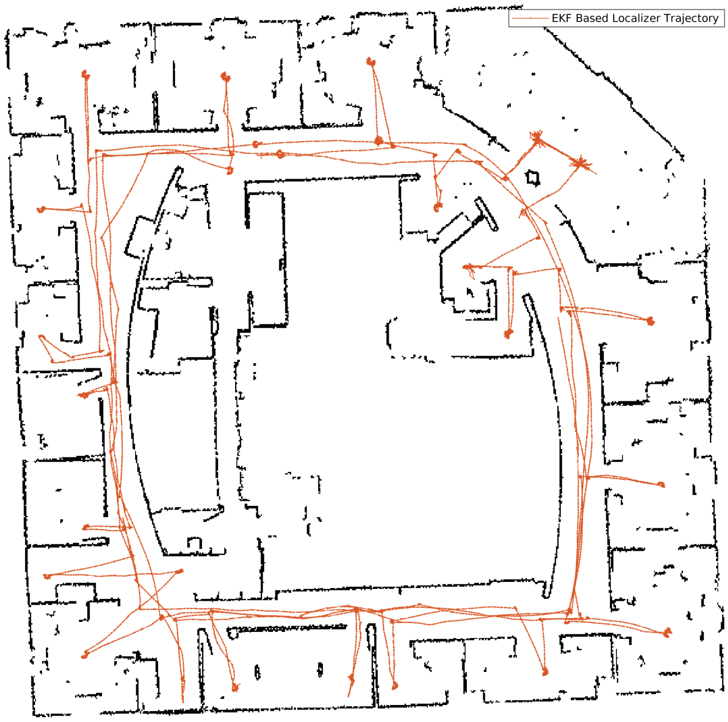


FIGURE 3.10: Complete trajectory of the robot in Dataset 2 estimated from the EKF based localizer

and *yaw* errors and the corresponding uncertainties at each step when the localizer is fed with a sufficient number of features. It can be observed that the pose error is sufficiently small for operations in a large, outdoor environment.

3.4.4 Dataset 4 : CAS Robotics Research Lab

This dataset is collected inside the CAS Robotics Research Lab 1 (Figure 3.15) in University of Technology Sydney. The goal of this experiment is to demonstrate that the proposed optimization based localization algorithm can be extended, in principle, to estimate the robot pose in a cluttered, indoor Three-Dimensional (3D) environment represented by a VDT.

A LIDAR-Lite V3 laser range measurement sensor (hereinafter referred to as the point laser) mounted on a pan-tilt unit (Fig. 3.16) is used to localize a Turtlebot robot inside the

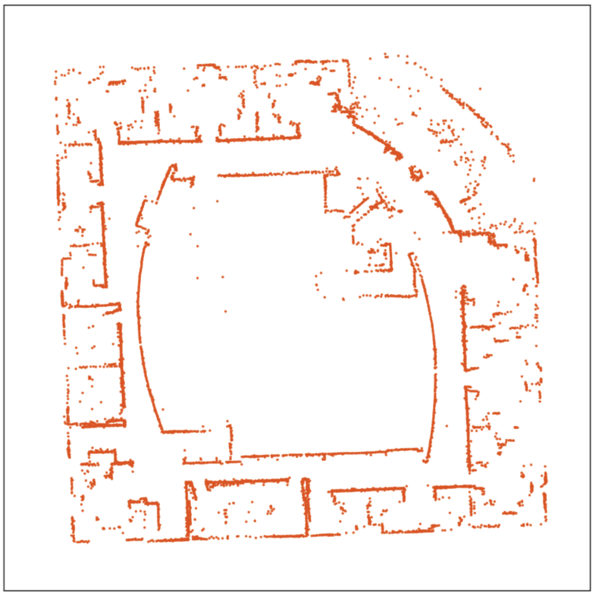


Figure 3.11: LiDAR measurements of Dataset 2 projected from the poses estimated using the EKF based localizer

Figure 3.12: Bogey5 : One of the UAVs custom built by Team VICTOR for MBZIRC 2017

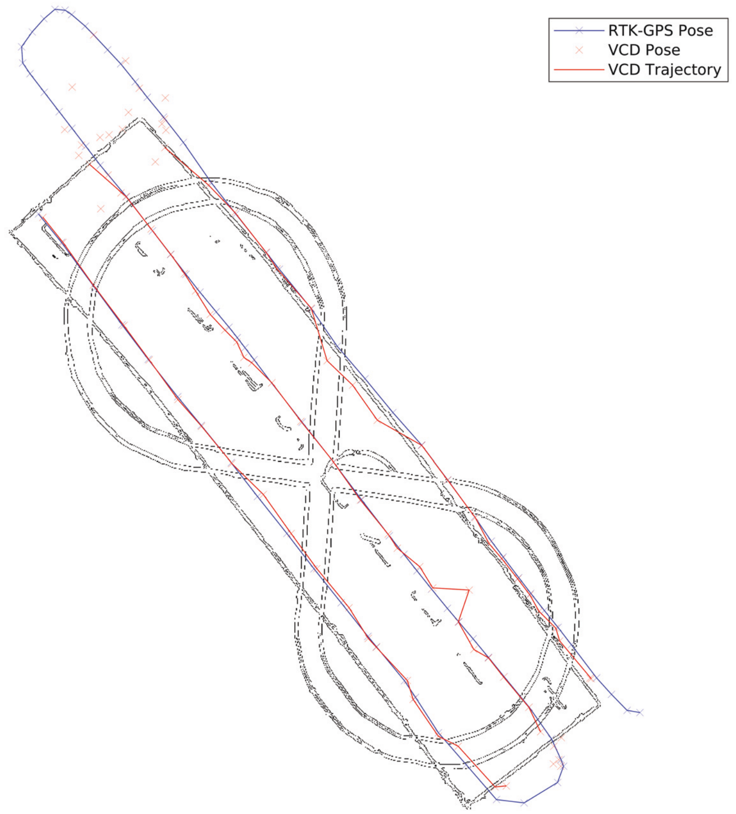


FIGURE 3.13: Localization of a UAV in Kentland Farms, VA, USA

CAS Robotics Research Lab 1 in UTS. Observation point clouds are generated while the Turtlebot is stationary by panning and tilting the point laser and transforming the range measurements to the robot frame using pan and tilt angles and calibration parameters defining the sensor geometry.

3.4.4.1 Building a map of the environment

To build a map of the environment, a Turtlebot is equipped with a Hokuyo URG-04LX-UG01 LiDAR mounted on the underside of the top plate of the robot (hereinafter referred to as the horizontal laser) that has a 5m range, and a 2D Hokuyo UTM-30LX-EW LiDAR (hereinafter referred to as the rotating laser) mounted on a rotating arm. First, the horizontal laser is used to create a 2D map of the environment using Cartographer [69].

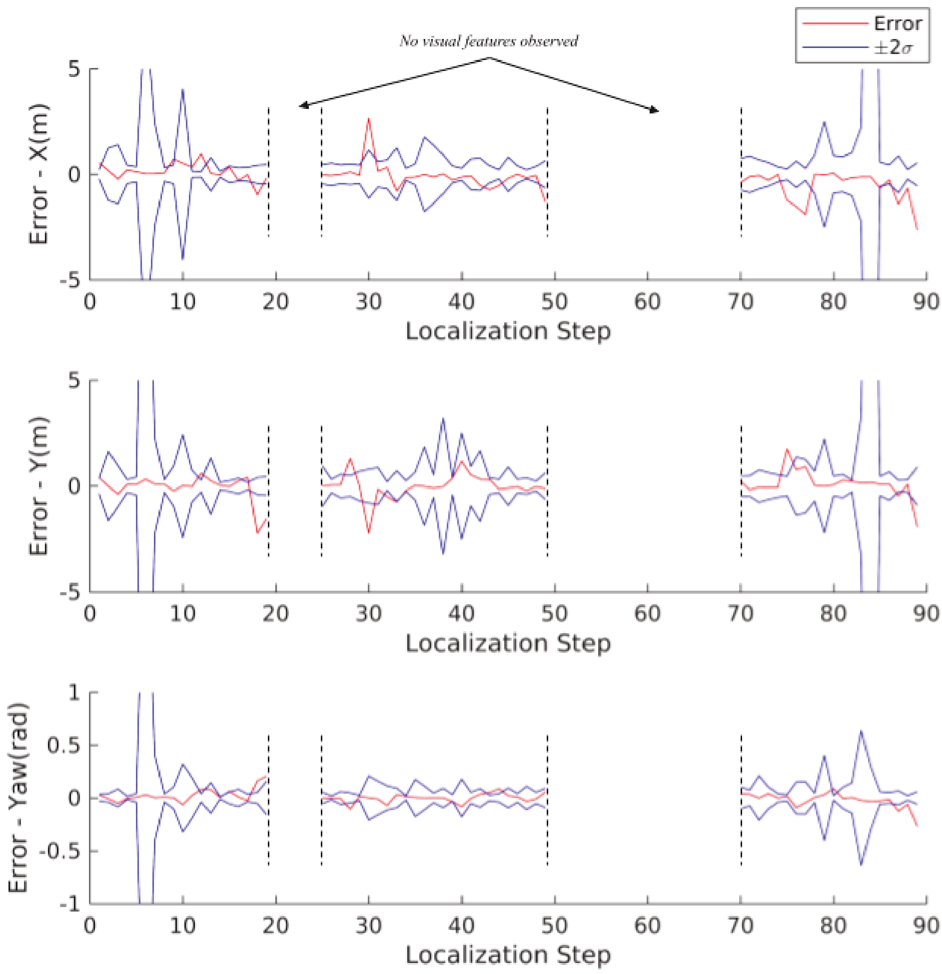


FIGURE 3.14: UAV Localization error compared to RTK-GPS

The rotating laser is then used to generate point clouds of the environment while Turtlebot is stationary. The robot poses associated with each point cloud is obtained using the Cartographer in pure localization mode with the previously generated 2D map. A complete point cloud of the environment is generated by stitching the point clouds using ICP refined pose estimations as seen in Figure 3.17. The resulting point cloud is sampled at 0.05m resolution and approximated by a cubic b-spline to obtain a continuous representation of its VDT.

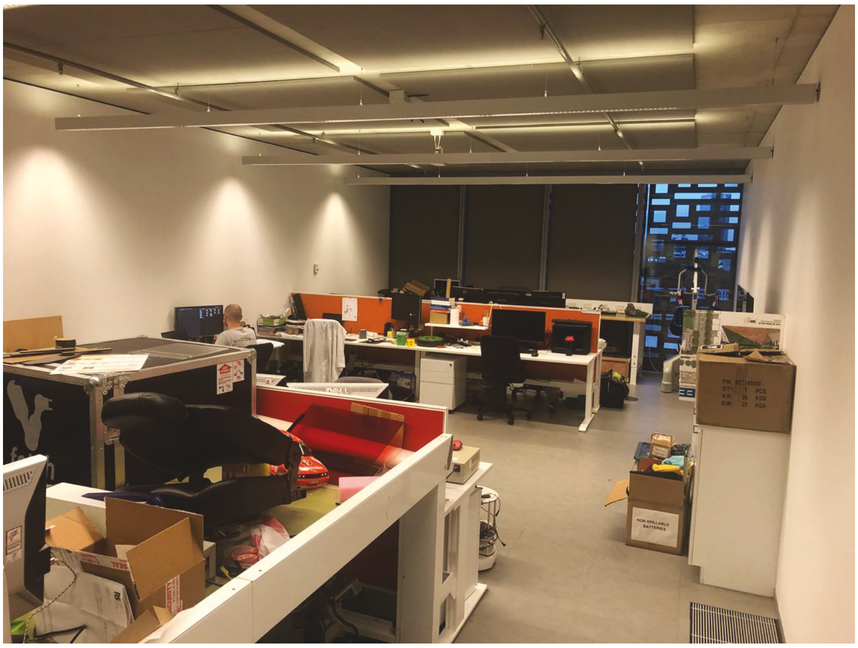


FIGURE 3.15: CAS Robotics Research Lab 1 in University of Technology Sydney

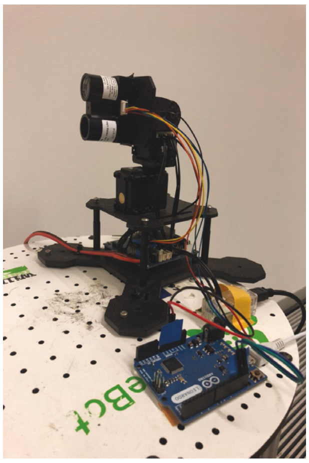


FIGURE 3.16: Point laser mounted on the pan-tilt unit

3.4.4.2 Estimating the pose of a Turtlebot using a LIDAR-Lite V3 laser range measurement sensor mounted on a pan-tilt unit

To demonstrate the capability of the proposed optimization based approach to estimate a 6DOF pose in a 3D environment represented by a VDT, observations are collected from the point laser at 3 different locations while the robot was stationary. The sensor system available was not suitable for acquiring measurements while the robot was in motion. The Turtlebot is driven to a location and then rolled and pitched to simulate uneven floor. The

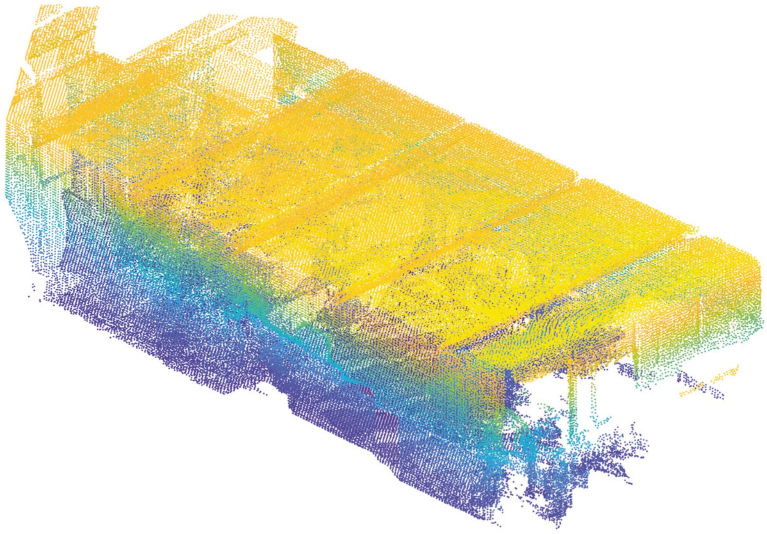


FIGURE 3.17: Stitched point cloud representation of the CAS Research Lab 1

TABLE 3.4: Difference between the Initial Guess and the Ground Truth in Each Dimension for the 4 scenarios

scenario	**X (m)**	**Y (m)**	**Z (m)**	**Roll (rad)**	**Pitch (rad)**	**Yaw (rad)**
A	+0.05	+0.05	+0.05	+0.025	+0.025	+0.025
B	+0.1	+0.1	+0.1	+0.05	+0.05	+0.05
C	+0.2	+0.2	+0.2	+0.1	+0.1	+0.1
D	+0.4	+0.4	+0.4	+0.2	+0.2	+0.2

2D robot pose obtained from Cartographer and the roll and pitch angles measured from an IMU are taken as the ground truth at each location. 5700 observations are collected at each location. The observed point cloud is downsampled by factors 2, 5, 10 and 100. The robot is localized at each point using the original observation and the downsampled ones. Four scenarios with offset between the initial guess and the true pose set to the values given in Table 3.4 are attempted. The localization root mean squared error is given in Figure 3.18. It can be observed that the localization error increases with each scenario. This is to be expected as the initial guess aggressively deviates from the true pose with each scenario. However, it can also be observed that the observation can be downsampled by a factor of 100 and still be close to the accuracy obtained using the dense observation.

