UNIVERSITY
OF TECHNOLOGY
SYDNEY

Efficient Behaviour based Information Driven Human Tracking System for Long Term Occlusion Recovery

by Zulkarnain Bin Zainudin

Thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Professor Sarath Kodagoda
and Emeritus Professor Gamini Dissanayake

University of Technology Sydney
The Faculty of Engineering and Information Technology

March 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Zulkarnain Bin Zainudin*, 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: 04/03/2024

Acknowledgements

I would like to express my deepest appreciation and gratitude to all the people who have helped and supported me during the past four years of my candidature.

First, I would like to express my sincere and heartfelt thanks to my two supervisors. To Professor Sarath Kodagoda, thank you so much for all your precious guidance and patience, setting aside many hours of your invaluable time, and allowing me to have many meaningful and significant discussion sessions from the beginning till the end of my research. To Emeritus Professor Gamini Dissanayake, my deepest appreciation to you for the hours of collaboration and insightful ideas towards possible research output. I would like to thank my colleague, Leo, for all your valuable contributions and wise thoughts during the various discussions we held together.

To my many colleagues at UTS-CAS, my corner fellows, Jason, Stephen, Kasra, Marc, Mitesh, and Mohamad, thank you. Research would have been much tougher and more challenging without the constant discussions and sharing of light moments when the going was tough.

To my beloved wife, Syahida Binti Mohtar, thank you for your undivided support and the belief in me to complete this new milestone in my life. To my children, Intan Humaira', Muhammad Zaki, and Intan Ruqayyah, you are the jewels in my heart.

Finally, I would like to thank my late father, Zainudin, and my mother, Ramlah, for their constant support. I hope that I have made you proud, and as a son, I am so privileged to have parents like you. To my parents-in-law, Mohtar and Siti Fatimah, thank you for their confidence in me, the support, and the motivation. To my late brother, Iskandar Mirza, and sisters, Zarinah, Zanariah, and Zuraidah, I hope my success will motivate you at least to follow in my footsteps, if not better.

Contents

List of Tables

List of Figures

Nomenclature

Formatting Style

Formatting	Description
$[\cdots]^T$	Vector or matrix transpose
$\lvert\cdot\rvert$	Absolute value

Subscript and Numbering

Subscript	Description
X_i	ith element in vector
X_j	jth element in vector

Symbol Usage

Symbol	Description	Units
α_T	Probability of false-track confirmation	none
β_T	Probability of true-track confirmation	none
P_T	Probability of a true person	none

Abbreviations

2D	Two Dimensional
ARMSE	Average Root Mean Square Error
EKF	Extended Kalman Filter
EM	Expectation-Maximization
GNN	Generalized Nearest Neighbour Filter
GP	Gaussian Process
GP-PF	Gaussian Process - Particle Filter
GP-EKF	Gaussian Process - Extended Kalman Filter
HMM	Hidden Markov Model
HRI	Human Robot Interaction
ICP	Iterative Closest Point
IMM	Interactive Multiple Model
IMMPDAF	Interacting Multiple Model Probabilistic Data Association Filter
JPDAF	Joint Probabilistic Data Association Filter

LiDAR	Light Detection and Ranging
LLR	Log-likelihood Ratio
LRF	Laser Range Finder
MD	Mahalanobis Distance
MI	Mutual Information
MIA	Mutual Information Approach
MMPDAF	Multiple Model Probabilistic Data Association Filter
NEES	Normalized Estimation Error Squared
NIS	Normalized Innovation Squared
NNSF	Nearest Neighbour Standard Filter
PDAF	Probabilistic Data Association Filter
PF	Particle Filter
RBF	Radial Basis Function
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROI	Region of Interest
SNSF	Strongest Neighbour Standard Filter
SVM	Support Vector Machine

Abstract

Efficient Behaviour based Information Driven Human Tracking System for Long Term Occlusion Recovery

Comprehending human motion patterns is pivotal for the development of tools designed to detect and track individuals. This thesis has devised a methodology for identifying people by analysing their torso height and employing classification algorithms. The Support Vector Machine (SVM) was chosen as the binary classifier, with four features assigned to it. Following an extensive experimental assessment of various classification methods, the SVM emerged as the preferred classifier due to its superior performance. The efficiency of the people tracking technique employing the Interacting-Multiple-Model Probabilistic Data Association Filter (IMMPDAF) was evaluated using both simulated and experimental data. The assessment relied on metrics such as Normalised Estimation Error Squared (NEES) and Normalised Innovation Squared (NIS). While the IMMPDAF exhibited robustness and consistency, it faced challenges with targets experiencing prolonged occlusion, thereby diminishing temporal prediction accuracy. Thus, researchers came up with Gaussian Processes (GP) to make tracking more accurate during long occlusions.

Experiments showed that the Gaussian Process-Particle Filter (GP-PF) was better at predicting the future in terms of time. Incrementally adding training data samples, despite their accuracy, resulted in an observed increase in the computational load.

A sample data management system was established to retain valuable data while discarding less informative data, utilizing techniques based on mutual information (MI) and Mahalanobis distance (MD). This approach significantly reduced sampling data while adhering to the average RMS error (ARMSE) limit.

The focus of this research was on observing people in indoor environments using a 2D laser range finder (LRF) or laser detection and ranging (LiDAR) as a sensor. The initial objective was to identify a suitable detection classification method and integrate it with the tracking technique. Subsequently, a specialised algorithm was developed to enhance temporal tracking capabilities, particularly in situations involving occlusions and partially absent observation data.

An effective technique for detecting and tracking people based on specified features was formulated in this thesis, integrating a learning algorithm with a tracking algorithm. The detection and tracking approaches employed parametric and non-parametric regression models along with learning algorithms. Using laser measurements to sort certain properties into groups using different types of classifiers made it easier to compare learning algorithms, and the confusion matrix helped find the best way to choose the suitable detection algorithm. The robustness and consistency of track generation and termination, based on the log-likelihood ratio (LLR) in conjunction with the Interacting Multiple Model Data Association Filter (IMMPDAF), were scrutinized. These investigations resulted in the development of Gaussian Process-Bayes Filters, which showcased proficiency in long-term occlusion tracking. Additionally, novel training data management approaches were established to minimize the number of samples required for training without compromising the tracker's effectiveness.

Chapter 1

Introduction

1.1 Problem Statement

The requirement for effective and precise monitoring systems in interior settings has become critical in today's world of rapid evolution. For several reasons, including security, safety, and resource optimisation, strong technologies to detect and monitor individuals are needed in indoor locations like malls, airports, offices, and public buildings. When dealing with busy or complicated indoor environments, traditional monitoring techniques sometimes fall short. Therefore, there is an urgent need for cutting-edge technology that can detect and track people in interior settings.

The successful development of a solution to this problem will result in an intelligent system capable of accurately detecting and tracking people in indoor environments. The system should offer real-time processing, adaptability to different settings, privacy preservation, and robustness against occlusions. By achieving these outcomes, the solution will enhance security measures, improve public safety, optimise resource management, and contribute to the overall efficiency of indoor spaces. The main goal of this thesis is to develop algorithms to detect and track people in natural indoor environments. In particular, the focus is on analysing the signals of stationary and moving objects from sensors observing complex environments.

1.2 Motivation

The ability to detect and track human motion is a useful tool for advanced robotic applications that rely on objects and visual sensors. Understanding human nature in motion and their interactions always forms a core of interest in intelligent systems such as automated surveillance systems, human-robot interactions and pedestrian detection in autonomous motor vehicles. Detection and tracking of human motion with occlusions or without observation information over a significantly long period of time is of immense interest in human-robotic interaction due to its implications for human safety. In an environment where robots and humans are in motion, robots must be able to identify and track human positions and travel on command while avoiding obstacles. However, when the observations are temporally occluded with any object, the tracking system of the robot fails to properly identify and track the targets. This is due to the nature of the parametric model of the tracker, where the track cannot be re-associated with the target for further tracking. Due to this limitation, a non-parametric model could be an appropriate answer to learn prediction and observation models for dynamical systems. One of the techniques that can be used for modelling is the Gaussian Process regression model. It provides uncertainty estimates for their predictions, which can be incorporated into trackers.

Object detection and tracking can be a time-consuming process due to the accumulative amount of data that is necessary to provide training data for the learning algorithm to learn those patterns. The increasing amount of accumulative data then leads to the process of learning the patterns becoming computationally expensive. Thus, the reduction of training data will alleviate the time-consuming computational process of learning predictions and observations. However, the information data needs to be appropriately managed while disregarding the non-informative data.

In this research, the scope of the study is detecting and tracking people in indoor environments where a static observation device known as a laser range finder (LRF) is positioned at torso height in a common area at the Centre of Autonomous Systems, University of Technology Sydney. Some of the areas are divided into cubicles, as they were being used as workstations for the researchers in the centre. In this area, there are many possibilities of obstacles and objects that might temporarily block the observation of the laser range finder towards detecting and tracking people on the walking paths between the cubicles. Thus, the temporal disappearance of observation data will lead to the tracker's failure to track moving people. In order to overcome this problem, a solution like a non-parametric model where uncertainty estimates for their predictions can be incorporated into trackers needs to be implemented.

1.3 Principal Contributions

This work addresses the problem of learning and utilising motion patterns in people detection and tracking in human-populated environments. The thesis explores approaches for exemplifying and learning human motion patterns. The principle contributions of this thesis arise from the development of a new methodology on temporal tracking that leads to the selection of informative data for a non-parametric estimation model of Gaussian Process-Particle Filter (GP-PF) to represent human motion patterns, which is suitable for online learning and can be deployed on a mobile robot that is fitted with sensors. The resulting non-parametric estimation model is implemented in people tracking to improve performance and accuracy.

The main contributions are:

- A novel approach for the establishment of selected features and parameters in a learning algorithm to represent people at the detection level is developed from sample-based representation. Among those features, segmented data fitted with ellipses has significantly contributed to eliminating unwanted segmented data that does not represent the torso cross section. The approach contains analyses on the selection of the most suitable learning algorithm, where eventually Support Vector Machine (SVM) is chosen.
- The application of SVM in conjunction with Extended Kalman Filter (EKF)

based people tracking is presented, where significant improvements are achieved. Among those are improved detection accuracy to lower the prediction task and increase the robustness of people tracking. This approach, which is associated with the Interacting Multiple Model (IMM) based Probabilistic Data Association Filter (PDAF), has been analysed using Normalized Estimation Error Squared (NEES) and Normalized Innovation Squared (NIS).

- Enhanced people tracking using Gaussian Processes-Particle Filter (GP-PF) which effectively learns human motion patterns, is presented to improve tracking performance, especially with long-term occlusions. It is shown that GP with a particle filter's ability to approximate multivariate posterior distributions is able to predict tracking even with long-term occlusions.
- Real-world data from experiments is presented to validate the approach and the applications of Gaussian Processes Particle Filter using Laser Range Finder (LRF) in an office-like environment.
- A novel approach to selecting and keeping the most informative data while discarding the least informative data was implemented with the implementation of the Information (MI) based technique along with the Mahalanobis Distance (MD). It significantly optimises the amount of training data that is necessary for Gaussian Processes to enhance their tracking performance.

1.4 Publications

Following is a list of publications from the work presented in this thesis:

- "Gaussian Processes-BayesFilters with Non-Parametric Data Optimization for Efficient 2D LiDAR Based People Tracking" Zainudin, Z. and Kodagoda, S., *International Journal of Robotics and Control Systems*, Vol. 3, No. 2, 2023.
- "Monte Carlo Simulations on 2D LRF Based People Tracking using Interactive Multiple Model Probabilistic Data Association Filter Tracker" Zainudin, Z. and

Kodagoda, S., *International Journal of Robotics and Control Systems*, Vol. 3, No. 1, 2023.

- "Non-Parametric Data Optimization for 2D Laser Based People Tracking" Zainudin, Z., Mat Ibrahim, M. and Kodagoda, S., *Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA 2017)*, Siem Reap, Cambodia, Jun. 2017.
- "Mutual Information Based Data Selection in Gaussian Processes for 2D Laser Range Finder Based People Tracking" Zainudin, Z., Kodagoda, S. and Dissanayake, G., *Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2013)*, Wollongong, Australia, Jul. 2013.
- "Mutual Information Based Data Selection in Gaussian Processes for People Tracking," Zainudin, Z., Kodagoda, S. and Van Nguyen, L., *Proceedings of the Australasian Conference on Robotics and Automation 2012 (ACRA 2012)*, Auckland, New Zealand, Dec. 2012.
- "Torso Detection and Tracking using a 2D Laser Range Finder," Zainudin, Z., Kodagoda, S. and Dissanayake, G., *Proceedings of the Australasian Conference on Robotics and Automation 2010 (ACRA 2010)*, Brisbane, Australia, Dec. 2010.

1.5 Thesis Overview

The remainder of this document is organised as follows:

Chapter 2 discusses an algorithm for people detection based on a two-dimensional laser range finder with the implementation of Support Vector Machines (SVM) as binary learning classifiers to classify a person or others.

Chapter 3 addresses the integration of people detection and tracking and the introduction of the Interacting Multiple Model Probabilistic Data Association filter (IMM-

PDAF) for people tracking with the tracking-by-detection technique. The performance of the tracker has been analysed using Normalized Estimation Error Squared (NEES) and Normalized Innovation Squared (NIS).

Chapter 4 presents a model to describe human motion patterns to enhance tracking performance with an application of Gaussian Processes. Theoretical justification and improvements to existing techniques are analysed in detail.

Chapter 5 concludes the work in this thesis with a summary of the outcomes presented and an outlook for future related research.

Chapter 2

People Detection

Detecting people in populated environments is an important tool in various machine vision applications, including robotics, health care, automotive, security, and defence. In this work, an algorithm for people detection has been introduced based on observations from a two dimensional (2D) laser range finder (LRF) or light detection and ranging (LiDAR). The LRF that was used for this experiment was mounted on a mobile robotic platform to scan the torso section of a person. Support Vector Machines (SVM) had been selected as binary learning classifiers to classify people and others based on analyses.

2.1 Introduction

This chapter explores the concept of people detection techniques and algorithms. Human Robot Interaction (HRI) has rapidly become an emerging area of research in recent years. Robots are gradually emerging as helpers, carers, security officers, and entertainers in today's life. Therefore, towards the realisation of these dreams, people detection and tracking can play a vital role in these situations.

It is impossible to overestimate the significance of people detection and tracking. Understanding people's movement patterns is essential for security officers in busy indoor areas like shopping centres, airports, and office buildings. Advanced detection techniques can be used to spot prospective dangers, keep an eye on suspicious activity, and act quickly in an emergency. Additionally, these devices significantly contribute to increased public safety by facilitating effective crowd control and quick emergency evacuation processes.

These technologies play a critical role in resource utilisation optimisation in addition to security. Tracking customer movements, for instance, in retail settings offers priceless information about consumer behaviour. This information can be used to create store layouts that improve customer satisfaction, increase sales, and simplify inventory control. Similar to how in home environments, tracking employee movements can result in the development of smarter workspaces where resources like lighting and climate control are optimised depending on occupancy patterns, helping with energy-saving efforts.

2.2 Related Works

In general, the people detection problem is handled by utilising classification algorithms. Thus, the selection of a classifier is an important stage in the detection process. Commonly used classifiers in people detection based on LIDAR or LRF are Support Vector Machines and AdaBoost. Arras et al. [2] used the AdaBoost algorithm with 14 features that were based on the characteristics of a laser segment. Further, Spinello et al. [3] applied Support Vector Machines classifier to 2D laser data and vision data. In this thesis, it compares the performance of different numbers of classifiers for a given set of features in order to choose the best classifier.

2.2.1 Sensors

Many researchers have been involved in developing algorithms utilising diverse sensor modalities with various levels of success. Sensors are key to various purposes of detection and tracking objects. Cameras and light detection and ranging (LIDAR) [4; 5] are commonly exploited sensors in those applications due to their portability features.

Camera-based sensors offer several advantages in their ability to detect and track objects since they can provide rich and detailed data, capturing not only spatial information but also colour, texture, and patterns, which makes them suitable for applications where visual information is crucial, such as computer vision and image recognition. Some advanced camera systems, such as those using stereo vision or depth-sensing technologies, can provide depth perception, which is crucial for applications like 3D mapping and robotics where understanding the distance to objects is essential [6]. However, they have a limited range of depths.

Camera-based sensors rely heavily on lighting conditions, and their performance can be affected by changes in ambient light. Low-light or high-glare environments can lead to degraded image quality and effectiveness. The effective range of camera-based sensors is limited by the lens and sensor specifications. In certain applications, such as medium-long-range monitoring and surveillance in poorly lit expansive indoor or outdoor areas, alternative sensor technologies like LiDAR may be more suitable. Camera-based sensors also raise privacy concerns, especially in public spaces where the constant monitoring and recording of visual data can infringe on individuals' privacy rights, leading to ethical and legal considerations [7].

Stereo cameras with depth perception, such as the Intel RealSense SR305, D415, and L515, use different algorithms to calculate depth. The SR305 employs coded light, wherein a predetermined pattern is projected onto the scene. By analysing the deformation of this pattern, the device calculates depth information. The D415 employs stereo vision technology, which involves capturing the scene using two imagers and calculating the difference between the two images to determine depth. The L515 sensor utilises time-of-flight technology to estimate depth by precisely calculating the delay between light emission and light reception. However, they have high depth errors at distances greater than 9 meters [8].

Thus, since LiDAR sensors can precisely measure distances by calculating the time it takes for laser pulses to travel to an object and return, which enables accurate ranging and distance measurements, they can be used as an observation device in this work.

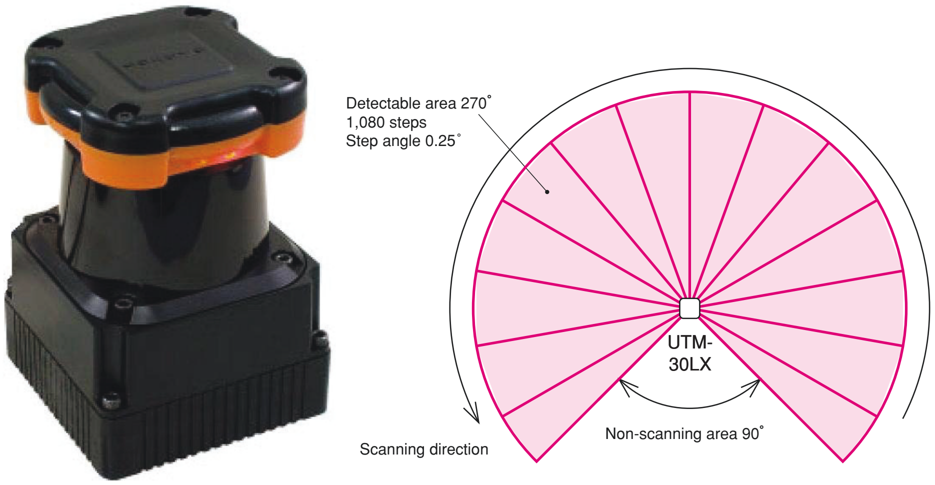
LiDAR sensors operate independently of ambient light conditions. They can function effectively in various lighting conditions, including complete darkness. LiDAR sensors also have a wide field of view, usually more than 180° which allows them to cover large areas in a single scan.

Selecting appropriate sensors is a fundamental prerequisite for achieving desired outcomes, as making a poor choice can lead to suboptimal system performance and increased expenses. The sensors must possess the capability to detect and identify all targets within a specified range with a satisfactory level of precision while also maintaining operational functionality for an extended period of time. For example, sensors that can detect objects within a range of less than 5 metres are quite affordable, whereas sensors with a larger detection range of up to 30 metres are more costly. Therefore, the selection of suitable sensors for particular applications is crucial for achieving the best possible performance and cost-effectiveness.

In recent studies, various types of LIDARs or LRFs were used in high performance applications such as 3D or multichannel LIDAR by many researchers[9–13]. However, these kinds of LIDARs are quite expensive since they are mainly used for autonomous vehicle applications [11]. Thus, 2D LIDAR or LRF is less expensive than 3D LIDAR and is preferred for use in domestic applications [4; 14–17].

There are several techniques that have been proposed for detecting people with laser range finders, such as motion-based, feature-based and heuristic approaches, as given in [2; 18–21]. In general, motion-based detection can have limitations due to some stationary people in the vicinity, for which temporal-difference features for multi-frames are not available [22]. Feature-based people detection in the literature uses single-layered or triple-layered approaches, which may detect legs, upper body, and head [2; 20; 21; 23] using laser range finders (LRFs). Leg detection is an appealing approach; however, it may lead to complex algorithms due to leg movements and the attire. Meanwhile, the triple layer approach has a higher computational cost for data synchronisation for three LRF devices. It is the belief that a single LRF could still be exploited as a cost effective solution to detect and track people. In this work, it was proposed to detect and track people at their torso height. Torso height generally has a cross section of an ellipse and a reasonably consistent shape formation. It also does not have complex dynamic movements such as legs and can be classified into standard torso categories [24] which means a template matching (as in computer vision) approach can be used.

Therefore, it was proposed in this research to use a laser range finder with an effective range of up to 30 meters which would be sufficient to scan a specific area of interest. As shown in Figure 2.1, the Hokuyo UTM-30LX Scanning Laser Range Finder has a range measurement distance of 30 metres and an angle field of view of 270 °.



(a) Hokuyo UTM-30LX

(b) Hokuyo Detection Area

Figure 2.1: Scanning Laser Range Finder

2.2.2 People Detection Techniques

People detection techniques using laser range finders (LRFs) can be divided into three main categories of observation: single-layered, double-layered, and triple-layered 2D range and bearing data.

Single-Layered 2D Range Data

People detection using leg motions with single-layered 2D range data with boosted features is erroneous and prone to error due to the complexity of the formation of instantaneous patterns on the legs of people. In early work, people detection is typically determined by subtracting two subsequent scans, and this technique can only identify moving people [25–27].

Later, Topp et al. [28] improvised the method that had been introduced by Schulz et al. [27] in order to detect stationary people. However, it had problems with the detection of multiple people in a cluttered environment. Arras et al. [2] introduced 14 features with the AdaBoost algorithm, which creates an accurate strong classifier with a combination of a set of weak classifiers.

Double-Layered 2D Range Data

Carballo et al. [23] introduced double-layered 2D Range data, which simultaneously implemented two laser range finder (LRF) at the legs and torso of the human body. This approach was adapted from Hashimoto's work [29] where sensors were set up in two parallel planes at different heights from the ground depending on the selected features. The sensors were set up at leg height and torso altitude. The fusion of sensors needs to deal with data duplication since each sensor has 270°angle of view. In the double layer fusion step, raw data from each layer is processed to extract features of people, and subsequently, people can be detected and tracked. However, synchronisation of data between each layer and fusion between sensors on data duplication have contributed to misclassification between legs and measurement noise, and at the torso part, measurement may contain data of people's hands, which causes occlusion [30].

Triple-Layered 2D Range Data

In 2009, Mozos et al. [21; 30] introduced a triple-layered laser range finder that simultaneously detected the heads, upper bodies, and legs of people with one classifier for each layer. With multiple layers, in comparison to a single layer, the detection rate is almost double the detection capabilities in a very cluttered environment. In this technique, the sensors are placed 160 cm above the floor for head detection, 140 cm above the floor for torso or chest detection, and 30 cm above the floor to detect legs. Even though multi-layer detection rates are higher than single-layer detection rates, this technique is highly dependent on the correct alignment of multiple LRFs, which could cause system failure if the misalignment is significantly large [30].

2.2.3 Torso Detection

The geometrical shape of human legs, hands, torso, and head are anatomical structures of the human body that can be seen by naked eyes and sensors. There are many approaches to the physical detection of humans using laser range finders, such as leg detection [31] [2], chest or upper body detection [32] and head detection [21].

Arras et al. [2] introduced detection on the legs of people with the stationary laser range finder mounted 30 cm above the floor in a corridor and an office environment. Zivkovic et al. [31] implemented legs detection using a laser range finder that was mounted 50 cm above the floor for corridors and cluttered offices. Carballo et al. [32] proposed detection on upper bodies and legs using a laser range finder that was mounted at 110 cm and 40 cm, respectively, from the ground.

In all sensor placement and height, the torso or upper body height is suitable for human detection since it has a reasonably consistent shape. In this research, torso detection has been chosen due to this reason.

2.3 Training and Detection Processes

In the data training process, there are three stages: feature extraction from segmented laser data, binary data labelling with people and others, and training with SVM data classifiers.

In the detection of people, there are three stages: feature extraction from segmented laser data, applying SVM for data classification, and people detection. The processes are shown in Figure 2.2.

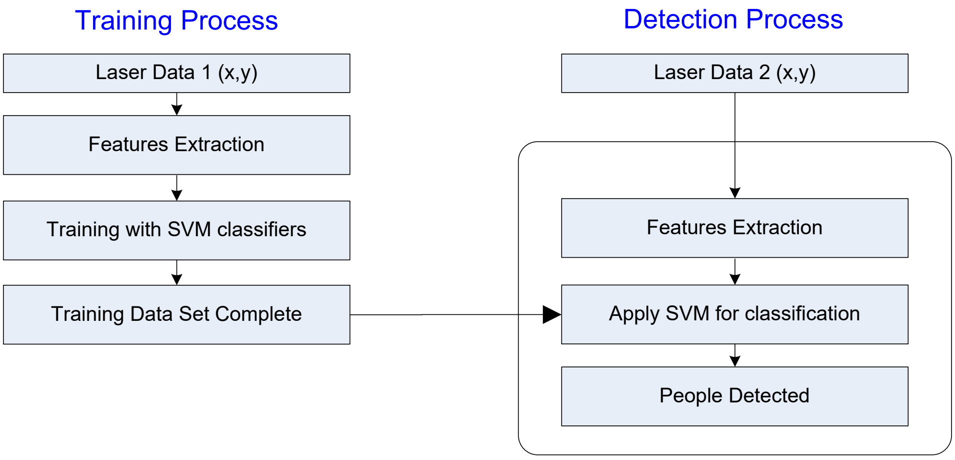


Figure 2.2: Training and Detection Process Flow

2.4 People Detection

Sequential processing steps of laser range/ bearing data are done based on detection range discontinuities in the laser scan.

2.4.1 Features Selection

The first processing step of laser range/bearing data based people detection is data segmentation. This is based on detecting range discontinuities in the laser scan. The laser range finder provides range and bearing, $\{r_i, \theta_i\}$ to objects in its field of view, where suffix i refers to a specific range/bearing data with $i = 1, \ldots, n$. By using a model based technique, which is realised using the Extended Kalman Filter (EKF) [33; 34], it is possible to partition the data into segments $S = \{s_1, s_2, \ldots, s_M\}$ as shown in Figure 2.3. M is the number of segmentations that are calculated by EKF. In Figure 2.3, symbol $'o'$ refers to discontinuity points, which define the start and end points of segments.

Once the data segmentation is performed, the next step is to extract meaningful features to be used in people detection as well as classification. The ellipsoidal torso shape, as observed by LRF (Laser Range Finder) or LiDAR (Light Detection and Ranging), can be analyzed to identify and select four primary features. When considering the ellipsoidal torso shape, the selected four primary features are based on cross-sectional length, major and minor axes length, surface curvature information, and intensity of data points.

Cross-sectional length is essential to differentiate the detection objects as people or others, such as chairs, walls, tables, and so on. Major and minor axes lengths represent the longest and shortest dimensions, respectively. These lengths provide essential information about the size and orientation of the torso. Surface curvature information gives insights into the shape and contours of the torso, which is valuable for understanding the overall geometry and structure of the object. Intensity data points are provided by the distance between the target and the observation point, where the longer distance will yield fewer data points. Compared to Arras et al. (2007), who used the AdaBoost algorithm to detect people on the legs using 14 features, our approach uses the Support Vector Machine technique to detect people on the torso using only 4 selected features. This results in reduced computing time and effort. The details of four features are listed below.

Feature 1: The cross-sectional length of the adult torso is one of the features that can represent characteristics of an adult human being. Length of a segment, l_s which is given by

$$l_s = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2}. \tag{2.1}$$

Feature 2: The ratio of the major to minor axes of the ellipse can represent the ellipsoidal like shape of the human torso. The laser range finder is mounted in such a way that it scans the torso of an average person. The cross-section of the torso of a human can generally be approximated by an ellipse. Therefore, an ellipse fitting algorithm [35] is implemented on segmented laser range data. The Cartesian coordinates of each element in an i^{th} segmented laser data, $s_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{in}\}$ can be transformed into a matrix, $\mathbf{D} = [x_{i1}, x_{i2}, ..., x_{in}]^T$. Then the solution for fitting ellipses is a general conic equation:

$$F(\mathbf{a}, \mathbf{x}) = \mathbf{a}.\mathbf{x} = ax^2 + bxy + cy^2 + dx + cy + f = 0 \tag{2.2}$$

$$\mathbf{Sa} = \lambda \mathbf{Ca} \tag{2.3}$$

$$\mathbf{a}^T \mathbf{Ca} = 1 \tag{2.4}$$

where, $\mathbf{a} = [a\ b\ c\ d\ e\ f]^T$, $\mathbf{x} = [x^2\ xy\ y^2\ x\ y\ 1]^T$, $\mathbf{S} = \mathbf{D}_i \mathbf{D}_i^T$,

and C is

$$\begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Here, the maximum and minimum values of λ, λ_{max} and λ_{min} define the length of the major and minor axes, respectively. The ellipses fitted for the segmented data in Figure 2.3 are shown in Figure 2.4. The features that are considered include the length of major and minor axes and the ratio of major and minor axes.

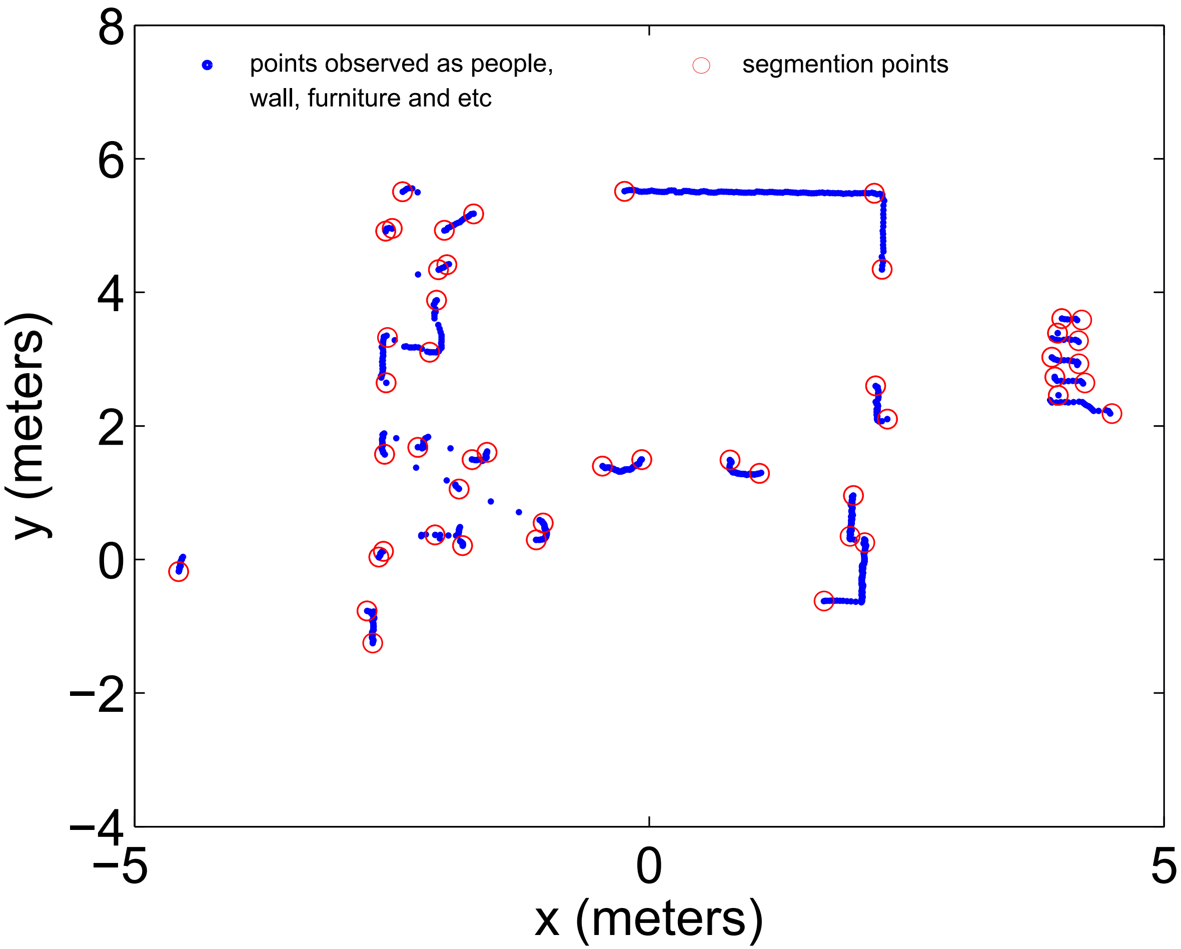


Figure 2.3: Segmentation of LRF data that represents people, walls, and furniture by EKF.

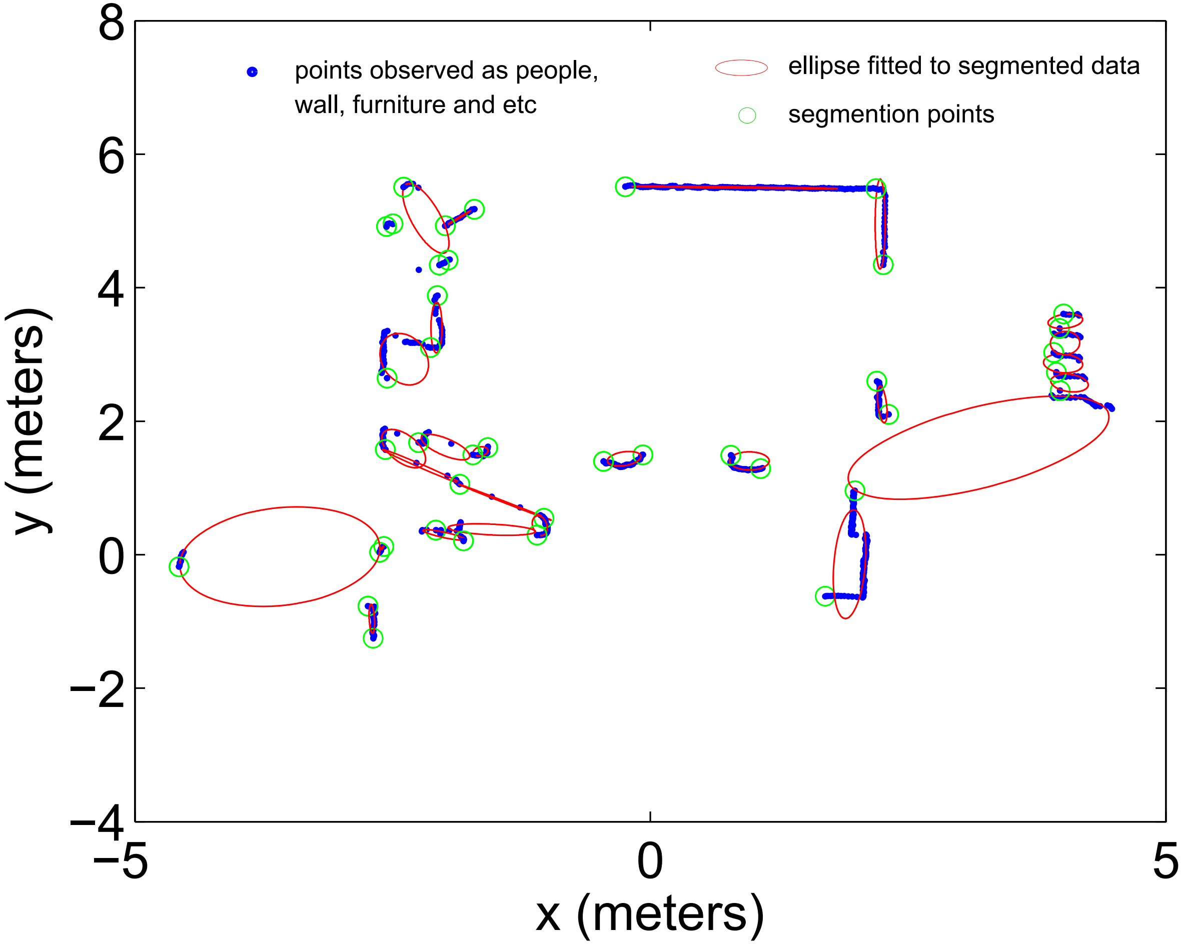


Figure 2.4: Segmented data represents people, walls and furniture fitted with ellipses.