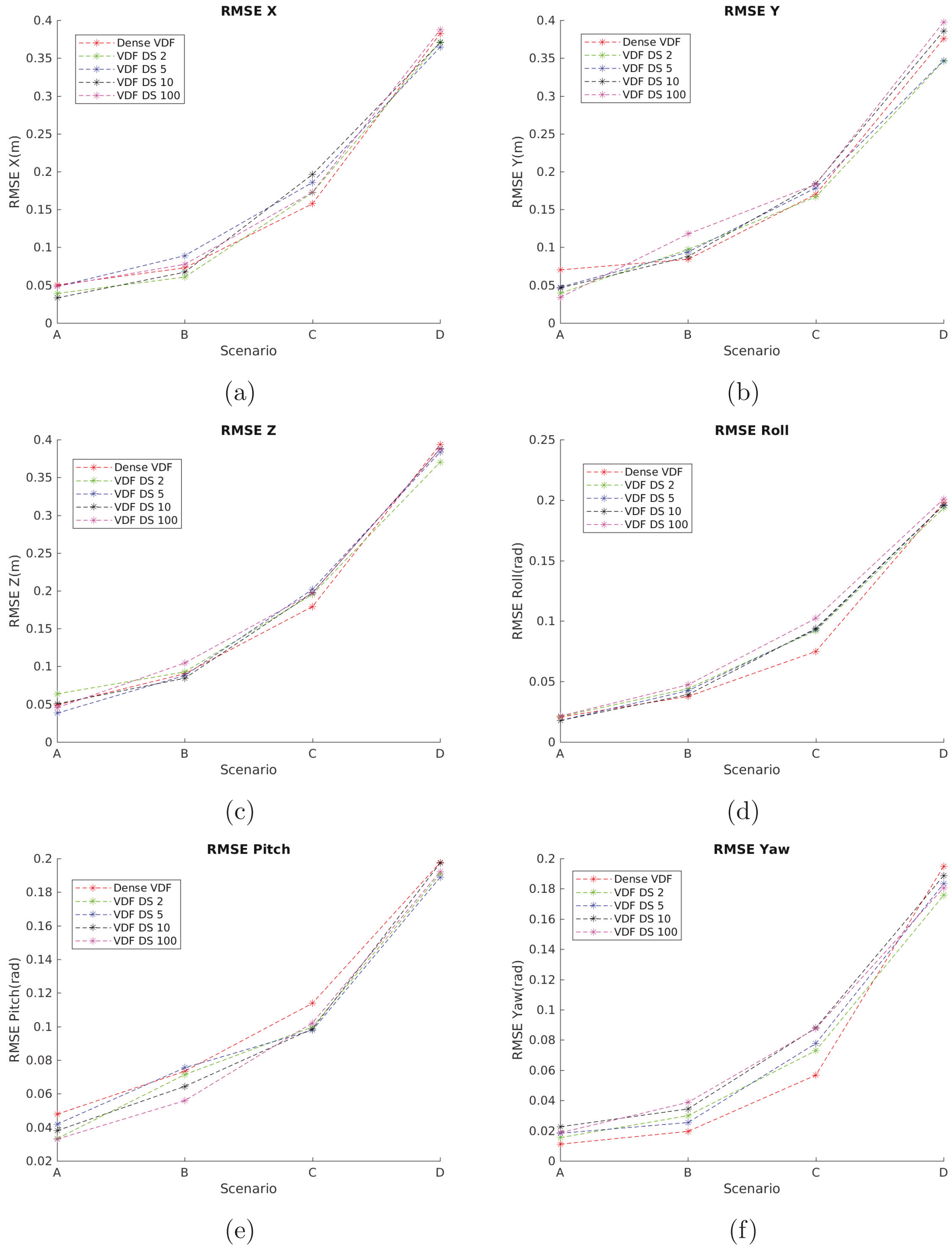


FIGURE 3.18: RMSE localization error for each initial guess scenario

3.4.5 Dataset 5 : Outdoor Localization of a Personal Mobility Scooter

The goal of this experiment is to demonstrate the capability of the proposed EKF based localization approach in an outdoor environment. The outdoor environment is represented by the VDT of its 2D ground surface structure, and monocular cameras are used for observations.

The EKF based proposed approach is integrated to a vision based localization framework of a Personal Mobility Device (PMD) shown in Figure 3.19. The initial framework, developed by Mr. Maleen Jayasuriya who is a PhD student at the Centre for Autonomous Systems (CAS), consists of a front end that utilizes deep learning network YOLO [90] to detect landmarks in the environment and use the bearings towards them to localize in an EKF back end as presented in [91]. The contribution of this thesis is the augmentation of the proposed EKF based localization approach with the ground surface structure observations based on VDT maps. The complete framework is shown in Figure 3.20.

A rear mounted tilted RGB-D camera (Figure 3.19) captures images of the ground surface, which is assumed to be locally planar. The RGB images are fed to a pre-trained HED [92] neural network to extract boundaries of the ground surface in the form of binary images (see figures 3.21a and 3.21b), and the depth information is used to project the extracted boundaries in the robot frame to obtain a vector of boundary points $Z_t = [(x^1_{z,t}, y^1_{z,t}), (x^2_{z,t}, y^2_{z,t}), ..., (x^n_{z,t}, y^n_{z,t})]$. It is then pruned to filter out any points that are not within a small distance from the ground level. A vector of boundary points in the robot frame is generated as X_z where the i^th element X_{zi} is obtained as

$$\begin{aligned} r_i &= \sqrt{(x^i_z)^2 + (y^i_z)^2} \\ \alpha_i &= arctan\Big(\frac{y^i_z}{x^i_z}\Big) \end{aligned} \tag{3.23}$$

$$X_{zi} = \begin{Bmatrix} x_{zi} \\ y_{zi} \end{Bmatrix} = \begin{Bmatrix} \hat{x}_{t|t-1} + r_i \sin(\alpha_i - \phi_{t|t-1}) \\ \hat{y}_{t|t-1} + r_i \cos(\alpha_i - \phi_{t|t-1}) \end{Bmatrix} \tag{3.24}$$

The observations are modelled similar to laser range measurements to simplify the characterisation of the observation noise. While it is theoretically possible to characterise the observation noise based on the actually observed parameters (pixels belonging to the boundaries and the corresponding depth information), characterising the measurement noise based on r (defined in (3.23)), is much simpler and still accurate. X_z is used in the observation equation as explained in Section 3.3.

Prior to conducting localization experiments, a mapping mission was carried out to construct a map by stitching ground surface information (Figure 3.21c), based on known vehicle poses obtained using RTK-GPS and Real-Time Appearance-Based Mapping (RTAB) RGBD and LIDAR Graph-Based SLAM framework [44].

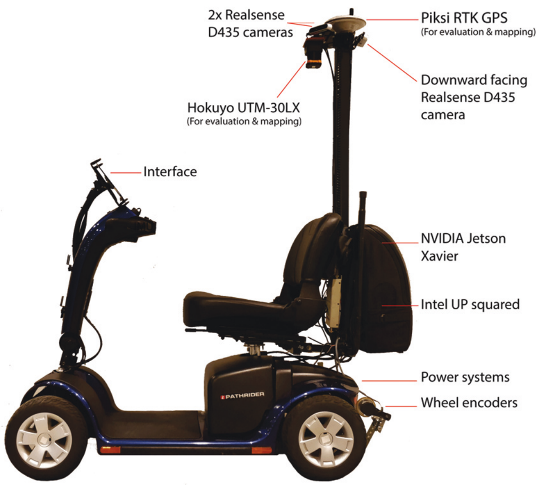


FIGURE 3.19: Hardware overview of retrofitted mobility scooter

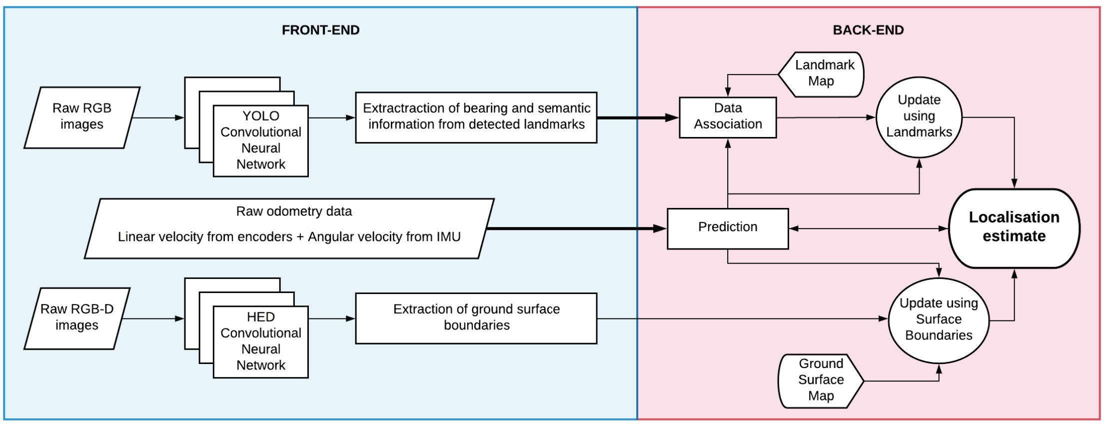
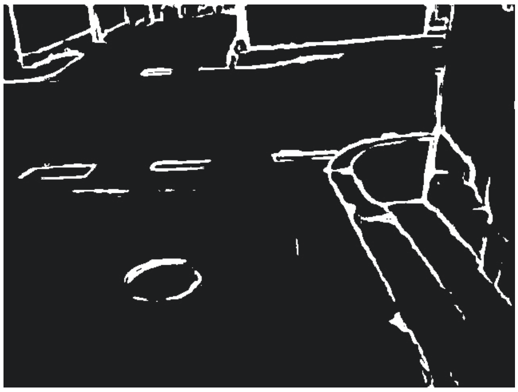
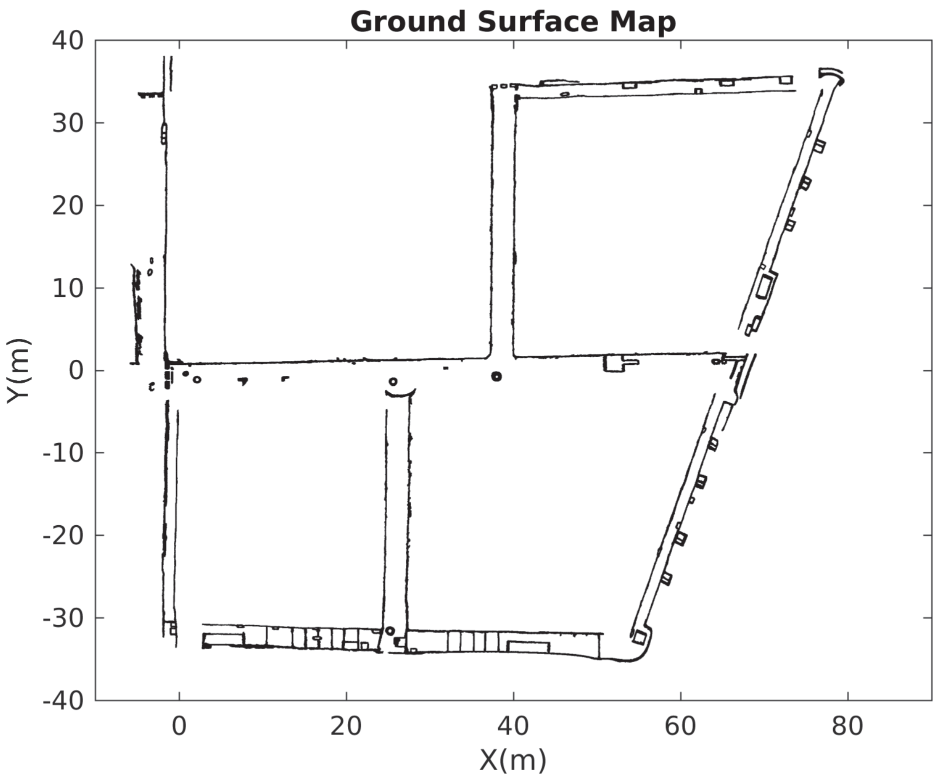


FIGURE 3.20: Localization framework for the PMD

(a)



(b)



(c)

FIGURE 3.21: (a) Ground Surface Image (b) the corresponding HED Output Image and (c) a stitched ground surface map

FIGURE 3.22: Localization of a PMD at Wentworth Park, Sydney

In order to characterise and validate the overall performance of the proposed framework, a location with reliable ground truth data was required. For this purpose a ~2000m^2 area of Wentworth Park, Sydney, Australia (Figure 3.22) was selected due to its clear skyline and the availability of high quality, centimetre-level accurate fixed RTK-GPS data. This continuous ground truth data is used to characterise the noise parameters of all sensors and to tune the overall EKF framework. The localization accuracy compared to RTK-GPS taken as ground truth is presented in Figure 3.23. It can be observed that the position error is sufficiently small for outdoor maneuvers and that the error is bounded by the uncertainty estimates.

In order to demonstrate the capabilities of the proposed framework in a less controlled and more real world urban setting, a second set of experiments were carried in a ~8000m^2 area in Glebe, Sydney, Australia. The mobility scooter was driven along the pavements amidst pedestrians and uneven terrain, covering a distance of ~550m. Unlike the Wentworth park

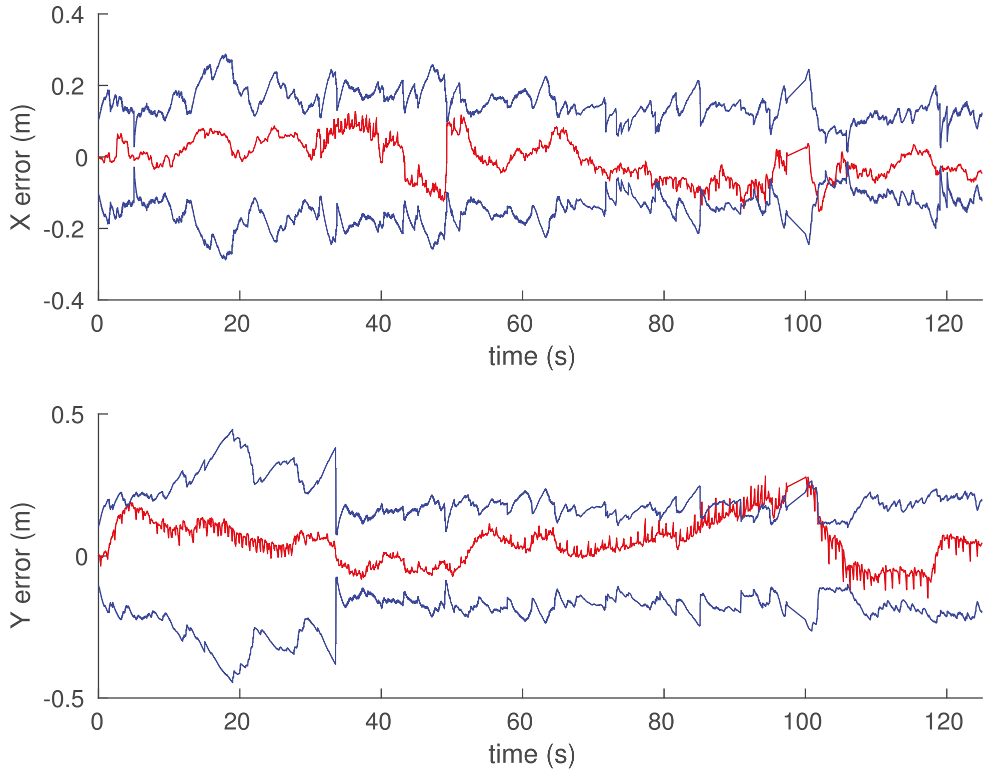


FIGURE 3.23: Localization error at Wentworth Park compared to RTK-GPS

environment, fixed RTK data was unavailable while in motion due to building and tree cover. The trajectory estimate of the proposed EKF is indicated in Figure 3.24 compared with GPS (where available) and RTAB pose estimate. The RTAB pose estimate is seen to contain large errors. This is due to the change in the illumination level in the environment between the mapping exercise and the localization exercise. However, the proposed EKF based localization framework demonstrates that it is able to estimate a trajectory that is qualitatively the most consistent with the actual trajectory of the PMD.

3.5 Discussion and Conclusions

This chapter presents two strategies that can be used to localize mobile robots when the environment is represented using a VDT, based on the sensor model presented in Chapter 2.

FIGURE 3.24: Localization of a PMD at Glebe, Sydney

The first strategy, presented in Section 3.2, is based on a non-linear least squares optimization that attempts to minimize the disparity between the actual sensor measurement and the expected sensor measurement. It is also capable of estimating the uncertainty of the pose estimate due to the noise in the sensor measurement. Experimental results are provided in both indoor and outdoor environments using range-bearing sensors as well as monocular cameras.

The second strategy is based on an EKF framework, presented in Section 3.3. In contrast to the traditional EKF localization framework used when the environment is represented using features, no explicit feature extraction from the sensor measurement is required. Since the disparity between the actual observation and the expected observation is implicitly captured by the VDT based sensor model, innovation vector can be calculated without resorting to techniques such as ray tracing. Experimental results are provided in both indoor and outdoor environments using range-bearing sensors and monocular cameras.

The VDT is capable of representing 3D environments as well. The localization of robots in 3D environments using the optimization framework has been investigated, and it has been demonstrated that localization of a robot in a 3D environment represented by a VDT using non-linear least squares optimization is possible, not only with dense point cloud observations but also from downsampled, sparse observations. However, in that work, robots are stationary when they observe the 3D environment using a One-Dimensional (1D) range sensor. Efficient localization of a mobile robot in a 3D environment represented by a VDT using either a non-linear least squares optimization framework or an EKF framework when the robot observe the environment during motion using a sensor such as a 3D LiDAR has to be further studied and implemented.

In the next chapter, an information gain based active localization strategy that makes use of the VDT environment representation introduced in Chapter 2 and the EKF framework introduced in Chapter 3 is presented.

Chapter 4

Information Gain Based Active Localization

4.1 Introduction

Robot localization approaches can be divided into two classes as passive localization and active localization. Passive localization approaches make use of information obtained using sensors to localize the robot without making an attempt to control the position of these sensors or the motion of robot. The active localization approaches exploit the capability to control the sensor position and/or the robot motion to efficiently localize the robot. Simply put, active localization attempts to answer the questions "where to look?" (active sensing) and "where to move?" (active navigation) so as to best localize the robot [93, 94]. The active maneuvering of a sensor or the robot is termed as an action.

Active localization methods can again be divided mainly into two classes as information-theoretic methods and learning-based methods. Information-theoretic methods attempt to determine the next action that would reduce the pose uncertainty and produce a better localization result by optimizing a value that quantifies the effect of that action. Entropy as defined by Shannon [95] is widely used as this value. Active Markov Localization (AML) [93, 94] is among the first published LiDAR based active localization strategies where the next navigation goal is actively selected. The Markov localization algorithm maintains a probability distribution that describes the robot's belief of being at any point in space. Next navigation goal is selected by minimizing the predicted future entropy of this belief. The same approach has been used with vision sensors as well [96, 97]. Another strategy is to use entropy maps [98]. The learning-based methods make use of reinforcement learning to learn policy models to determine the next action [99, 100]. Alternatively, simpler heuristics have also been used to determine actions such as identifying and moving closer to new features for active localization [101]. These strategies are capable of global localization. In other words, these strategies do not require an initial guess of the pose to converge to the true robot pose.

On feature maps, using active sensing to continuously observe the same feature to improve the localization result in an Extended Kalman Filter (EKF) framework has also been attempted with a sonar [102] and with cameras [103].

Active sensing and active navigation are also used for efficient exploration of the environment during Simultaneous Localization and Mapping (SLAM). In an active SLAM system, appropriate actions are decided not only to reduce the uncertainties of the pose estimate and the map estimate but also to optimize coverage [104]. Cost functions based on the trace of the covariance matrix [105], the D-optimality criterion [106], information gain [107] and entropy [108] have been used to search for optimal action during SLAM.

Active navigation for active localization hinders a robot's ability to efficiently perform high level navigation functions. For an example, a Personal Mobility Device (PMD) similar to that discussed in Section 3.4.5 in Chapter 3 cannot make use of active navigation as it needs to move towards the driver's destination as efficiently as possible. On the other hand, active sensing can be used with a sensor dedicated for localization to obtain accurate pose estimates efficiently.

The EKF based passive localization framework presented in Section 3.3 of Chapter 3 can be used with a dense Two-Dimensional (2D) LiDAR with a wide field of view and a Vector Distance Transform (VDT) representation of the environment. However, if the robot is equipped with a sparse range-bearing sensor such as a solid state LiDAR with a narrow field of view (a sensor with a lower information rate), passive localization is not adequate to obtain a sufficient amount of information to accurately localize the robot. In such cases, active localization becomes a necessity.

This chapter investigates whether a sensor with a significantly lower information rate than a conventional LiDAR can be used for accurate localization on an environment represented by a VDT with the EKF framework proposed in Section 3.3 in Chapter 3 using active sensing. A strategy is proposed for the localization of a robot that follows a predefined trajectory using a single range-bearing measurement at a time, by actively controlling the sensor to observe an area of the map that produces information that can improve the quality of the robot pose estimate. The main motivation for this work comes from the emergence of solid state LiDARs. Although solid state LiDARs produce sparse range measurements, they are attractive due to the fact that they have no mechanical components and therefore are of low cost. As a suitable solid state LiDARs was not available, sparse sensing is simulated in two ways. During the simulations a single point laser range finder that can be actively controlled and pointed at a desired bearing parallel to the map plane is used while during the real world experiments, a single measurement from a desired angle using a conventional LiDAR range sensor is used. A cost function based on information gain is used to control the sensor. As in the case of the localization strategies proposed in Chapter 3, the proposed active localization strategy, described in Section 4.2, also requires an initial guess. Section 4.3 provides experimental validation of the proposed system. Section 4.4 ends the chapter with a discussion.

4.2 Methodology

This section presents a strategy for active localization of an autonomous mobile robot in a 2D environment represented by a VDT when the robot is equipped with a range measuring sensor that can be rotated parallel to the map plane, using an EKF framework. The flow diagram of the proposed active localization framework is shown in Figure 4.1. The EKF framework for localization and the active observation of the environment are discussed in the following subsections.

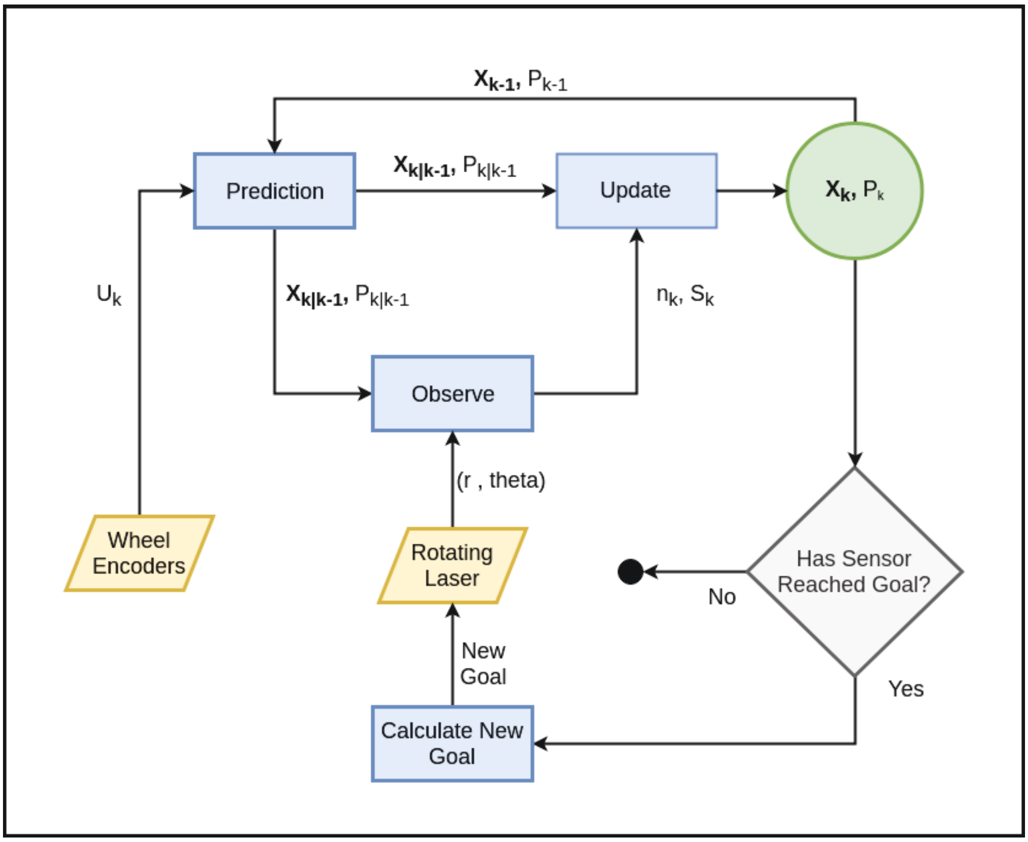


FIGURE 4.1: Active localization framework

4.2.1 The EKF Framework

The EKF framework used in this localization approach follows the EKF presented in Section 3.3 of Chapter 3. A motion model based on angular and linear velocities obtained from wheel encoders is used in the prediction step. The bearing and the range measurements obtained from the rotating laser are used in the observation step. However, since only a single range-bearing measurement pair is used at each iteration in the observation step, it should be noted that the innovation vector would be of a length 2. The update equations also follow those defined in Section 3.3 of Chapter 3, with all matrix dimensions consistent with the length of the innovation vector.

4.2.2 Active Sensing of the Environment

Active sensing is implemented by determining the direction along which the sensor is to be aimed so that the information gain is maximised. To discretize the problem, the range of rotation of the sensor is divided into n segments. The sensor is given a segment as the goal, and it continuously rotates towards the furthest end of the goal segment in a sweeping motion while providing its current range and bearing measurements as observations to the EKF. Once it reaches its goal, a new bearing goal is calculated based on predicted information gain. For an example, Figure 4.2 shows an example where a rotation range of 270° is divided into 6 segments. If the current bearing of the range sensor is within segment n_2 and the goal is selected as segment n_5, the range sensor would rotate from its current bearing to a bearing of $-90°$ in a sweeping motion.

4.2.2.1 Selection of Goal Segment

The current pose estimate $\hat{\boldsymbol{X}}_{\boldsymbol{k}}$ and its covariance matrix P_k are obtained from the EKF to calculate the next goal. Given these values, for each segment n_i, a hypothetical observation vector X_{o_i} is generated by ray tracing at fixed angular resolution. Based on each of these hypothetical observation vectors, an updated covariance matrix Pk_i is calculated for each segment n_i using the standard observation and update steps of the EKF framework.

A cost function that combines the information gain as well as the cost of motion is then defined for the i^{th} segment as

$$cost(i) = trace(P_{ki}) + (\lambda\ d) \tag{4.1}$$

where d is the distance from the current segment j to the i^{th} segment calculated as $d_{ij} = |i - j|$ and λ is a tuning parameter. The segment associated with the minimum cost is selected as the next goal segment.

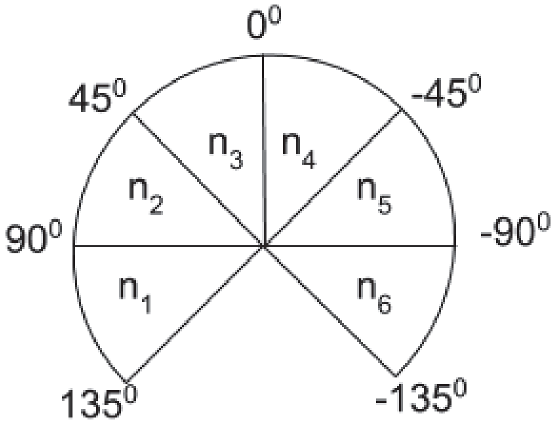


FIGURE 4.2: An example of a 270° rotation range divided into 6 segments

Parameter	Value
Range of rotation	360 °
Angular velocity	0.5250 $rads^{-1}$
Number of segments (n)	8
λ	0.00001
Max laser range	3 m
Min laser range	0.5 m

TABLE 4.1: Sensor parameters for the Dataset 1

4.3 Experiments

This section presents results from a dataset collected in a simulation environment and a dataset collected from a real world experiment.