Feature 3: The curvature of the torso of an adult human can be represented in terms of mean curvature. Mean curvature characteristic of a segment, S_i that is taken from the measurement. Given three sequential Cartesian coordinates, $\mathbf{x}_1$, $\mathbf{x}_p$ and $\mathbf{x}_n$, let $\mathbf{A}$ denote the area of the triangle enclosed by $\mathbf{x}_1\mathbf{x}_p\mathbf{x}_n$ and d_1, d_p and d_n denote the distance of three legs of the triangle. Then, an approximation of the discrete curvature of the boundary at $\mathbf{x}_p$ is given by [2],

$$\mathbf{x}_p = \frac{4A}{d_1 d_p d_n} \tag{2.5}$$

Feature 4: The number of points in the segmentation depends on the distance between the target (human) and the laser source. Therefore, the fourth feature is the ratio of the distance between the laser source and the centre of segmentation over a number of points, l_c which is given by,

$$l_c = \frac{\sqrt{x_C^2 + y_C^2}}{n} \tag{2.6}$$

where x_C, y_C and n are the centre points of x and y, and the number of points, respectively.

2.4.2 Classifier Selection

Once the features have been extracted, a classification routine is implemented. In order to compare the performance of different classifiers, Weka [36], a popular open source machine learning software was used. The data for these comparative analyses was captured using a Hokuyo laser range finder while people were freely wandering in an office like environment, as shown in Figure 2.8. The laser range finder was mounted to scan the torso of a person. Cross-validation is used as a resampling procedure to evaluate classifiers performance. The dataset is partitioned into two subsets as training and testing data, as 50% of the scans were used for training and the other 50% were used for testing. This process is repeated multiple times, and the performance metrics are averaged over the runs.

The evaluation results of different classifiers based on a confusion matrix generated from true detections. A confusion matrix is a table that is often used to evaluate the performance of a classification algorithm. It compares the predicted classifications of a model against the true classifications. It typically consists of four values: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The term true detections indicates that the classifiers correctly identified and classified the presence of people in the environment. True detections contribute to the true positive (TP) count in the confusion matrix.

Various classification algorithms or models have been employed for the tasks of people detection, such as Support Vector Machine, AdaBoost, Simple Logistic, Bayesian Networks, and others. The results of detection were compared in a scenario where there is only one person moving around the environment and more than one person, such as two or three people moving in the vicinity of LRF. This condition can be relevant in evaluating the classifiers' performance, especially in terms of their ability to correctly detect and classify instances in a less crowded or dynamic setting. The outcomes obtained from the various classifiers are being compared. It includes accuracy, precision, and recall derived from the confusion matrix. These metrics provide a quantitative measure of how each classifier performs in terms of true positive detections and avoiding false positives and false negatives.

Table 2.1 shows the results of different classifiers, which were extracted from the true detection of confusion matrix with few people wandering in the environment as shown in Figures 2.11 and 2.12. As expected, when there was only one person in the vicinity of the laser range finder, all classifiers were performing well. However, with more people, the classifiers tend to have poorer performances.

This could be mainly due to the differences in sizes, costumes, and artefacts due to occlusion. Out of the given classifiers, it could be seen that the Radial Basis Function Support Vector Machines (RBFSVM) performed better, and since it can also handle

Table 2.1: Comparison of classifiers

No. of Person	Classifier	Training Data Accuracy (%)	Testing Data Accuracy (%)
One	RBFSVM	98.60	96.86
	AdaBoostM1	96.58	92.59
	Simple Logistic	97.98	90.94
	MultiBoostAB	94.12	62.60
	BayesNet	98.60	95.68
	Complement Bayes Net	94.12	63.23
	Naive Bayes	98.51	98.06
	Naive Bayes Simple	98.51	98.06
	Naive Bayes Updateable	98.51	98.06
More than one	RBFSVM	96.71	95.22
	AdaBoostM1	96.82	64.30
	Simple Logistic	96.93	61.37
	MultiBoostAB	93.65	69.54
	BayesNet	97.46	64.84
	Complement Bayes Net	87.93	57.13
	Naive Bayes	96.51	75.63
	Naive Bayes Simple	96.40	70.70
	Naive Bayes Updateable	96.50	68.60

Note: The percentage value displays the accuracy of the features that are truely classified as a person and others.

non-linear classification problems, it was chosen as the classifier to be used in this study.

The LRF data consists of various furniture, structures, people and their poses. As given in Section 2.4, the LRF data was first segmented and filtered, and ellipses were fitted for feature extraction. Ellipses fitted on the torso of a person with different poses are shown in Figure 2.5a and Figure 2.5b. Although there are slight changes due to the position of hands (this could also happen due to different types of clothing), the ellipses were fitted reasonably well.

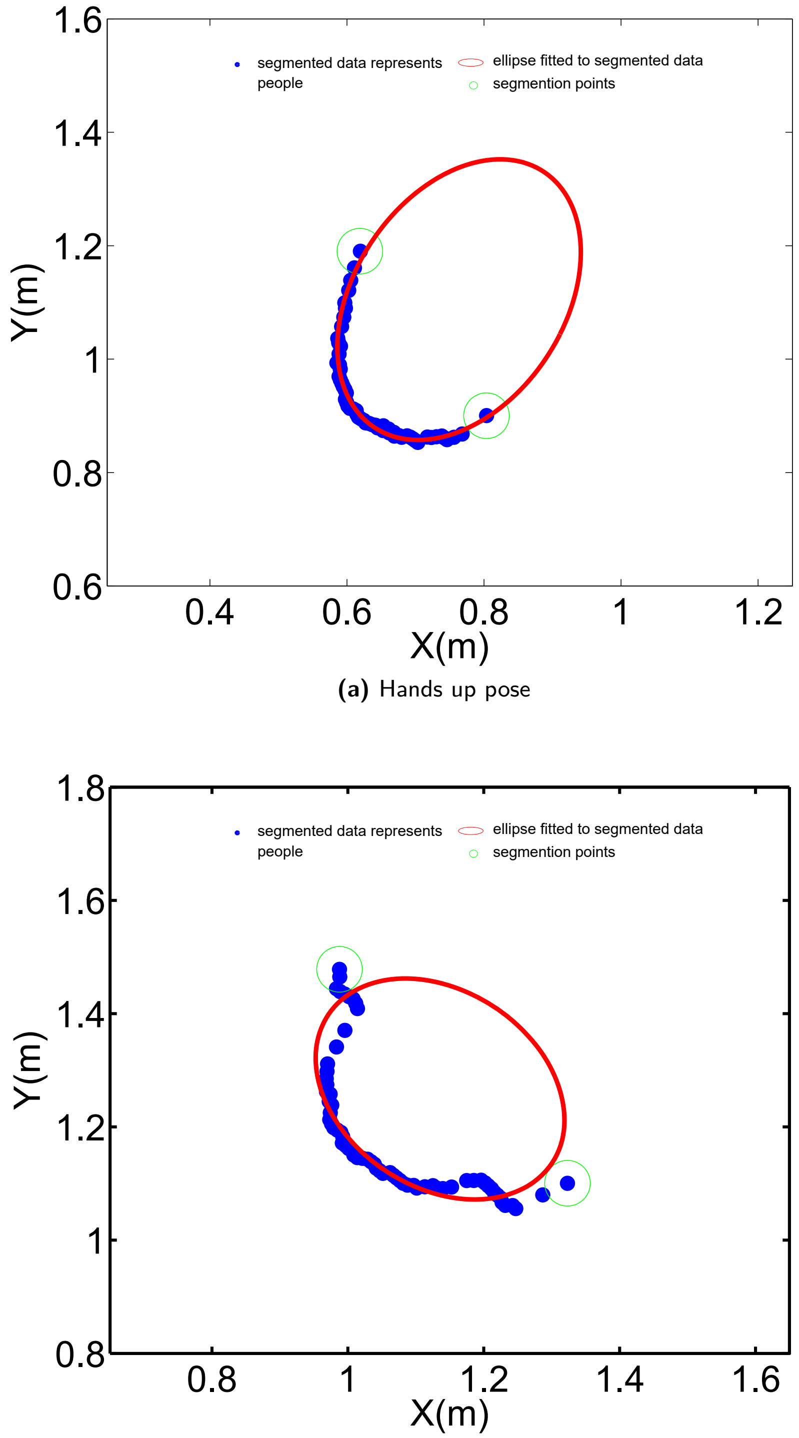
The features described in Section 2.4.1 were estimated and used in Weka [36] with several numbers of classifiers as shown in Table 2.1. The data was analyzed by categorizing the scenarios into three cases based on the number of people present in the environment (and hence possible occlusions).

In general, it could be seen that the classifier performance degraded with the increased number of people due to the rise in occlusions. Although in simple scenarios, classifiers such as BayesNet perform well, they are susceptible to errors with increased complexity. On the other hand, classifiers such as radial basis function support vector machines (RBFSVM) leads to better classification accuracy in both simple and complex scenarios, as referred to Table 2.1.

2.4.3 Support Vector Machine

People's detection problems can generally be resolved by using a supervised learning classification algorithm. The selection of a suitable classifier technique is a major challenge for the detection process. In this thesis, Support Vector Machines (SVM) is chosen as supervised learning classifiers for people detection based on 2D laser range finder [37].

Support Vector Machines (SVM) is a supervised learning technique based on statistical learning theory for classification and regression problems. SVM performs classification by estimating hyperplanes in multidimensional spaces to separate data into



(a) Hands up pose

(b) Hands down pose

Figure 2.5: A Segmented LRF data for human pose that fitted with an ellipse

different classes [38]. As a supervised learning technique, SVM requires correctly labeled data to learn the pattern for further generalization. The main idea of SVM is to find the optimum hyperplane in a high-dimensional space to divide into different classes. For a case of linearly separable classification and only two classes, SVM estimates the separating hyperplane with the largest margin between two samples or classes.

In most practical situations, labeled data are not linearly separable and provide no separating hyperplane. In order to handle non-linear classification problems, use the function (kernel) to map nonlinearly separable data to a different Euclidean space for the data to be linearly separable [39; 40]. In this research application, binary classification for people and others was chosen and implemented since classification involves only two classes.

Given a training data set $T = \{(\mathbf{F}_i, l_i) | i_i \in (-1, 1)\}$, where $i = 1, 2, ..., n$ and SVM requires the following optimization [41]

$$\frac{1}{2}(\mathbf{w}_T\mathbf{w}) + G\sum_{i=1}^{n}\xi_i \tag{2.7}$$

subject to $l_i(\mathbf{w}^T\phi(F_i) + b) \geq 1 - \xi$ where $\xi_i \geq 0$. $\mathbf{F}$ and l are the features and the label of the data set. Training vectors $\mathbf{F}_i$ are mapped into a higher dimensional space by function ϕ. G is the penalty parameter of the error term. For the radial basis function SVM, the kernel function is

$$K(\mathbf{F}_i, \mathbf{F}_j) = exp(-\gamma||\mathbf{F}_i - \mathbf{F}_j||^2), \gamma > 0\,, \tag{2.8}$$

where γ is the kernel parameter.

2.4.4 Scan matching by using Iterative Closest Point (ICP)

Some of the scans are taken from a moving observer fitted with a laser range finder (LRF), and therefore, an implementation of scan matching such as Iterative Closest Point (ICP) [1] is used in order to have common global coordinates for consecutive scans. A range scan at discrete time k can be defined as a set of points $\{\mathbf{q}_k\}$ which represent the range and bearing in polar coordinates $\{d_k, \alpha_k\}$.

The scans are indexed by $j = 1,2,3,...,N$, Φ denotes the field of view, N is given by Φ/ρ for partial field of view scan since maximum scan angle is $(\Phi = 270^o)$. Let XY_k be a coordinate system referred to the laser scan at discrete time k. Assuming that the X axis is aligned with the laser beam at $\alpha_k(0) = 0^o$ as seen in Figure 2.6.

$$\alpha_k(j) = \rho_j \tag{2.9}$$

and the cartesian coordinates of the jth point $\mathbf{q}_k(j)$ are;

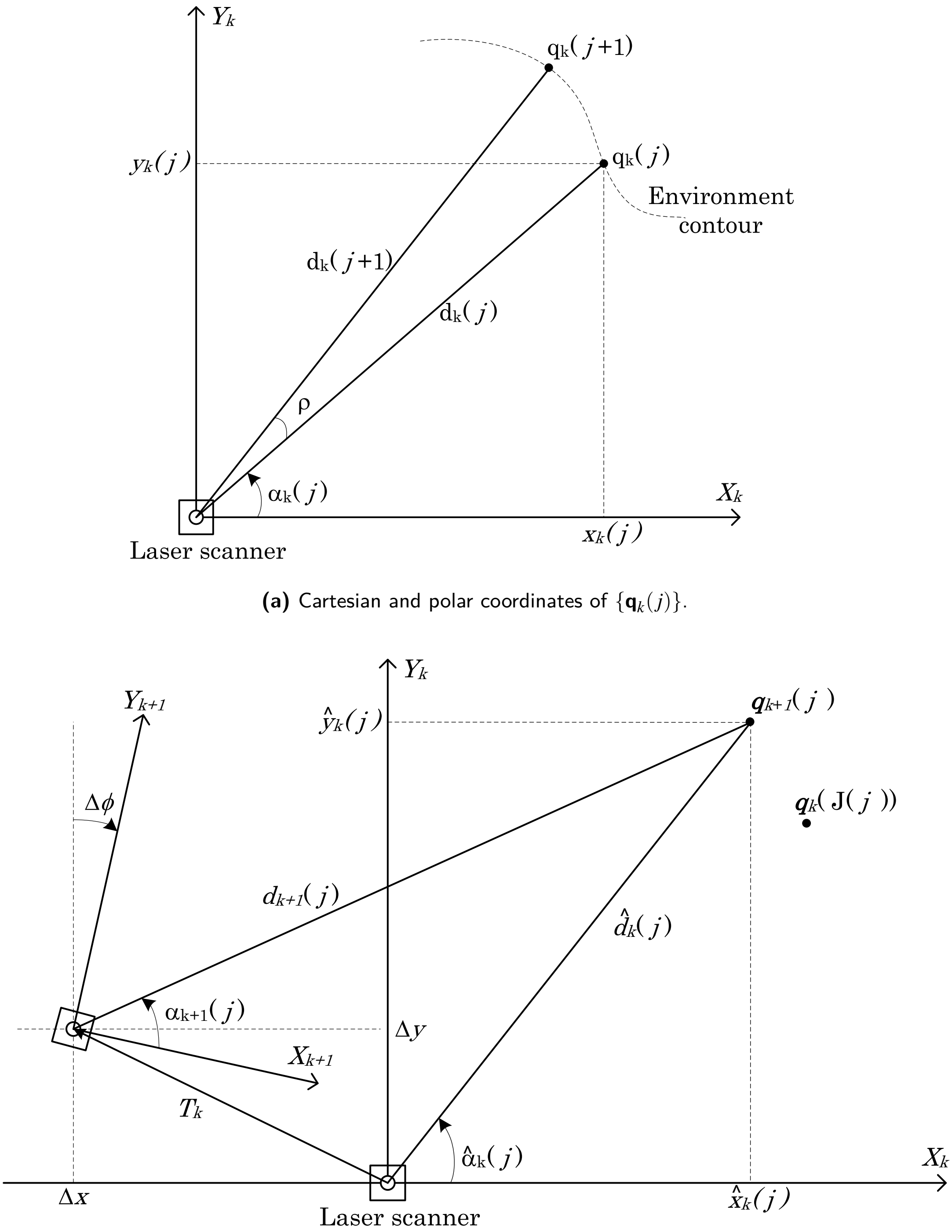
$$x_k(j) = d_k(j)cos(\alpha_k(j)) \tag{2.10a}$$

$$y_k(j) = d_k(j)sin(\alpha_k(j)) \tag{2.10b}$$

When the observer is in motion, two consecutive scans at discrete instants k and $k+1$ will be recorded from different poses of the observer. Thus, $\{\mathbf{q}_k\}$ and $\{\mathbf{q}_{k+1}\}$ will refer to the laser frames at those instants, denoted as XY_k and XY_{k+1}, respectively. Both frames are defined by the global coordinate system.

$\{\mathbf{q}_{k+1}\}$ must be projected onto XY_k frame to find correspondence of two frames, which result in $\{\hat{\mathbf{q}}_k\}$ according to a tentative transformation T_k as shown in Figure 2.6. T_k is a combination of the relative displacements $(\Delta x, \Delta y)$ and the rotation increment $\Delta\phi$ between XY_k and XY_{k+1}.

Cartesian coordinates for $\{\hat{\mathbf{q}}_k\}$ as shown in Figure 2.6 are defined as:



(a) Cartesian and polar coordinates of $\{\mathbf{q}_k(j)\}$.

(b) Projection of $\{\mathbf{q}_{k+1}(j)\}$ onto XY_k frame.

Figure 2.6: Scan Matching using Iterative Closest Point (ICP), adopted from article written by Martínez et al [1]

$$\begin{bmatrix} \hat{x}_k(j) \\ \hat{y}_k(j) \end{bmatrix} = \begin{bmatrix} cos(\Delta\phi) & -sin(\Delta\phi) \\ sin(\Delta\phi) & cos(\Delta\phi) \end{bmatrix} \begin{bmatrix} x_{k+1}(j) \\ y_{k+1}(j) \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2.11}$$

In the same manner, the projected polar coordinates for $\{\hat{\mathbf{q}}_k\}$ can be described as:

$$\hat{\alpha}_k(j) = \Delta\phi + tan^{-1}\left(\frac{y_{k+1}(j)+\Delta y}{x_{k+1}(j)+\Delta x}\right) \tag{2.12a}$$

$$\hat{d}_k(j) = \sqrt{(x_{k+1}(j)+\Delta x)^2 + (y_{k+1}(j)+\Delta y)^2} \tag{2.12b}$$

Scan matching is a suitable approximation problem since exact correspondence of points from different scans is difficult due to random noise, spurious ranges, and occluded areas. A general matching index I_{T_k} for a given transformation T_k can be formulated as:

$$I_{T_k} = \frac{\sum_{j=0}^{N}[p_{T_k}(j)e_{T_k}(j)]}{n_{T_k}P_{T_k}} \tag{2.13}$$

where a match error function e_{T_k} can be defined as:

$$e_{T_k}(j) = e_{T_k}[\hat{\mathbf{q}}_k(j), \mathbf{q}_k(J(j))], \tag{2.14}$$

a boolean function for outlier detection p_{T_k} can be defined as:

$$p_{T_k}(j) = \begin{cases} 0 & if\,|e_{T_k}(j)| \geq E \\ 1 & otherwise, \end{cases} \tag{2.15}$$

the number n_{T_k} of valid correspondence is given by:

$$n_{T_k} = \sum_{j=0}^{N} p_{T_k}(j) \tag{2.16}$$

the ratio that shows the degree of overlap of any possible transformation is:

$$P_{T_k} = \frac{n_{T_k}}{N+1}. \tag{2.17}$$

In the Iterative Closest Point (ICP) Algorithm, it has four-step iterations. T_k is necessarily to be initialized with the odometric motion estimation T_k^o prior to the first step, which calculates the cartesian coordinates of $\hat{\mathbf{q}}_k$ and $\mathbf{q}_{k+1}$ onto XY_k according to Eq. 2.11.

The second step is computing the squared distances for every possible combination of $\hat{\mathbf{q}}_k$ and $\mathbf{q}_k$ points:

$$e(i,j) = (x_k(i) - \hat{x}_k(j))^2 + (y_k(i) - \hat{y}_k(j))^2 \tag{2.18}$$

The third step is calculating the correspondence index function $J(j)$ based on minimum squared distances:

$$J(j) = m, \text{ if } e(m,j) \ = \text{min}_{i=0}^{N}[e(i,j)] \tag{2.19}$$

Therefore, the match error function of Eq. 2.14 is given by:

$$e_{T_k}(j) = (x_k(J(j)) - \hat{x}_k(j))^2 + (y_k(J(j)) - \hat{y}_k(j))^2 \tag{2.20}$$

and outlier detection is carried out with Eq. 2.15.

In the final step, motion parameters are updated by minimizing Eq. 2.13 with the error definition of Eq. 2.20. This optimization can be solved as follows:

$$\frac{dI_{T_k}}{d\Delta\phi} = 0 \Rightarrow \Delta\phi^{new} = tan^{-1}\left(\frac{S_{x_k}S_{y_{k+1}} + n_{T_k}S_{y_kx_{k+1}} - n_{T_k}S_{x_ky_{k+1}} - S_{x_{k+1}}S_{y_k}}{n_{T_k}S_{x_kx_{k+1}} + n_{T_k}S_{y_ky_{k+1}} - S_{x_k}S_{x_{k+1}} - S_{y_k}S_{y_{k+1}}}\right) \tag{2.21a}$$

$$\frac{dI_{T_k}}{d\Delta x} = 0 \Rightarrow \Delta x^{new} = \left(\frac{S_{x_k} - \cos(\Delta\phi^{new})S_{x_{k+1}} + \sin(\Delta\phi^{new})S_{y_{k+1}}}{n_{T_k}}\right) \tag{2.21b}$$

$$\frac{dI_{T_k}}{d\Delta y} = 0 \Rightarrow \Delta y^{new} = \left(\frac{S_{y_k} - \sin(\Delta\phi^{new})S_{x_{k+1}} - \cos(\Delta\phi^{new})S_{y_{k+1}}}{n_{T_k}}\right) \tag{2.21c}$$

where the S terms stand for the following sums:

$$S_{x_k} = \sum_{j=0}^{N}[p_{T_K}(j)x_k(J(j))] \qquad S_{x_{k+1}} = \sum_{j=0}^{N}[p_{T_K}(j)x_{k+1}(j)] \tag{2.22a}$$

$$S_{y_k} = \sum_{j=0}^{N}[p_{T_K}(j)y_k(J(j))] \qquad S_{y_{k+1}} = \sum_{j=0}^{N}[p_{T_K}(j)y_{k+1}(j)] \tag{2.22b}$$

$$S_{x_kx_{k+1}} = \sum_{j=0}^{N}[p_{T_K}(j)x_k(J(j))x_{k+1}(j)] \qquad S_{x_ky_{k+1}} = \sum_{j=0}^{N}[p_{T_K}(j)x_k(J(j))y_{k+1}(j)] \tag{2.22c}$$

$$S_{y_kx_{k+1}} = \sum_{j=0}^{N}[p_{T_K}(j)y_k(J(j))x_{k+1}(j)] \qquad S_{y_ky_{k+1}} = \sum_{j=0}^{N}[p_{T_K}(j)y_k(J(j))y_{k+1}(j)] \tag{2.22d}$$

ICP guarantees convergence to a local minimum, which is not necessarily the globally optimal solution that is close to the odometric estimation, where the most expensive computation is to find the closest points at each iteration [42; 43].

2.5 Experimental Results

The robot used in the experiments is a Segway equipped with sensors and computers, which is shown in Figure 2.7. It has an onboard computer, an AMD Athlon II X2 255/ Dual Core/ 3.1 GHz with 4GB of DDR3 running on the Linux Ubuntu 9.10 operating system. The robot system uses a HOKUYO UTM-30LX laser range finder that has 30 meters of detection range, 0.25^o angular resolution, an angular field view of 270^o and a 25 millisecond sampling period. The Segway robot is used to monitor the environment while people are in motion. The experiments were carried out in a common area of the laboratory, as shown in Figure 2.8.

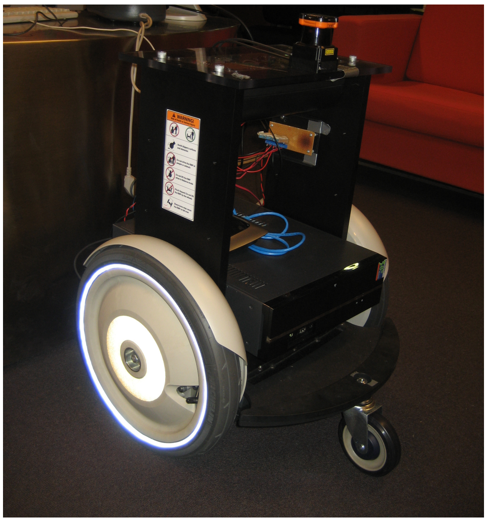


Figure 2.7: Robot used in the experiments

2.5.1 People Detection

Experiments were conducted to assess the performance of the people detection algorithm in a 5×7 square meters room fitted with office-related furniture such as tables and chairs. The distances between the laser range finder (LRF) and targets were from 0.5 to 4 meters. In order to have a better understanding of the errors and their causes, data was collected in specific scenarios, such as people facing the sensor with hands up or down and people facing sideways with hands up or down. In all cases, the laser

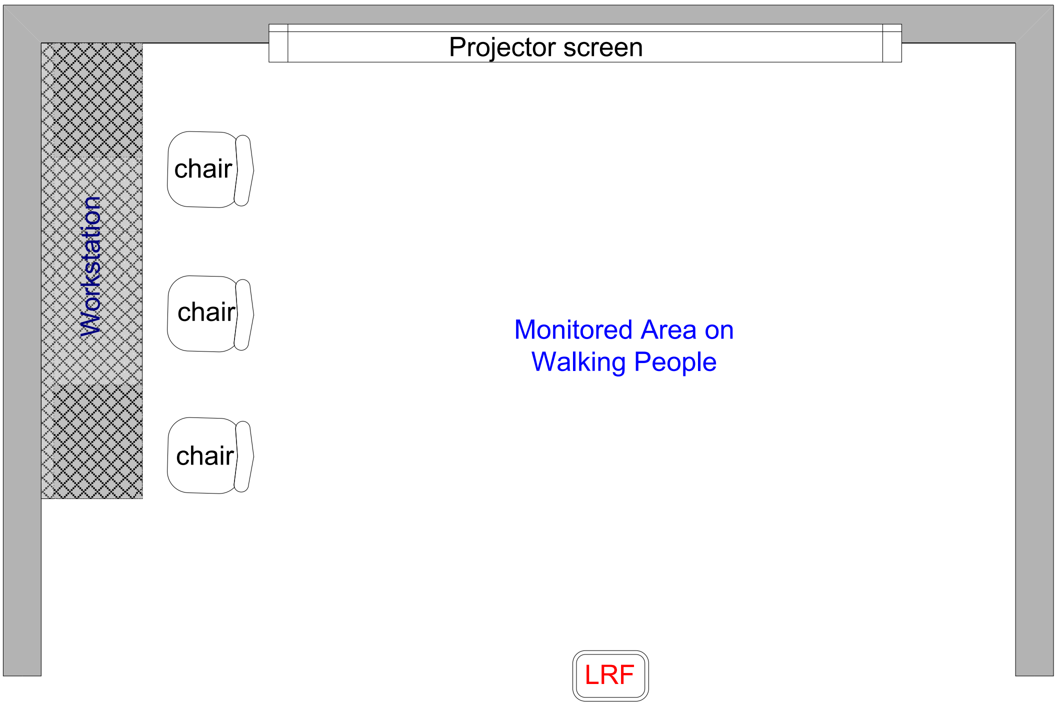


Figure 2.8: Graphical Illustration of the experiment location

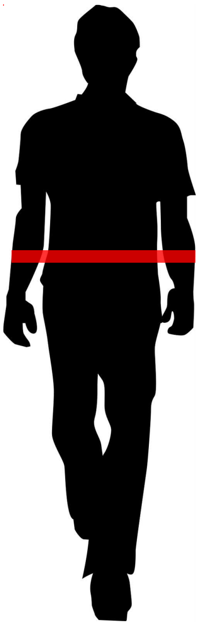


Figure 2.9: Laser data taken at torso height as shown on red marking.

range finder data were taken at the torso height of the human, as shown in Figure 2.9. Table 2.2 summarizes the accuracies, precisions, and recalls on the straight and side facing faces of the people. Detection accuracy with the side facing is slightly similar to that with the straight facing of people because of its shape similarity. Precision and recall of both types facing are generally higher than 80% where precision represents the number of selected labels that are relevant and recall represents the number of relevance labels that are selected. In Table 2.3, it is the summarization of the accuracies, precisions, and recalls of two and three people in vicinity. It shows that precision and recall on the predicted label on the hands up and down pose are higher than 86%.

On the other hand, not surprisingly, people in the hands up pose have higher accuracy than those in the hands down pose (normal pose). It could also be noted that the false positives are always smaller than the false negatives. Therefore, the algorithm provides more candidates, which can be further filtered to improve the detection accuracy. Total computing time is not more than 0.09s in all scenarios.

Although, in general the ellipse fitting algorithm worked well, it had some problems with segmented data relevant to occluded scenarios. It can be identified by observing the change in shape from one person overlapping to the other. This is explained in Figure 2.10. In the figure, ellipse fitting was done reasonably well from (a) to (d). The problem started at (e), where one ellipse had undergone a significant change to its size and shape. In (f), one ellipse completely disappeared due to an occlusion and started to re-appear in (d).

Table 2.2: Confusion Matrix for Straight and Side Facing

STRAIGHT FACING				
	HANDS UP POSE		NORMAL POSE	
	True Label			
Predicted Label	Person	Others	Person	Others
Person	86.41 %	0.77 %	83.33 %	3.00 %
Others	13.59 %	99.23 %	16.67 %	97.00 %
Total Accuracy of True Detection	96.54 %		94.77 %	
Precision (Person)	93.00 %		84.34 %	
Recall (Person)	90.29 %		83.33 %	
Computing Time	0.03 s		0.06 s	

SIDE FACING				
	HANDS UP POSE		NORMAL POSE	
	True Label			
Predicted Label	Person	Others	Person	Others
Person	91.97 %	2.14 %	86.75 %	3.56 %
Others	8.03 %	97.86 %	13.25 %	96.44 %
Total Accuracy of True Detection	97.22 %		92.84 %	
Precision (Person)	84.00 %		93.51 %	
Recall (Person)	91.97 %		86.75 %	
Computing Time	0.06 s		0.02 s	
Note: Percentage indicates on true and false detection.				

Table 2.3: Confusion Matrix for Two People and Three People

TWO PEOPLE				
	HANDS UP POSE		NORMAL POSE	
	True Label			
Predicted Label	Person	Others	Person	Others
Person	95.86 %	0.00 %	90.71 %	8.33 %
Others	4.14 %	100.00 %	9.29 %	91.67 %
Total Accuracy of True Detection	97.25 %		91.12 %	
Precision (Person)	100.00 %		93.38 %	
Recall (Person)	95.86 %		90.71 %	
Computing Time	0.01 s		0.01 s	

THREE PEOPLE				
	HANDS UP POSE		NORMAL POSE	
	True Label			
Predicted Label	Person	Others	Person	Others
Person	90.35 %	4.93 %	86.19 %	2.63 %
Others	9.65 %	95.07 %	13.81 %	97.37 %
Total Accuracy of True Detection	93.18 %		94.20 %	
Precision (Person)	92.41 %		92.84 %	
Recall (Person)	90.35 %		86.19 %	
Computing Time	0.09 s		0.08 s	
Note: Percentage indicates on true and false detection.				

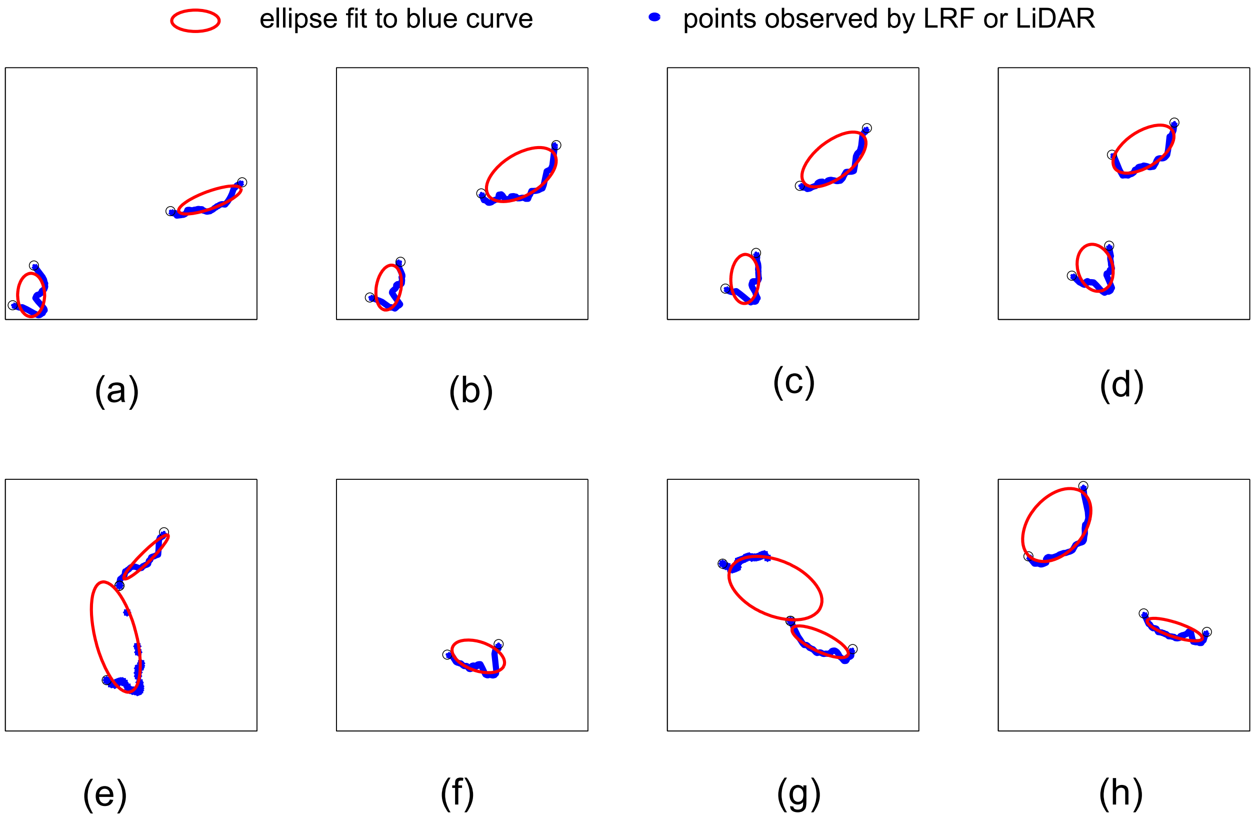


Figure 2.10: Occlusion of two people that is shown in successive time frames

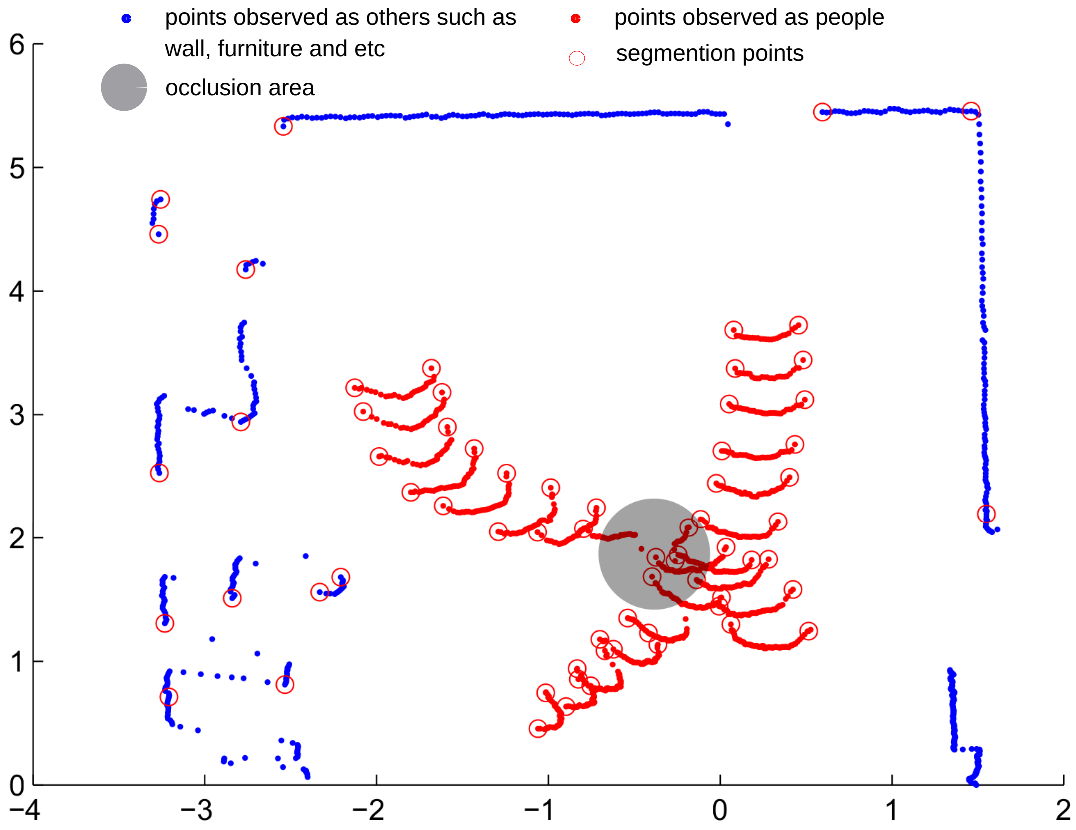


Figure 2.11: Detection of two people with occlusions that are shown in the grey circle

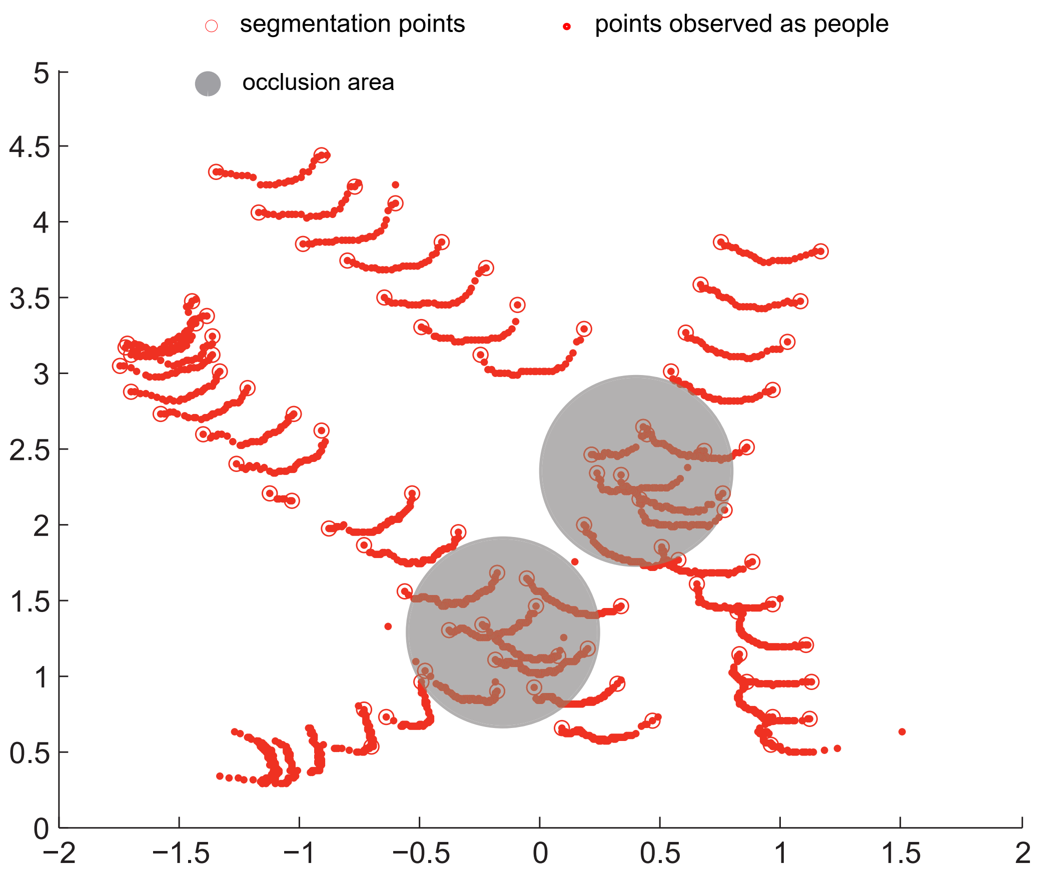


Figure 2.12: Detection of three people with occlusions that are shown in the grey circle

Clearly, this phenomenon can be further understood by referring to Figure 2.11 and Figure 2.12. The occlusion occurs in the grey circle area where the shape of the torso on detected people is not representing the appropriate shape of people. Therefore, this process causes a detector to have difficulties handling the targets.

2.6 Conclusion

In this chapter, a number of features were implemented on segmented data using Kalman Filter and a number of techniques for learning algorithms proposed in the literature that were suitable for detecting people were examined. The techniques were evaluated against several key criteria in the context of the research problems addressed. The number of selected learning algorithms is evaluated by various numbers of people in the office environment. From the evaluation, the Support Vector Machine (SVM) classifier has shown promising results with a high percentage of positive detection in comparison with other learning algorithms.

Various scenarios on people looking at the sensor and people looking sideways with hands up and down with occluded scenarios have been chosen, and surprisingly, the detection rate produces a higher precision and recall percentage even with two and three people in the vicinity. The following chapter discusses the use of the IMMPDAF tracker for people tracking, and the performance of the tracker was analysed using simulation and experimental findings.

Chapter 3

People Tracking

Efficient people tracking is one of the important tasks in dealing with Human-Robot-Interaction (HRI) in real-world scenarios. In various applications, people detection and tracking are integrated into a module using the so-called tracking-by-detection technique. In this research work, the Interacting-Multiple-Model Probabilistic Data Association Filter (IMMPDAF) for people tracking with the tracking-by-detection technique was introduced. The performances of the tracker using Normalized Estimation Error Squared (NEES) and Normalized Innovation Squared (NIS) were then analysed.

3.1 Introduction

This chapter explains the concept and implementation of people tracking techniques and algorithms. A variety of sensors and algorithms are used in people tracking, particularly in dynamic scenarios, to effectively identify and forecast people's movements. Numerous uses, including control of crowds, autonomous vehicles, smart buildings, and surveillance, depend on this mechanism. Precision tracking and environmental sensing have advanced significantly with the use of LRF technology for people tracking. As technology develops, overcoming present obstacles and guaranteeing ethical application will open the door for creative applications in a variety of industries that will further improve safety, effectiveness, and user experiences.

3.2 Related Works

Tracking people is an important aspect of security, surveillance and human robot interaction. There has been much research and interest in populated environments using various sensors such as cameras and laser range finders (LRFs) [5] with various detection and tracking techniques.

In laser-based tracking, it is faster to process data and less sensitive to lighting conditions. It can provide good accuracy in sparsely populated environment. However, the tracking accuracy gradually decreases when there are many interactions and occlusions of multiple people [44]. To overcome this drawback, multiple people tracking has been implemented using Bayesian filters with data association, such as the probabilistic data association filter (PDAF), joint probabilistic data association filter (JPDAF) and multiple hypothesis tracker (MHT) for effective tracking [45].

However, a multiple-model-based approach in which different models run in parallel and describe different aspects of human models, such as the Interactive Multiple Model (IMM) estimator, is an effective methodology to deal with manoeuvring people.

Therefore, it is important to measure the performance of the state estimator for target tracking. There are several techniques to validate and tune different sensors and process models. Among these techniques, Normalized Estimation Error Squared (NEES) and Normalized Innovation Squared (NIS) are useful to measure the consistency of the filter [46]. NEES requires the ground truth of the tracking data and predicted data, which should be applied using Monte Carlo runs. NIS is the difference between the actual and predicted observation. These will help to detect and improve noise characteristics.

3.2.1 Importance of People Tracking

The ability to avoid colliding with people is highly important in robotic environments. The detection of a person or more makes the robot aware of a potential collision in its vicinity and predicts the course of the people in the environment. Apart from predicting the target, the robot will also be able to change its trajectory. In this task, the most important aspect is how to deal with the prediction and motion model of the people and the data association of multiple targets.

3.2.2 Methods of Data Association

Most people's tracking algorithms for dealing with multiple-targets are associated with methods of data association. Data association algorithms can be classified into 3 groups:

1. Target-oriented approach, which assumes each measurement originated from a known target.
2. Track-oriented approach, which hypothesises that each track is either undetected, terminated, associated with a measurement, or linked to the start of manoeuvre.
3. Measurement-oriented approach, which generates a number of candidate hypotheses based on the measurement received and evaluates these hypotheses as more measurement data are received.

In association, algorithms specifically used for track formation are classified into non-Bayesian and Bayesian association techniques. Non-Bayesian association techniques can be listed below.

- Nearest Neighbour Standard Filter (NNSF). The classification and pattern recognition algorithm NNSF is straightforward and easy to understand. It adds a fresh data point to the training dataset's nearest neighbour's class. A distance metric, such as Euclidean distance, is used to identify who is the "nearest" neighbour. Although NNSF is computationally effective, it is sensitive to data noise and outliers.
- Strongest Neighbour Standard Filter (SNSF). The Nearest Neighbour Standard Filter (SNSF) is an extension of that filter. SNSF takes into account numerous nearest neighbours as opposed to only one nearest neighbour. Based on a voting

system, the class with the most support from the closest neighbours is chosen to be assigned to the new data point. By lessening the effect of outliers or noisy data, this method can increase the classification's robustness.

- Generalized Nearest Neighbour Filter (GNN). The nearest neighbour algorithms have been improved upon by the GNN. For various features in the dataset, it enables the evaluation of several distance metrics or even various weighting schemes. The generalised framework offered by GNN enables users to alter the association criteria in accordance with the particulars of the data being examined. GNN may be tailored to fit a variety of applications with varying data properties, thanks to its versatility.

These non-Bayesian association approaches are useful in situations where the underlying assumptions of Bayesian methods might not hold true or when simplicity and computing effectiveness are essential. It is crucial to remember that these methods do have some drawbacks, particularly when dealing with noisy or high-dimensional data. As a result, to address particular problems with non-Bayesian association methods, researchers and practitioners frequently investigate hybrid approaches or more sophisticated methodologies.

For examples of Bayesian association techniques, they can be listed as follows.

- Probabilistic Data Association Filter (PDAF). A Bayesian filtering technique for multi-target tracking is called the Probabilistic Data Association Filter (PDAF). It deals with the issue of data association, which entails connecting sensor readings with pre-existing target tracks. As new measurements are received, the PDAF maintains a probability distribution over potential data associations and recursively updates this distributions. It is helpful in situations where the number of targets is unknown beforehand and may change over time.
- Multiple Model Probabilistic Data Association Filter (MMPDAF). The numerous Model Probabilistic Data Association Filter (MMPDAF) expands on the fundamental PDAF concept by using numerous dynamic models to reflect the potential

manoeuvres of the tracked targets. Each model represents a particular type of motion behaviour, such as constant acceleration or velocity. In cases where the target's motion characteristics change over time, MMPDAF can manage the scenario by retaining numerous hypotheses regarding the target's motion. With this method, the filter is more capable of adjusting to changing target dynamics.

- Interacting Multiple Model Probabilistic Data Association Filter (IMMPDAF). The Interacting Multiple Model Probabilistic Data Association Filter (IMM-PDAF) further enhances the MMPDAF method by allowing interactions between many target models. In this method, information from one model can affect the forecasts and likelihoods of data associations from other models. IMMPDAF is especially helpful in cases where targets exhibit complicated manoeuvres or behaviours because it can more precisely capture complex target behaviours by including interactions.

These Bayesian association approaches are essential for target tracking, particularly when uncertainty, variable target dynamics, and ambiguity in the data association are common. These techniques address the problems of multi-target tracking in applications including radar systems, autonomous cars, and surveillance technologies by utilising probabilistic frameworks. Therefore, in this thesis, IMM-PDAF was chosen to address agile and obstructed target tracking.

3.2.3 Probabilistic Data Association Filter (PDAF)

Bar-Shalom and Tse [46] proposed that the algorithm assigned a probability, called the association probability, to every hypothesis associating a validated measurement to a target. Validated measurements refer to measurements that are within the validation gate of a target in real time. A validation gate centered around the predicted measurement of the target set up to select the set of validated measurements is

$$[z(k) - \hat{z}(k|k-1)]^T S^{-1}(k)[z(k) - z(k|k-1)] \leq \gamma \tag{3.1}$$

where $S(k)$ is the covariance of the innovation and γ determines the size of the gate. The set of validated measurements at time k is

$$Z(k) = z_i(k), \quad i = 1, \cdots, m_k \tag{3.2}$$

where $z_i(k)$ is the ith measurement in the validation region at time k.

The standard PDAF equations are described as the followings:

State Prediction. The state prediction step estimates the next state of the system based on the previous state estimate. In this equation, $\hat{x}(k|k-1)$ represents the predicted state at time k, given the state estimate $\hat{x}(k-1|k-1)$ at the previous time step. F is the state transition matrix.

$$\hat{x}(k|k-1) = F\hat{x}(k-1|k-1) \tag{3.3}$$

Measurement Prediction. The measurement prediction step estimates the expected measurement at the current time step based on the predicted state $\hat{x}(k|k-1)$ and $\hat{z}(k|k-1)$ representing the predicted measurement at time k, and H is the measurement matrix mapping the state space to the measurement space.

$$\hat{z}(k|k-1) = H\hat{x}(k|k-1) \tag{3.4}$$

Innovation of ith measurement. The innovation represents the difference between the actual measurement $z_i(k)$ and the predicted measurement $\hat{z}(k|k-1)$. It is denoted as $v_i(k)$.

$$v_i(k) = z_i(k) - \hat{z}(k|k-1) \tag{3.5}$$

Covariance Prediction. This equation predicts the error covariance of the predicted state at time k. $P(k-1|k-1)$ represents the predicted covariance at time k, F is the state transition matrix, Q is the process noise covariance and G is the process noise gain matrix.

$$P(k|k-1) = FP(k-1|k-1)F^T + GQG^T \tag{3.6}$$

Innovation Covariance. The innovation covariance $S(k)$ quantifies the uncertainty in the innovation (the difference between the actual measurement and the predicted measurement) at time k. H is the measurement matrix, and R is the measurement noise covariance.

$$S(k) = HP(k|k-1)H^T + R \tag{3.7}$$

Kalman Gain. The Kalman Gain $K(k)$ determines the amount of weight given to the current measurement for updating the state estimate. It is calculated using the predicted covariance $P(k|k-1)$, innovation covariance $S(k)$, and the measurement matrix H.

$$K(k) = P(k|k-1)H^T S(k)^{-1} \tag{3.8}$$

Updated covariance if target originated measurements were known. $P^\circ(k|k)$ represents the updated state covariance if the measurements were known to originate from the target. It is calculated using the Kalman Gain $K(k)$and innovation covariance $S(k)$.

$$P^\circ(k|k) = P(k|k-1) - K(k)S(k)K(k)^T \tag{3.9}$$

Overall covariance update. This equation represents the overall update of the state covariance matrix $P(k|k)$ at time k. It takes into account the predicted covariance $P(k|k-1)$, Kalman gain $K(k)$, innovation covariance $S(k)$ and the association probabilities $\beta_i(k)$ for different measurements.

$$\nu(k) = \sum_{i=0}^{m_k} \beta_i(k)\nu_i(k) \tag{3.10}$$

$$P(k|k) = P^{\circ}(k|k) + K(k)[\beta_{\circ}(k)S(k) + \sum_{i=0}^{m_k}[\beta_i(k)\nu_i(k)\nu_i(k)^T] - \nu(k)\nu(k)^T]K^T(k) \tag{3.11}$$

where m_k is the number of validated returns at kth instant.

The updated state estimate. $\hat{x}(k|k)$ represents the updated state estimate at time k. It is obtained by combining the predicted state $\hat{x}(k|k-1)$and the innovation weighted by the Kalman gain.

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\nu(k) \tag{3.12}$$

The PDAF association probabilities. $\beta_i(k)$ represents the association probabilities for different hypotheses associating a validated measurement to a target. $p_i(k)$ represents the likelihood of the ith measurement given the target. These probabilities are used to weigh the contribution of each measurement to the overall covariance update, as shown in equation (3.11).

$$\beta_i(k) = \frac{p_i(k)}{\sum_{i=0}^{m_k} p_i(k)} \tag{3.13}$$

where **measurement likelihood**.

$$p_i(k) = \begin{cases} \lambda(1 - P_d P_g), & \text{if } \ i = 0 \\ \dfrac{P_d}{(2\pi)^{M/2}|S(k)|^{1/2}} exp[-\dfrac{1}{2}r_i(k)^2], & \text{if } \ [\Omega(k)] = 1; i \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

$p_i(k)$ represents the likelihood of the ith measurement given the target at time k. When $i = 0$, indicating the null hypothesis with no valid measurement associated with the target, $p_i(k)$ is calculated based on clutter parameters λ, probability of detection P_d, and probability of gating P_g. For non-null hypotheses $[\Omega(k)] = 1$, meaning the measurement belongs to the validation gate of the target, $p_i(k)$ is calculated using the measurement residual $r_i(k)$ and innovation covariance $S(k)$;

where

$$\lambda = \frac{m_k}{V(k)},$$

$$V(k) = \frac{\pi^{M/2}}{\Gamma(M/2+1)}\gamma^M |S(k)|^{1/2},$$

λ represents the clutter density, calculated based on the number of validated returns (m_k) and the clutter volume $(V(k))$. $V(k)$ is determined by the dimensionality of the state vector M and the innovation covariance S_k. It is used in the calculation of λ, Γ denotes the Gamma function, and γ is a constant parameter; and

$$\Omega(k) = \begin{cases} 1, & \text{if the return belongs to the validation gate of the target} \\ 0, & \text{otherwise.} \end{cases}$$

$\Omega(k)$ is an indicator function that evaluates whether a return belongs to the validation gate of the target which is denoted as 1 or otherwise, which is denoted as 0.

PDAF may perform poorly when tracking crossing targets or when the targets are close to each other. It also needs to provide separate track initiation and deletion algorithms. It is mainly good for non-manoeuvring targets in cluttered environment. A combination of IMM and PDAF, called IMMPDAF [47] can be used to overcome those issues. It can be used for track initiation, track maintenance on manoeuvring targets, and track termination [48].

In this particular task, it can be accomplished with a constant velocity model and a constant turn rate model, which can be assigned to several models such as turning left and turning right of the targets [49].

3.2.4 Interacting Multiple Model Tracker

The idea of all multiple model approaches is to overcome bad prediction by tracking manoeuvring targets with abrupt deviations from a straight-line motion. This process is hard to represent with a single manoeuvre model; thus, the representation of various potential target manoeuvre states, run in parallel and continuously evaluated by using previous states of filters' residuals, can be carried out in multiple models. The Interacting Multiple Models algorithm is widely accepted as one of multiple models approaches [48].

Bayes's rule and the residuals are applied to specify the relative probabilities of the validity of the models. Typically, a probability-weighted composite of the individual filters is the output; otherwise, it may prove to be more accurate to output estimates from the filter with the highest probabilities. For the multiple model approach, the ith dynamics model and measurement equations are

$$x_i(k+1) = F_i(k)x_i(k) + v_i(k) \tag{3.14}$$

$$z_i(k+1) = H_i(k+1)x_i(k+1) + w_i(k+1) \tag{3.15}$$

where x is the state of the target, defined as $x(k) = \begin{bmatrix} x & \dot{x} & y & \dot{y} & \omega \end{bmatrix}^T$ with x and y denotes the Cartesian coordinates of target; ω is a turn rate; $F(k)$ is the state transition model with a constant speed and a turn rate model; $v(k)$ is a zero-mean Gaussian white noise (process noise) with appropriate covariance Q; H_{k+1} is the measurement model; $x(k+1)$ is measured coordinates at scan $k+1$; and $w(k+1)$ is random noise on measurements at scan $k+1$.

The IMM algorithm can be divided into four parts:

- An input mixer (Interaction).
- A filter for each model (Updates).
- A model probability evaluator.
- An output mixer.

Input Mixer (Interaction)

The input state estimate mixer merges the previous cycles of mode-conditioned state estimates and covariance, using mixing probabilities, to initialise the current cycle of each mode-conditioned filter. The filtering process starts with a priori state estimates $x_j^{\circ}(k-1|k-1)$, state error covariance $P_j(k-1|k-1)$, and the associated probabilities $\mu_j(k-1)$ for each model. The initial state estimate and covariance for model j at time k is computed as follows:

$$\hat{x}_j^{\circ}(k-1|k-1) = \sum_{i=1}^{N} \hat{x}_i(k-1|k-1)\mu_{i|j}(k-1|k-1) \tag{3.16}$$

$$P_j^{\circ}(k-1|k-1) = \sum_{i=1}^{N} \mu_{i|j}(k-1|k-1)\{(P_i(k-1|k-1)+$$

$$[\hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)][\hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)]^T\} \tag{3.17}$$

where

$$\mu_{i|j}(k-1|k-1) = \frac{1}{c_j} p_{ij} \mu j(k-1)$$

$$\bar{c}_j = \sum_{i=1}^{N} p_{ij} \mu j(k-1)$$

and p_{ij} is assumed transition probability for switching from model i to j, and $\bar{c}_j$ is a normalization constant.

Filtering Updates

The updates for each subfilter or model are performed using the Kalman filter or Extended Kalman filter equations. Different models are used, and usually there is a second-order model with a few third order models. The second-order model is dominant when the target is in a non-manoeuvring state. This model is more straightforward and computationally efficient because it often represents constant velocity motion. It works well for following gradually moving targets without significant acceleration or manoeuvres.

The third-order model for the manoeuvring state has different process noise levels, which allow it to capture and account for varying uncertainties in position, velocity, and acceleration dynamics. When the target is thought to be in a state of manoeuvring, the third-order model is used. When targets are being manoeuvred, they experience acceleration or changes in velocity, which necessitate a more complicated model to appropriately represent their behaviour. Third-order models take these manoeuvres into account, enabling the system to quickly adjust to variations in target motion. The set of Kalman filtering equations that provide the model updates is shown as below. For more details of the process, refer to [50].

$$
\begin{aligned}
x_j(k|k-1) &= F_j(k-1)x_j^0(k-1|k-1) + G(k-1)U(k-1) \\
P_j(k|k-1) &= F_j(k-1)P_j^0(k-1|k-1)(F_j(k-1))^T + Q_j(k) \\
S_j(k) &= H_j(k)P_j(k|k-1)(H_j(k))^T + R(k) \\
K_j(k) &= P_j(k|k-1)(H_j(k))^T(S_j(k))^{-1} \\
\tilde{z}_j(k) &= z_j(k) - H_j(k)x_j(k|k-1) \\
x_j(k|k) &= x_j(k|k-1) + K_j(k)[\tilde{z}_j(k)] \\
P_j(k|k) &= [I - K_j(k)H_j(k)]P_j(k|k-1)
\end{aligned}
$$

3.2.5 Model Probability Evaluator

The likelihood of $\Lambda_j(k)$ is computed with the filter residual $\tilde{z}_j(k)$, the covariance of the filter residuals $S_j(k)$ and the assumption of a Gaussian distribution [51]. The likelihood of $\Lambda_j(k)$ is given by

$$\Lambda_j(k) = \frac{1}{\sqrt{2\pi|S(k)|}} e^{-0.5(\tilde{z}_j(k)^T(S_j(k))^{-1}\tilde{z}_j(k))}$$

The model probabilities update is

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\bar{c}_j \tag{3.19}$$

$$c = \sum_{j=1}^{r} \Lambda_j(k)\bar{c}_j \tag{3.20}$$

Output Mixer

The output mixer combines all the state estimates and covariances from the individual filter output as follows:

$$\hat{x}(k|k) = \sum_{j=1}^{N} \hat{x}_j(k|k)\mu_j(k) \tag{3.21}$$

$$P(k|k) = \sum_{j=1}^{N} \mu_j(k)\{P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)][\hat{x}_j(k|k) - \hat{x}(k|k)]^T\} \tag{3.22}$$

Various dynamic motion models that describe aspects of target motion build the IMM filter. For a particular target manoeuvre, the filter will automatically choose the mix of models.

3.2.6 Consistency Analysis of IMM Tracker

The finite-sample consistency property states that estimation errors based on a finite number of measurements should be consistent with their theoretical statistical properties, as in the following [52]:

1. mean zero.
2. covariance matrix as calculated by the IMM tracker.

The consistency criteria of an IMM tracker are that:

1. state errors should be acceptable as a zero mean, and their magnitude should be comparable to the state covariance measured by the tracker.
2. innovations should also have the same property.
3. innovations should be acceptable as independent and identically distributed random variables.

where the innovation is the difference between the observed value of a variable at k and the optimal forecast of that value based on information available prior to k.

The first criteria can be tested only in simulations. By using notation

$$\tilde{x}(k|k) = x(k) - \hat{x}(k|k) \tag{3.23}$$

to define the normalised estimation error squared (NEES) [53]

$$\varepsilon(k) = \tilde{x}(k|k)^T P(k|k)^{-1} \tilde{x}(k|k) \tag{3.24}$$

The test can simultaneously verify both properties (1) mean zero and (2) covariance matrix.

3.2.7 Monte Carlo Simulation-Based Tests

The test is based on the results of Monte Carlo simulations that provide N independent samples $\varepsilon^i(k), i = 1, ..., N$, of the random variable $\varepsilon(k)$. For the sample average of $\varepsilon(k)$, the N-run average NEES [53] is

$$\tilde{\varepsilon}(k) = \frac{1}{N} \sum_{i=1}^{N} \varepsilon^i(k). \tag{3.25}$$

Then $N\tilde{\varepsilon}(k)$ will have a chi-square density with Nn_x degrees of freedom, where n_x is the dimension of the measurement. A hypothesis that the state estimation errors are consistent with the filter-calculated covariances, which is also called the chi-square test, is accepted if $\tilde{\varepsilon}(k) \in [r_1, r_2]$ where the acceptance interval is determined such that $P\{\tilde{\varepsilon}(k) \in [r_1, r_2] | H_0\} = 1 - \alpha$. The interval of r_1 and r_2 is the 95% probability concentration region for $\tilde{\varepsilon}(k)$.

The correspondence of the innovations with their filter-calculate covariances is tested in a similar manner. Under the hypothesis that the filter is consistent, the normalised innovation squared (NIS) can be shown as the following equation. It has a chi square distribution with n_z degrees of freedom, where n_z is the dimension of the measurement. The following equation, cited from [53], calculates the average NIS from N independent samples $\varepsilon_v^i(k)$.

$$\tilde{\varepsilon}_v(k) = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_v^i(k). \tag{3.26}$$

3.2.8 Types of IMM Tracker Testing

Consistency tests for off-line multiple-run (Monte Carlo simulation) tests of NEES and NIS demonstrate the following types of tests. Figure 3.2 shows the test statistic obtained from $N = 50$ Monte Carlo runs with the two-sided probability regions. It shows the state's N-run average NEES. Note that in this case, the two-sided 95% region is [0.05,7.38]. Since the lower limit is practically zero, only the upper limit is of interest, and it is taken for the 5% tail rather than the 2.5% tail, which is 7.38.

3.3 Tracking using Interacting Multiple Model (IMM) Tracker

Once people were detected based on the laser data, it was temporally tracked based on an Interactive Multiple Model (IMM) tracker [47; 48]. Constant velocity and constant turn rate models have been used to model human motion.

3.3.1 Dynamic Model

The tracking represents the target (i.e., people detected) as a curve of torso denoted by the midpoint (x,y) and orientation ϕ. For the constant velocity and constant turn rate model that is applicable to human motion, the coordinate system is shown [54] as

$$X_{k+1} = \begin{bmatrix} 1 & \frac{sin\omega(k)\Delta T}{\omega(k)} & 0 & -\frac{1-cos\omega(k)\Delta T}{\omega(k)} & 0 \\ 0 & cos\omega(k)\Delta T & 0 & -sin\omega(k)\Delta T & 0 \\ 0 & \frac{1-cos\omega(k)\Delta T}{\omega(k)} & 1 & \frac{sin\omega(k)\Delta T}{\omega(k)} & 0 \\ 0 & sin\omega(k)\Delta T & 0 & cos\omega(k)\Delta T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} X_k + \begin{bmatrix} \frac{\Delta T^2}{2} & 0 & 0 \\ \Delta T & 0 & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T \end{bmatrix} v_k \tag{3.27}$$

where ΔT is the sampling time and ω is the turn rate of the target can be assigned a few values to define several models, including $\omega = 0$ for the constant velocity model (straight path trajectory). The target state vector is defined as $\mathbf{s} = [x, v_x, y, v_y, \phi]^T$, where position vector $\mathbf{x} = [x, y]^T$, velocity vector $\mathbf{v} = [v_x, v_y]^T$ and the orientation of target ϕ. The system noise vector is given by v_k.Â

The measurement model of a person is then:

$$Z_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} X_{k+1} + \begin{bmatrix} v_x(k+1) \\ v_y(k+1) \end{bmatrix} \tag{3.28}$$

where $X_k = [x\ v_x\ y\ v_y\ \omega]^T$.

3.3.2 Track Formation and Termination

Due to the large scatter present in the environment due to various furniture, glass walls and various metal parts, there were obvious false detections. The tracking problem was complex and nontrivial to handle due to the disappearance, reappearing and manoeuvring of the target in the clutter. This problem was handled by an IMM Probabilistic Data Association Filter (PDAF) tracker with track confirmation and deletion. Using the Markov relationship, the probability of existence of a true person $P_T(k+1|k)$ before receiving data in scan $k+1$ is given by,

$$P_T(k+1|k) = P_{22}P_T(k|k) + P_{12}(k|k) \tag{3.29}$$

where P_{22} is the transition probability from an unobservable to an observable state, and P_{12} is the transition probability from an unobservable to an observable state. Then, the probability update of person existence [48] is

$$P_T(k+1|k+1) = \frac{1-\delta_{k+1}}{1-\delta_{k+1}P_T(k+1)} P_T(k+1|k) \tag{3.30}$$

where

$$\delta_{k+1} = \begin{cases} P_D P_G, & N_{k+1} = 0 \\ P_D P_G \left[1 - \overline{V} \sum_{i=1}^{N_{k+1}} \frac{1}{P_G (2\pi)^{M/2} \sqrt{S|k+1|}} e^{-d_i^2/2} \right], & \text{otherwise} \end{cases}$$

and $V = V_{G_{k+1}} / (N_{k+1} - P_D P_G P_T(k+1|k))$, P_D is the probability of detection, P_G is the gate probability, V_G is the gate volume, N_{k+1} is the number of measurements inside the validation gate, S is the innovation covariance, and d_i^2 is the normalised innovation squared of the ith measurement.

The log-likelihood ratio (LLR) [48] is defined as:

$$LLR_{k+1} = \ln\left(\frac{P_T}{1 - P_T}\right). \tag{3.31}$$

Once the LLR is obtained, confirmation and termination of track thresholds are determined as

$$\begin{cases} LLR_{k+1} \geq \ln\left(\frac{1-\beta_T}{\alpha_T}\right), & \text{declare track confirmation} \\ \ln\left(\frac{\beta_T}{1-\alpha_T}\right) < LLR_{k+1} < \ln\left(\frac{1-\beta_T}{\alpha_T}\right), & \text{continue test} \\ LLR_{k+1} \leq \ln\left(\frac{\beta_T}{1-\alpha_T}\right), & \text{delete track} \end{cases}$$

where α_T and β_T are the probability of false-track confirmation and the probability of true-track termination, respectively.

3.4 Simulation Results

A simulation study has been performed to analyse the robustness and consistency of the IMMPDAF tracker in tracking moving people. The sensor used is capable of detecting objects up to 30 meters range. The observation is assumed to be static.

There are three motion models considered for tracking people, where each model depicts a distinct motion behaviour with a particular emphasis on turning motions. Model 1 refers to constant velocity with $\omega = 0$; Model 2 refers to left turns with $\omega = 1.4$ rad/s; and Model 3 refers to left turns with $\omega = -1.4$ rad/s. The mode transition-probability matrix used [48] for the simulation **T** is

$$\begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}.$$

The algorithm performance of the IMMPDAF tracker is evaluated by Monte Carlo experiments for 50 runs with random error and variance Q = diag(10,10). A simulation of two people with a manoeuvring movement in the opposite direction and having an occlusion on the first track is shown in Figure 3.1. The person is assumed to have been detected correctly.

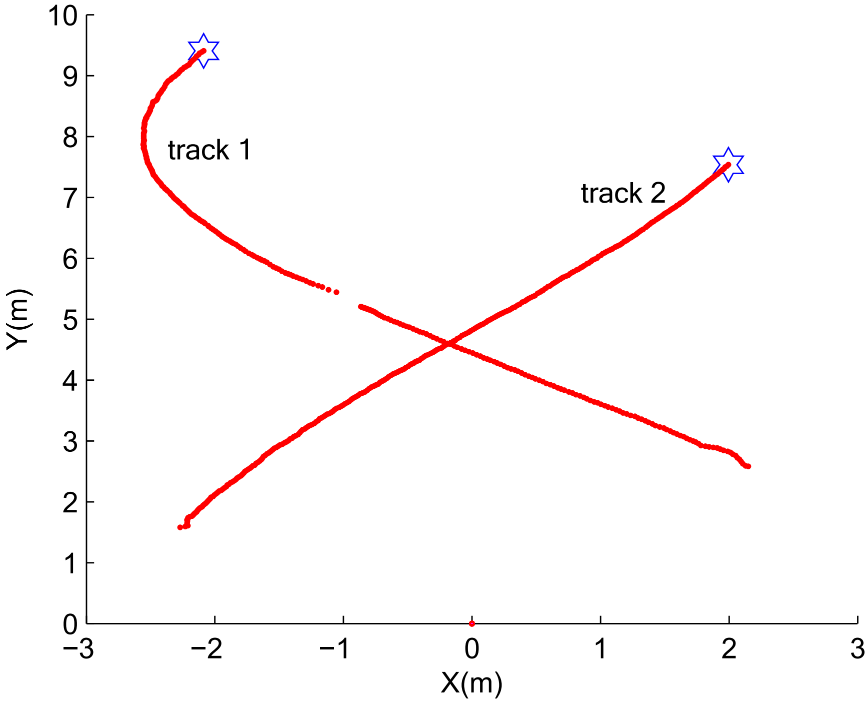


Figure 3.1: Tracking two people with occlusion

3.4.1 Performance Analysis with Normalised Estimation Error Squared (NEES)

The results of Normalized Estimation Error Squared (NEES) and RMS error on both the x and y axes are presented. The following equations are used to calculate both NEES and RMS error on both x and y axes:

$$\text{NEES} = \tilde{x}(k|k)^T S(k|k)^{-1} \tilde{x}(k|k) \tag{3.32}$$

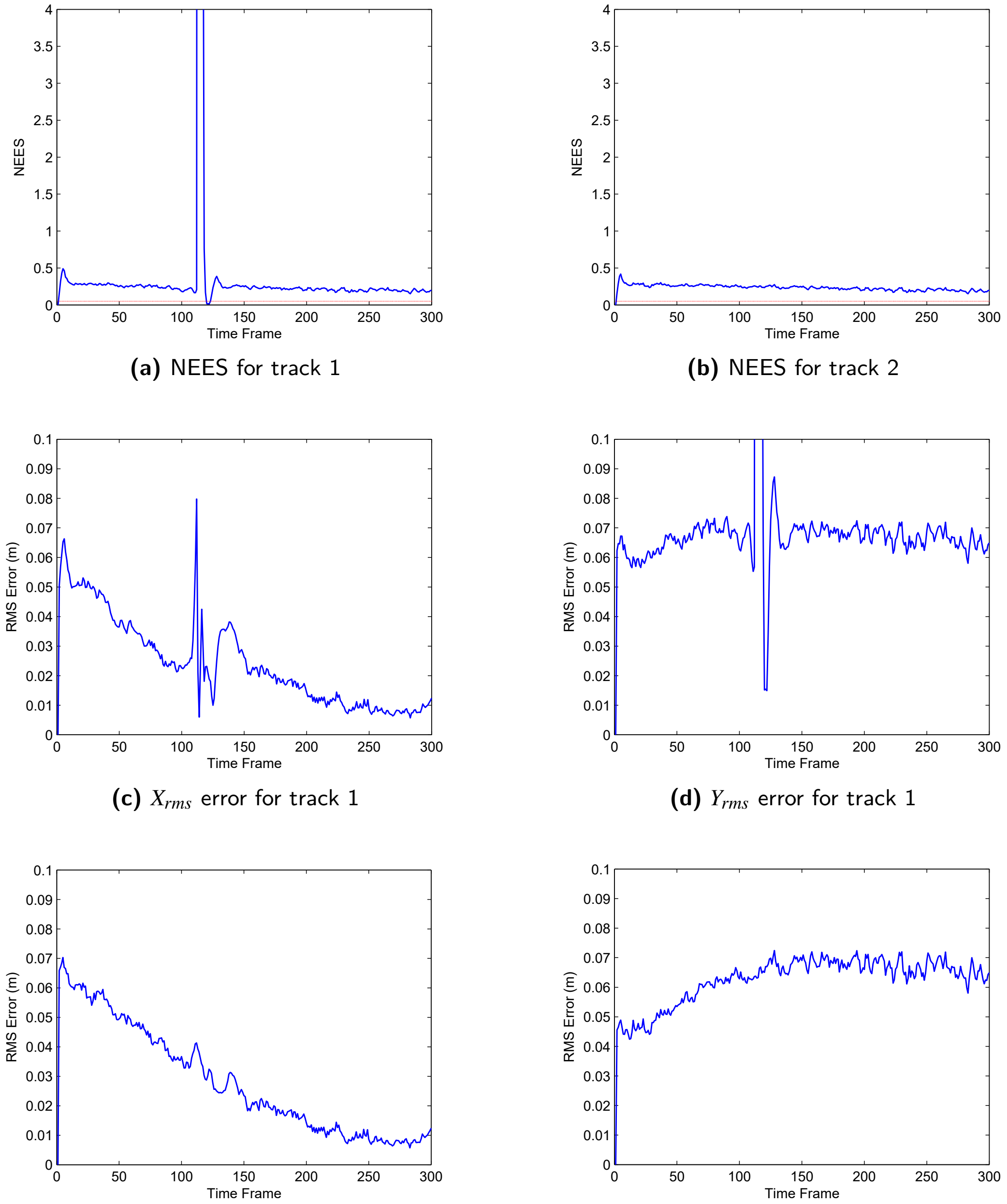
where $\tilde{x}(k|k) = x(k) - \hat{x}(k|k)$, $x(k)$ is the true position and $\hat{x}(k|k)$ is the predicted position by the tracker in the k-th time.

$$\text{RMS Error} = \sqrt{\frac{\sum_{k=1}^{n} \left(x(k) - \hat{x}(k)\right)^2}{n}}. \tag{3.33}$$

According to [52], the lower and upper limits of the two-sided 95% region are [0.05, 7.38]. The upper limit is of interest since the lower limit is practically near zero. In track 1, as shown in Figure 3.2a, the value of NEES is higher when the occlusion occurs. On the other hand, in track 2, as shown in Figure 3.2b, the value of NEES is between 0 and 1, which shows the IMM state estimation is consistent.