4.3.1 Dataset 1

This simulation dataset was generated in the Willow Garage environment using Gazebo. A Turtlebot robot equipped with a rotating range sensor navigates in a room and a corridor inside the Willow Garage environment. The parameters of the rotating range sensor are listed in Table 4.1. The sensor has a rotation range of 360 ° and it rotates at an angular velocity of 0.5250 $rads^{-1}$. A map of the environment is created by navigating the robot equipped with a 2D LiDAR through the environment and stitching the scans using ground truth poses.

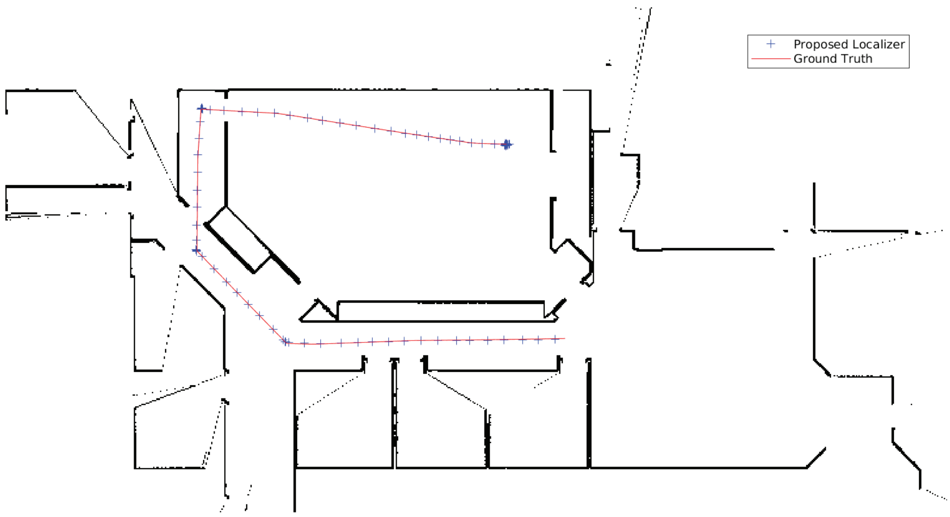


FIGURE 4.3: Trajectory of the robot in simulation estimated using the proposed framework and compared with the ground truth

Dimension	RMSE	MAE
X (m)	0.0562	0.0453
Y (m)	0.0343	0.0248
ϕ (rad)	0.0084	0.0065

TABLE 4.2: Error values for simulation

Figure 4.3 shows the position estimates provided by the proposed localization framework (sparse for clarity) overlapped by the true trajectory. The root mean squared error (RMSE) and mean absolute error (MAE) metrics are displayed in Table 4.1.

Figure 4.4 shows the error and the uncertainty values of each component of the pose estimate provided by the proposed localization framework. At $t = 300s$ it can be observed that the uncertainty and the error along the X direction are higher compared to the uncertainty and error along the Y direction. At $t = 365s$ it can be observed that the uncertainty along the X direction starts to decrease. At $t = 500s$ it can be observed that the uncertainty and the error along the Y direction are higher compared to the uncertainty and the error along the X direction. These behaviors can be explained using Figures 4.5, 4.6 and 4.7.

At $t = 300s$, as seen in Figure 4.5, only a wall parallel to the X direction is within the range of the sensor. The sensor is directed towards that wall because it is the only region within

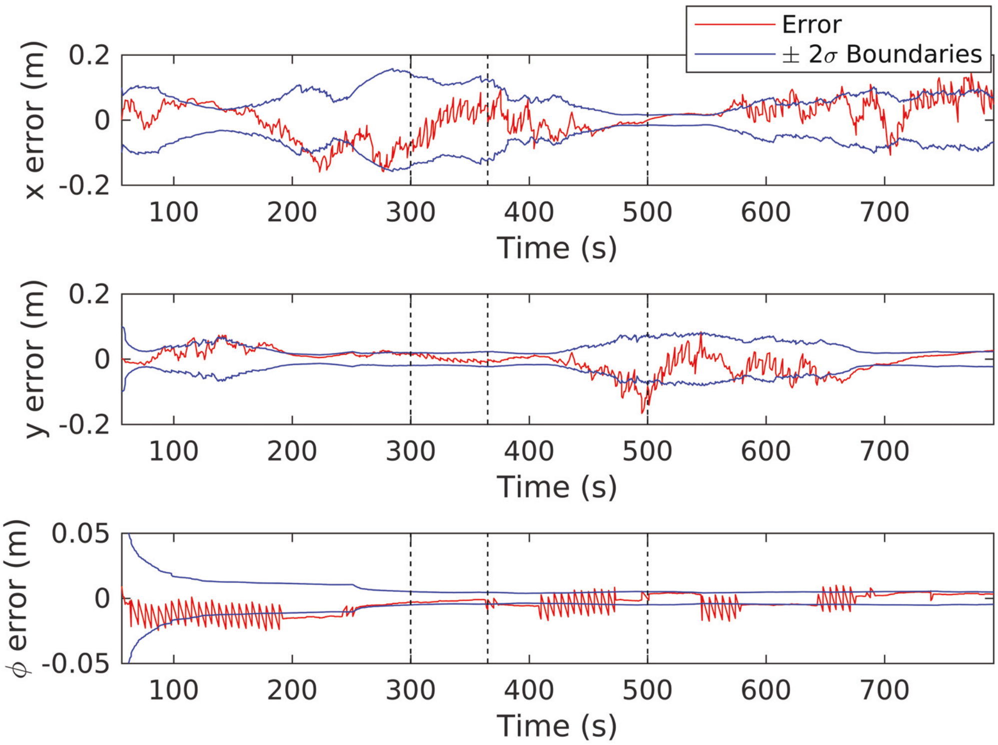


FIGURE 4.4: Error of the robot pose estimates for simulation with the proposed localization framework compared with ground truth

the observable area of the map (area inside the green circle) that can produce information. The EKF framework continues to accumulate information about the position of the robot along the Y direction while the uncertainty and the error along the X direction increase. At $t = 365s$ it can be seen in Figure 4.6 that the robot reaches the end of the room and a wall parallel to the Y direction is now within the range of the sensor. Since the robot pose uncertainty along the X direction is high, the sensor is now directed towards that wall. The uncertainty along the X direction gradually decreases when the sensor is kept directed at that wall. At $t = 500s$ it can be seen in Figure 4.7 that the sensor continues to observe a wall parallel to the Y direction, increasing the estimator's information about the position of the robot along the X direction. However, during this period, the uncertainty and the error along the X direction increases. Information about the orientation of the robot is consistently obtained, regardless of the orientation of the sensor. In turn the uncertainty estimate of the yaw is consistent. The oscillation of the yaw error indicates the EKF estimator converging to the true yaw value during fast rotations.

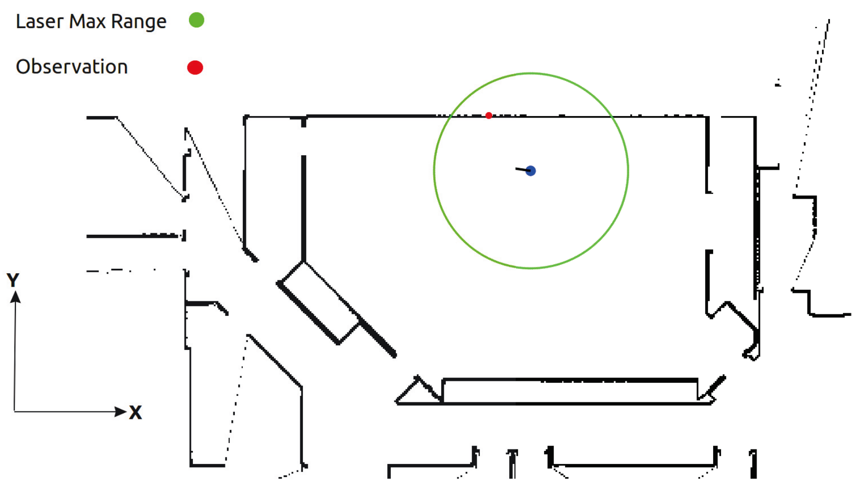


FIGURE 4.5: Observation, estimated robot pose and maximum sensor range at $t = 300s$. Only a wall parallel to X direction is within range and the sensor is pointed towards it. Pose uncertainty along the X direction increases.

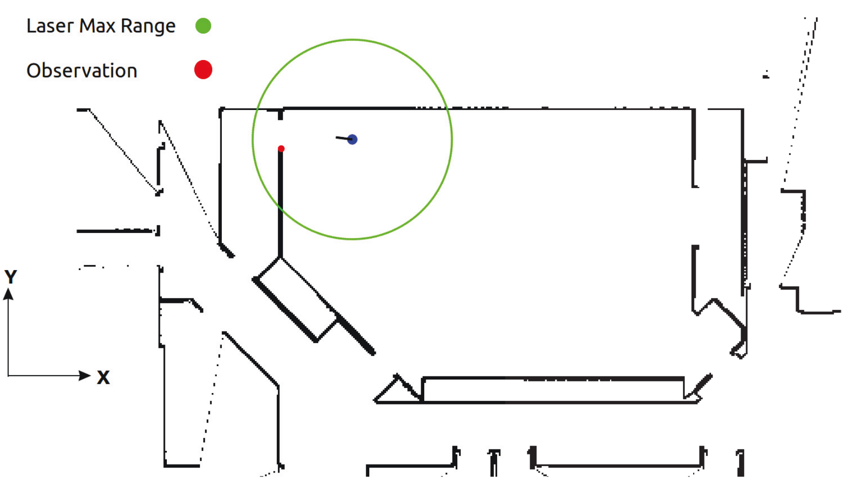


FIGURE 4.6: Observation, estimated robot pose and maximum sensor range at $t = 365s$. A wall parallel to Y direction is within range. To reduce the pose uncertainty along X direction, the sensor is directed towards that wall.

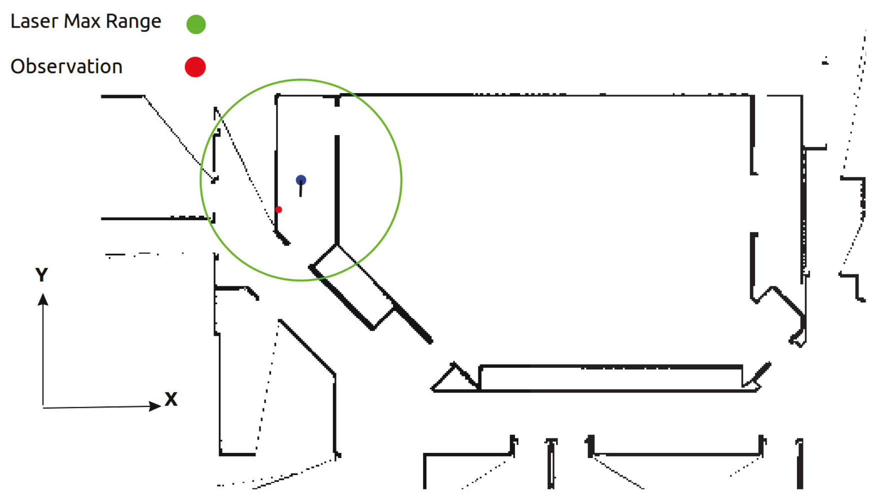


FIGURE 4.7: Observation, estimated robot pose and maximum sensor range at $t = 500s$. The sensor is directed towards a wall parallel to Y direction. Pose uncertainty along Y direction grows.

Parameter	**Value**
Range of rotation	270 °
Angular velocity	0.1745 $rads^{-1}$
Number of segments (n)	23
λ	0.00001
Max laser range	30 m
Min laser range	0.5 m

TABLE 4.3: Sensor parameters for the Dataset 2

4.3.2 Dataset 2

This dataset was collected at Level B1 of Building 11, University of Technology Sydney using the mobility scooter platform described in Section 3.4.5 of Chapter 3. The rotating range measurement sensor is simulated using a Hokuyo 2D LiDAR. The sensor parameters are described in Table 4.3. As illustrated in Figure 4.8, a single range measurement (shown in red) is obtained from the LiDAR scan (shown in green) to simulate the range sensor. To simulate the rotation of the sensor, between consecutive scans, consecutive laser beams are picked. Since the Hokuyo LiDAR used in this experiment operates at a frequency of 40 Hz and covers the field of view of 270 ° using 1081 laser beams, the rotation of the sensor is simulated at an angular velocity of 0.1745 $rads^{-1}$. A map of the environment is created using GMapping [68].

FIGURE 4.8: An example for a laser scan from the Hokuyo 2D LiDAR and the single observation used for active localization

Since the true robot poses are not available in this dataset, robot poses are obtained using the passive EKF based localization algorithm proposed in Section 3.3 in Chapter 3 to compare with the proposed active localization approach.

A comparison of the trajectories estimated using the two algorithms is demonstrated in Figure 4.9. It can be seen that in spite of only using a single pair of range-bearing measurements at the update step, the trajectory estimated by the active localization strategy and the trajectory estimated by a passive EKF based localizer that uses the dense observations from the 2D LiDAR with 270° field of view are similar. This is further demonstrated in

FIGURE 4.9: A comparison between the trajectories estimated for Dataset 2 by the proposed active localization algorithm and the passive EKF localizer proposed in Chapter 3

Figure 4.10 where the difference of each component of the two pose estimates are plotted over time.