For RMS errors for 50 Monte Carlo runs in the x and y axes for tracks 1 and 2, respectively, which are presented from Figure 3.2c to Figure 3.2f, they are consistently small, except for sudden changes when the occlusion occurs. This shows that the tracking performance is undeviating for IMMPDAF except for the period of the occlusion.Â

The simulation of a person who moves in a circular motion was carried out. Figure 3.3a shows the pose of the human torso from various angles in a circular motion. The person is assumed to move at a constant velocity with a constant turn rate. The value of NEES for 50 Monte Carlo runs is within the lower and upper limits of the two-sided 95% region as referred to Figure 3.3b.



(a) NEES for track 1

(b) NEES for track 2

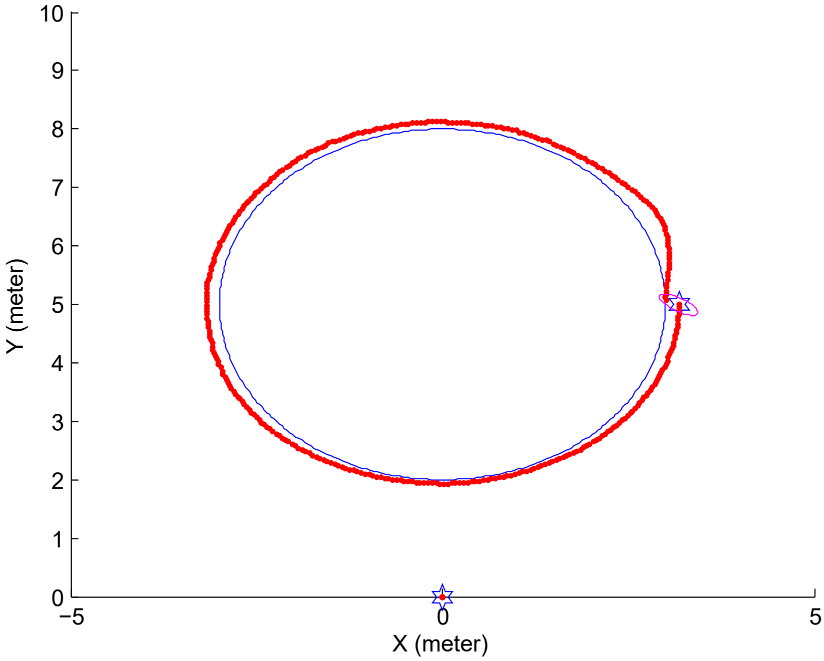
(c) X_{rms} error for track 1

(d) Y_{rms} error for track 1

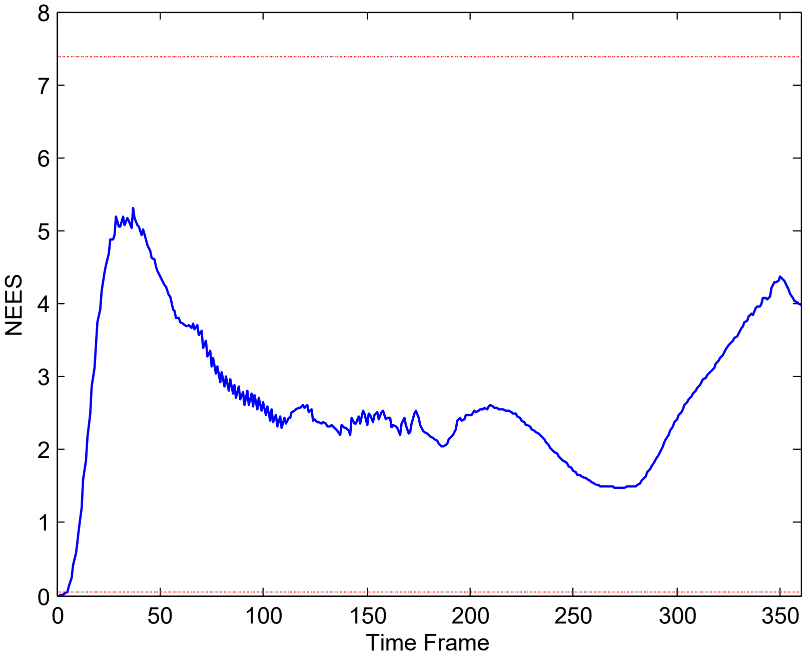
(e) X_{rms} error for track 2

(f) Y_{rms} error for track 2

Figure 3.2: NEES and RMS errors for tracks 1 and 2



(a) Red-line is an IMMPDAF tracker and blue-line is a groundtruth



(b) NEES of a person

Figure 3.3: Pose of a human torso that was being observed from various angles in a circular motion

3.4.2 Normalised Estimation Error Squared (NEES) with Multiple Tracking Occlusions

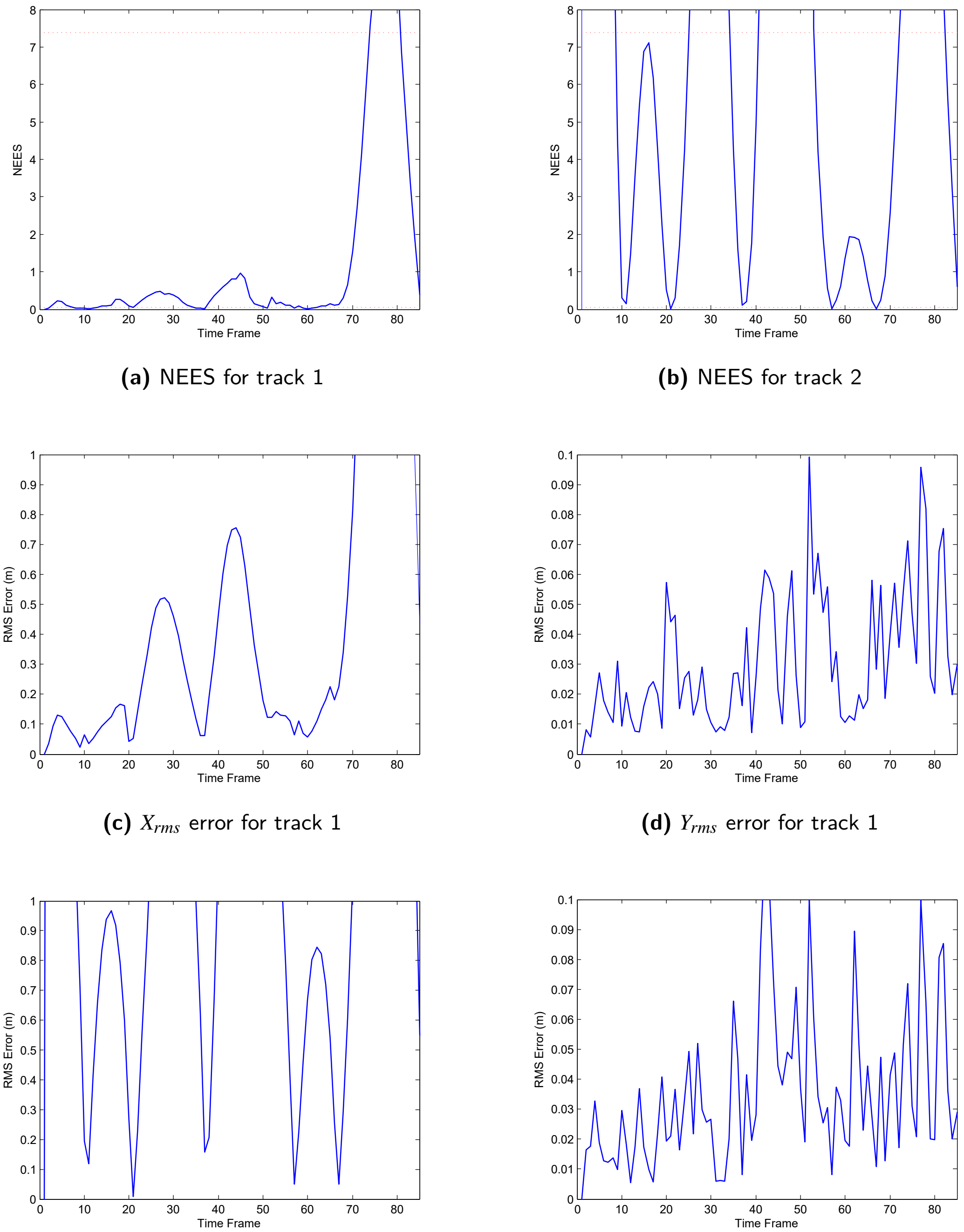
The simulation of two people walking in zig-zag (track 1) and curve (track 2) motions with magenta-line stated as the groundtruth is shown in Figure 3.5. Two people with labels as track 1 and track 2 are assumed to walk at a speed of 1.4 ms^{-1} with 3 modes: a value of $\omega = 0$ for a straight path, $\omega = 2.6$ rad/s for the right turns, and $\omega = -2.6$ rad/s for the left turns. The detection and tracking in this simulation include multiple occlusions when the two people cross each other with a sharp turn on the zig-zag route. From the observation, the IMMPDAF tracker has difficulties tracking the sharp turns and occlusions when referring to a visual comparison between the trackings and the groundtruths on the plot shown in Figure 3.5.

By referring to Figure 3.4a and Figure 3.4b, the values of NEES for 50 Monte Carlo runs are outside of the lower and upper limit of the two-sided of 95% region on various time frames, as it can be seen in track 1 from time frame 70 to 80 and in track 2 from time frame 1 to 9, 25 to 34, 41 to 53 and 72 to 82.

Referring to those time frames, it can be seen that the RMS error for 50 Monte Carlo runs in the x and y axes for track 1 and track 2, respectively, as shown from Figure 3.4c to Figure 3.4f are significantly large, especially on the X axes, where it exceeds the value of 1 on various time frames. Those values on NEES and RMS errors show that the tracking performance is inconsistent, which leads to inaccurate tracking ability. Thus, the result on tracking performance in this simulation emphasises that the state estimation errors of the IMMPDAF tracker are inconsistent with the filter-calculated covariances.

3.5 Experimental Results

The experiments on tracking three people that were tracked in a common area of the faculty were carried out. Fig. 3.6 shows the tracking performances. The segway robot was used in a stationary position to monitor the environment. To evaluate the



(a) NEES for track 1

(b) NEES for track 2

(c) X_{rms} error for track 1

(d) Y_{rms} error for track 1

(e) X_{rms} error for track 2

(f) Y_{rms} error for track 2

Figure 3.4: NEES and RMS errors for tracks 1 and 2

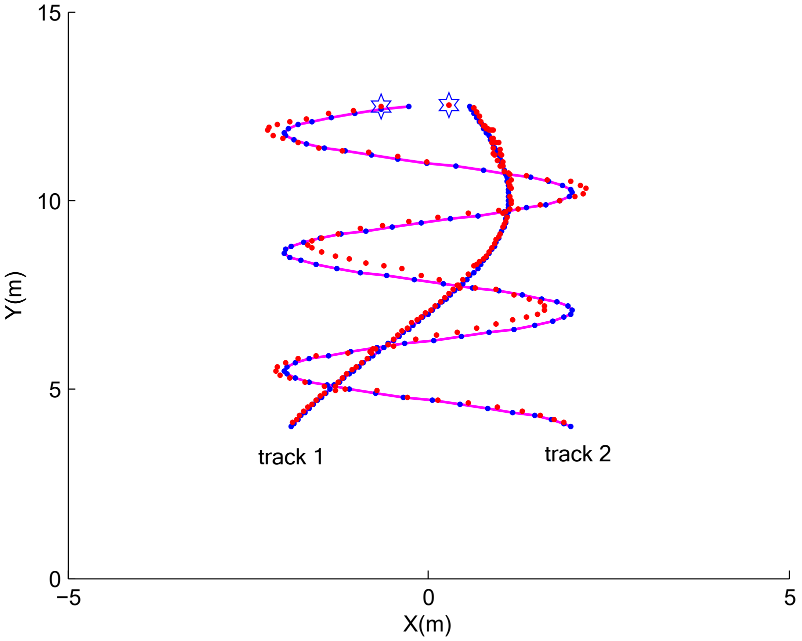


Figure 3.5: Tracking of two people with multiple occlusions (red-dot is an IMMPDAF tracker, and magenta-line with blue-dot is a groundtruth)

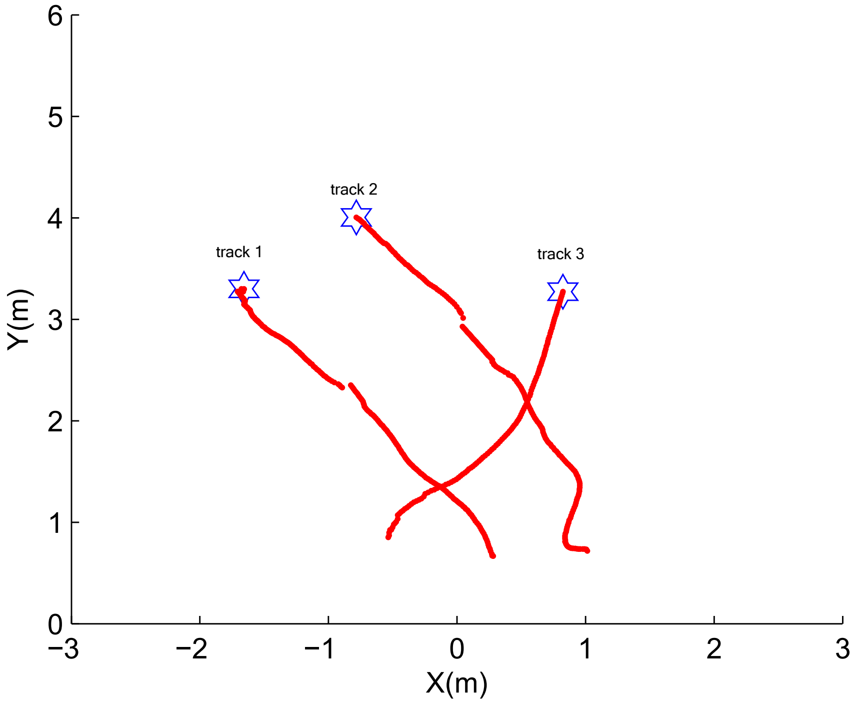


Figure 3.6: The tracking performance of three people

performance of this tracker, the results of Normalized Innovation Squared (NIS) [52] are presented subsequently.

The equation of NIS is

$$\text{NIS} = \tilde{x}(k|k)^T S(k|k)^{-1} \tilde{x}(k|k) \tag{3.34}$$

where $\tilde{x}(k|k) = x(k) - \hat{x}(k|k)$, $x(k)$ is the position from the sensor measurements and $\hat{x}(k|k)$ is the predicted position by the tracker in the k-th time.

However, sensor fusion, which is the process of merging data or information from several sensors to create a more precise and thorough picture of an environment or a system, can considerably help in overcoming difficulties caused by occlusion in various ways.

The idea of placing multiple sensors in various locations can cover various areas of the environment. Other sensors may still have a clear line of sight to the objects of interest if one sensor's view is obscured by occlusion. Having multiple sensors ensures that the system can still rely on data from other sensors even if one sensor is temporarily blinded.

Data fusion from various sensors (such as cameras, LiDAR, radar, or ultrasonic sensors) can provide more accurate and dependable object tracking and localization. By integrating information from sensors with distinct sensing principles (such as vision-based and depth-based sensors), the system can estimate the positions and movements of objects even when they are partially or completely obscured from one sensor's view.

Sensors can provide temporal data, capturing how objects move and change over time. By integrating temporal data from multiple sensors, the system can predict the probable locations of occluded objects based on their previous movements, enabling more accurate predictions.

3.5.1 Occlusion Handling with Probabilistic Data Association Filter (PDAF)

Section 3.3.2 discusses the integration of the people detection part into the IMM-based temporal tracking algorithm. Figure 3.7 shows the tracking of two people (T1 and T2) using a stationary observer. The motions of T1 and T2 caused an occlusion, where T1 disappeared from observations. However, the predictions of the IMM tracker allowed for the re-association of the track with T1 for further tracking.

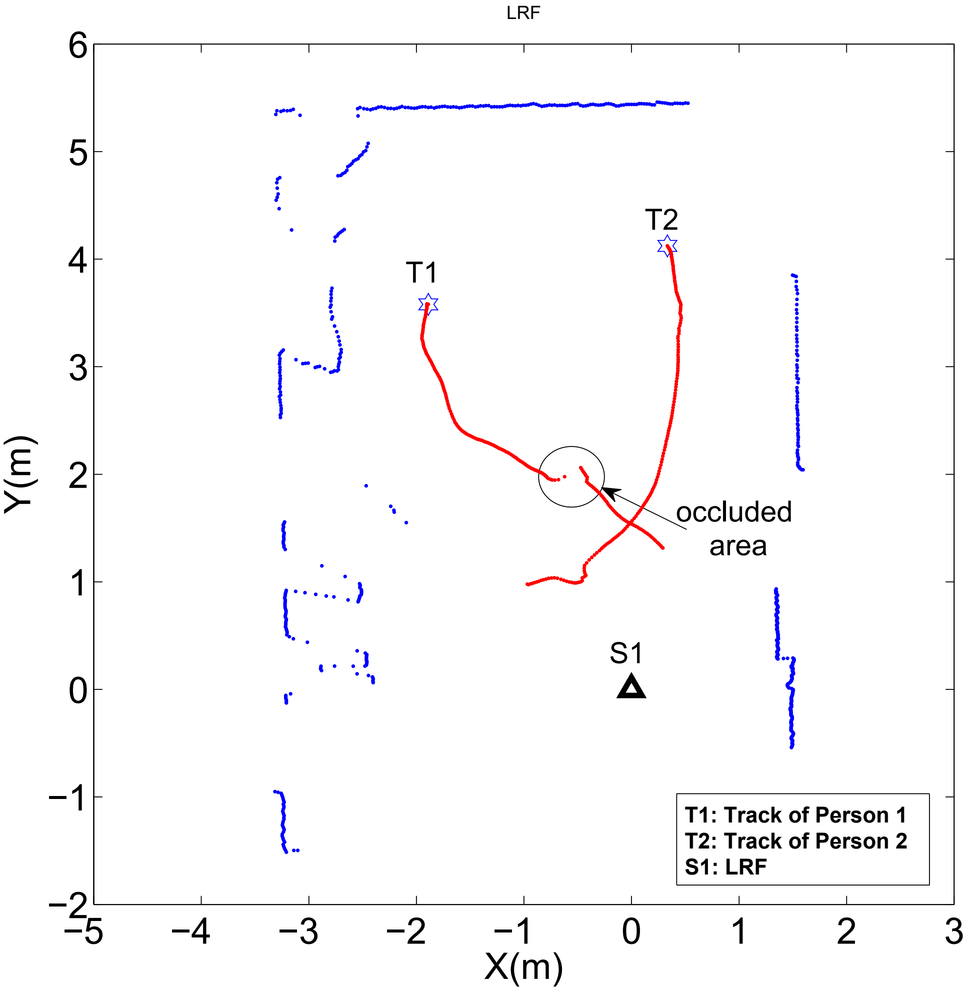


Figure 3.7: People track results with a stationary observer (T1 and T2 denote the tracks of two people)

Figure 3.8 shows the results of tracking two people with a dynamic observer. The motions of T1 and T2 again cause the occlusion where T2 disappears from the observation, and the scenario is quite similar to that of the stationary observer. If T2 has the possibility of being terminated and disappears for a long time, it will reappear as a new target.

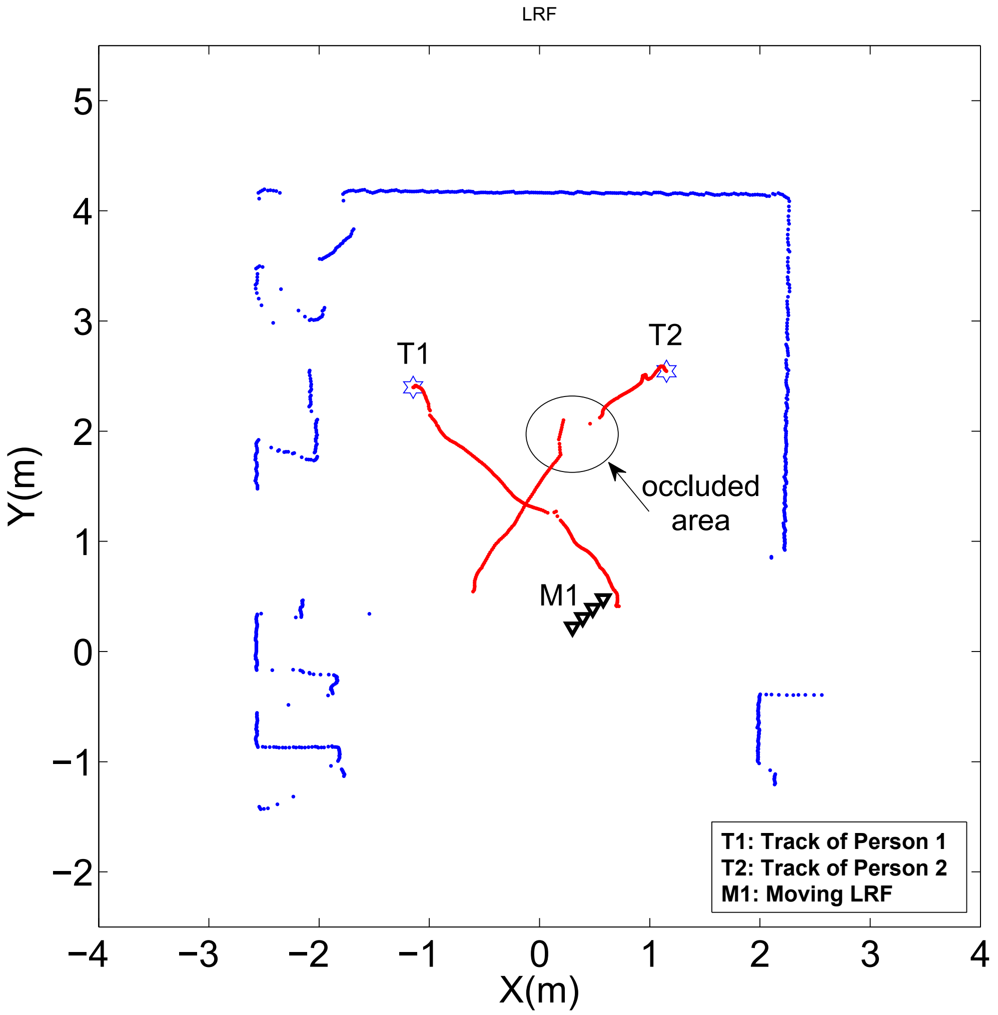


Figure 3.8: People track results with a moving observer (T1 and T2 denote the tracks of two people)

The process of determining the tracks is based on the log-likelihood ratio (LLR), which is shown in Figure 3.9. A new track is confirmed if the LLR is higher than an upper threshold, and a track is deleted if it falls below a lower threshold (as defined in Section 3.3.2).

The tracks that have been occluded for a long time have the possibility of being

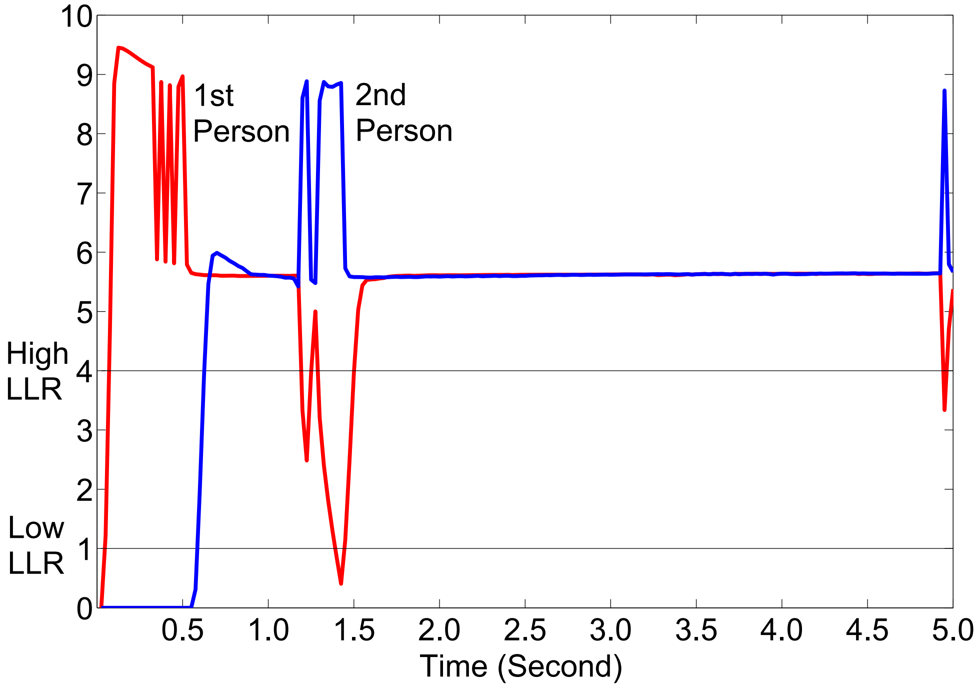
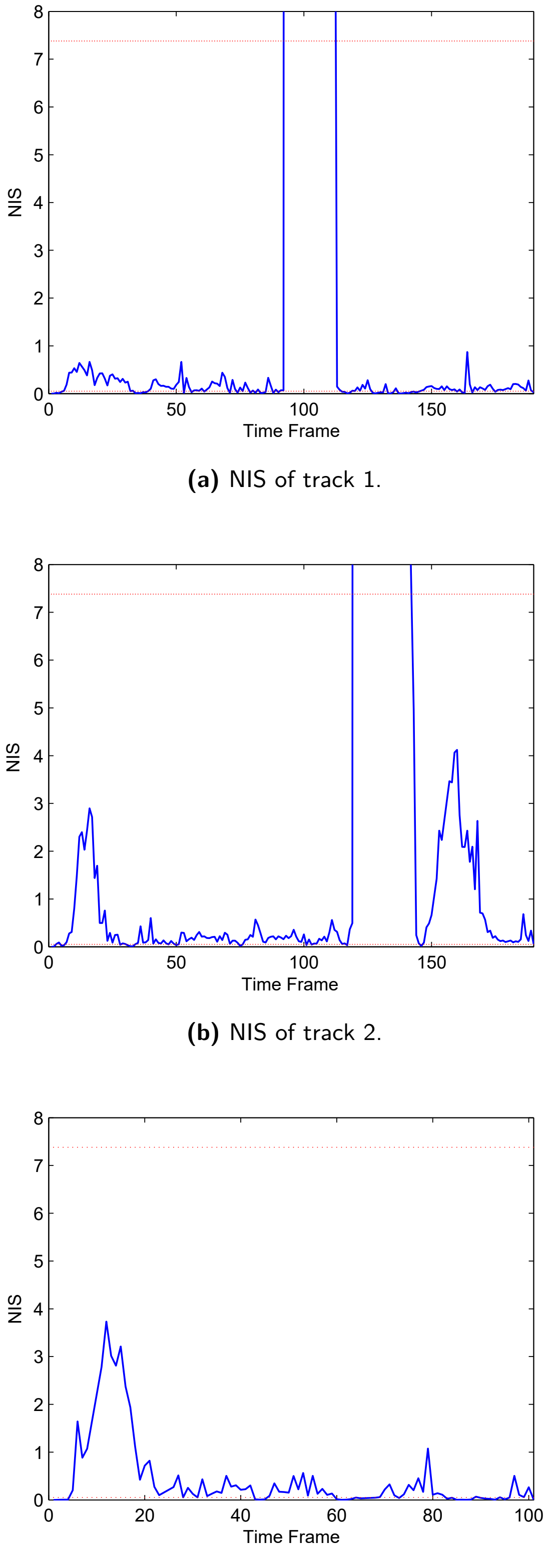


Figure 3.9: The log-likelihood ratio (LLR) of two people tracking

deleted and re-appearing as new tracks. It is reasonable as long as the application does not require the identification and tracking of a particular individual.

3.5.2 Performance Analysis with Normalised Innovation Squared (NIS)

In this experiment, there are two occlusions where the tracks are lost for a short period: track 1 at a time frame of 100 to 115 and track 2 at a time frame of 120 to 140 as shown in Fig. 3.10a and Fig. 3.10b. In Fig. 3.10c, there is no occlusion, and thus the value of NIS is within the upper and lower limits.



(a) NIS of track 1.

(b) NIS of track 2.

(c) NIS of track 3.

Figure 3.10: Normalized Innovation Squared (NIS) of tracks 1,2 and 3

For overall performance, NIS values for three tracks fall within the 95% probability region. Therefore, the tracker's consistency is proven. However, when two occlusions occur, the track is lost for a short period. Therefore, for this reason, it can be confirmed that the current tracker has difficulties handling any target with a long period of occlusion.

3.6 Conclusion

This chapter presents a review of the tracking techniques and methods of data association and addresses the performance of the tracker with simulated and real data. The review discussed and listed various types of data association techniques and trackers. The consistency of the tracker was also discussed and analysed using Monte Carlo simulation based tests. From the simulations and tests, even though most of the NEES and NIS of the tracks fall within the 95 % probability region, the tracker still has difficulties handling targets when there are occlusions where the NEES and NIS are out of the lower and upper limits of the two-sided 95% region.

The next chapter discusses the implementation of Gaussian Processes with the tracker to improve temporal prediction of the target and effectively maintain the optimum amount of training data.

Chapter 4

Non-Parametric People Tracking

In general, a model can be developed to describe human motion patterns that has the capability to enhance tracking performance even with long-term occlusions. One way to effectively learn these patterns is to apply Gaussian Processes (GP). However, with the increase in the amount of training data over time, the GP becomes computationally expensive. In this work, a Mutual Information (MI) based technique along with the Mahalanobis Distance (MD) measure to keep the most informative data while discarding the least informative data has been proposed. The algorithm is tested with data collected in an office environment with a Segway robot equipped with a laser range finder. It leads to more than 90% data reduction while keeping the limit of average RMS errors. A GP based Particle filter tracker for long-term people tracking with occlusions was implemented. The comparison results with the Extended Kalman Filter based tracker show the superiority of the proposed approach.

4.1 Introduction

An element on state estimation of a dynamical system mainly in human motion is a niche problem in various applications in security systems and robotics. The most successful and widely used techniques for these purposes are Bayesian filters such as particle filters or Kalman filters. Bayes filters repeatedly estimate posterior probability distributions over the state of a system, where the main components are the prediction and observation models. It probabilistically defines the temporal progression of the process and the measurements captured by the sensors. Nonetheless, one can estimate the parameters and noise components of these models from manual approximation or training data, as they represent the ongoing processes in a parametric manner [55]. Despite the effectiveness of such parametric models, it is hard to find accurate parametric models since their predictive capabilities may be limited to certain aspects of the process. For instance, it is difficult to translate a model of human motion for people tracking [56], object tracking using visual sensory data [57], deep learning human tracking recognition [58] and human legs motion tracking using a Kalman filter-based tracker [59]. These systems are based on parameters with complex relationships among various features. Some conditions that are associated with the tracked object make it difficult to interact with each anterior progression, such as the position and speed of two feet [60; 61].

Due to the limitations of parametric models, non-parametric models such as Gaussian process regression models [62] can be substituted to learn prediction and observation models for dynamical systems. GPs models have been successfully tested to solve the problem of learning predictive state models [63; 64]. GP regression models provide uncertainty estimates for their predictions, which can be incorporated into particle filters as observation models [63] or for improved sampling distribution [65].

4.2 Particle Filter

There are a number of ways of estimating the target's position, as reported in the literature. One of the most successful and promising approaches is using particle filters, which have solved several hard perceptual problems in robot visions. Particle filters are approximate techniques for calculating posterior in partially observable controllable Markov chains with discrete time. They are usually highly geometric and generalize classical robotics notions such as kinematics and dynamics by adding non-deterministic noise. If states, controls and measurements are discrete, the Markov chain is equivalent to hidden Markov models (HMM) [66; 67] and can be implemented exactly. The posterior requires space exponential in the number of state features, though more efficient approximations exist that can exploit conditional independence that might exist in the model of the Markov chain [68]. Robotic applications typically utilize particle filters in continuous state spaces. For a continuous state space, closed form solutions for calculating are only known for highly specialized cases. A common approximation in non-linear non-Gaussian systems is to linearize the actuation and measurement models. If the linearization is obtained via a first-order Taylor series expansion, the result is known as the extended Kalman filter or EKF [33; 69–71]. Unscented filters often obtain a linear model through non-random sampling [72]. However, cases where the Gaussian-linear assumption is a suitable approximation confine all these techniques. Particle filters address the more general case of nearly unconstrained Markov chains. Particle filters are attractive because they can be applied to nearly any probabilistic detection and tracking model that can be formulated as a Markov chain. Additionally, particle filters at any moment do not require a fixed computation time and their accuracy generally increases with the available computational resources.

In the context of these works, particle filters estimate the posterior over unobservant state variables from sensor measurements. For instance, measurements are taken from a laser range finder (LRF) or laser detection and ranging (LiDAR) where the state refers to the position of the human torso relative to its environment along with the number and location of objects in the region of interest.

4.3 Human Motion Prediction and Learning

People tracking is one of the most important aspects for mobile robots to be efficiently deployed in populated environments. Most techniques used for people tracking are based on weak assumptions about human motion such as the constant velocity motion model or the Brownian model which predicts future states simply based on the history of past states. In other words, human motion is driven by the physical and social constraints of the environment. However, even over a short period, human motion follows more complex and non-linear patterns when influenced by various factors, such as the intentional objective, other people and objects along the path in the environment, and following social rules. By considering these requirements, it is proposed to explore more sophisticated motion models for people tracking since humans may frequently undergo lengthy occlusion events.

Bruce et al. [56] have proposed better human motion models for people tracking, where the robots initially learn goal locations in the environment from people trajectories. Human motion is predicted along the paths computed by a planner from the actual location of the person to the estimated goal location. Liao et al. [73] extract a Voronoi graph from a map of the environment and constraint the state of the people to lie on the edges of the graph. The motion of people is then predicted along those edges, following the topological shape of the environment. Bennewitz et al. [74] learn typical motion patterns that people follow in an environment where this approach accumulates trajectories of people and combines them to motion patterns using Expectation-Maximization (EM) clustering. Each motion pattern is used to derive a Hidden Markov Model (HMM) for the mobile robot to predict the motion of people.

Models for pedestrian dynamics have also been developed and applied in communities, such as in quantitative sociology or spatial cognition. These models are used for crowd simulation, evacuation dynamics or building design. The first model which is typically deterministic and force-based employs fluid dynamic and gas kinetic models in which people are considered particles with their motion being described by a fluid-dynamic equation. The second model is based on a ruled-based dynamical model which is usually described by a set of rules specifying the probability of moving to neighboring cells and discrete cellular automata, which discretizes space into cells that can be occupied only by one person. However, motion models based on their discrete nature cannot be readily applied within a probabilistic tracker that requires proper propagation error in the prediction state.

Helbing et al. [75] proposed the concept of social forces or social fields, where the forces model different aspects of motion behaviours, such as the motivation of people to reach a goal, the repulsive effect of walls and other people as well as physical constraints.

In this work, a people tracker with gaussian process regression with GP prediction and observation models can be combined with particle filters which have been introduced by Ko et al. [63] was integrated. The resulting GP-Particle Filter (GP-PF) inherits features of GP regression. The underlying models and all their parameters can be learned from training data using non-parametric regression. Incorporating such models into the GP regression typically allows the filter to learn the parameters from significantly less training data. When the process of tracking enters areas in which not enough training data is available, the filter naturally increases its uncertainty estimation. However, GP becomes inefficient when large training data sets are available. Due to this reason, it was proposed to keep the most informative data while discarding the least informative data by using the Mutual Information (MI) based technique along with the Mahalanobis Distance (MD).

4.4 Gaussian Process

A Gaussian Process is a powerful non-parametric technique for learning regression functions from sample data, which contains a collection of random variables with any subset of them having a joint Gaussian distribution [62; 76] and represents posterior distributions over functions based on training data. First, assume a set of training data, $D = \langle X, \mathbf{y} \rangle$, where $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ is a matrix containing d-dimensional input examples and $\mathbf{y} = [y_1, y_2, ..., y_n]$ is a matrix containing scalar output. Gaussian Process assumes that the data is derived from a noisy process whose regression output is modeled using a noisy version of the function,

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon, \tag{4.1}$$

where ε is zero mean additive Gaussian noise with a variance of σ_n^2 . With training data $D = \langle X, \mathbf{y} \rangle$ and a test input, $\mathbf{x}_*$, a GP defines a Gaussian predictive distribution over the output $\mathbf{y}_*$ with mean

$$\text{GP}_\mu(\mathbf{x}_*, D) = \mathbf{k}_*^T [K + \sigma_n^2 I]^{-1} \mathbf{y} \tag{4.2}$$

and variance

$$\text{GP}_\Sigma(\mathbf{x}_*, D) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T [K + \sigma_n^2 I]^{-1} \mathbf{k}_*. \tag{4.3}$$

$\mathbf{k}_*$ is a vector defined by kernel values between the test input $\mathbf{x}_*$ and the training inputs $\mathbf{x}$. K is the $n \times n$ kernel matrices of training input values $\mathbf{k}[m] = k(\mathbf{x}_*, \mathbf{x}_m)$ and $K[m,n] = k(\mathbf{x}_m, \mathbf{x}_n)$. The variance GP_Σ, which is the prediction uncertainty, depends on the process noise and the correlation between the test input and the training data. The widely used kernel function, squared exponential is selected for this process, which is given by,

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}')W(\mathbf{x}-\mathbf{x}')^T} \tag{4.4}$$

where σ_f^2 is the signal variance. The diagonal matrices W contain the length scales for each input dimension.

The GP parameters describing the kernel function and the process noise, respectively are called hyperparameters of the Gaussian Process. These hyperparameters are learned by maximizing the log likelihood of the training data using numerical optimization techniques such as conjugate gradient decent [62]. Consider a d-dimensional trajectory V that has $|V|$ number of points. If observation is made on a set of points, $A \subset V$, based on the GP model, it can predict the value at any point $y \in V \backslash A$. Let Z_A denote a set of values at the finite set A, and z_y denote a value at y. In probabilistic terms, the conditional distribution is derived at the predicted point of y where Z_A is given as follows:

$$\mu_{y|A} = \mu_y + \Sigma_{yA}\Sigma_{AA}^{-1}(Z_A - \mu_A) \tag{4.5}$$

$$\sigma_{y|A}^2 = k(y,y) - \Sigma_{yA}\Sigma_{AA}^{-1}\Sigma_{Ay} \tag{4.6}$$

where Σ_{yA} is a covariance vector with one entry for each $\mathbf{x} \in A$ with value $k(y,\mathbf{x})$; $\mu_{y|A}$ and $\sigma_{y|A}^2$ are conditional mean and variance at y; μ_A is a mean vector of Z_A ; and Σ_{AA}^{-1} is a covariance matrix of Z_A with every entry calculated by $k(\mathbf{x},\mathbf{x})$.

4.5 Data Selection and Management

In regular practice, it is necessary to incorporate all the samples into the training phase of the GP. However, it becomes inefficient whenever the GP needs to be trained to accommodate new observations. In the people tracking scenario, the motion models are learned, and they need to be adapted in time to accommodate variations. Thus, when the new observations are available, the GP needs to be learned with an increasing number of samples. Intuitively, the new observations need to be included, if only they are informative. This problem is proposed to be resolved using a Mutual Information (MI) based strategy and Mahalanobis Distance (MD) based criteria. The MI picks the most informative measurements that are given by the whole scan and uses them to represent the GP surface. Whenever new data is available, MD is calculated between the new measurement and the GP. If it is within the 95 % of the confidence interval, the new measurement is discarded as the GP is already capable of representing the data. However, if the MD is greater than 95 % of the confidence interval, the data is not representative of the GP and hence, it needs to be included in order to represent the new GP. This process will govern data management and adapt to new scenarios. The process flow for data selection in Gaussian Processes variables is shown in Figure 4.1.

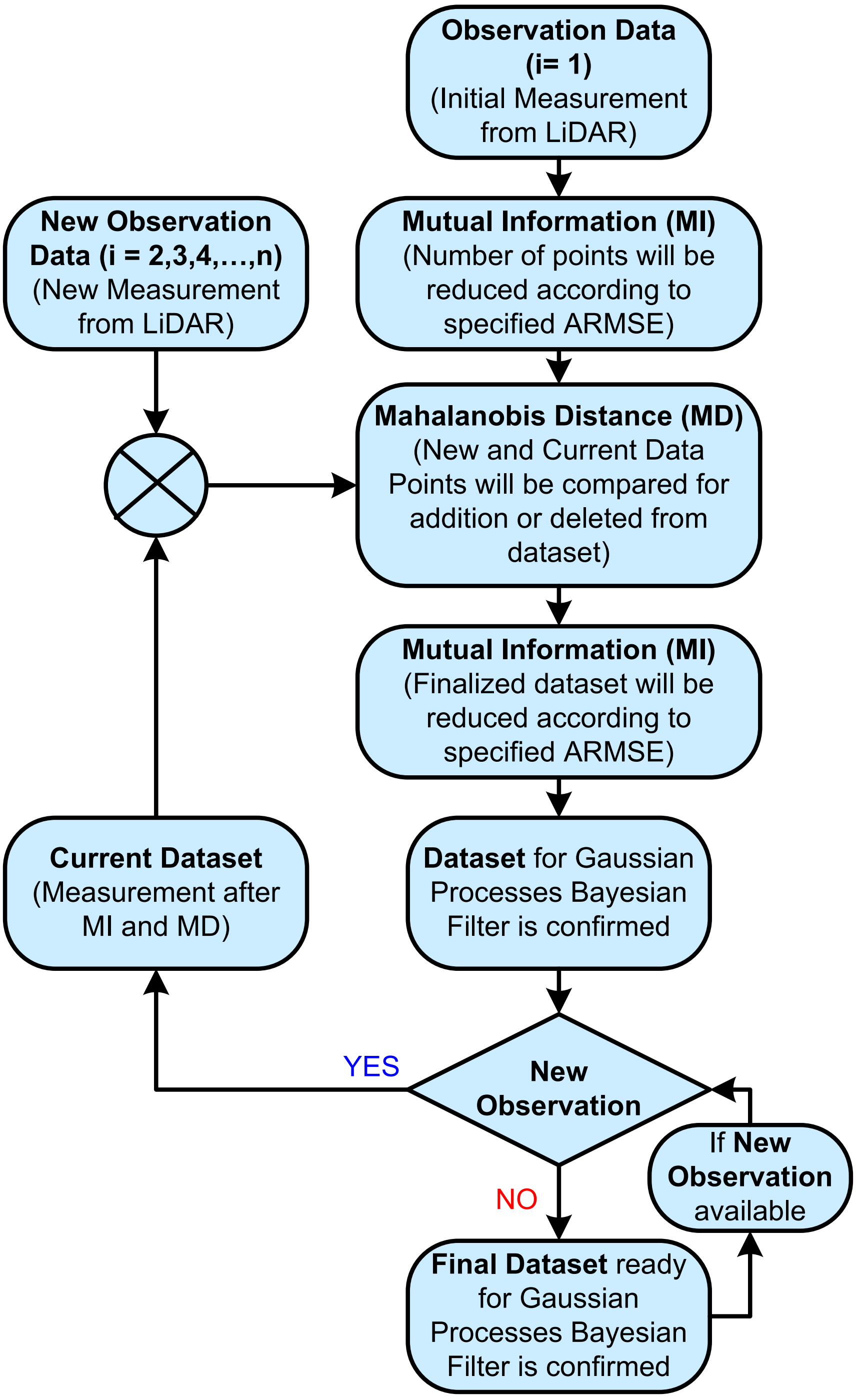

Observation Data (i= 1)
(Initial Measurement from LiDAR)
Mutual Information (MI)
(Number of points will be reduced according to specified ARMSE)
New Observation Data (i = 2,3,4,...,n)
(New Measurement from LiDAR)
Mahalanobis Distance (MD)
(New and Current Data Points will be compared for addition or deleted from dataset)
Mutual Information (MI)
(Finalized dataset will be reduced according to specified ARMSE)
Current Dataset
(Measurement after MI and MD)
Dataset for Gaussian Processes Bayesian Filter is confirmed
YES
New Observation
If **New Observation** available
NO
Final Dataset ready for Gaussian Processes Bayesian Filter is confirmed

Figure 4.1: The process flow on the selection of data points in Gaussian Processes.

4.5.1 Mutual Information

In the previous discussion, the most informative data points were selected based on the Mutual Information algorithm [77]. In order to find k best points in the whole trajectory V, it starts with an empty set of locations $A = \phi$ and greedily adds placements in sequence until $|A| = k$ based on the angle or magnitude of displacement of each point. The algorithm chooses the next point that produces the maximum increase in mutual information. More specifically, the MI between the subset A and the rest of the trajectory $V \backslash A$ can be formulated by:

$$F(A) = I(A; V \backslash A)$$

Once $y \in V \backslash A$ is chosen and added to A, the variation of MI can be calculated by:

$$F(A \cup y) - F(A) = = H(A \cup y) - H(A \cup y|\bar{A}) - [H(A) - H(A|\bar{A} \cup y)] = H(y|A) - H(y|\bar{A}) \tag{4.7}$$

4.5.2 Mahalanobis Distance

The MD is used to decide the importance of new data to be incorporated in the GP learned model. As per the previous discussion, this will allow the GP to represent dynamically changing environments and hence improve its adaptability. Assume a new measurement value of mean μ_{x_m} and variance σ_{x_m} Â at a location $\mathbf{x}_i$, where $\mathbf{x} = \langle x, y \rangle$, was received. The GP can now predict the mean μ_{x_p} and variance σ_{x_p} at that location as well. Thus, the MD can be calculated as

$$d(\mathbf{x}) = \sqrt{\frac{(\mu_{\mathbf{x}_m} - \mu_{\mathbf{x}_p})^2}{\sigma^2_{\mathbf{x}_m} + \sigma^2_{\mathbf{x}_p}}}. \tag{4.8}$$

The measurement used in this application is one dimensional and therefore, chi-square tables the threshold for $d(\mathbf{x})$ to be within a 95 % confidence interval, which can be chosen as 3.84 [78].

4.6 Gaussian Process - Particle Filter (GP-PF)

A particle filter (PF) is a very flexible Sequential Monte Carlo (SMC) approach that allows the implementation of a recursive Bayesian filter through Monte Carlo simulations [79–82]. It can be applied to a wide range of dynamic state-space models: linear and non-linear, Gaussian and non-Gaussian, stationary and dynamic, discrete and continuous. Particle filters require learning prediction and observation models, which can be achieved by directly applying Gaussian process regression. In this work, it is restricted to training prediction models as at a later stage, the observer is dynamic, making it difficult to learn observation models. The prediction model maps the state and control, $(\mathbf{x}_k, \mathbf{u}_k)$ to the state transition, $\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. The next state is found by simply adding the state transitions to the previous state. Therefore, appropriate forms of prediction and observation training data sets are given by,

$$D_p = \langle (X, U), X' \rangle \tag{4.9}$$

where X is a matrix containing the locations and $X' = [\Delta x_1, \Delta x_2, ..., \Delta x_k]$ is a matrix containing transitions made from those states when applying the controls stored in U.

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \approx N(\text{GP}_\mu([x_{k-1}, \mathbf{u}_{k-1}], D_p), \text{GP}_\Sigma([x_{k-1}, \mathbf{u}_{k-1}], D_p)) \tag{4.10}$$

Â where $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ represents the probability of the variable $\mathbf{x}_k$ given the values of $\mathbf{x}_{k-1}$ and $\mathbf{u}_{k-1}$. $N(\text{GP}_\mu([x_{k-1}, \mathbf{u}_{k-1}], D_p), \text{GP}_\Sigma([x_{k-1}, \mathbf{u}_{k-1}], D_p))$ represents a normal distribution N characterized by its mean $\text{GP}_\mu([x_{k-1}, \mathbf{u}_{k-1}], D_p)$ and variance $\text{GP}_\Sigma([x_{k-1}, \mathbf{u}_{k-1}], D_p)$. Thus, $\text{GP}_\mu([x_{k-1}, \mathbf{u}_{k-1}], D_p)$ represents the mean of the Gaussian Process, which is evaluated at the point $[x_{k-1}, \mathbf{u}_{k-1}]$ using the data D_p. Similarly, $\text{GP}_\Sigma([x_{k-1}, \mathbf{u}_{k-1}], D_p)$ represents the covariance of the Gaussian Process at the same point. The basic task of a particle filter is to represent posteriors over the state $\mathbf{x}_k$ by setting X_k of weighted samples:

$$X_k = \{\langle \mathbf{x}_k^m, w_k^{(m)} \rangle | m = 1, ..., M\}. \tag{4.11}$$

Here each $\mathbf{x}_k^m$ is a sample and each $w_k^{(m)}$ is a non-negative numerical factor called importance weight. Thus, $\mathrm{GP}_\mu([x_{k-1}, \mathbf{u}_{k-1}], D_p)$ is the short form of the Gaussian represented by $(\mathrm{GP}_\mu([x_{k-1}, \mathbf{u}_{k-1}], D_p), \mathrm{GP}_\Sigma([x_{k-1}, \mathbf{u}_{k-1}], D_p))$.

Note that the covariance of this prediction is typically different for each sample, taking the local density of training data into account. The complete step can be found in [63].

4.7 Gaussian Processes - Extended Kalman Filter (GP-EKF)

An incorporation of GP models into the EKF requires a linearization of the GP function, which follows the interpretation that was specified by A. Girard et al [83] besides utilising GP mean and covariance. The derivative of the GP mean function for each output dimension can be described as:

$$\frac{d(\mathrm{GP}_\mu)(x_*, D)}{d(x_*)} = \frac{d(\mathbf{k}_*)^T}{d(\mathrm{x}_*)}[K + \delta_n^2 I]\mathrm{y}. \tag{4.12}$$

Note that $\mathbf{k}_*$ is the vector of kernel values between query input x_* and the training inputs X.

The partial derivatives of the kernel vector function are

$$\frac{d(\mathbf{k}_*)}{d(\mathrm{x}_*)} = \begin{bmatrix} \frac{d(k(\mathrm{x}_*, \mathrm{x}_1))}{d(\mathrm{x}_*[1])} & \cdots & \frac{d(k(\mathrm{x}_*, \mathrm{x}_1))}{d(\mathrm{x}_*[d])} \\ \vdots & \ddots & \vdots \\ \frac{d(k(\mathrm{x}_*, \mathrm{x}_n))}{d(\mathrm{x}_*[1])} & \cdots & \frac{d(k(\mathrm{x}_*, \mathrm{x}_n))}{d(\mathrm{x}_*[d])} \end{bmatrix}. \tag{4.13}$$

where d is the dimensionality of input space and n is the number of training points. The partial derivatives are depend on the type of kernel function. For the squared exponential kernel, the expression will be as:

$$\frac{d(k(\mathrm{x}_*,\mathrm{x}))}{d(\mathrm{x}_*[i])} = -W_{ii}\sigma_f^2(\mathrm{x}_*[i]-\mathrm{x}[i])\exp^{-\frac{1}{2}(\mathrm{x}_*-\mathrm{x})W(\mathrm{x}_*-\mathrm{x})^T}. \tag{4.14}$$

Stacking l Jacobian vectors together can determine the full $l \times d$ Jacobian of a prediction or observation model, which is one for each of the output dimensions. A comprehensive explanation can be found in [63].

4.8 Experimental Analysis

The experiments were carried out in a common area of the university, as shown in Figure 4.2 and Figure 4.12. In these experiments, a subject is a person who is walking around a cubicle multiple times and walking multiple times in four trajectories while the robot is stationarily observing the subject. A trace of trajectories containing 20 paths in circular motion is shown in Figure 4.3. Meanwhile, a trace of 4 trajectories contains 10 paths in one direction, as shown in Figure 4.13.

The method of detecting people applied prior to tracking is based on laser data taken at the torso height of a human, followed by an extraction of significant features and a classification process as shown in chapter 2. Once a person was detected based on laser data, it was then used in GP modelling. The main idea was to keep the average root mean square error (ARMSE) of prediction below 5 cm, as the environment contains corridors of width ranging from 130 to 150cm. Finally, the GP-PF was used for tracking people.

In this experiment, the training data had been processed in three stages. Firstly, applying Mutual Information Approach (MIA) on the first set of data in the trajectory to determine the least number of points that could be used to represent the GP with the given ARMSE. Secondly, the MD was applied to each new measurement to ensure

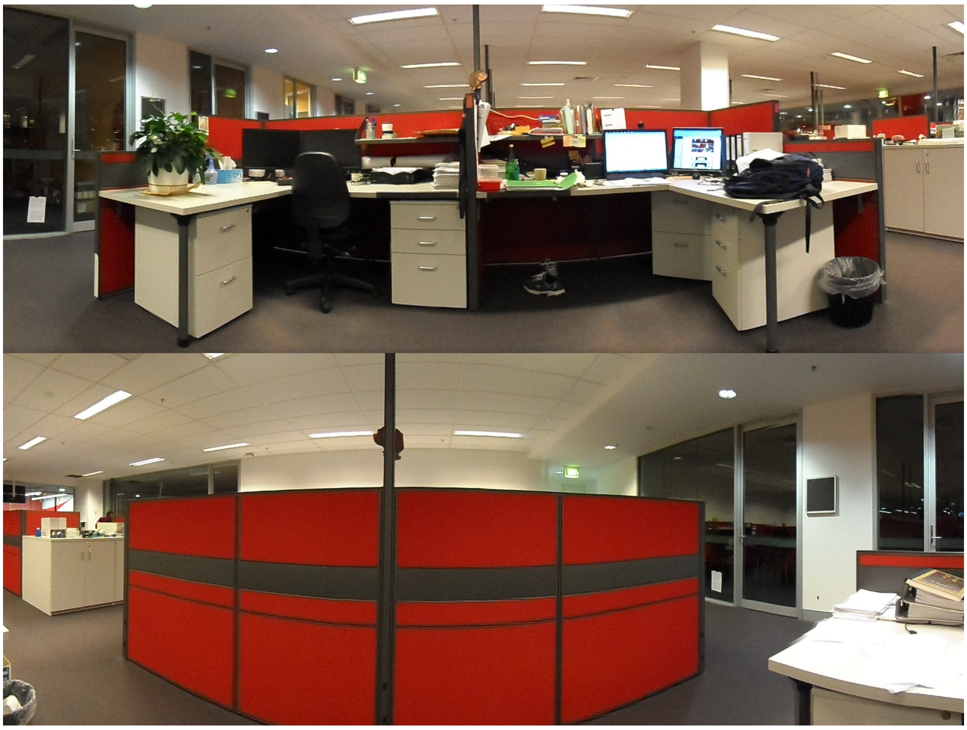


Figure 4.2: Office environment in the Centre of Autonomous Systems

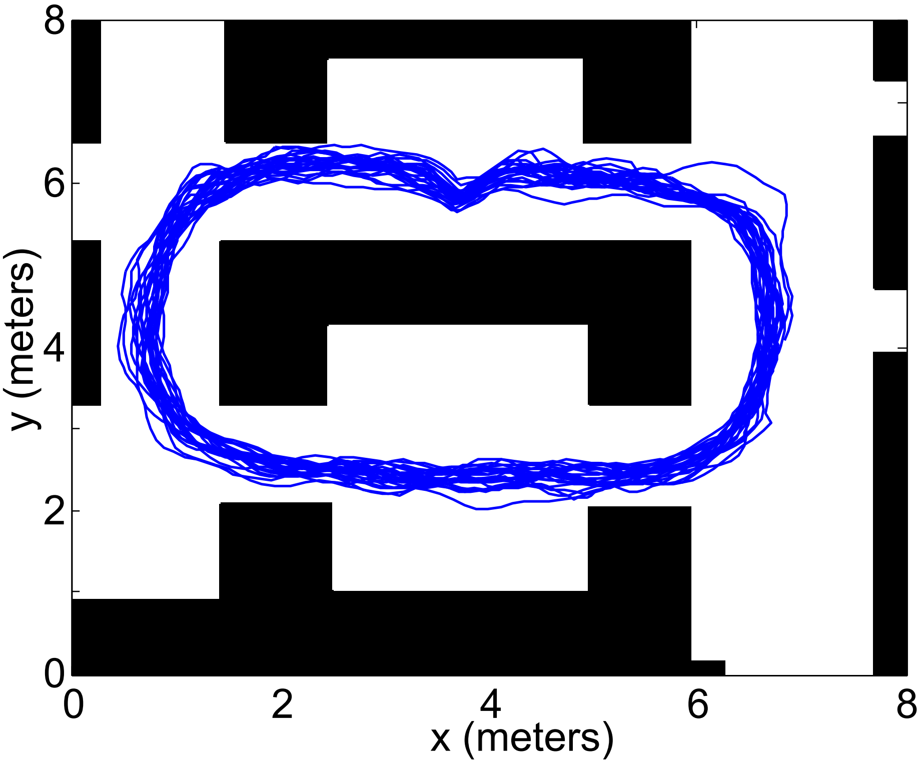


Figure 4.3: Circular Trajectories of a walking person

its contribution to new knowledge. Thirdly, the MIA was applied again to remove any redundant data points as a result of adding new data.

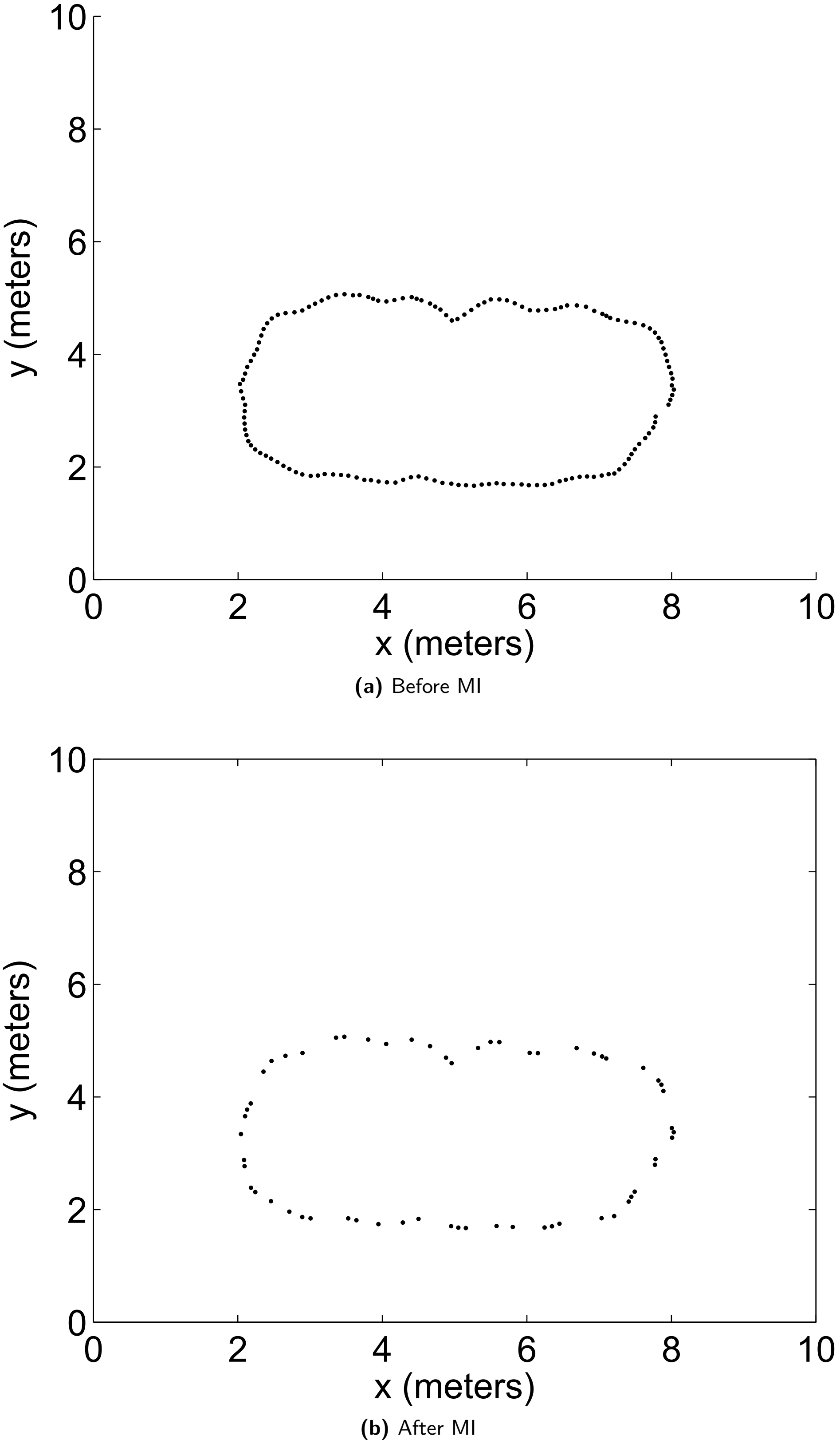
4.8.1 Circular Trajectories Tracking

A single trajectory contains 150 points, as shown in Figure 4.4a is the first loop of the observation data. From this observation, the learned GP means and covariances prior to implementation of MIA and MD are shown in Figure 4.5a and Figure 4.5b, respectively. The GP predictions exhibit high uncertainty towards the corners of the plots where no measurements are available. The data was then processed using MIA data selection to achieve the set of points that represented the same RMS error. The resulting points are shown in Figure 4.4b which leads to a 60% reduction of data points. Figure 4.6a shows the RMSE calculated at measured points using optimized GP, which is less than 0.045 meter.

When a new trajectory is available for the training data, the predicted and measured values of the mean and covariance of each point in the x and y axes are compared using MD. Those points that have MDs less than 3.84 (threshold) are discarded. For example, most of the new measurements shown in Figure 4.7a provide less MDs than the threshold obeying the learned model and hence measurements that are less than the threshold are discarded. However, more variations of data, as can be seen in Figure 4.7b, give rise to higher MDs than the threshold. Those points are incorporated into the training samples for retraining purposes.

After the second set of observations, once the data to be added to the model has been decided based on the MD process, MIA is used for selecting the most informative data points, as shown in Figure 4.9a. Figure 4.8a and Figure 4.8b show the final mean and final covariance, and after retraining, the data still has less than 0.04 meters, as referred to Figure 4.4b.

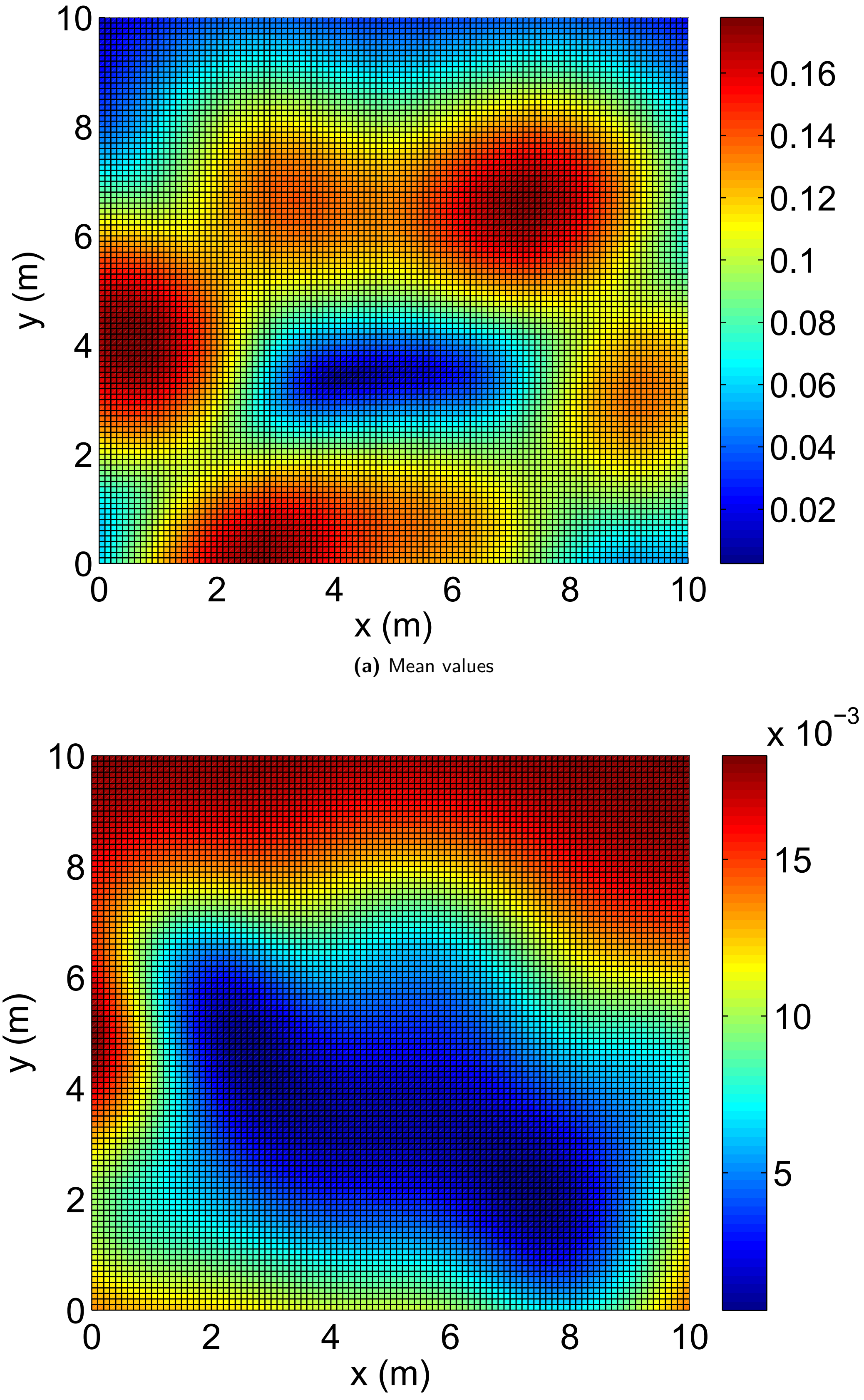
For testing the overall accuracy, referring to Figure 4.4a as the initial training data and Figure 4.9a as the final training data, the RMSE of the mean value is as shown in Figure 4.9b, at each point of the 150 training data. As can be seen, the RMSE



(a) Before MI

(b) After MI

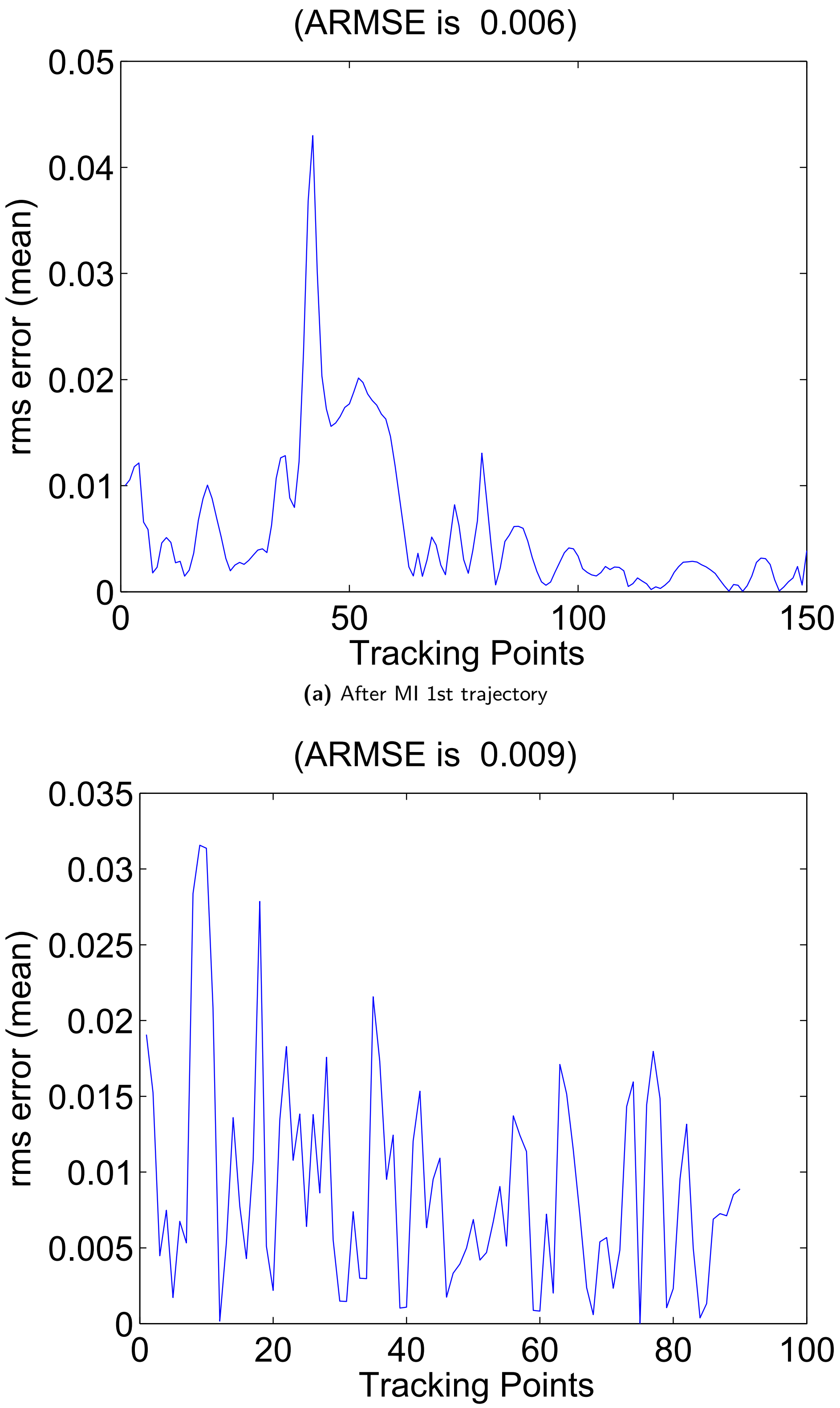
Figure 4.4: Points in trajectory prior to and post-MI implementation



(a) Mean values

(b) Covariance values

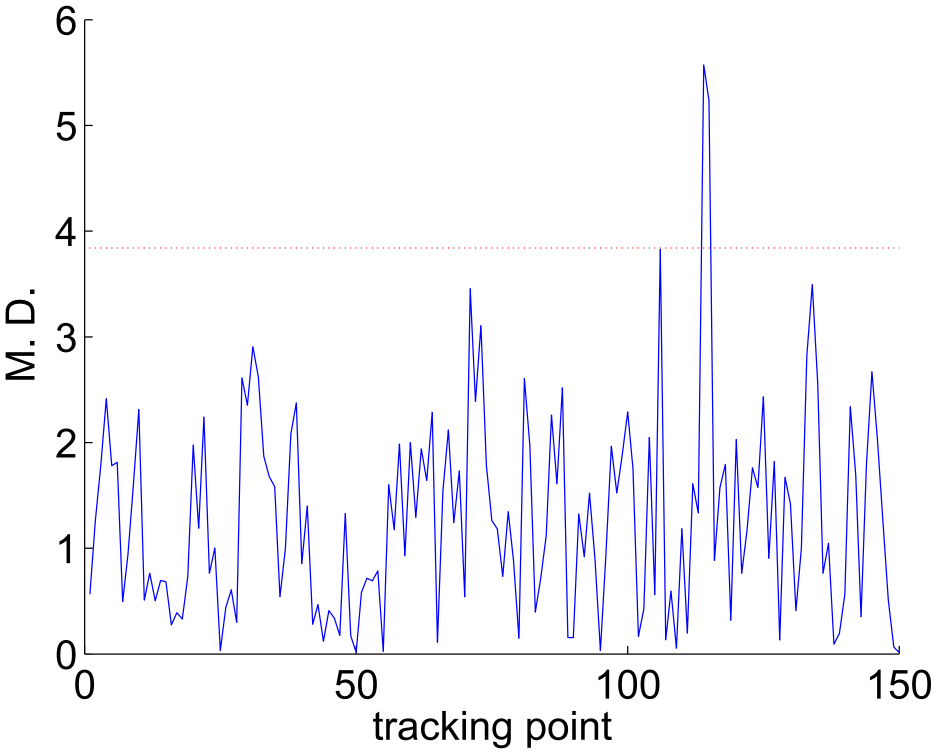
Figure 4.5: GP regression with 150 data points