Accurate localization using a single pair of range-bearing measurements with an EKF framework on an environment represented by a VDT is made possible by active sensing. If the direction of the range sensor is fixed (i.e., passive sensing), successful localization cannot be guaranteed. To demonstrate this, passive localization was attempted using a single observation obtained at the fixed direction of robot heading. The results of this

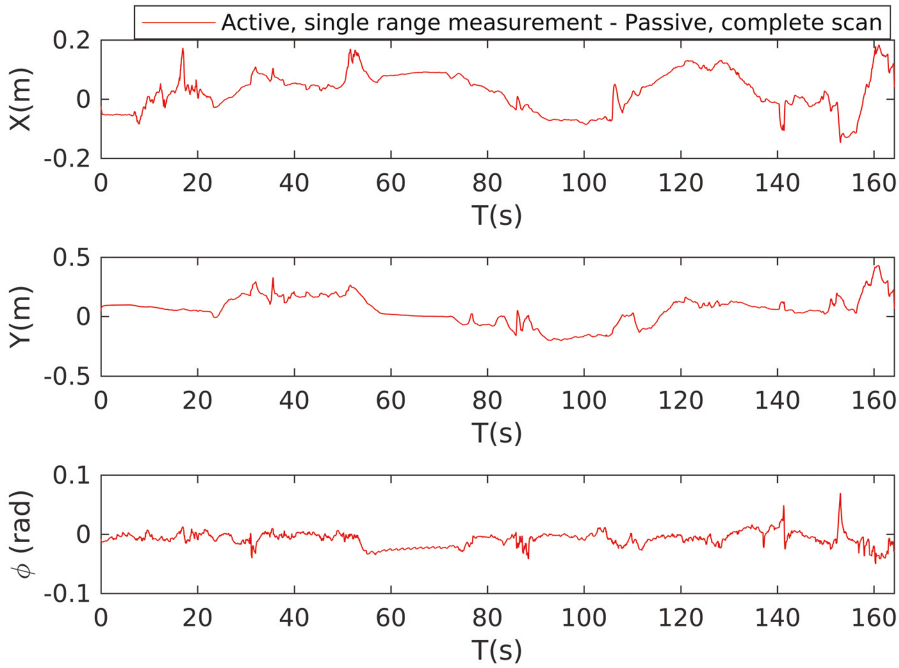


Figure 4.10: A comparison between the poses estimated by the active localization algorithm and the passive EKF localizer proposed in Chapter 3 for Dataset 2

exercise are demonstrated in Figures 4.11 and 4.12. It can be seen that, with passive sensing and a single range measurement at a time, the EKF can continue to estimate the robot pose with a high degree of accuracy up to a point since the observations up to that point provide sufficient information and the odometry noise is low. However, it can be seen that passive sensing leads to the eventual failure of localization.

4.4 Discussion and Conclusions

This chapter presents an active localization framework based on information gain which can localize a mobile robot that follows a predefined trajectory, in an environment represented by a 2D VDT map when the robot is equipped with a range sensor that can be rotated parallel to the map plane. The sensor provides a single pair of range-bearing measurements at a time that is used by an EKF framework to estimate the robot pose.

FIGURE 4.11: Trajectory of the robot in Dataset 2 estimated by passive localization using a single observation at a fixed direction compared with the EKF localizer proposed in Chapter 3

The EKF can accurately estimate a robot pose and its uncertainty using a sparse set of range-bearing measurements, provided that each range-bearing measurement pair contributes to improve the estimates. Localization using only a single pair of range-bearing measurements and actively sensing regions of the environment that contributes to a stronger estimate of the pose is demonstrated. Using a simulation dataset that contains true robot poses, the accuracy of the proposed strategy is verified. A real world dataset is then used to validate the proposed active localization strategy by comparing with the pose estimates provided by an EKF based localization framework that is provided with measurements

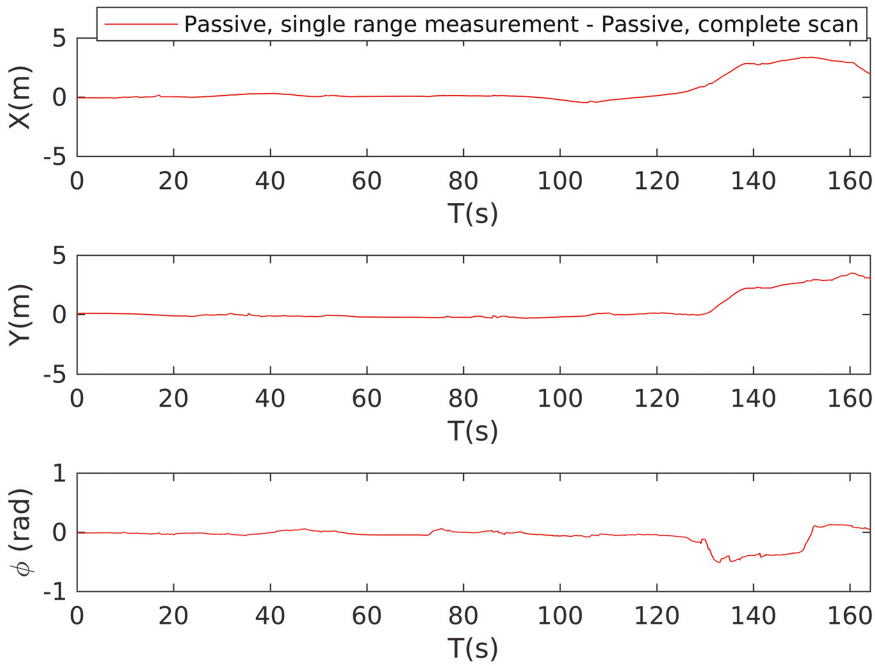


FIGURE 4.12: Difference between the poses estimated by passive localization using a single observation at a fixed direction compared with the EKF localizer proposed in Chapter 3

from a 2D LiDAR. The results pave way for the use of low cost sparse sensors such as solid state LiDARs for mobile robot localization.

In the proposed active localization strategy, the trace of the predicted pose covariance matrix is used to evaluate the information gain. The suitability of alternative metrics such as the predicted entropy of the pose, the predicted area of the covariance ellipse, and the predicted volume of the covariance ellipsoid remains to be investigated. Furthermore, it remains to be investigated whether there are alternatives to the greedy minimization approach.

The successful localization of a robot using the proposed approach depends on several factors such as the maximum range of the sensor, the angular velocity of the sensor, the accuracy of the odometry measurements and the velocity of the robot. As demonstrated in the simulation experiment, the pose estimate starts to accumulate errors when the sensor is unable to observe areas of the environment that can provide a sufficient amount of information. The rate of error accumulation corresponds to the magnitude of the noise in the odometry measurements. Furthermore, it is intuitive that the sensor rotation velocity has to be high relative to the linear and angular velocities of the robot for successful localization. However, further investigation is needed to develop a mathematical relation between the sensor and robot velocities.

The environment representations so far discussed are all assumed to be perfect. In practice, all real world representations contain errors. In the next chapter, how the error of a VDT map representing an environment can be captured, is presented.

Chapter 5

Representation of Uncertain Vector Distance Transform Maps

5.1 Introduction

In autonomous mobile robot navigation, a map is a mathematical representation of an environment that a robot can “understand”. The importance of the availability of a map for autonomous mobile robot navigation is emphasized in Chapter 2. However, these maps may not represent the environment completely accurately and they may contain errors. Therefore, it is necessary for an autonomous mobile robot to have a measure of accuracy of the map for safe and efficient navigation.

Feature maps, which describe the environment using geometric primitives, have an efficient and compact way of expressing the error of each feature location estimate (and in turn, the error of the map), as a covariance matrix. In a feature based Extended Kalman Filter (EKF) Simultaneous Localization and Mapping (SLAM) framework, when the map converges, a fully correlated covariance matrix that represents the error in feature pose estimates that is determined only by the initial covariance in the robot is naturally produced [28]. When such a map is used to localize an autonomous mobile robot in an EKF framework, the map error can be taken into account by incorporating the covariance matrix in the innovation uncertainty. However, as explained in Section 2.1 in Chapter 2, the feature maps are not suitable for some autonomous navigation tasks such as path planning, as information about free space is not explicitly encoded in the map.

Alternatively, an Occupancy Grid Map (OGM) [3, 4] which represents the environment using a set of finite-sized cells, each of which is encoded with a value describing the probability of it being occupied, free, or not known, inherently is a better alternative for autonomous robot navigation tasks such as path planning as the definition of free space a robot can navigate in is clear. However, there are several limitations in OGMs, as discussed below.

Since the OGM is a discrete representation of the environment, its accuracy is dependent on the resolution. Therefore, to a achieve higher accuracy, a higher resolution is required, which in turn increases the memory usage. Furthermore, the OGM is essentially assumed to be perfect. That is, it does not capture an estimate of the map error. While the probability of occupancy can be considered as a measurement of error, most applications of OGMs threshold the probability of occupancy to obtain a binary map. Another limitation of OGMs is that OGMs assume the probability of occupancy of each cell to be independent from its neighbours. This assumption is necessary to reduce the high dimensionality of the mapping problem [8]. In reality however, this assumption does not hold as the occupied points in space are not randomly distributed and there exists a spatial correlation between occupied cells.

Normal Distribution Transform (NDT) [5, 6] is another method of environment representation. Similar to OGMs, the environment is discretized into constant sized cells. However, instead of each cell being encoded with the probability of it being occupied, each cell in NDT describes the probability of measuring a sample for any position within itself using a normal distribution. It is piecewise continuous and differentiable. This approach captures the map error within each cell as a covariance matrix. However, it still suffers from the discretization error as there is no correlation between the normal distributions of two neighbouring cells.

Gausian Process (GP) representation of OGMs is a continuous, non-linear and non parametric representation of the environment that can overcome the limitations of OGM mentioned above [8, 9]. A GP is essentially a natural generalization of the Gaussian distribution where the mean and the covariance of a Gaussian distribution are a vector and a matrix whereas the mean and the covariance of a GP are functions [7], which are learned using training data obtained from a range-bearing sensor such as a 2D LiDAR at known locations. The GP, after training, can be queried at a desired resolution for probability of occupancy and be thresholded to obtain the usual OGM. While the errors in sensor measurements and pose estimates used to train the GP are captured in the posterior probability of occupancy, this information is partially lost during thresholding as the output is a binary OGM.

Vector Distance Transform (VDT) maps presented in this work so far have been derived from OGM maps and are interpolated to obtain a continuous representation. Therefore they do not contain any information about the accuracy of the map. The main contribution of this chapter is a technique for building a Two-Dimensional (2D) VDT map together with its uncertainty from a 2D point cloud. The uncertainty of each point in the point cloud is computed by propagating the sensor noise and the robot location uncertainties. The fact that the distance transform is the entropy satisfying solution to the Eikonal equation is explored. The focus is to generate a compact, parameterized representation, where the mean and the covariance matrix of the parameter vector naturally captures the map uncertainty, exactly the same way as in the case of a geometric feature based map. In effect, these parameters can be thought of as high level features, making it possible to write appropriate observation equations relating these “features”, robot locations and sensor observations. As a result any existing probabilistic localization technique may be used to generate robot location estimates that incorporate the uncertainty of the map.

This chapter is organized as follows. Section 5.2 presents the proposed map building process. Section 5.3 presents experimental results to validate the proposed approach. Section 5.4 concludes the chapter by discussing the findings.

5.2 Representation of Uncertain Vector Distance Transform Maps Using High Level Feature Vectors

Representing environments using a VDT is extensively discussed in Section 2.2 in Chapter 2. This section proposes a novel method to use the VDT to build a 2D map by incorporating the uncertainties of the sensor measurements and the pose estimates which can later be used in a VDT based localization framework. A set of range-bearing measurements (such as 2D LiDAR measurements) and the corresponding poses at which these measurements are obtained, together with their uncertainties are assumed to be available. For an example, a conventional Pose-SLAM algorithm is able to generate this information.

If $u(x),\ x \in \Re^2$ is the distance from point x to a manifold S where $dim(S) = 0$ (i.e. a set of points), then $u(x)$ describes the Unsigned Distance Transform (UDT). It has been shown that the UDT is also the entropy satisfying solution of an Eikonal equation at unit speed [109], as shown in (5.1).

$$|\nabla u| = 1, u|_S = 0 \tag{5.1}$$

For an environment represnted by a VDT denoted by $DT_v : \Re^2 \to \Re^2$ and a UDT denoted by $DT_u : \Re^2 \to \Re^1$ for any $(x, y) \in \Re$, the relationship between DT_v and DT_u is given as

$$DT_u(x,y)^2 = DT_x(x,y)^2 + DT_y(x,y)^2 \tag{5.2}$$

where DT_x and DT_y are the two orthonormal components of DT_v.

Using (5.1) and (5.2), a Partial Differential Equation (PDE) can be derived as

$$\frac{(DT_x \frac{\partial DT_x}{\partial x} + DT_y \frac{\partial DT_y}{\partial x})^2 + (DT_x \frac{\partial DT_x}{\partial y} + DT_y \frac{\partial DT_y}{\partial y})^2}{DT_x^2 + DT_y^2} = 1 \tag{5.3}$$

where the boundary conditions are given by

$$DT_x(x, y) = 0, DT_y(x, y) = 0, \forall(x, y) \in S \tag{5.4}$$

The solution to this PDE is a VDT. However, this problem cannot be analytically solved and an alternative approach has to be pursued to solve it. The following section presents a generic approach to solve a PDE when an analytical solution is not straightforward.

5.2.1 Parametric Approximation of a Solution to a PDE

Let $\mathscr{L}$ and $\mathscr{B}$ be differential operators, f and g be given functions and $\partial\Omega$ be the boundary of a bounded open domain Ω. For a problem modeled as (5.5) and (5.6), an approximate solution to the PDE can be derived by reducing the problem to an unconstrained optimization problem if a function with enough parameters that can be arbitrarily close to any element in the underlying space $\mathscr{V}$ is available [110].

$$\mathscr{L}u = f, x \in \Omega \tag{5.5}$$

$$\mathscr{B}u = g, x \in \partial\Omega \tag{5.6}$$

Given such a function $u_a(x, \beta_i)$ where β_i represents the parameters, the objective function of the unconstrained optimization problem that results in the approximate solution $u = u_a(x, \beta_i)$ is given by

$$h = \int_\Omega ||\mathscr{L}u_a - f||^2 d\mathscr{V} + \int_{\partial\Omega} ||\mathscr{B}u_a - g||^2 dS \tag{5.7}$$

Therefore, an approximate solution for the PDE described by (5.3) and (5.4) can be obtained if a good approximation function is determined. The following section explores such an approximation function that can be used with a VDT representation of an environment.

5.2.2 Cubic Spline Surface Approximation of VDT

Cubic splines are widely used for representing complex shapes. A cubic spline surface is constructed by a set of surface patches that connect with C^2 continuities. A surface patch can be defined as (5.8)

$$P(p,q) = (\frac{1}{6})^2 \begin{bmatrix} p^3, p^2, p, 1 \end{bmatrix} MPM^T \begin{bmatrix} q^3 \\ q^2 \\ q \\ 1 \end{bmatrix} \tag{5.8}$$

where M is the basis matrix defined as (5.9), P is a 4x4 control point grid defined as (5.10) and $0 \leq p, q \geq 1$ are the parameters that describe the relative location within the surface patch with four corner points $K_{00} = P(0,0)$, $K_{01} = P(0,1)$, $K_{10} = P(1,0)$, $K_{11} = P(1,1)$. The set of control points P_{ij} is the sole set of parameters that govern the behavior of the cubic spline surface.