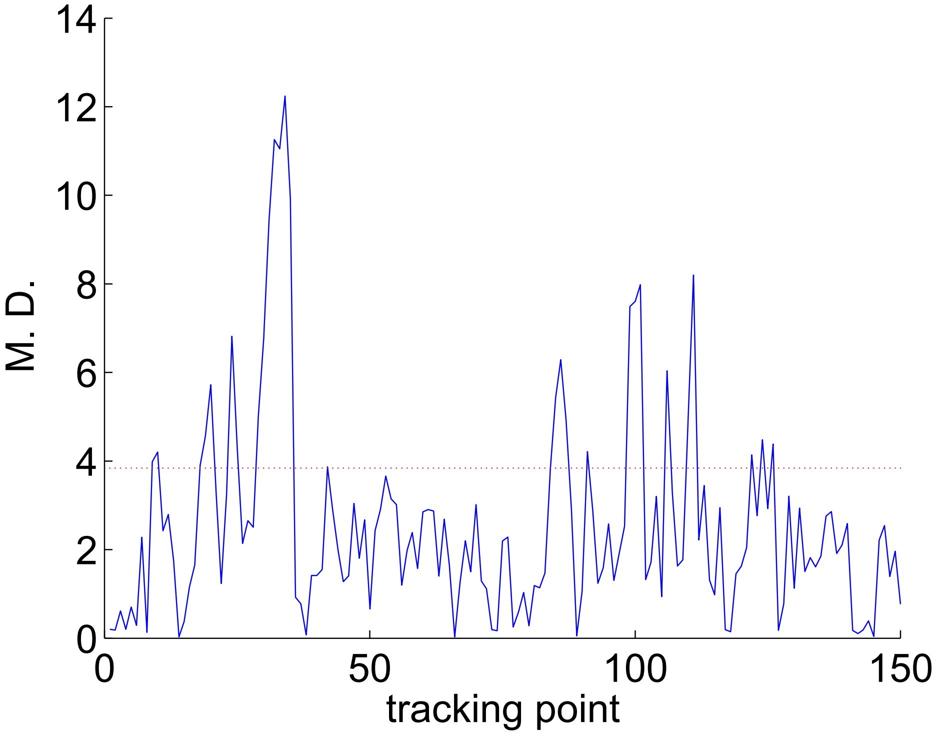
(a) After MI 1st trajectory

(b) After MD 2nd trajectory

Figure 4.6: RMSE between predicted mean and the measurements

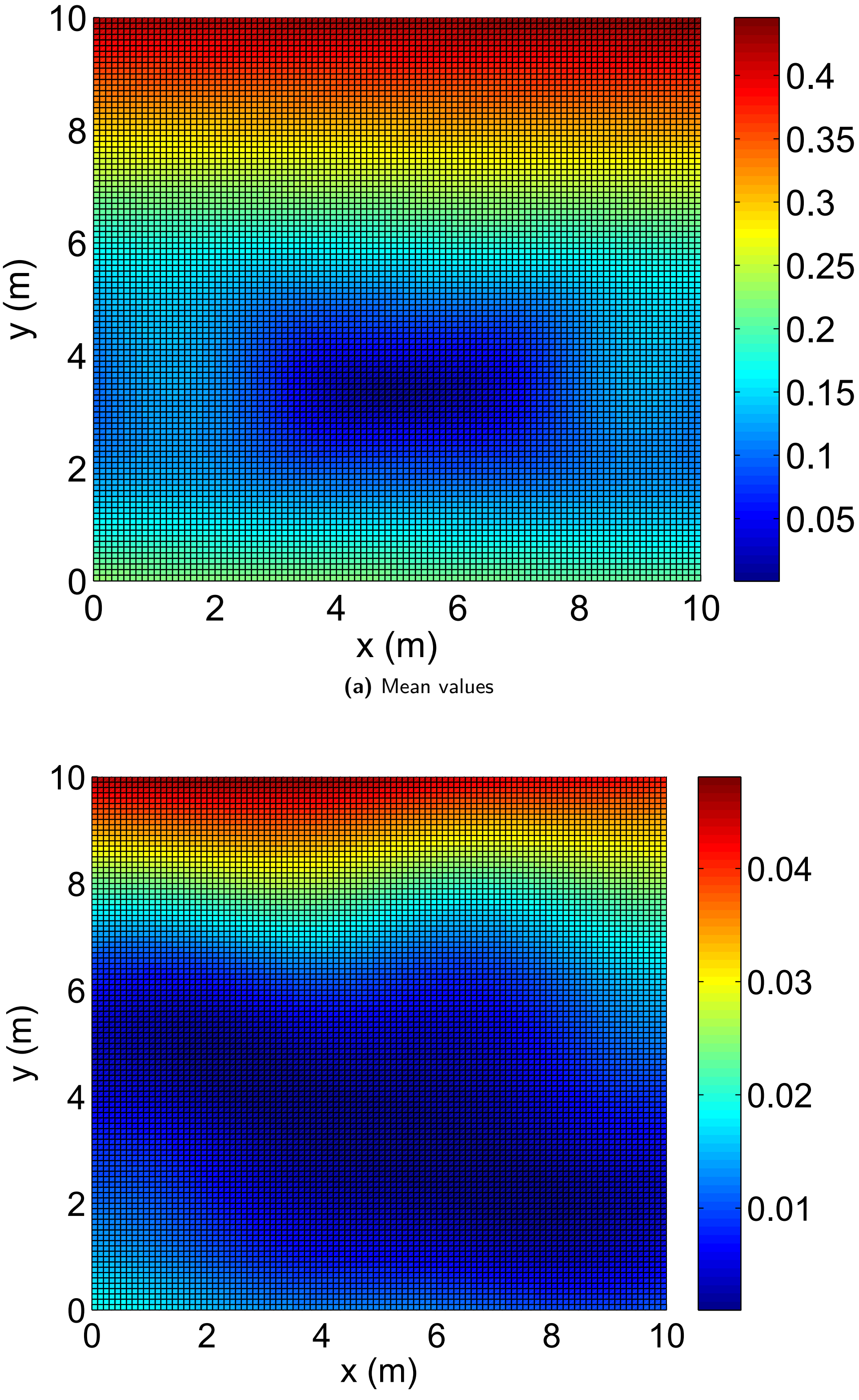


(a) X direction



(b) Y direction

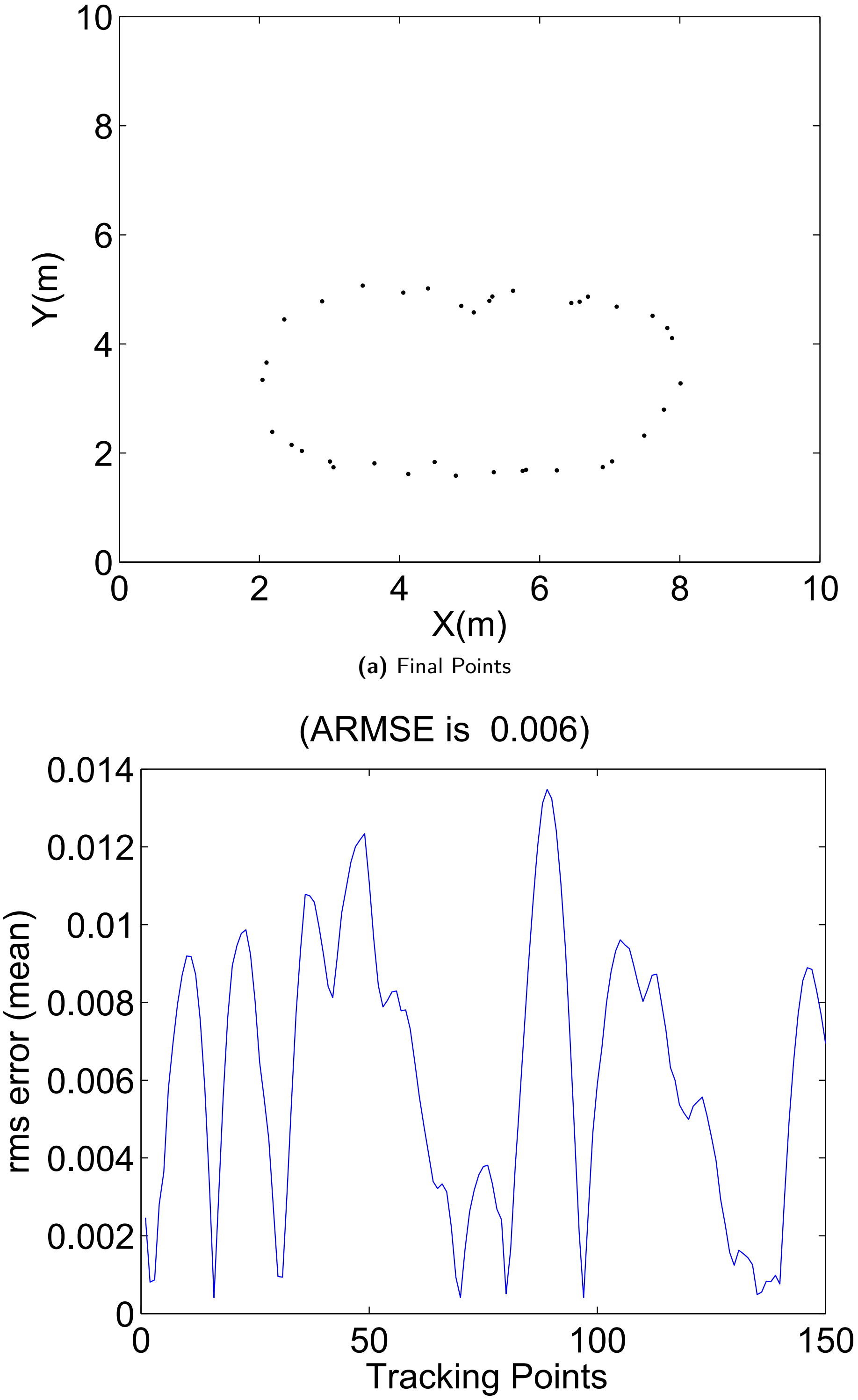
Figure 4.7: Implementation of Mahalanobis Distance



(a) Mean values

(b) Covariance values

Figure 4.8: GP Regression of final points after MD and MIA



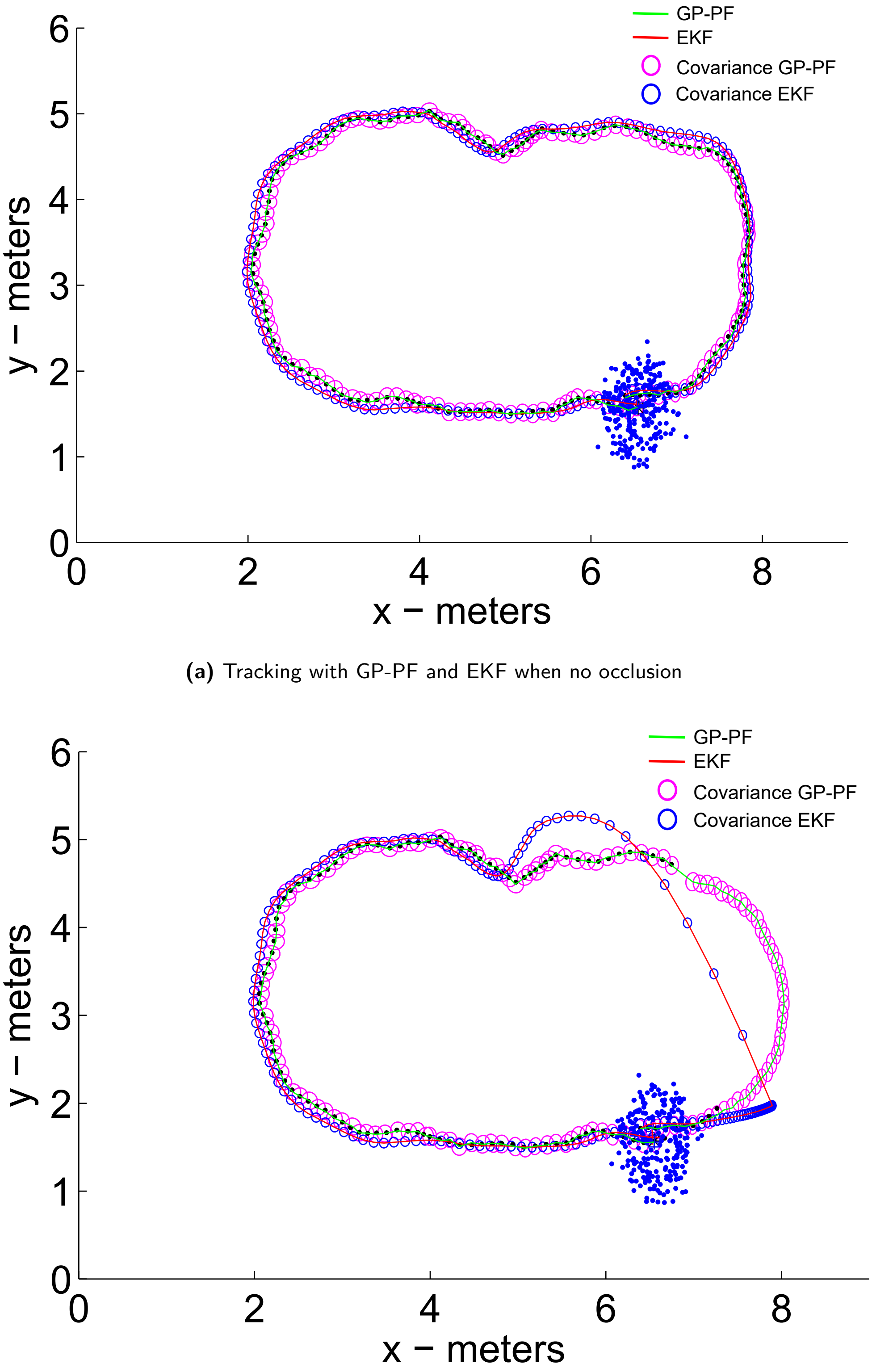
(a) Final Points

(b) RMSE between initial and final GP

Figure 4.9: Final points and RMSE

at each point is less than 0.02 meters, which is reasonably good as the environment contains corridors of width ranging from 1.3 to 1.5 meters. The number of training data is reduced to 37 from the total of 300 points, which is more than 85% data reduction.

The GP learned model is then used for predicting the PF based tracker and evaluating its long term tracking ability. Figure 4.10 shows the comparison results of model based EKF and the proposed GP-PF trackers with an occlusion. As it can be seen, the EKF has poor tracking performance, while the GP-PF still manages to perform well. It is to be noted that the EKF tracker has lost track, but with growing covariance, it could finally converge to the track again.This is possible as the experiment only involves one track (data association is assumed) and the fact that it goes around the loops. The comparison of four types of trackers: EKF, PF, GP-EKF and GP-PF is shown in Figure 4.11. Figure 4.11 clearly demonstrates that GP-EKF and GP-PF outperform the other trackers.



(a) Tracking with GP-PF and EKF when no occlusion

(b) GP-PF (green line) and EKF (red line) when occlusion occurs

Figure 4.10: Tracking performance of the Gaussian Process-Particle Filter (GP-PF)

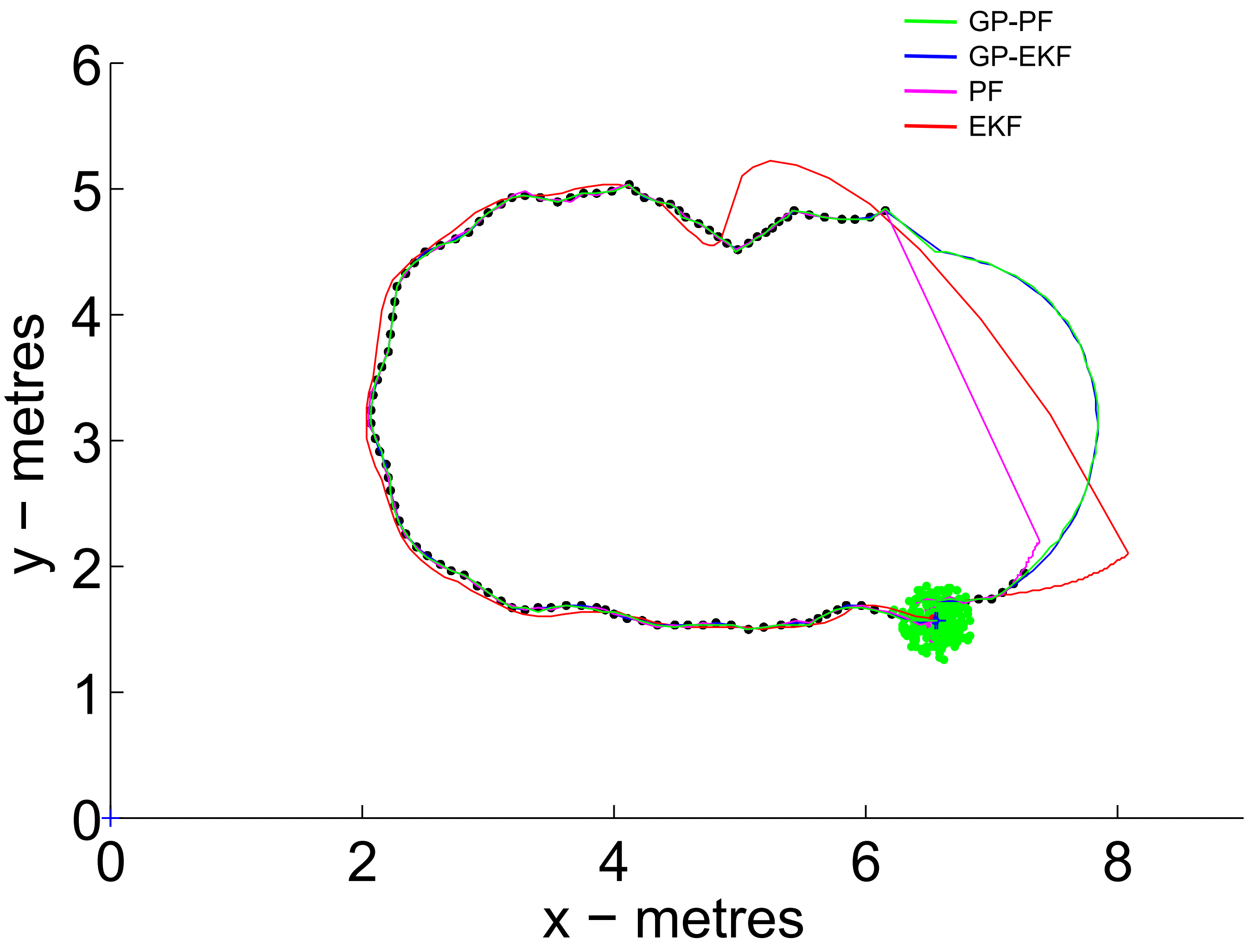


Figure 4.11: Comparison of four types of trackers: EKF (red line), PF (magenta), GP-EKF (blue line), and GP-PF (green line)

4.8.2 Multiple Trajectories Tracking

Figure 4.14 contains 3868 points collected on 10 trajectories of four routes. The learned GP means and covariances are shown in Figure 4.16a and Figure 4.16b, respectively. The GP predictions are highly uncertain towards the corners of the plot and the values of means and covariance are uniformly distributed in the areas where no observations were made. The data was then processed using MI-based data selection to achieve the set of points that represented the same RMSE. When a new trajectory was available for the training data, the predicted and measured values of the mean and covariance of each point in the x and y axes were compared using MD. Those points that had MDs less than 3.84 (threshold) were discarded. Once the data that were to be added to the model had been decided based on MD; MI was then applied for selecting the most informative data points as shown in Figure 4.15.

The algorithm starts with an empty set of points and greedily adds placements in sequence until a designated number of points are found based on the angle or magnitude of displacement of each point, in order to find the best points in the whole selective data set. The MI algorithm chooses the succeeding point that produces the maximum increase in mutual information. The mean and covariance after retraining are shown in Figure 4.17a and Figure 4.17b. As shown in Figure 4.18, most of the data had an RMSE of less than 0.02 meters and an ARMSE of 0.0038 meters. Referring to Figure 4.14 and Figure 4.15, the number of training data was reduced to 610 points from the total of 3868 points, which was reduced to 15.77% or more than 80% of the data reduction.

Figure 4.19a to Figure 4.20b refer to people tracking on routes 1 to 4, respectively. It can be seen that the GP-PF efficiently tracks people with reduced data from the GP learned model. The GP learned model is then used for predicting the PF based tracker for evaluating its long term tracking ability. The comparison results of model based EKF and the proposed GP-PF with an occlusion are as shown in Figure 4.21. As it can be seen, the EKF tracker has poor tracking performance, while the GP-PF tracker still manages to perform well. It is to be noted that the EKF tracker has lost track; however, it manages to continue predicting the track with growing covariance. Covariance ellipses grow as the prediction continues, as shown in Figure 4.22.

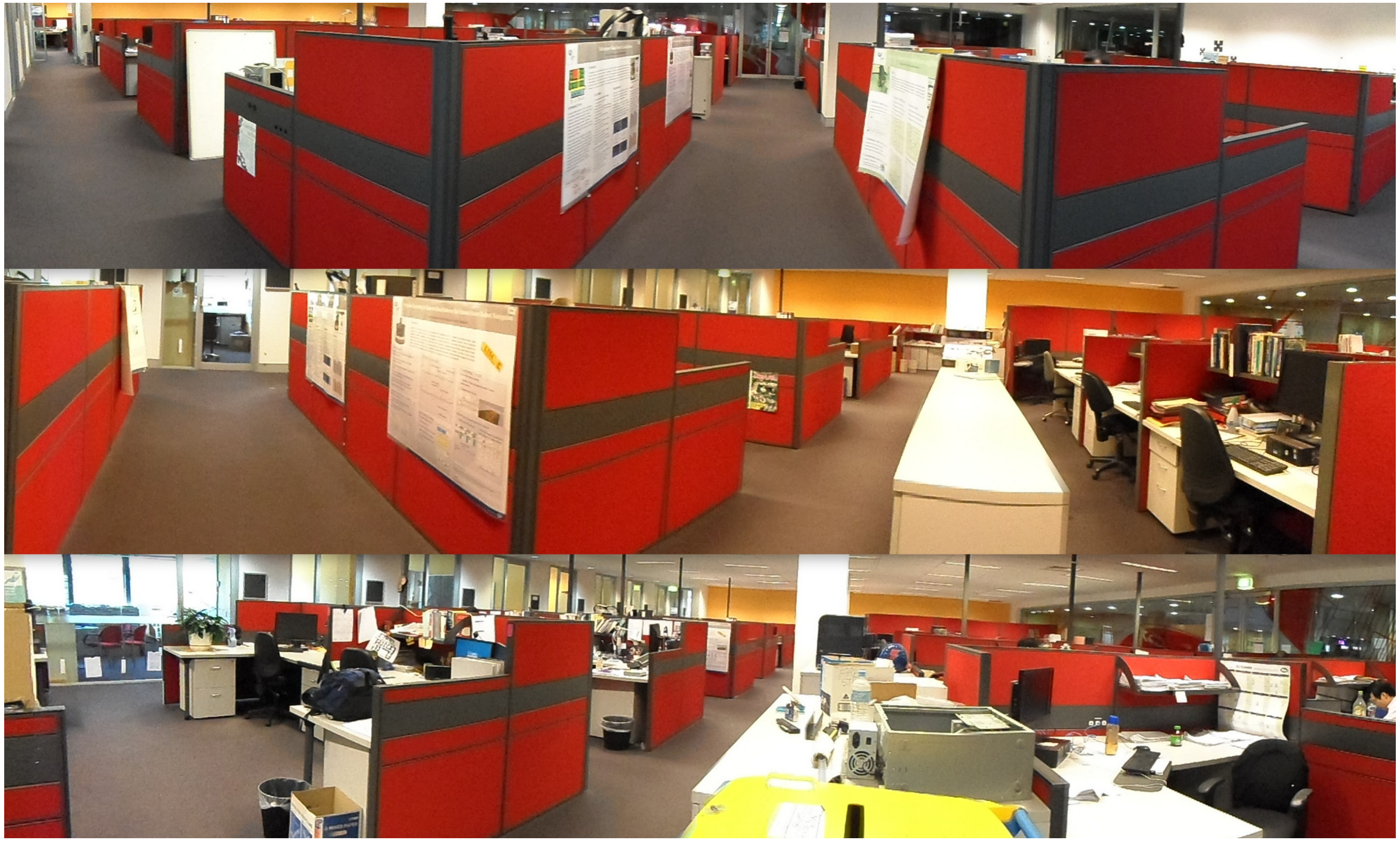


Figure 4.12: Office environment in the Centre of Autonomous Systems

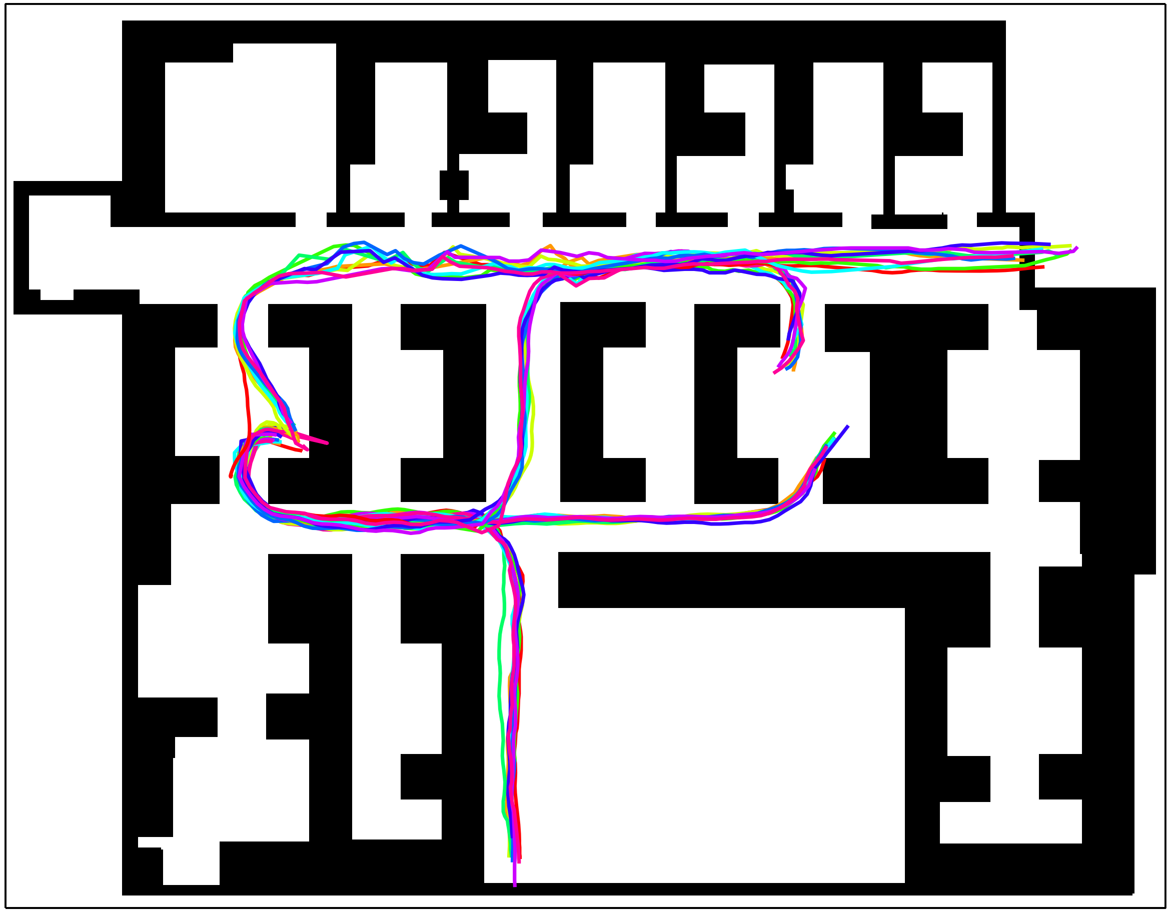


Figure 4.13: Multiple trajectories of a walking person

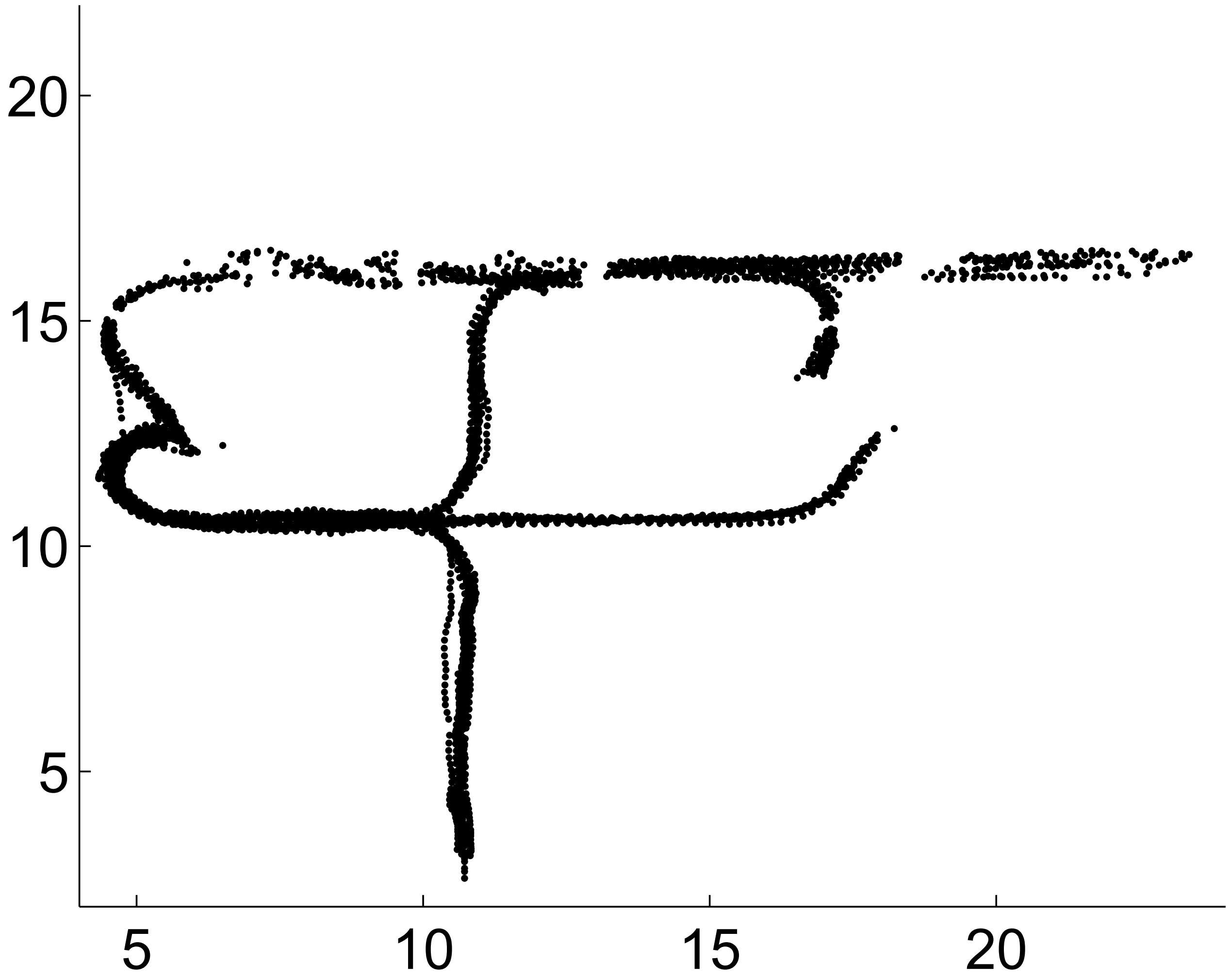


Figure 4.14: The initial points of a subject are represented as dots as they walk

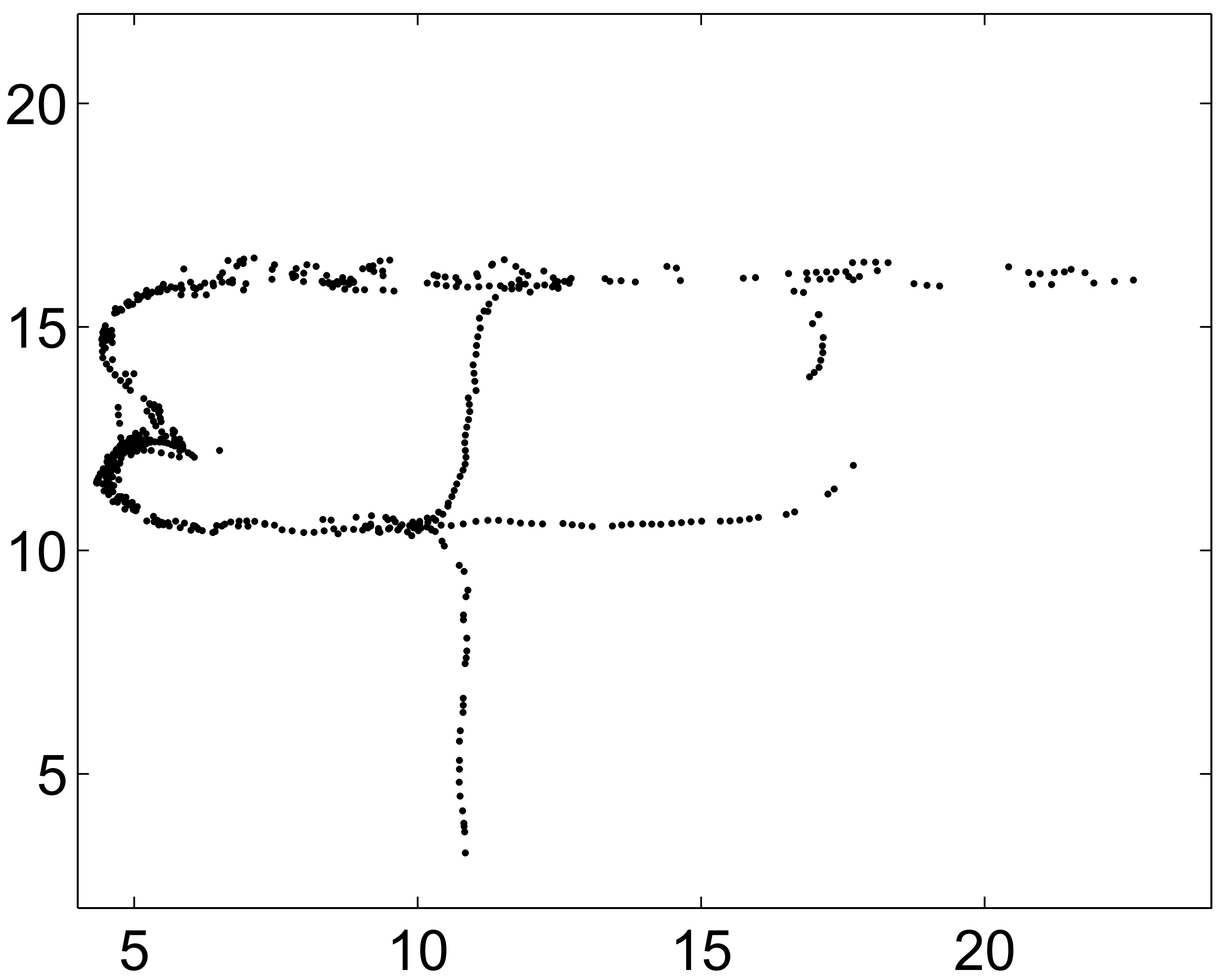
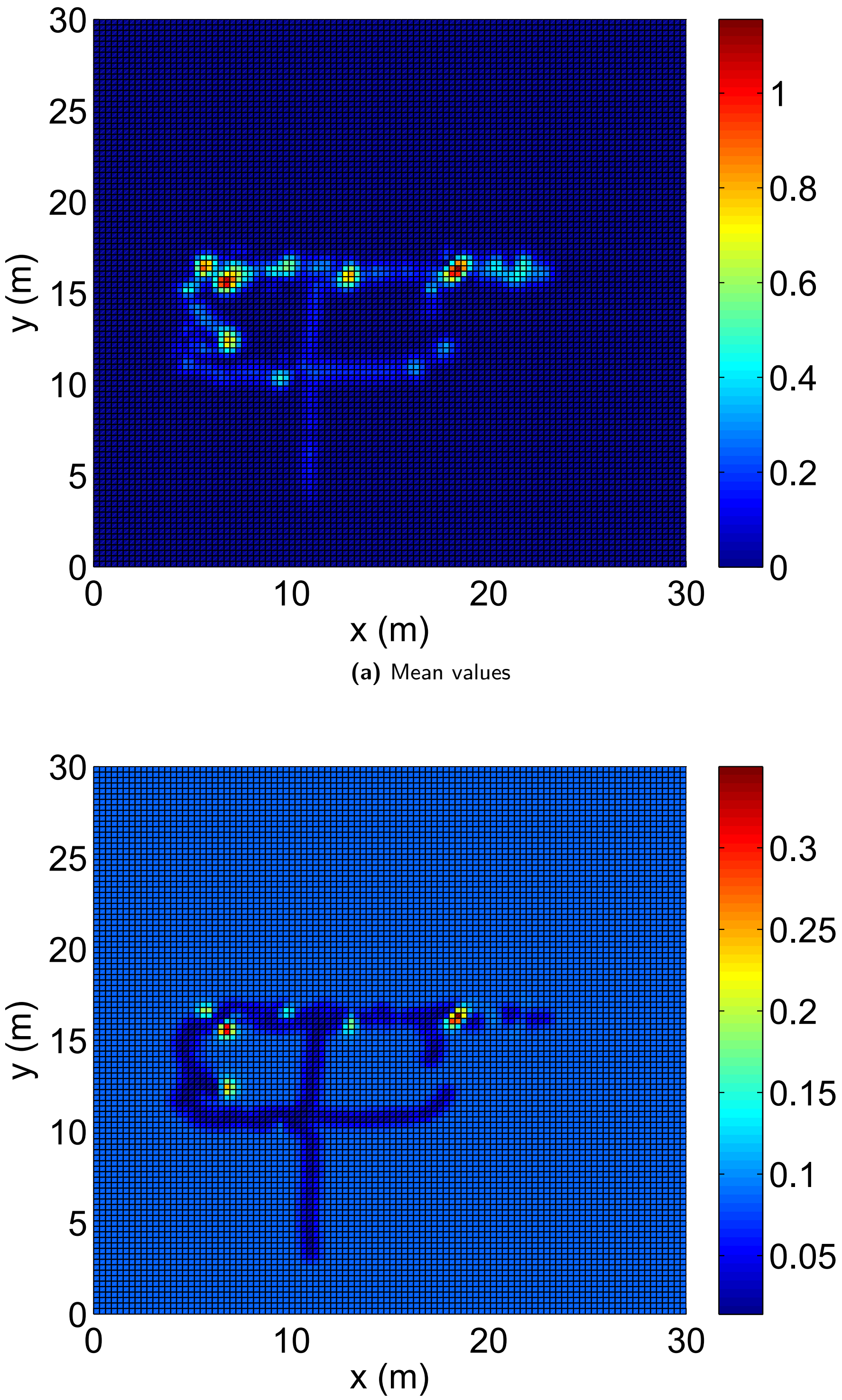