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{5.9}$$

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \tag{5.10}$$

It will be numerically illustrated in Section 5.3 that cubic splines provide a suitable approximation to the distance function given a sufficiently dense set of control points. Furthermore as they have local support, the Jacobians associated with the optimization problem defined by (5.7) become sparse.

Using cubic splines as the approximation function, the PDE defined by (5.3) and (5.4) to which the solution is a VDT can now be solved using unconstrained optimization with the objective function h_v defined as (5.11).

$$h_v = \int_{\Omega} \left|\left| \left(\frac{(DT_x \frac{\partial DT_x}{\partial x} + DT_y \frac{\partial DT_y}{\partial x})^2 + (DT_x \frac{\partial DT_x}{\partial y} + DT_y \frac{\partial DT_y}{\partial y})^2}{DT_x^2 + DT_y^2} \right) - 1 \right|\right|^2 d\mathscr{V} + \int_{\partial\Omega} \left(DT_x^2 + DT_y^2 \right) dS \quad (5.11)$$

As discussed in Section 2.2 in 2, the VDT has discontinuities along the cut-loci. Close to these cut-loci, the Eikonal equation does not hold in the cubic spline approximation. Therefore, sample points in the underlying space $\mathscr{V}$ that fall along the cut-loci need to be detected and removed.

However, at this point, the uncertainty of the point cloud (the set of points x) has not been taken into account in the map building process. The following subsection describes how the uncertainty of the point cloud can be taken into account in the map building process.

5.2.3 Estimating Map Uncertainty

Let $\hat{\boldsymbol{X}}_{\boldsymbol{R}} = (x_r, y_r, \phi_r)^T$ be a mobile robot pose estimate in a 2D environment with covariance $cov(\hat{\boldsymbol{X}}_{\boldsymbol{R}})$ and $S_{r\theta} = \{(r_i, \theta_i\})$ be the observations obtained from a 2D LiDAR with a range uncertainty of σ_r at $\hat{\boldsymbol{X}}_{\boldsymbol{R}}$. Each observation from the sensor can be projected into the world coordinate frame using (5.12) to obtain a 2D point cloud.

$$X_{oi} = \begin{Bmatrix} x_{oi} \\ y_{oi} \end{Bmatrix} = \begin{Bmatrix} x_r + r_i \cos(\theta_i - \phi_r) \\ y_r + r_i \sin(\theta_i - \phi_r) \end{Bmatrix} \quad (5.12)$$

The uncertainties of the robot pose estimate and range measurements can be propagated to each of the points in the point cloud $cov(X_{oi})$ as (5.13).

$$cov(X_{oi}) = \Big(\frac{\partial X_{oi}}{\partial \hat{\boldsymbol{x}_{\boldsymbol{R}}}}\Big) \cdot cov(\hat{\boldsymbol{x}_{\boldsymbol{R}}}) \cdot \Big(\frac{\partial X_{oi}}{\partial \hat{\boldsymbol{x}_{\boldsymbol{R}}}}\Big)^T + \Big(\frac{\partial X_{oi}}{\partial r}\Big) \cdot \sigma_r^2 \cdot \Big(\frac{\partial X_{oi}}{\partial r}\Big)^T \tag{5.13}$$

The uncertainty of the observations can be integrated as a set of weights ρ_{oi}, derived as (5.14) to remodel objective function described in (5.11).

$$\rho_{oi} = \frac{1}{\Big(\frac{\partial(DT_x^2+DT_y^2)}{\partial X_{oi}}\Big) cov(X_{oi}) \Big(\frac{\partial(DT_x^2+DT_y^2)}{\partial X_{oi}}\Big)^T} \tag{5.14}$$

Although the Eikonal equation described in (5.3) should be satisfied everywhere in $\mathscr{V}$, there will be a residue that depends on the quality of the approximation. Accordingly, a weight ρ_{ei} is empirically determined to be associated with the relevant elements of the objective function. This leads to the weighted non linear least squares problem h_{v^*} defined in (5.15).

$$\begin{aligned} h_{v^*} = \int_{\Omega} \rho_{ei} \Big|\Big| \Big(\frac{(DT_x \frac{\partial DT_x}{\partial x} + DT_y \frac{\partial DT_y}{\partial x})^2 + (DT_x \frac{\partial DT_x}{\partial y} + DT_y \frac{\partial DT_y}{\partial y})^2}{DT_x^2 + DT_y^2} \Big) - 1 \Big|\Big|^2 d\mathscr{V} \\ + \int_{\partial\Omega} \rho_{oi} \Big(DT_x^2 + DT_y^2 \Big) dS \end{aligned} \tag{5.15}$$

The approximate solution consists of two cubic spline surfaces that correspond to DT_x and DT_y. As the parameters representing the environment are obtained through solving an optimization problem, an explicit function relating the inputs to this solution does not exist. Therefore the covariance matrices for the optimized parameter vectors corresponding to DF_x and DF_y, denoted by Px_{ij}^* and Py_{ij}^* are given by (5.16) and (5.17)

$$cov(Px_{ij}^*) = J_x * cov(X_{oi}) * J_x^T \tag{5.16}$$

$$cov(Py_{ij}^*) = J_y * cov(X_{oi}) * J_y^T \tag{5.17}$$

where J_x and J_y are the corresponding Jacobians at the optimal solution. J_x and J_y are derived using the implicit function theorem [87] as (5.18) and (5.19) where H_x is the Hessian matrix of h_{v^*} with respect to Px_{ij} and H_y is the Hessian matrix of h_{v^*} with respect to Py_{ij}.

$$J_x = -H_x^{-1} * \left[\frac{\partial^2 h_{v^*}}{\partial P x_{ij}^* \partial r_k} \right] \tag{5.18}$$

$$J_y = -H_y^{-1} * \left[\frac{\partial^2 h_{v^*}}{\partial P y_{ij}^* \partial r_k} \right] \tag{5.19}$$

5.3 Experimental Results

This section illustrates the proposed approach using two simulation experiments.

5.3.1 Experiment 1

A robot equipped with a 180° FOV LiDAR observing an L-shaped environment from 3 poses is simulated as shown in Figure 5.1.

To determine an appropriate control point density required to approximate the VDT for this environment, numerical experiments in which the true VDT is approximated with cubic spline surfaces with different resolutions were conducted. Figure 5.2 demonstrates the differences between the true VDT and its least squares approximations at different resolutions. As discussed in Section 5.2.2, this demonstrates that given a sufficiently high control point resolution, a cubic spline surface can approximate the VDT highly accurately.

To simulate the impact of the sensor noise and the pose estimate noise, the range observations and pose estimates are corrupted using zero mean Gaussian noise with the parameters given in Table 5.1. The point cloud map generated from the resulting pose estimates and range measurements is shown in in Figure 5.3.

Using a control point resolution of 0.2m, the VDT of the environment is approximated and is demonstrated in Figure 5.4. Figure 5.5 demonstrates the zero contour of the VDT

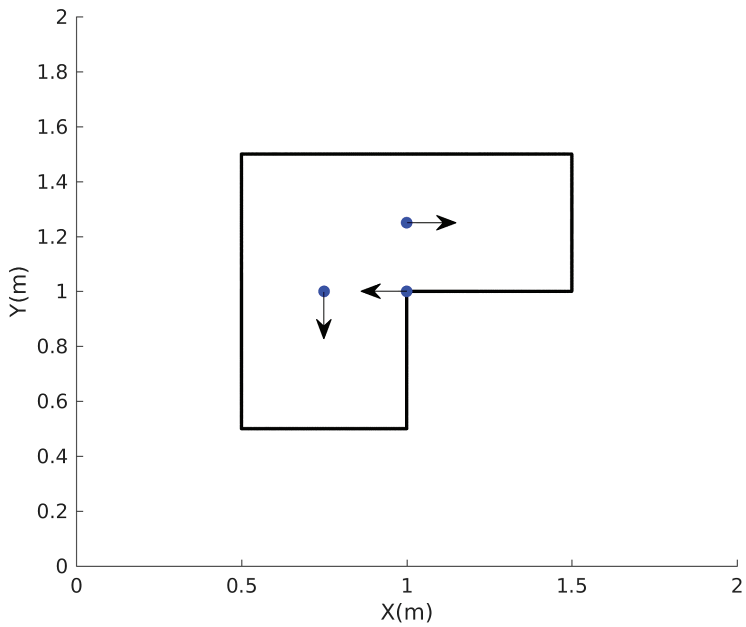


FIGURE 5.1: Experiment 1 : The environment and the robot poses

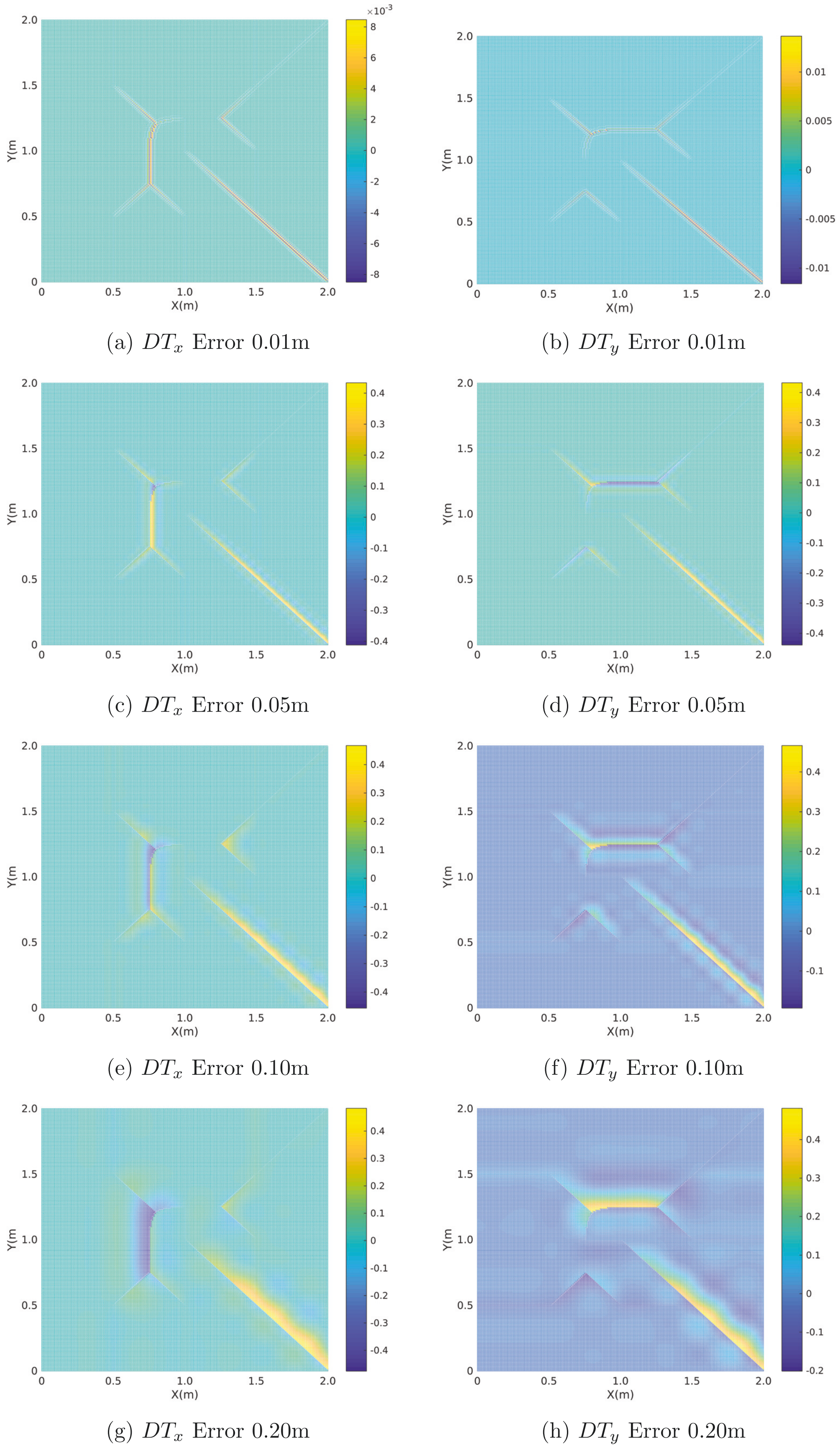
Measurement	Noise Parameter
Position Estimates	$\sigma_x = 0.01\text{m}$ $\sigma_y = 0.01\text{m}$
Orientation Estimates	$\sigma_\phi = 0.005\text{rad}$
Range Measurements	$\sigma_r = 0.02\text{m}$

TABLE 5.1: Noise parameters used in Experiment 1

approximation. It can be seen that the boundary closely follows the observations and the edges are smooth as the cubic spline has a control point resolution of 0.2m. The approximated VDT is queried at the true boundary points. The distance values given by the VDT indicate the error in the map, as ideally the VDT should be zero at true boundary points. The error and its uncertainty are demonstrated in Figure 5.6 and it can be seen that the error is bounded by the 2σ uncertainty estimates.

5.3.2 Experiment 2

A Turtlebot robot equipped with a 270° FOV LiDAR is simulated to navigate through a corridor in an example office environment (Willow Garage office) in Gazebo (Figure 5.7). As in the case of Experiment 1, the range measurements and the true robot pose are corrupted using zero mean Gaussian noise to simulate a real world situation. The noise



(a) DT_x Error 0.01m

(b) DT_y Error 0.01m

(c) DT_x Error 0.05m

(d) DT_y Error 0.05m

(e) DT_x Error 0.10m

(f) DT_y Error 0.10m

(g) DT_x Error 0.20m

(h) DT_y Error 0.20m

FIGURE 5.2: Error (meters) in approximating the VDT using cubic splines at different control point resolutions

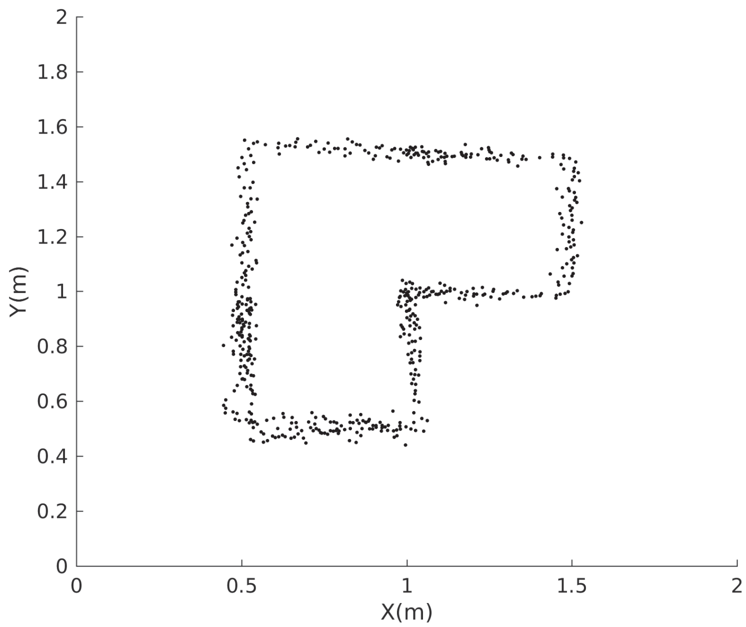
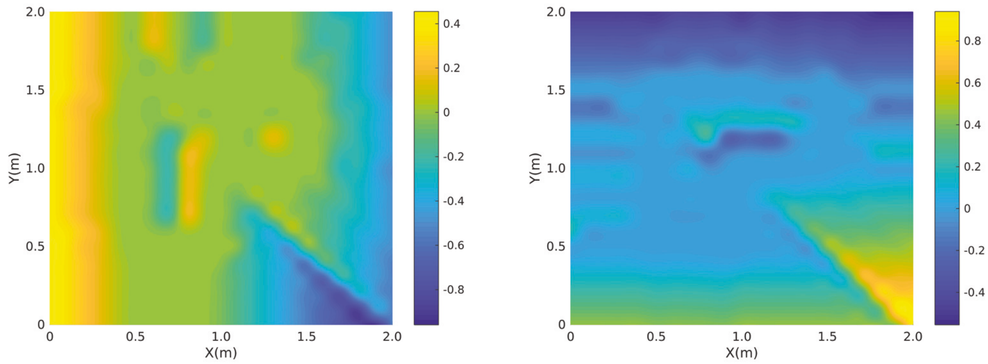


FIGURE 5.3: Experiment 1 : Noisy observations reprojected using noisy pose estimates



(a) DT_x (b) DT_x

FIGURE 5.4: Experiment 1 : Optimal VDT at 0.2m control point resolution

parameters are given in Table 5.2. The point cloud map generated from the corrupted pose estimates and range measurements is shown in Figure 5.8.

To manage the computational load, the VDT is approximated with a control point resolution of 0.25m. Figure 5.9 shows the resulting OGM at the same resolution.

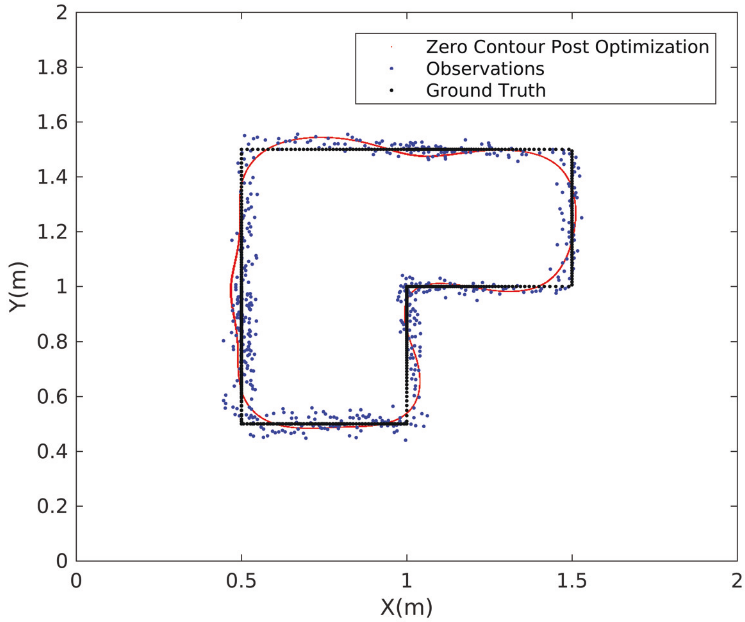


FIGURE 5.5: Experiment 1 : Zero contour of the approximated VDF, observations, and true boundary

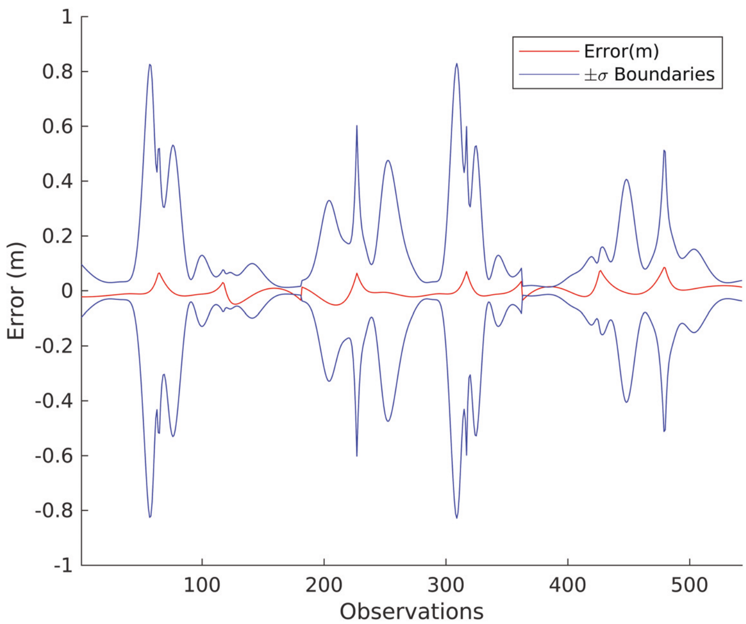


FIGURE 5.6: Experiment 1 : Error of the VDT approximation bounded by uncertainty

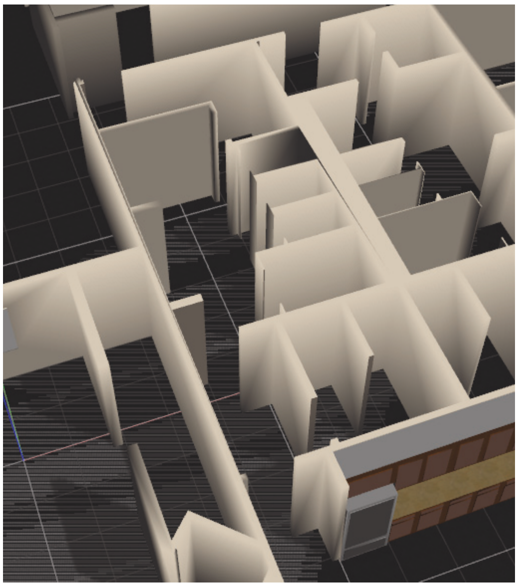


FIGURE 5.7: Experiment 2 : A corridor in the Gazebo Willow Garage environment

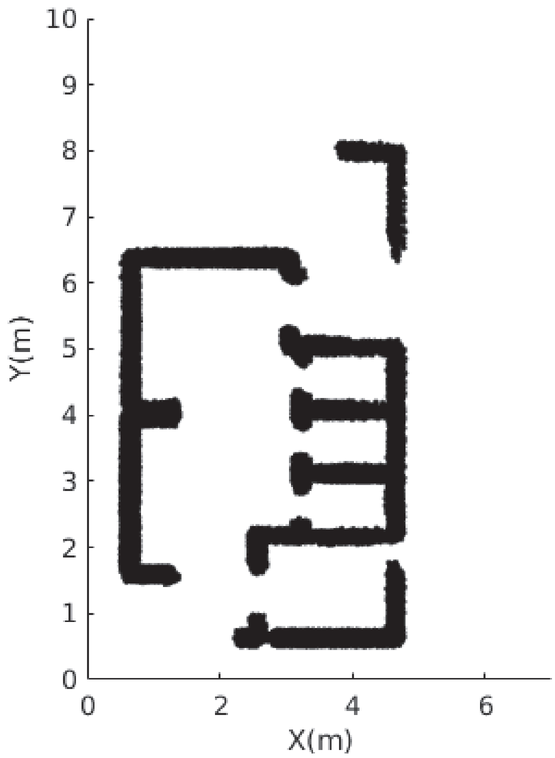


FIGURE 5.8: Experiment 2 : Point Cloud Map

Measurement	**Noise Parameter**
Position Estimates	$\sigma_x = 0.05\text{m}$ $\sigma_y = 0.05\text{m}$
Orientation Estimates	$\sigma_\phi = 0.005\text{rad}$
Range Measurements	$\sigma_r = 0.02\text{m}$

TABLE 5.2: Noise parameters used in Experiment 2

FIGURE 5.9: Experiment 2 : OGM of the Office Space Corridor after Optimization

5.4 Discussion and Conclusion

The method for representing environments using a parameter vector and its covariance matrix proposed in this chapter can be seen as an attempt to unify occupancy grid maps that capture environment geometry using aggregation of point clouds, with maps that are a collection of geometric features such as points and lines. The ability to represent point clouds with a parameter vector and associated uncertainties makes it possible to extend probabilistic techniques that can only be used with feature maps, for example extended Kalman filters, to deal with uncertain point clouds.

While the work in this chapter has presented a suitable algorithm for representing uncertain point cloud maps and that the results demonstrate the viability of the proposed algorithm, a number of questions remain. During the experimental evaluation, it was observed that weights in the objective function of the weighted non linear least squares problem has a significant impact on the accuracy of the resulting map. In the examples presented, these were derived by trial and error. The set of weights ρ_{oi} were normalized into a range between 1.0-2.0 inclusive. The weight ρ_{ei} was set to 0.125. For practical use, it is essential that strategies to automatically determine these values are developed. The sharp corners in the environment are made smooth by the approximation function. The effect of such smoothening on localization algorithms has to be investigated. The proposed methodology is also computationally heavy. One way to address this is to represent environments with submaps that are tractable and develop a strategy to merge these while preserving all the associated uncertainties. In the spirit of truncated distance transforms that have become a popular in representing 3D point clouds, it is clearly worth investigating whether non-uniform splines can be used to reduce the dimensionality of the problem without a significant loss of utility. Exploring whether there are different basis functions that naturally satisfy the Eikonal equation, in order to reduce the dimensionality of the representation, to make the optimization process converge faster and also provide guarantees that the true optimum has been reached, could be valuable. Another possibility is to explore whether it is feasible to use other universal approximators such as deep neural networks for which there are very efficient and fast implementations. Using the parameterized representation and the associated covariance matrix in localisation and SLAM algorithms that are currently limited to environments with geometric features, for example extended Kalman filters, and evaluate their relative merits when compared with traditional solutions in occupancy grids is also likely to be a fruitful avenue for further research.

Chapter 6

Conclusions

For a mobile robot to be autonomous, it requires knowledge about the environment in the form of a map. Maps are mathematical models of the environment that the robot can understand. Given a map, estimating the pose of the robot within that map is known as localization.

In this thesis, the Vector Distance Transform (VDT) is proposed as a superior alternative than its counterparts, the Unsigned Distance Transform (UDT) and the Signed Distance Transform (SDT), to represent the environment for autonomous mobile robot navigation as the VDT contains more information about the environment than the UDT and the SDT. Given a VDT map, a novel sensor model for a laser range finder that can model the measurement error is proposed. A non-linear least squares optimization algorithm and an Extended Kalman Filter (EKF) based algorithm are then proposed to localize a mobile robot using the proposed sensor model on a VDT map. Using a Turtlebot robot, a personal mobility scooter and a custom built unmanned aerial vehicle (UAV), the localization capabilities of the proposed localization algorithms are demonstrated in indoor and outdoor environments. It is then demonstrated that a sensor that produces a single pair of range-bearing measurements at a time is adequate to localize a mobile robot on a VDT map if the sensor can actively observe regions of the environment that increase the information about the robot pose. Finally, an algorithm for generating a VDT representation of an environment based on a set of noisy sensor measurements and the corresponding set of noisy poses, that can also capture the uncertainty of the map, is presented. It allows the localization algorithms to incorporate the errors in the map into the uncertainty of the pose.

6.1 Summary of Contributions

6.1.1 Environment Representation Using Vector Distance Transform Maps

Given a set of boundary points, a distance transform describes the distance from the domain to the closest boundary points. Hence distance transforms encode more information about the environment than the conventional occupancy grid maps and feature maps. Distance transform maps are continuous representations of the environment that capture the geometry of the environment as the zero level set. With distance transforms, explicit feature extractions from the observations and associating these features with the map are also not required. As demonstrated in Chapter 5, distance transforms allow the uncertainty of the map to be represented in a convenient way. Therefore, distance transforms can be preferred over feature maps and occupancy grid maps for autonomous mobile robot applications.

There are three main variants of the distance transform; the unsigned, the signed, and the vector distance transforms. It is argued that the well behaved nature of the vector distance transforms at the boundary makes it preferred over unsigned distance transform. Furthermore, unlike the signed distance transform, VDT can represent complex environments. The VDT also contains more information about the environment than its unsigned and signed counterparts as the UDT and the SDT encode only the distance to the closest boundary point whereas the VDT also encodes the direction.

Based on the VDT representation of the environment, a novel sensor model that associates the robot state and the measurement is then proposed. The disparity between the actual observation and the expected observation based on a hypothetical robot pose can be obtained from this sensor model as a vector and as a scalar. The resulting disparity vector and the scalar disparity value can be used in localization algorithms.

6.1.2 Vector Distance Transform Based Robot Localization

Based on VDT maps and the proposed sensor model, two strategies to localize mobile robots are presented.

The first strategy is based on minimizing the scalar disparity measure between the expected and actual observations using non-linear least squares optimization. Given the well behaved nature of the gradients along the boundaries of the VDT map, it is also possible to estimate a covariance for the optimized robot pose.

The second strategy is based on the EKF framework. In contrast to traditional EKF based localization frameworks, no explicit feature extraction or association is required. An innovation vector can be calculated using the disparity vector produced by the sensor model that allows using an EKF framework for localization when the robot is equipped with a LiDAR.

The proposed two strategies are evaluated using simulation and real world datasets collected from a range of robots including an indoor navigating Turtlebot, an outdoor navigating personal mobility scooter and a custom built UAV.

6.1.3 Information Gain Based Active Localization

With a range sensor that can be rotated parallel to the map plane, an information gain based active localization strategy is proposed. It is demonstrated that even when only a single pair of range-bearing measurements is available at a time, a robot can be localized using an EKF, given that the sensor is directed to observe regions of the environment that results in improved pose estimates. The proposed active localization strategy allows the localization of robots using sensors with low spatial resolutions.

The proposed information gain based active localization strategy is validated using simulations and real world datasets.

6.1.4 Representation of Uncertain Vector Distance Transform Maps

Localization algorithms that use occupancy grid maps or distance transform maps usually assume the map to be perfect. However, the errors in the representation of the environment result in errors in the localization. To mitigate this, a method to represent a VDT map using a vector of parameters and its covariance matrix is proposed. Using cubic splines as an approximation function to the solution of a partial differential equation, an uncertain VDT is derived from uncertain point cloud data.