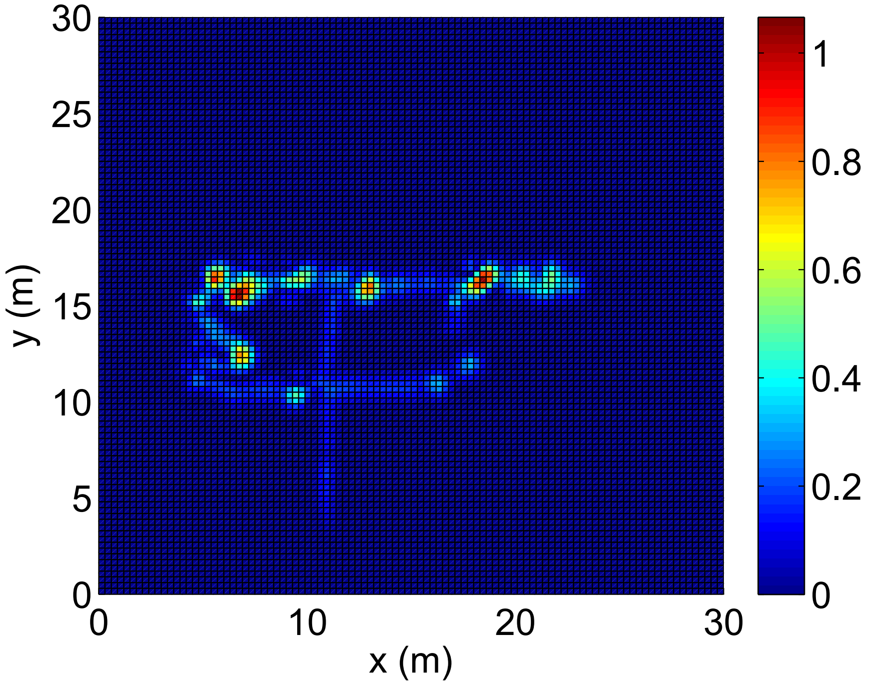
Figure 4.15: Final points after MIA and MD



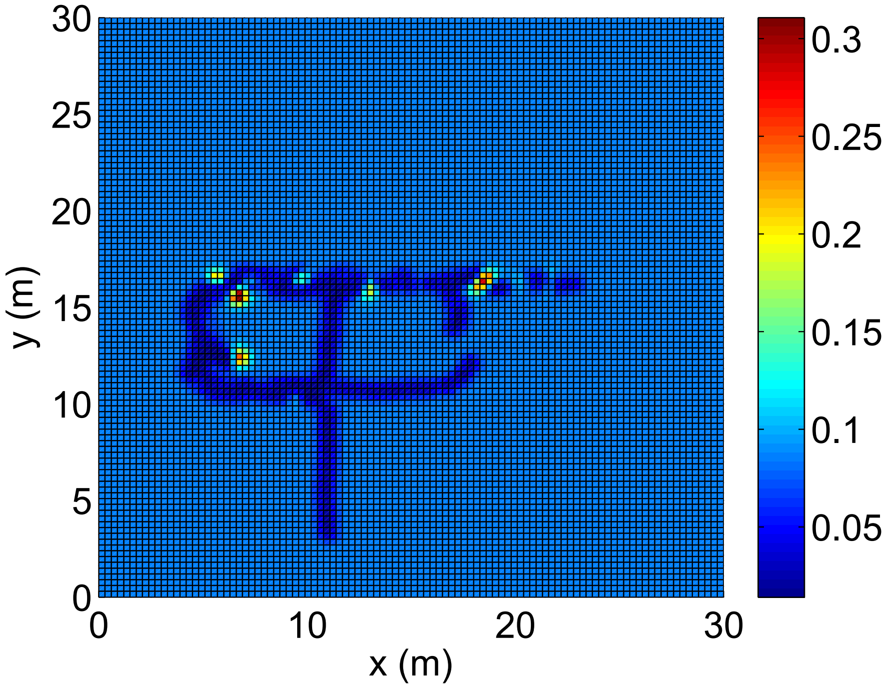
(a) Mean values

(b) Covariance values

Figure 4.16: Gaussian Process Regression before MIA and MD



(a) Mean values



(b) Covariance values

Figure 4.17: Gauss Process Regression after MIA and MD

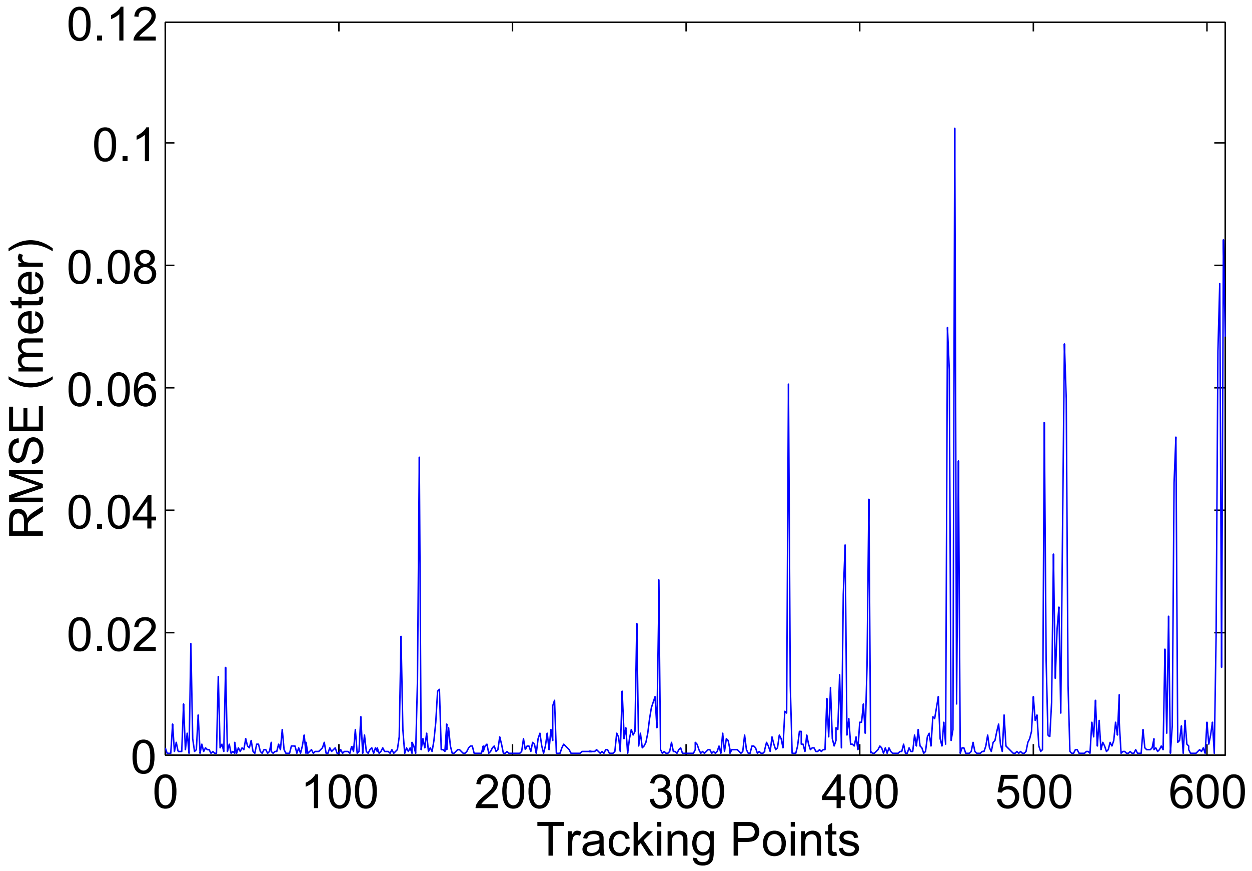
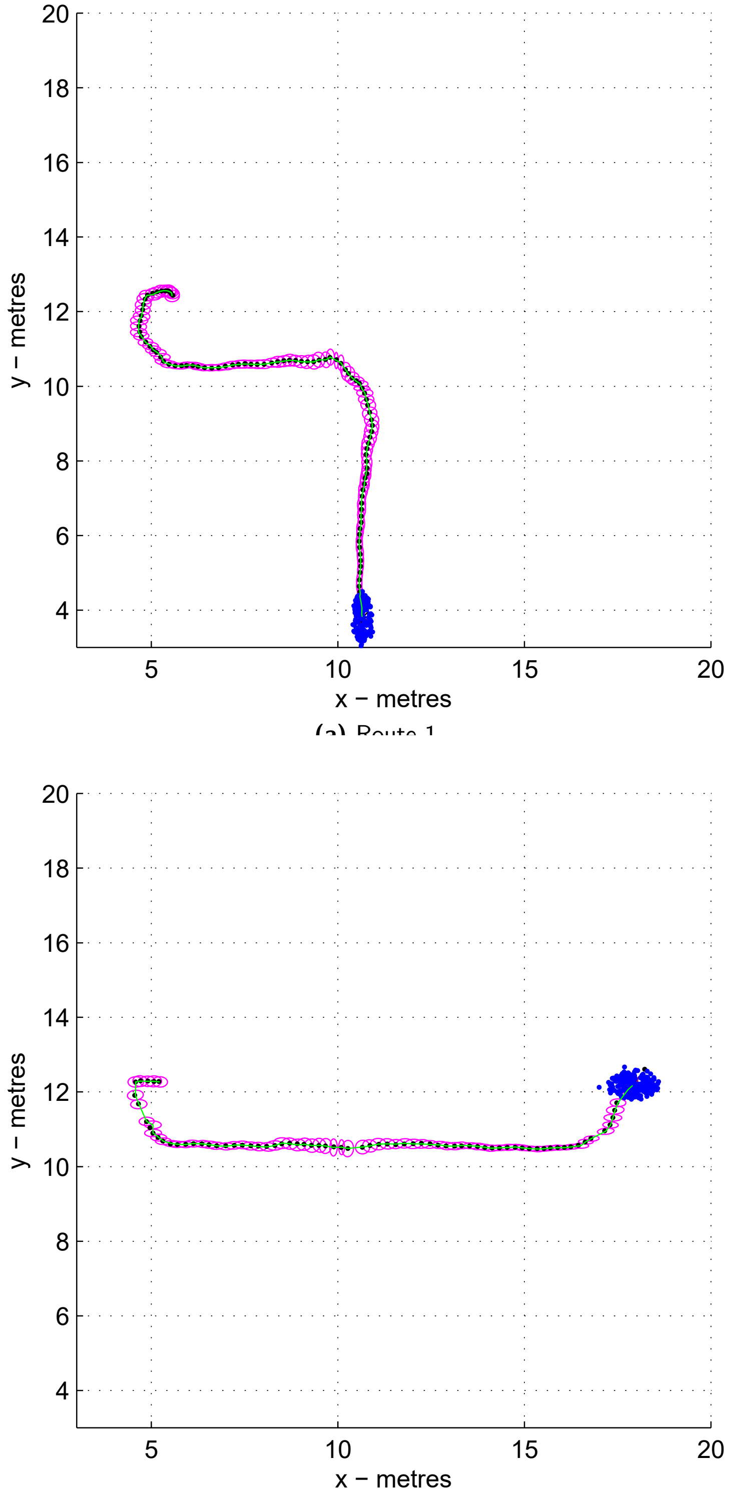


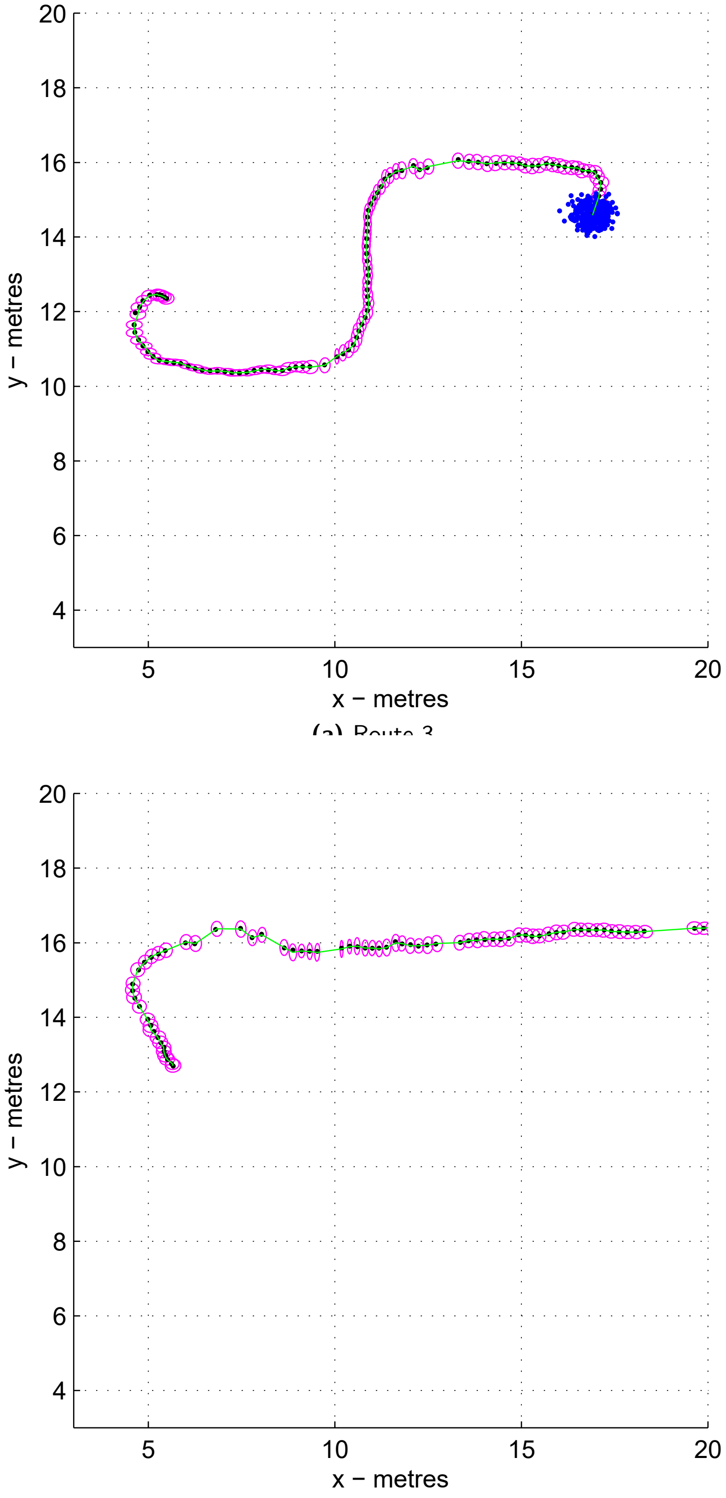
Figure 4.18: RMSE between predicted mean and the measurement



(a) Route 1

(b) Route 2

Figure 4.19: Tracking route 1 and route 2 with GP-PF



(a) Route 3

(b) Route 4

Figure 4.20: Tracking route 3 and route 4 with GP-PF

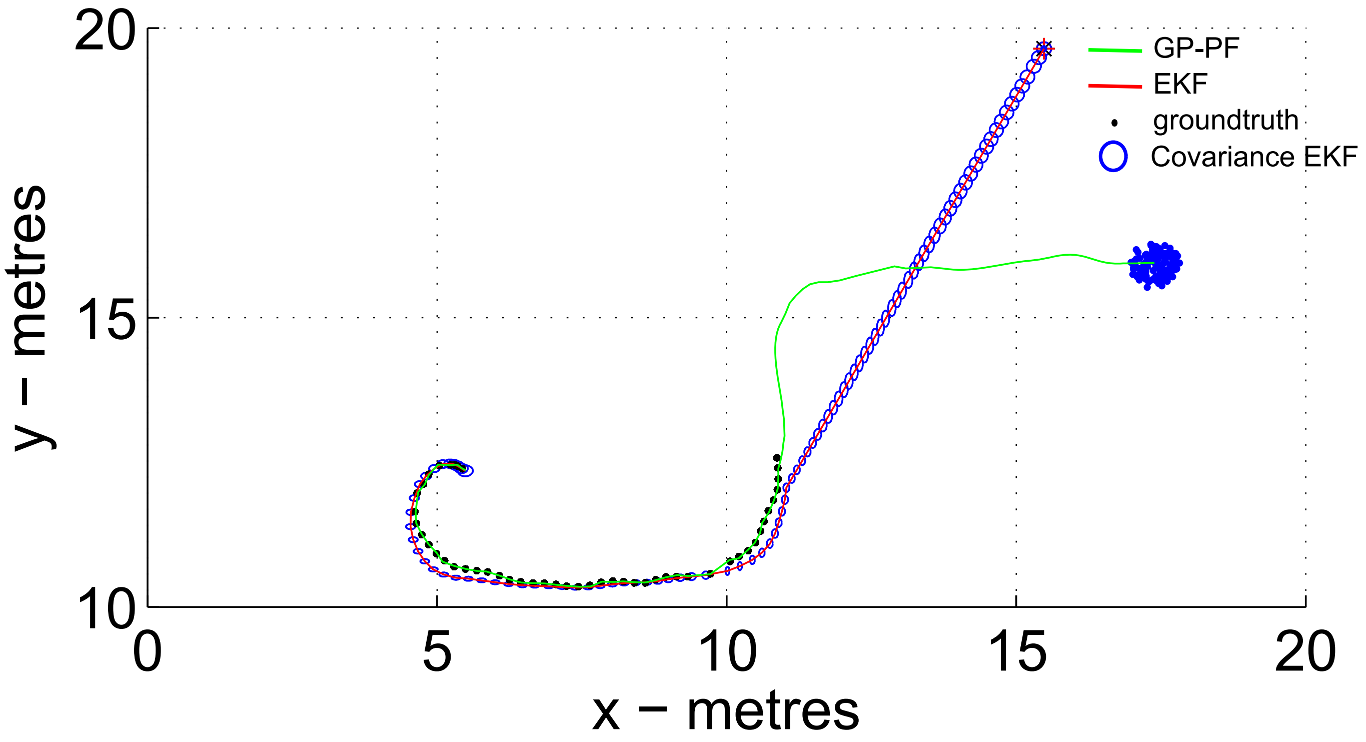


Figure 4.21: Occlusion: tracking with GP-PF (green line) and EKF (red line)

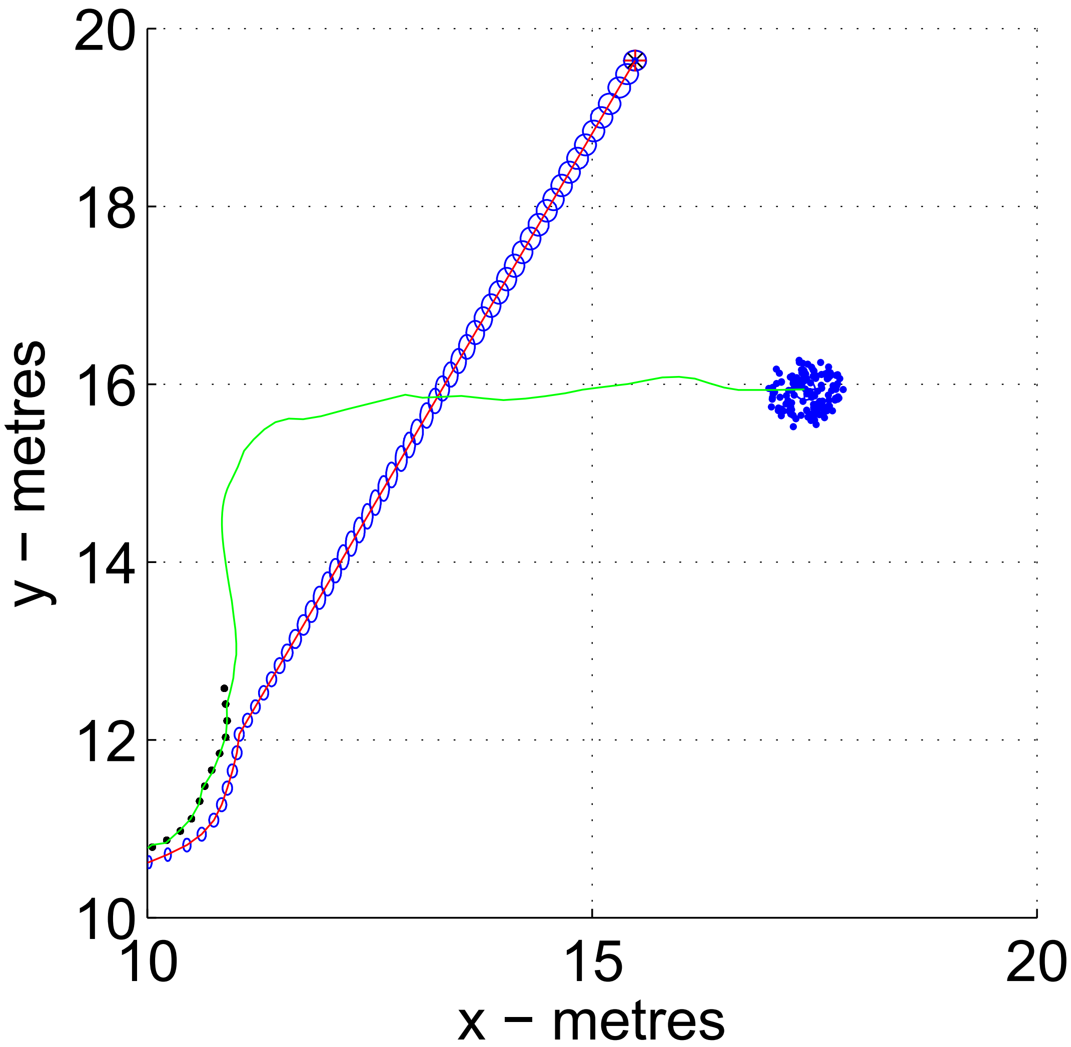


Figure 4.22: Zoom in on the growing covariance ellipses of the occlusion period with the EKF tracker

4.8.3 Simultaneous Trajectories Tracking

For 2 and 4 people who are simultaneously tracked in the vicinity, various scenes of people trajectories are shown in Figure 4.23, Figure 4.24 and Figure 4.27. In those figures, the green line represents trajectories tracked by GP-PF and the red line represents trajectories tracked by EKF. The black dotted point represents the observation point. In Figure 4.23, two people are tracked with the same trajectories in the starting phase and in the latter phase, they walk in the opposite direction. Both trajectories have their observations partially disrupted due to the object momentarily blocking the vision of LRF. It has been clearly visualised in Figure 4.23 for both trajectories; trajectories that are tracked by GP-PF show better prediction performance compared to the trajectories tracked by EKF. The trajectories tracked by EKF deviated significantly from the reference track. Thus, the GP-PF tracker shows superior tracking performance over the EKF tracker. Figure 4.24 shows the tracking of four people in their tracks without any occlusion. It shows that the trajectories tracked by the GP-PF and EKF trackers obviously lie within the observation points. However, on the sharp manoeuvring curve, it shows that the GP-PF tracker has better tracking ability than the EKF tracker. In Figure 4.25a, the comparison is made on four tracking methods: EKF, PF, GP-EKF and GP-PF. Green, blue, red, and magenta lines represent the GP-PF, GP-EKF, EKF, and PF trackers, respectively. The black dots represent the routes' reference points or ground truth. When it comes to maneouvring conditions, EKF performs the worst among the four tracking methods. GP-PF clearly outperforms EKF in terms of maneuvring conditions.

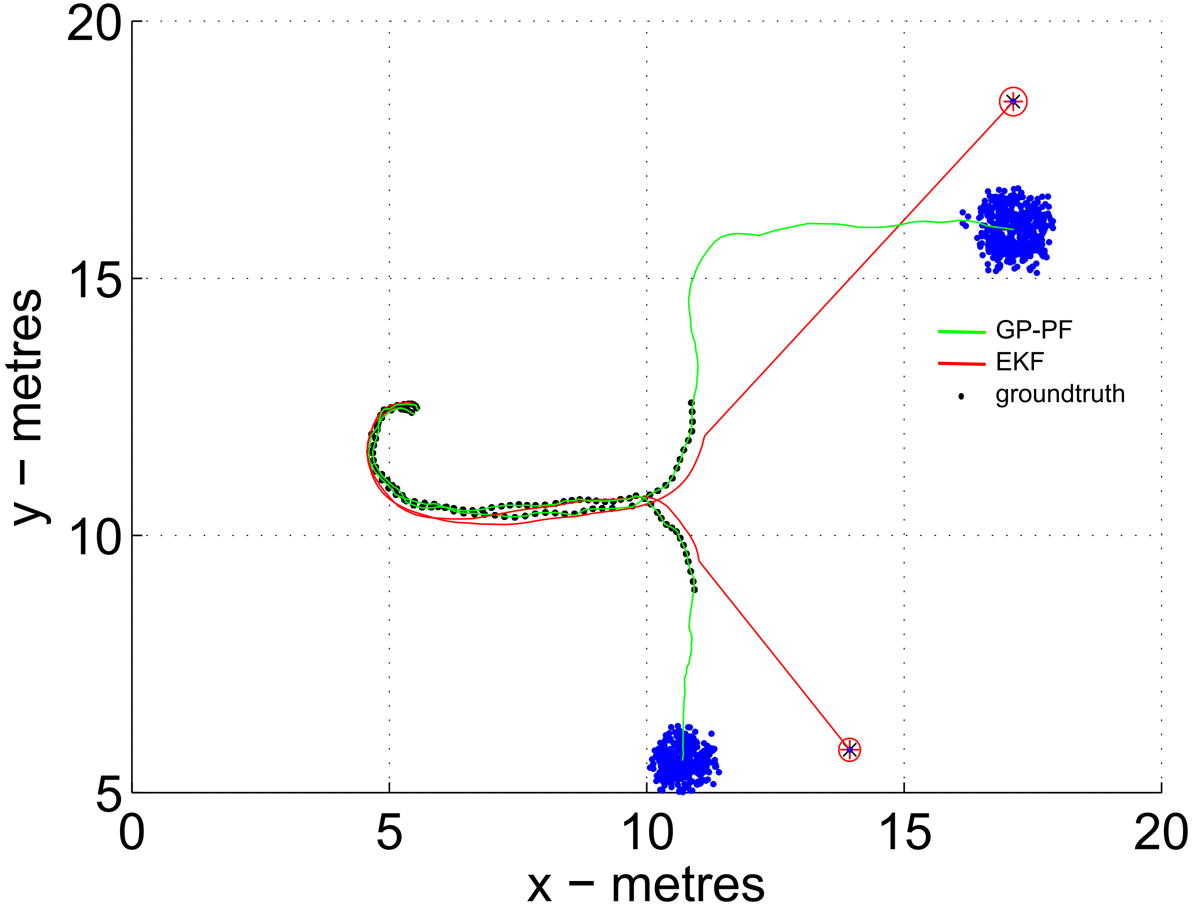


Figure 4.23: Two people were simultaneously tracked with partial occlusions by GP-PF (green line) and EKF (red line)

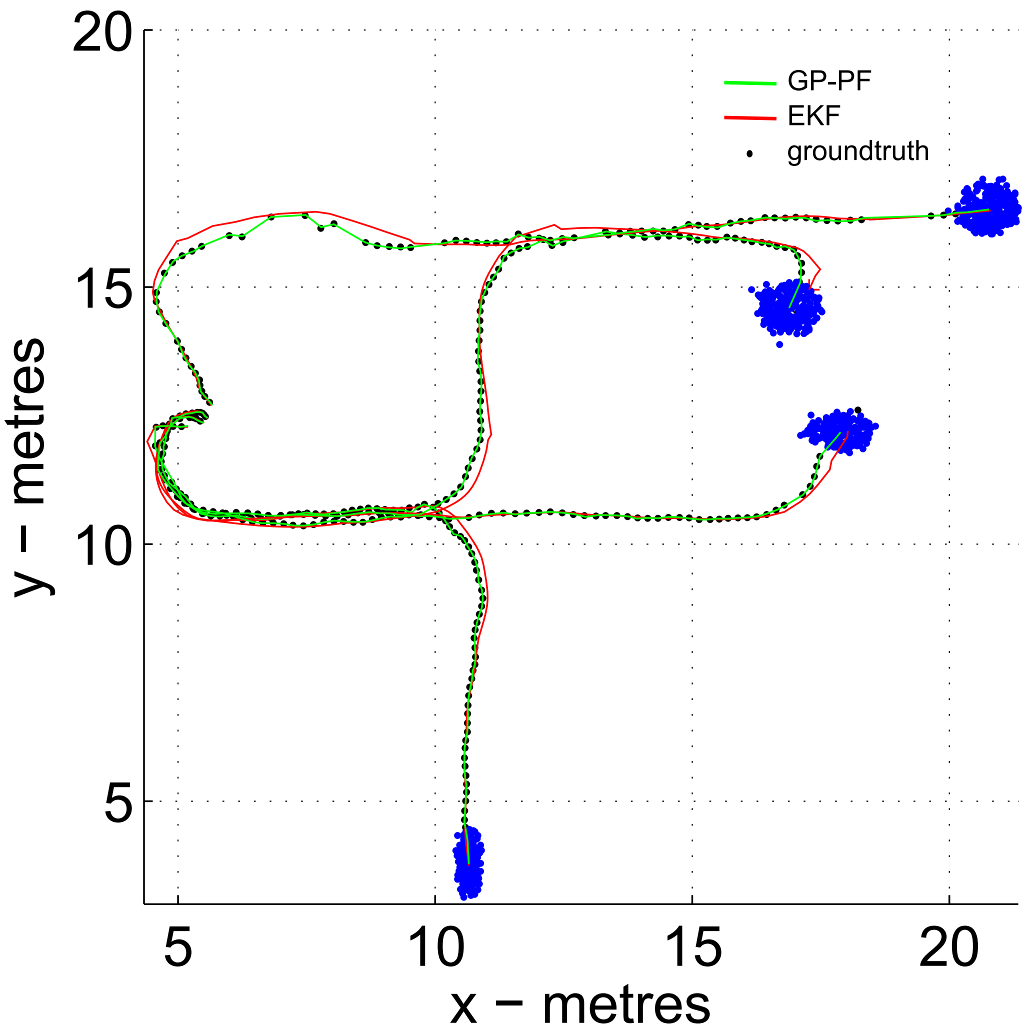


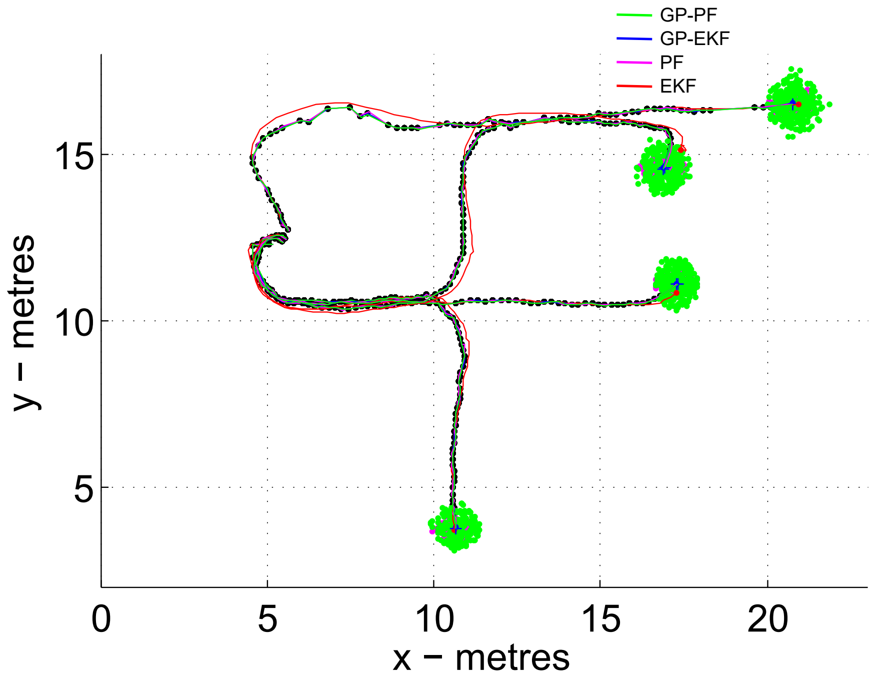
Figure 4.24: Four people were simultaneously tracked with no occlusions by GP-PF (green line) and EKF (red line)

A study was conducted to compare the RMSE on 50 Monte Carlo runs, as depicted in Figure 4.25b, and evaluate each tracker's performance on a single route, as illustrated in Figure 4.26a. GP-PF and GP-EKF have convincingly demonstrated that they have greater tracking performance than EKF and PF because of their lower RMSE.

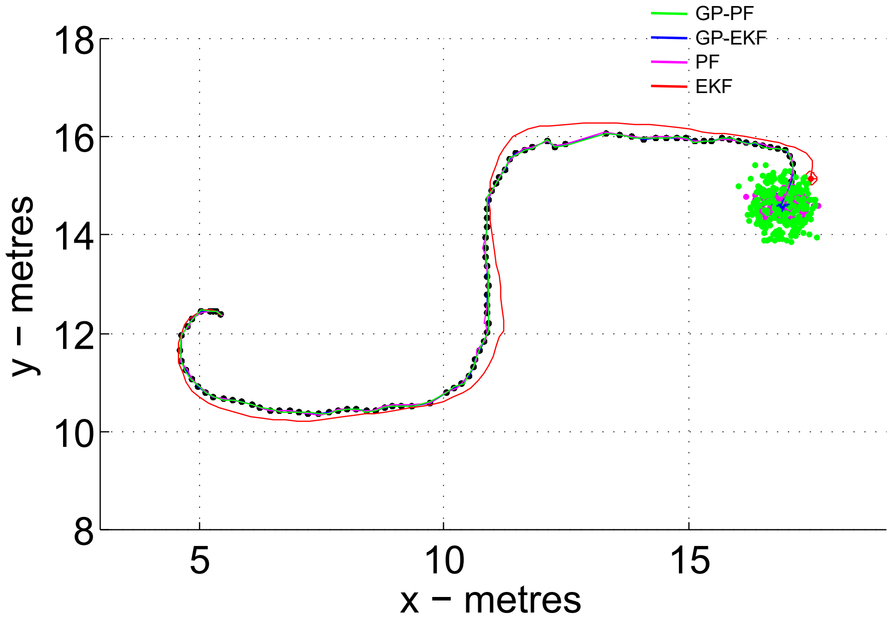
The comparison on only GP-PF and GP-EKF trackers in Figure 4.26b shows that GP-PF performs marginally better than GP-EKF because GP-PF has a lower RMSE than GP-EKF.

In Figure 4.27, four people are tracked in their designated direction and two of them have their trajectories partially disrupted due to objects momentarily blocking the vision of LRF. It is clearly seen that partial disrupted moments have been better predicted using the GP-PF tracker. The prediction that was done by the EKF tracker obviously deviated from the original track.

In Figure 4.28, the comparison of the performance of four tracking methods has been shown. It is visually proven that GP-PF which is shown in the green line, has better performance than the others.