This can be viewed as an attempt to unify occupancy grid maps and feature maps. The uncertainty of the derived VDT map can be incorporated into probabilistic localization frameworks such as the EKF framework. Using a set of simulation experiments, the proposed framework is validated.

6.2 Discussion of Limitations and Future Work

In Chapter 2, VDT maps and a complementing sensor model for laser range finders are presented. While cameras have also been used in the experiments in Chapter 3, the camera observation noise model has been experimentally determined. A possible future research direction is the formulation of a theoretically sound sensor model for monocular cameras that can be used with VDT maps. Furthermore, preliminary work on extending the VDT representation to Three-Dimensional (3D) environments has already been completed. Efficient localization of mobile robots in 3D environments represented by VDT using non-linear least squares optimization and EKF frameworks is a possible future direction.

The cost function used to select the goal segment in the information gain based active localization strategy proposed in Chapter 4 is based on the trace of the covariance matrix, as this has been a metric preferred in the literature [104]. The determinant of the covariance matrix, the area of the covariance ellipse, and the volume of the covariance ellipsoid can also be used as alternatives to the trace of the covariance matrix. Further investigation is needed to determine the best approach to obtain a scalar value to describe the information contained in the covariance matrix. Furthermore, it is intuitive that the sensor has to rotate fast relative to the linear and angular velocities of the robot for successful localization. Further investigation is needed to determine a model for the correspondence between the speed of the rotation of the sensor and the maximum linear and angular velocities the robot can achieve while maintaining accurate pose estimates.

In Chapter 5, a framework for generating an uncertain VDT map using a set of noisy range-bearing measurements and the corresponding noisy pose estimates. While the viability and the validity of the proposed framework are verified, this proof of concept requires a significant amount of research effort before it can be used to represent large scale maps. In the simulation experiments, the weights of the weighted least squares optimization cost function are derived by trial and error. How these weights can be automatically calculated remains unsolved. Cubic splines are used as the approximation function in the proposed approach. However, a number of other basis functions that can satisfy the Eikonal equation such as neural networks exist and whether such a function is better suited to approximate the VDT remains to be explored. The parameter vector becomes computationally intractable when the size of the map increases. It remains to be seen whether the optimization of submaps would result in an accurate map. The proposed approach requires a set of noisy range-bearing measurements and the corresponding noisy pose estimates as the input that can be obtained using a typical Pose-SLAM. The possibility of online generation of uncertain VDT maps by extending the proposed framework to a Simultaneous Localization and Mapping (SLAM) framework remains to be investigated.

Bibliography

[1] Hugh F Durrant-Whyte and John J Leonard. Navigation by correlating geometric sensor data. In *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and System (IROS'89) The Autonomous Mobile Robots and Its Applications*, pages 440–447. IEEE, 1989.

[2] 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.

[3] Alberto Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. *Autonomous mobile robots: Perception, mapping, and navigation*, 1991.

[4] Alberto Elfes. *Occupancy grids: A probabilistic framework for robot perception and navigation.* PhD thesis, 1991.

[5] 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)*, volume 3, pages 2743–2748. IEEE, 2003. ISBN 0-7803-7860-1. doi: 10.1109/IROS.2003.1249285.

[6] Kunjin Ryu, Lakshitha Dantanarayana, Tomonari Furukawa, and Gamini Dissanayake. Grid-based scan-to-map matching for accurate 2D map building. *Advanced Robotics*, 30(7):431–448, 4 2016. ISSN 0169-1864. doi: 10.1080/01691864.2015.1124025.

[7] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[8] Simon O'Callaghan, Fabio T Ramos, and Hugh Durrant-Whyte. Contextual occupancy maps using gaussian processes. In *2009 IEEE International Conference on Robotics and Automation*, pages 1054–1060. IEEE, 2009.

[9] Simon T O'Callaghan, Fabio T Ramos, and Hugh Durrant-Whyte. Contextual occupancy maps incorporating sensor and location uncertainty. In *2010 IEEE International Conference on Robotics and Automation*, pages 3478–3485. IEEE, 2010.

[10] 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*, pages 303–312. ACM, 1996.

[11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, volume 11, pages 127–136, 2011.

[12] Felix Hausdorff. *Grundzuge der mengenlehre*, volume 61. American Mathematical Society, 1978.

[13] Lakshitha Dantanarayana. *Navigation and control for assistive robotics*. PhD thesis, 2016.

[14] Danielsson Per-Erik. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3):227–248, 1980.

[15] Azriel Rosenfeld and John L Pfaltz. Distance functions on digital pictures. *Pattern recognition*, 1(1):33–61, 1968.

[16] Danielson Per-Erik. A new shape factor. *Computer Graphics and Image Processing*, 7(2):292–299, 1978.

[17] Yen-Hsi Richard Tsai. Rapid and accurate computation of the distance function using grids. *Journal of Computational Physics*, 178(1):175–195, 2002.

[18] Hossam Abd El Munim and Aly A Farag. Shape representation and registration using vector distance functions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[19] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. In *Computer Graphics Forum*, volume 29, pages 1733–1741. Wiley Online Library, 2010.

[20] Lakshitha Dantanarayana, Gamini Dissanayake, and Ravindra Ranasinge. C-log: A chamfer distance based algorithm for localisation in occupancy grid-maps. *CAAI transactions on intelligence technology*, 1(3):272–284, 2016.

[21] F Donoso-Aguirre, J-P Bustos-Salas, Miguel Torres-Torriti, and Andres Guesalaga. Mobile robot localization using the hausdorff distance. *Robotica*, 26(2):129–141, 2008.

[22] Miguel Torres-Torriti and Andres Guesalaga. Scan-to-map matching using the hausdorff distance for robust mobile robot localization. In *2008 IEEE International Conference on Robotics and Automation*, pages 455–460. IEEE, 2008.

[23] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[24] Hugh Durrant-Whyte. Multi sensor data fusion. Technical report, 2001. URL `http://www.mit.bme.hu/eng/system/files/oktatas/targyak/7357/MultiSensorDataFusionNotes_DurrantWhyte_2011.pdf`.

[25] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[26] Michail Tsagris, Christina Beneki, and Hossein Hassani. On the folded normal distribution. *Mathematics*, 2(1):12–28, 2014.

[27] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching.

Technical report, Sri International, Menlo Park, California, Artificial Intelligence Centre, 1977.

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

[29] Bradford W Parkinson. The global positioning system (navstar). *Bulletin Géodésique*, 53(2):89–108, 1979.

[30] Han-Shue Tan and Jihua Huang. Dgps-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):415–428, 2006.

[31] Goran M Djuknic and Robert E Richton. Geolocation and assisted gps. *Computer*, 34(2):123–125, 2001.

[32] Richard B Langley. Rtk gps. *GPS World*, 9(9):70–76, 1998.

[33] Christian Weiss, Andreas Masselli, Hashem Tamimi, and Andreas Zell. Fast outdoor robot localization using integral invariants. In *Proceedings. International Conference on Computer Vision Systems*, 2007.

[34] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation*, pages 1643–1649. IEEE, 2012.

[35] Shuji Oishi, Yohei Inoue, Jun Miura, and Shota Tanaka. Seqslam++: View-based robot localization and navigation. *Robotics and Autonomous Systems*, 112:13–21, 2019.

[36] David G Lowe et al. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[37] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[38] Pubudu N Pathirana, Andrey V Savkinb, Samitha W Ekanayake, and Nicholas J Bauer. A robust solution to the stereo-vision-based simultaneous localization and mapping problem with steady and moving landmarks. *Advanced Robotics*, 25(6-7): 765–788, 2011.

[39] Taihú Pire, Thomas Fischer, Javier Civera, Pablo De Cristóforis, and Julio Jacobo Berlles. Stereo parallel tracking and mapping for robot localization. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1373–1378. IEEE, 2015.

[40] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[41] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1935–1942. IEEE, 2015.

[42] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31 (5):1147–1163, 2015.

[43] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[44] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[45] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.

[46] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR 2004)*, volume 1, pages I–I. Ieee, 2004.

[47] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1456, 2013.

[48] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[49] Juan Jose Tarrio and Sol Pedre. Realtime edge-based visual odometry for a monocular camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 702–710, 2015.

[50] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[51] Andrew Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3946–3952. IEEE, 2008.

[52] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *2010 IEEE Intelligent Vehicles Symposium*, pages 486–492. IEEE, 2010.

[53] Igor Cvišić and Ivan Petrović. Stereo odometry based on careful feature selection and tracking. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015.

[54] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 4007–4012. IEEE, 2004.

[55] Peidong Liu, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. Direct visual odometry for a fisheye-stereo camera. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1746–1752. IEEE, 2017.

[56] Mingyang Li and Anastasios I Mourikis. Improving the accuracy of ekf-based visual-inertial odometry. In *2012 IEEE International Conference on Robotics and Automation*, pages 828–835. IEEE, 2012.

[57] Mingyang Li and Anastasios I Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[58] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304. IEEE, 2015.

[59] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[60] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016.

[61] Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Direct visual-inertial odometry with stereo cameras. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1885–1892. IEEE, 2016.

[62] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[63] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. *arXiv preprint arXiv:1910.02490*, 2019.

[64] Luka Teslić, Igor Škrjanc, and Gregor Klančar. Ekf-based localization of a wheeled mobile robot in structured environments. *Journal of Intelligent & Robotic Systems*, 62(2):187–203, 2011.

[65] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[66] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[67] Amcl, Nov 2019. URL `http://wiki.ros.org/amcl`.

[68] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2432–2437. IEEE, 2005.

[69] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016.

[70] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[71] Dirk Haehnel, Sebastian Thrun, and Wolfram Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI*, volume 3, pages 915–920, 2003.

[72] Hara Yoshitaka, Kawata Hirohiko, Ohya Akihisa, and Yuta Shin'ichi. Mobile robot localization and mapping by scan matching using laser reflection intensity of the sokuiki sensor. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 3018–3023. IEEE, 2006.

[73] Simona Nobili, Raluca Scona, Marco Caravagna, and Maurice Fallon. Overlap-based icp tuning for robust localization of a humanoid robot. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4721–4728. IEEE, 2017.

[74] Vicon Motion Systems Ltd UK. Tracker, Nov 2019. URL `https://www.vicon.com/software/tracker/`.

[75] Inc Natural Point. Motive optical motion capture software, Nov 2019. URL `https://optitrack.com/products/motive/`.

[76] Davide Dardari, Andrea Conti, Jaime Lien, and Moe Z Win. The effect of cooperation on uwb-based positioning systems using experimental data. *EURASIP Journal on Advances in Signal Processing*, 2008:124, 2008.

[77] Andrea Conti, Davide Dardari, and Moe Z Win. Experimental results on cooperative uwb based positioning systems. In *2008 IEEE International Conference on Ultra-Wideband*, volume 1, pages 191–195. IEEE, 2008.

[78] Faheem Zafari, Athanasios Gkelias, and Kin K Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 2019.

[79] Wondimu K Zegeye, Seifemichael B Amsalu, Yacob Astatke, and Farzad Moazzami. Wifi rss fingerprinting indoor localization for mobile devices. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1–6. IEEE, 2016.

[80] Asok Perera, Janindu Arukgoda, Ravindra Ranasinghe, and Gamini Dissanayake. Localization system for carers to track elderly people in visits to a crowded shopping mall. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2017.

[81] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, pages 483–494. ACM, 2014.

[82] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 269–282. ACM, 2015.

[83] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 01 1965. ISSN 0010-4620. doi: 10.1093/comjnl/7.4.308.

[84] Mary Ann Branch, Thomas F Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999.

[85] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[86] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2): 431–441, 1963.

[87] John C Clarke. Modelling uncertainty: A primer. *Tutorial of Department of Eng. Science*, pages 1–21, 1998.

[88] 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, 2015.

[89] ROS.org. stage, Nov 2019. URL `http://wiki.ros.org/stage`.

[90] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[91] Maleen Jayasuriya, Gamini Dissanayake, Ravindra Ranasinghe, and Nathanael Gandhi. Leveraging deep learning based object detection for localising autonomous personal mobility devices in sparse maps. In *2019 Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019.

[92] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[93] Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *International Joint Conferences on Artificial Intelligence Organization*, pages 1346–1352, 1997.

[94] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.

[95] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[96] Josep M Porta, Bas Terwijn, and B Krose. Efficient entropy-based action selection for appearance-based robot localization. In *2003 IEEE International Conference on Robotics and Automation*, volume 2, pages 2842–2847. IEEE, 2003.

[97] Stefan Czarnetzki, Sören Kerner, and Michael Kruse. Real-time active vision by entropy minimization applied to localization. In *Robot Soccer World Cup*, pages 266–277. Springer, 2010.

[98] Tal Arbel and Frank P Ferrie. Viewpoint selection by navigation through entropy maps. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 248–254. IEEE, 1999.

[99] Devendra Singh Chaplot, Emilio Parisotto, and Ruslan Salakhutdinov. Active neural localization. *arXiv preprint arXiv:1801.08214*, 2018.

[100] Sai Krishna Gottipati, Keehong Seo, Dhaivat Bhatt, Vincent Mai, Krishna Murthy, and Liam Paull. Deep active localization. *IEEE Robotics and Automation Letters*, 4(4):4394–4401, 2019.

[101] Patric Jensfelt and Steen Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760, 2001.

[102] J. M. Manyika and H. F. Durrant-Whyte. A tracking sonar sensor for vehicle guidance. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 424–429 vol.3, May 1993. doi: 10.1109/ROBOT.1993.292209.

[103] Andrew J Davison and David W Murray. Mobile robot localisation using active vision. In *European Conference on Computer Vision*, pages 809–825. Springer, 1998.

[104] R. Valencia, J. Valls Mir, G. Dissanayake, and J. Andrade-Cetto. Active pose slam. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1885–1891, Oct 2012. doi: 10.1109/IROS.2012.6385637.

[105] Ruijie He, Sam Prentice, and Nicholas Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *2008 IEEE International Conference on Robotics and Automation*, pages 1814–1820. IEEE, 2008.

[106] Henry Carrillo, Yasir Latif, José Neira, and José A Castellanos. Fast minimum uncertainty search on a graph map representation. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2504–2511. IEEE, 2012.

[107] Henry Carrillo, Philip Dames, Vijay Kumar, and José A Castellanos. Autonomous robotic exploration using occupancy grid maps and graph slam based on shannon and rényi entropy. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 487–494. IEEE, 2015.

[108] Rafael Valencia, Juan Andrade-Cetto, and Josep M Porta. Path planning in belief space with pose slam. In *2011 IEEE International Conference on Robotics and Automation*.

[109] Sean Mauch. A fast algorithm for computing the closest point and distance transform. Technical report, Caltech, 2000.

[110] G Dissanayake and N Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10(3):195–201, 1994.