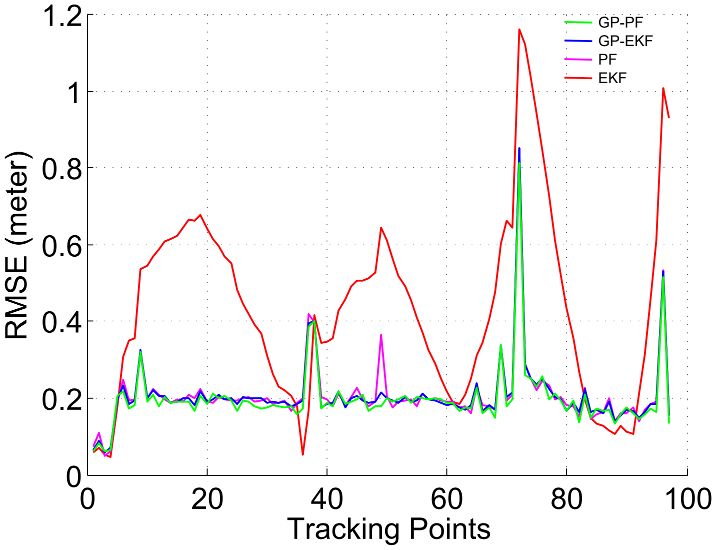


(a) Four people were simultaneously tracked with no occlusions by EKF (red line), PF (magenta line), GP-EKF (blue line) and GP-PF (green line)

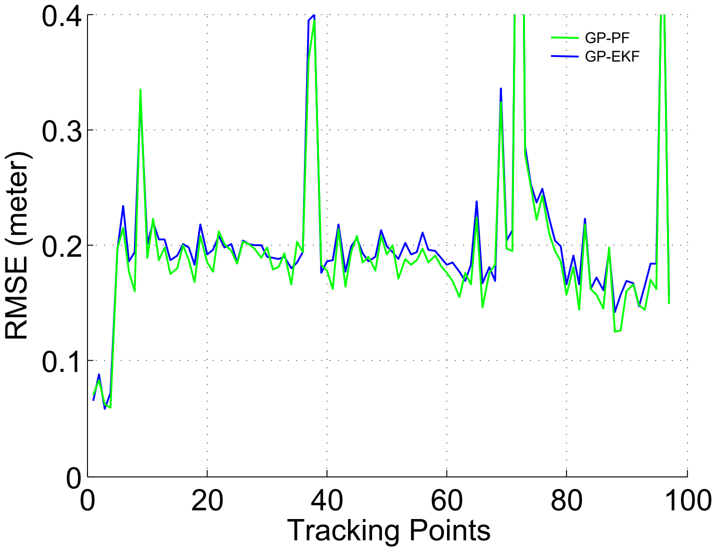


(b) Tracking a person without occlusions by EKF (red line), PF (magenta line), GP-EKF (blue line) and GP-PF (green line)

Figure 4.25: Tracking people with four methods



(a) Comparison of RMS Error on 1 route without occlusions for tracking by EKF (red line), PF (magenta line), GP-EKF (blue line) and GP-PF (green line)



(b) Comparison of RMS Error on 1 route without occlusions for tracking by GP-EKF (blue line) and GP-PF (green line)

Figure 4.26: RMS Error on 1 route

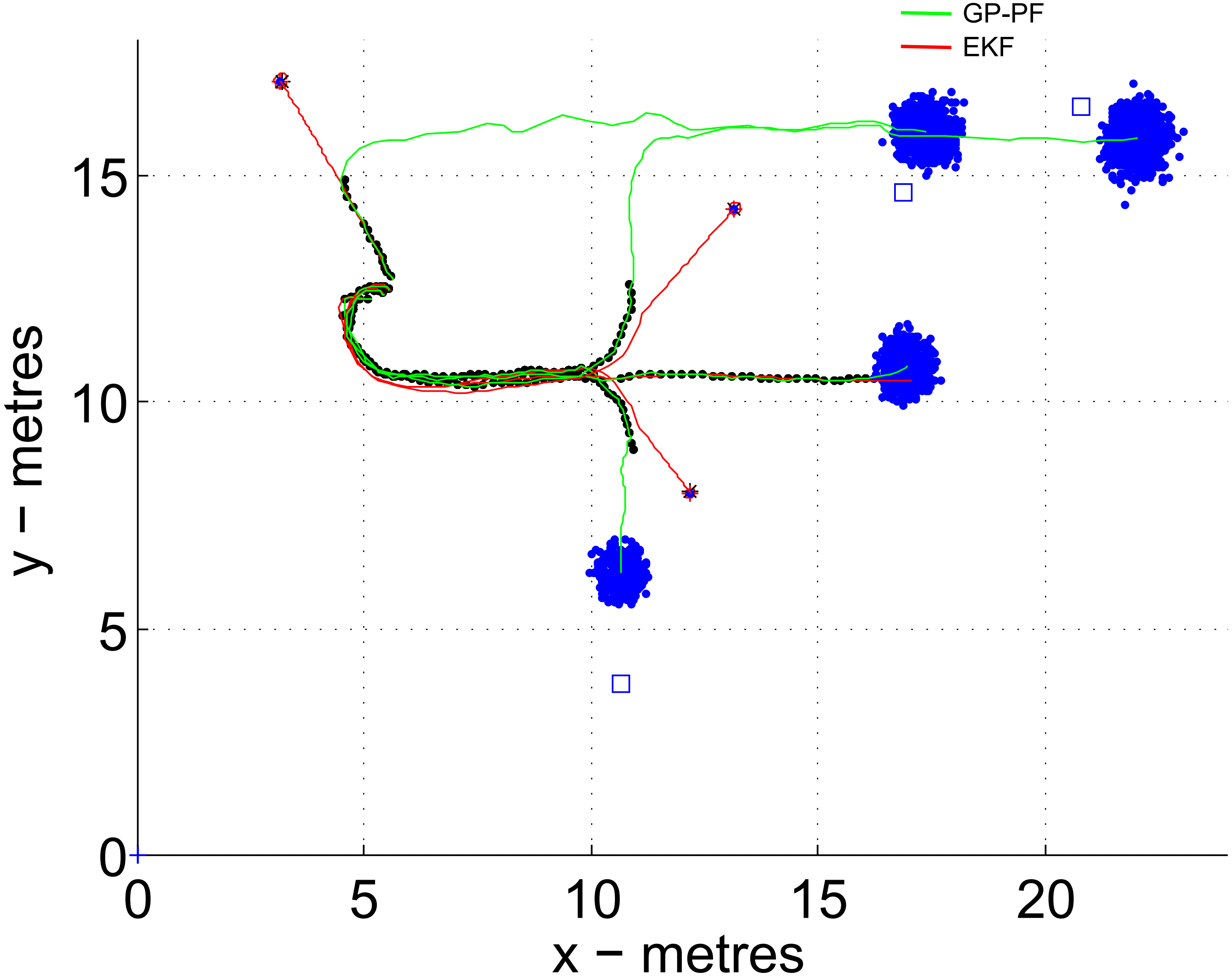


Figure 4.27: Four people were simultaneously tracked with partial occlusions by GP-PF (green line) and EKF (red line)

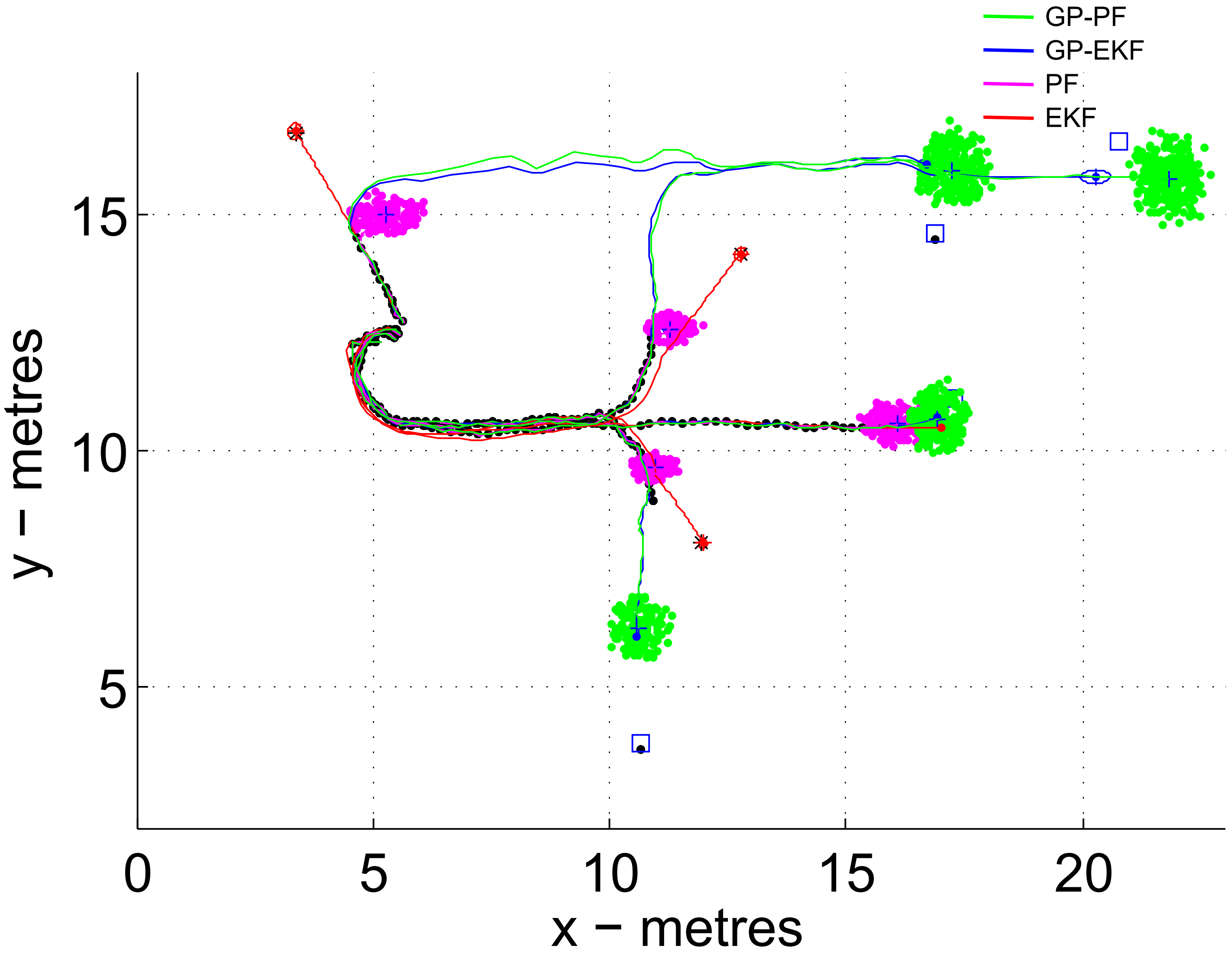


Figure 4.28: Four people were simultaneously tracked with partial occlusions by EKF (red line), PF (magenta line), GP-EKF (blue line) and GP-PF (green line)

4.9 Tracking the Freely Walking People Scenario

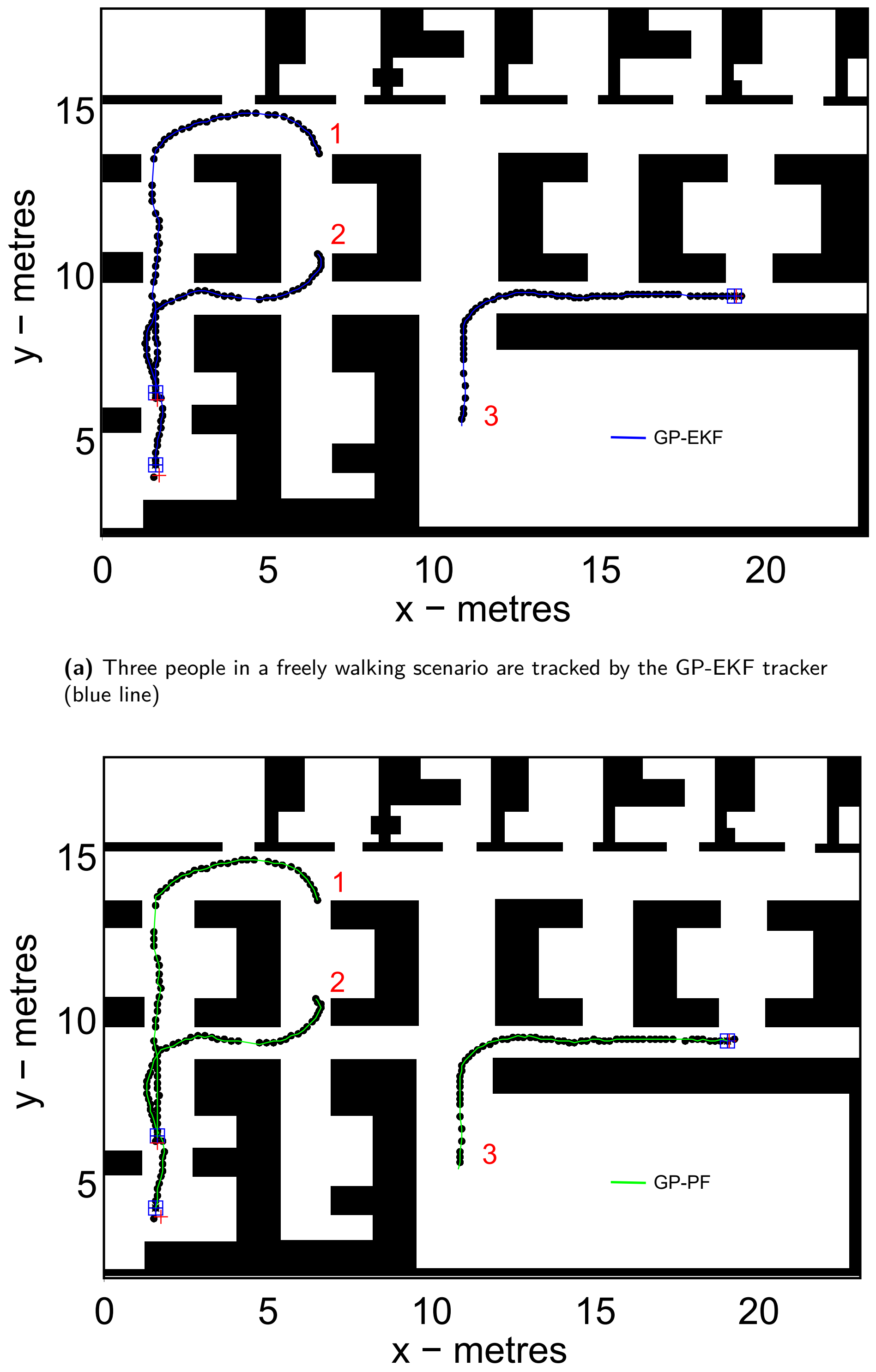
Figures 4.29a and 4.29b demonstrate the tracking of three people in a freely walking situation using GP-EKF and GP-PF. It is obvious that both trackers have shown remarkable tracking abilities. Figure 4.30 provides a visual representation of the accuracy on both trackers. Both the GP-EKF and GP-PF trackers perform equally well.

4.10 Conclusion

This chapter presents the results on the tracking performances of Gaussian Processes-BayesFilters such as GP-Extended Kalman Filter and GP-Particle Filter in comparison with conventional trackers such as Extended Kalman Filter and Particle Filter. The Gaussian Process and its implementation have been discussed and the experimental results have been analyzed and visualized to verify the validity of these trackers. It can be concluded that GP-EKF and GP-PF managed to handle the long term better than EKF and PF. Further, it shows that the GP-EKF and GP-PF were effective in learning people's navigation patterns and using them efficiently in occlusion handling.

This approach reduced data points by more than 90 percent while keeping the ARMSE within acceptable limits. This is a promising data optimization that will reduce computational time when dealing with periodic accumulative data set. The learned GP which was incorporated with Bayesian Filters was then used to track people along the various paths in the vicinity. When compared to PF and EKF trackers, both GP-PF and GP-EKF have achieved higher tracking performance when dealing with occlusions.

Comparing both Gaussian Process-BayesFilters, GP-PF has slightly performed better than GP-EKF. Furthermore, the performance of Gaussian Process-BayesFilters is not affected by the walking speed of people since state transitions are based on displacements in x and y coordinates. However, the Gaussian Process model needs to be trained for specific environments and different scenarios.



(a) Three people in a freely walking scenario are tracked by the GP-EKF tracker (blue line)

(b) Three people in a freely walking scenario are tracked by the GP-PF tracker (green line)

Figure 4.29: Tracking on three people

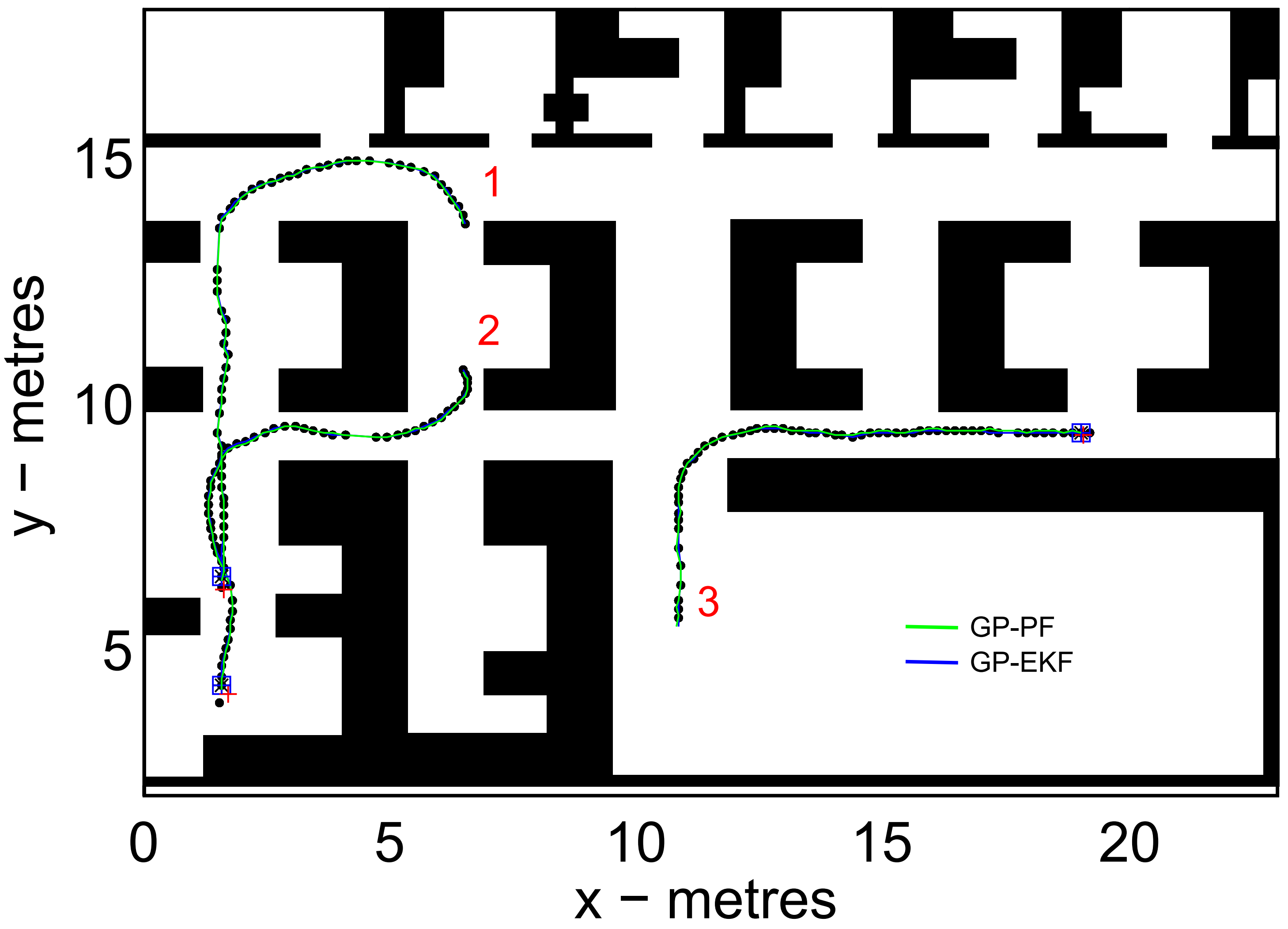


Figure 4.30: Three people in a freely walking scenario are identified by GP-EKF (blue line) and GP-PF (green line)

Chapter 5

Conclusion

The objective of this thesis is to provide solutions on the performance of laser tracker towards people tracking technique when observations are partially absent and blocked by obstacles. This chapter summarises the contributions of this thesis. Section 5.1 highlights the major theoretical practical solutions it has offered. Future research directions are addressed in Section 5.2.

5.1 Summary of Contributions

The major contributions arise from the issue of temporally failed people tracking by various tracking techniques: analysis of an underlying problem on tracking techniques attributed to tracking failure, experimental analysis on the non-parametric estimation method using the Gaussian Process Tracker, and data optimisation methods. The main contributions are summarised in the following subsections.

5.1.1 Tracking by Detection Technique

A single-layer 2D laser range finder was chosen as an observation device, and it was set up at torso height in order to have faster computing time and to avoid synchronisation issues in a multiple-layered 2D laser range finder. Torso height between 110 cm and 140 cm was chosen as the place of detection due to the cross section of the torso, which could be generally approximated as an ellipse. By comparison, support vector machines performed better than other classifiers, and they were chosen to train extracted features for training data. In the classification process, due to non-linear classification problems, binary classification for people and others was implemented. For the observation data that is taken from moving observers (LRF), scan matching such as iterative closest point (ICP) was implemented in order to have global coordination for consecutive scans. The detection of people and predicting their course were carried out using the Interacting Multiple Model Probabilistic Data Association Filter (IMMPDAF) tracker. The IMMPDAF tracker was used to overcome the poor performance of tracking on crossing targets and track maintenance on manoeuvring targets. Tracker was analysed for consistency by off-line multiple-run (Monte Carlo simulation) tests of the normalised estimation error squared (NEES) and the normalised innovation squared (NIS). In the simulation and experimental tests, the results show that the IMMPDAF tracker has difficulties handling any target with a long period of occlusion. Thus, it is important to introduce estimation that is able to improve temporal prediction of the target.

5.1.2 Non-Parametric Estimation Method

State estimation of a dynamical system mainly in human motion is one of the problems in various applications of robotics and security systems. Most state estimation models are parametric representations of the ongoing processes with parameters and noise components. However, it is hard to establish accurate parametric models since predictive state estimation has limited capabilities. Thus, non-parametric methods such as the Gaussian process regression model are an option to provide uncertainty estimates for their predictions. These non-parametric prediction and observation models can be combined with particle filters as the Gaussian Process-Particle Filter (GP-PF), where the underlying models and all the parameters can be learned from training data using non-parametric regression.

5.1.3 Data Optimisation and Management

It is regularly needed to incorporate all the samples into the training phase of the GP. This process becomes inefficient whenever the GP needs to be trained to accommodate new observations with an increasing number of samples since it will extend the duration of the training phase and increase the need for larger computer memory, which will eventually lengthen the computing time. Thus, a mutual information (MI)-based strategy and Mahalanobis distance (MD)-based criteria are proposed to optimise the number of samples that are necessary for the training phase of the GP. The MI sequentially picks the most informative measurements to represent the GP surface. Whenever a new observation is available, the MD is calculated between the new measurement and the GP. If MD is within 95% of the confidence interval, the new measurement is not counted as the GP is already capable of representing the data. Otherwise, if the MD is higher than 95% of the confidence interval, the data is not representative of the GP, and it needs to be counted. This process will optimise the data and adapt it to new scenarios.

5.1.4 Experimental Validation

Experimental results are used to validate the proposed approach for circular trajectory tracking and multiple trajectory tracking with occlusions. The GP-learned model incorporated with Partical Filter (PF) and Extended Kalman Filter (EKF) is initially demonstrated with circular trajectories with occlusions. Later, multiple trajectories of tracking with occlusions are carried out under various conditions. Experiments have demonstrated that Gaussian Process-Bayes filters with optimised training data are feasible even under multiple target interactions and occlusions.

5.2 Future Research

Research in several directions to extend the work presented in this thesis is discussed below.

5.2.1 Improvements to the Optimisation Method

Area of GP Learned Model may be made further computationally efficient by investigating better representations, such as discretized into small cells with dimensions of 5 cm × 5 cm, to reduce the number of observation points that are necessarily counted for the GP surface. Within the pixel, the number of observation points can be reduced to a single point if they share the same information prior to the Mutual Information (MI) based strategy and Mahalanobis Distance (MD) based criteria data management process. The technique to determine information similarity with the number of observation points within the pixel shall be decided in succeeding research.

5.2.2 Bidirectionally Optimised GP Learned Model

In the present work, the scope of the GP learned model was established only for human motion in one direction. In order to handle the opposite-direction traffic, it is needed to form a separate set of Optimised GP-Learned Models. that represent human motion in the return direction. Eventually, two sets of Optimized GP Learned Model are setup to provide GP regression that can be directly applied to the problem of learning prediction and observation models required by particle filters. As a future direction of work, it is interesting to explore how this problem could be formulated to handle multi-directional traffic with varying speeds.

5.2.3 Partially Area-Oriented, Optimised GP Learned Model

In order to develop human motion patterns for prediction and learning, it takes some computational time to generate the GP surface, even with the latest available computing processors. In mathematical computing, there are quite a number of matrices and their size expands with the number of samples. Since it has to accommodate new observations and if the area to be covered expands, the GP needs to be learned with a large and increasing number of samples which leads to a longer computational period. The investigation of managing computational and information resources when expanding into very large areas is an interesting future research topic.

5.2.4 Fusion of Optimised GP Learned Model with Images

Further development on improving the accuracy of tracking ability, fusion of the Optimised GP Learned Model with images related to the possible motion, trajectory and direction of a person (single tracking) or people (multiple tracking) can be studied. Location-oriented images that contain information related to trajectory are useful to increase the probability of possible motion. As images of a person holding something related to the direction that he or she is heading, these images can be attributed to location such as a pantry room if the person holds a mug or a printing room if the person carries paper. Incorporating multi-modal sensory information to further enhance the tracking capability and performance is another interesting area of further research.

5.2.5 Integration of Two-Dimensional and Three-Dimensional Data

Neither camera calibration information nor 3D data are used in the suggested detection and tracking techniques. However, 3D data may be useful to enhance tracking and detection capabilities. Researchers are aware of how difficult it may be to detect and track people when they are occluded. Relying on 2D visual patterns is insufficient for managing occlusion problems. Our ability to learn templates for the visible parts that capture appearance and 3D information is enhanced by the utilisation of human 3D geometry. Accordingly, 3D information can be used to distinguish between people who are occluded by other people or objects in indoor and outdoor environments.

5.2.6 Real-time Tracking and Detecting Algorithm

It is preferred to use fast detection and tracking algorithms to meet the requirements of real-time applications. However, more time is required to execute additional computations in complicated models when some efficient but computationally expensive components are added to the detection and tracking methods. New approaches are needed to meet the real-time criterion since online features extraction and learning are the main computational limits of existing algorithms.

5.2.7 Fusion of Multiple Sensors

Sensor fusion methods improve the accuracy, dependability, and resilience of systems that track people. This makes them useful for surveillance, security, navigation, and human-computer interaction to solve problems caused by occlusion. The monitored area can be covered by integrating a camera, LiDAR, motion detector, and other sensor data. The system can identify and monitor individuals in real-time via sensor fusion, improving accurate detection and reducing false alarms. This enhances the overall efficiency of detection and tracking. Utilising sensor fusion techniques with data from several sensors enhances the accuracy, reliability, and durability of people-monitoring systems. They are extremely advantageous in numerous practical situations, such as surveillance, security, navigation, and human-computer interface, among other fields.

Bibliography

[1] J. L. Martínez, J. González, J. Morales, A. Mandow, and A. J. García-Cerezo, "Mobile robot motion estimation by 2d scan matching with genetic and iterative closest point algorithms," *Journal of Field Robotics*, vol. 23, no. 1, pp. 21–34, 2006. [pp. vii, 25, 26]

[2] K. Arras, O. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *2007 IEEE International Conference on Robotics and Automation.*, pp. 3402–3407, IEEE, 2007. [pp. 8, 10, 12, 13, 19]

[3] L. Spinello and R. Siegwart, "Human detection using multimodal and multidimensional features," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008*, pp. 3264–3269, 2008. [pp. 8]

[4] K. Koide, J. Miura, and E. Menegatti, "Monocular person tracking and identification with on-line deep feature selection for person following robots," *Robotics and Autonomous Systems*, vol. 124, p. 103348, 2020. [pp. 8, 10]

[5] E. Chebotareva, K.-H. Hsia, K. Yakovlev, and E. Magid, "Laser rangefinder and monocular camera data fusion for human-following algorithm by pmb-2 mobile robot in simulated gazebo environment," 2021. [pp. 8, 40]

[6] X. Li, C. Liu, J. Li, M. Baghdadi, and Y. Liu, "A multi-sensor environmental perception system for an automatic electric shovel platform," *Sensors*, vol. 21,

p. 4355, 2021. [pp. 9]

[7] P. Wang, “Research on comparison of lidar and camera in autonomous driving,” *Journal of Physics: Conference Series*, vol. 2093, p. 012032, 2021. [pp. 9]

[8] F. Lourenco and H. Araujo, “Intel realsense sr305, d415 and l515: Experimental evaluation and comparison of depth estimation,” in *16th International Conference on Computer Vision Theory and Applications*, 2021. [pp. 9]

[9] A. Borcs, B. Nagy, and C. Benedek, “Instant object detection in lidar point clouds,” *IEEE Geoscience and Remote Sensing Letters*, pp. 992–996, 2017. [pp. 10]

[10] Z. Dai, A. Wolf, P. Ley, T. Glück, M. C. Sundermeier, and R. Lachmayer, “Requirements for automotive lidar systems,” *Sensors*, vol. 22, no. 19, p. 7532, 2022. [pp.]

[11] Y. Li and J. Ibañez-Guzmán, “Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems,” *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, 2020. [pp. 10]

[12] M. E. Warren, “Automotive lidar technology,” in *2019 Symposium on VLSI Circuits*, pp. C254–C255, IEEE Xplore, 2019. [pp.]

[13] D. Katkoria and J. Sreevalsan-Nair, “Rosels: Road surface extraction for 3d automotive lidar point cloud sequence,” in *Proceedings of the 3rd International Conference on Deep Learning Theory and Applications, DeLTA 2022, Lisbon, Portugal, July 12-14, 2022* (A. L. N. Fred, C. Sansone, O. Gusikhin, and K. Madani, eds.), pp. 55–67, SCITEPRESS, 2022. [pp. 10]

[14] J. Chen, P. Ye, and Z. Sun, “Pedestrian detection and tracking based on 2d lidar,”

The 2019 6th International Conference on Systems and Informatics (ICSAI 2019), pp. 421–426, 2019. [pp. 10]

[15] H. Beomsoo, A. A. Ravankar, and T. Emaru, "Mobile robot navigation based on deep reinforcement learning with 2d-lidar sensor using stochastic approach," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, 2021. [pp.]

[16] H. Beomsoo, A. A. Ravankar, and T. Emaru, "Mobile robot navigation based on deep reinforcement learning with range only sensor," in *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, vol. 2021, pp. 1P1–L07, 2021. [pp.]

[17] D. L. Tomasi and E. Todt, "Cbnav: Costmap based approach to deep reinforcement learning mobile robot navigation," in *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, 2021. [pp. 10]

[18] L. Cambuim and E. Barros, "Fpga-based pedestrian detection for collision prediction system," *Sensors*, vol. 22, p. 4421, 06 2022. [pp. 10]

[19] Z. Zhang and K. Kodagoda, "Multi-sensor approach for people detection," in *Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005.*, pp. 355–360, IEEE, 2005. [pp.]

[20] Z. Zivkovic and B. Krose, "Part based people detection using 2d range data and images," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 214–219, IEEE, 2007. [pp. 10]

[21] O. Mozos, R. Kurazume, and T. Hasegawa, "Multi-part people detection using

2d range data," *International Journal of Social Robotics*, vol. 2, no. 1, pp. 31–40, 2010. [pp. 10, 12, 13]

[22] D. S. Michael J. Jones, "Pedestrian detection using boosted features over many frames," in *19th International Conference on Pattern Recognition 2008*, 2008. [pp. 10]

[23] A. Carballo, A. Ohya, and S. Yuta, "Multiple people detection from a mobile robot using double layered laser range finders," in *2009 IEEE International Conference on Robotics and Automation (ICRA2009) Workshop*, 2009. [pp. 10, 12]

[24] C. Harrison and K. Robinette, "Caesar: Summary statistics for the adult population (ages 18-65) of the united states of america," tech. rep., DTIC Document, 2002. [pp. 11]

[25] A. Fod, A. Howard, and M. Mataric, "A laser-based people tracker," in *Proceedings. ICRA'02. IEEE International Conference on Robotics and Automation, 2002.*, vol. 3, pp. 3024–3029, IEEE, 2002. [pp. 12]

[26] M. Scheutz, J. McRaven, and G. Cserey, "Fast, reliable, adaptive, bimodal people tracking for indoor environments," in *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004).*, vol. 2, pp. 1347–1352, IEEE, 2004. [pp.]

[27] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *International Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003. [pp. 12]

[28] E. A. Topp and H. I. Christensen, "Tracking for following and passing persons," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005).*, pp. 2321–2327, IEEE, 2005. [pp. 12]

[29] M. Hashimoto, Y. Matsui, and K. Takahashi, "People tracking with in-vehicle multi-laser range sensors," in *SICE, 2007 Annual Conference*, pp. 1851–1855, IEEE, 2007. [pp. 12]

[30] A. H. A. Rahman, K. A. Z. Ariffin, N. S. Sani, and H. Zamzuri, "Pedestrian detection using triple laser range finders.," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 7, no. 6, 2017. [pp. 12, 13]

[31] Z. Zivkovic and B. Krose, B.se, "People detection using multiple sensors on a mobile robot," in *Unifying perspectives in computational and robot vision*, pp. 25–39, Springer, 2008. [pp. 13]

[32] A. Carballo, A. Ohya, and S. Yuta, "Reliable people detection using range and intensity data from multiple layers of laser range finders on a mobile robot," *International Journal of Social Robotics*, vol. 3, no. 2, pp. 167–186, 2011. [pp. 13]

[33] I. Ullah, X. Su, X. Zhang, and D. Choi, "Simultaneous localization and mapping based on kalman filter and extended kalman filter," *Wirel. Commun. Mob. Comput.*, vol. 2020, pp. 2138643:1–2138643:12, 2020. [pp. 14, 72]

[34] K. R. S. Kodagoda, W. S. Wijesoma, and A. P. Balasuriya, "Road curb and intersection detection using a 2D LMS," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 19–24, 2002. [pp. 14]

[35] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999. [pp. 16]

[36] E. Frank, M. Hall, and L. Trigg, "Weka3: Data mining software in java," *The University of Waikato*, 2006. [pp. 19, 22]

[37] L. Spinello, R. Triebel, and R. Siegwart, "Multimodal people detection and tracking in crowded scenes," in *Proc. of The AAAI Conference on Artificial Intelligence (Physically Grounded AI Track)*, 2008. [pp. 22]

[38] V. Vapnik, "Estimation of dependences based on empirical data. 1982," *NY: Springer-Verlag*, 1982. [pp. 24]

[39] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992. [pp. 24]

[40] A. Y. Hata and D. F. Wolf, "Terrain mapping and classification using support vector machines," in *Proc. 6th Latin American Robotics Symp. (LARS)*, pp. 1–6, 2009. [pp. 24]

[41] C. Hsu, C. Chang, C. Lin, *et al.*, "A practical guide to support vector classification," 2003. [pp. 24]

[42] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992. [pp. 29]

[43] Z. Zhang, *Iterative point matching for registration of free-form curves*. PhD thesis, Inria, 1992. [pp. 29]

[44] Y. Nakamori, Y. Hiroi, and A. Ito, "Multiple player detection and tracking method using a laser range finder for a robot that plays with human," *ROBOMECH Journal*, vol. 5, pp. 1–15, 9 2018. [pp. 40]

[45] F. D. Bar-Shalom and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, pp. 82–100, 2009. [pp. 40]

[46] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975. [pp. 40, 43]

[47] B. Kovacevic, D. Ivkovic, and Z. Radosavljevic, "Immpdaf approach for l-band radar multiple target tracking," in *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–5, IEEE Xplore, 2020. [pp. 47, 54]

[48] K. R. S. Kodagoda, S. S. Ge, W. S. Wijesoma, and A. P. Balasuriya, "Immpdaf approach for road-boundary tracking," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 478–486, 2007. [pp. 47, 48, 54, 55, 56, 57]

[49] D. Lerro and Y. Bar-Shalom, "Interacting multiple model tracking with target amplitude feature," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 494–509, 1993. [pp. 48]

[50] X. R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," *IEEE Transactions on Control Systems Technology.*, vol. 1, no. 3, pp. 186–194, 1993. [pp. 50]

[51] N. G. Wah, *Intelligent Systems: Fusion, Tracking and Control*. Research Studies Press Ltd., 2003. [pp. 51]

[52] Y. Bar-Shalom and X.-R. Li, "Estimation and tracking- principles, techniques, and software," *Norwood, MA: Artech House, Inc, 1993.*, 1993. [pp. 52, 58, 64]

[53] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004. [pp. 53]

[54] Y. Bar-Shalom and X.-R. Li, *Estimation and tracking: principles, techniques, and software*, vol. 393. Artech House Norwood, 1993. [pp. 54]

[55] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems," *Boston, MA: Artech House*, 1999. [pp. 71]

[56] A. Bruce and G. Gordon, "Better motion prediction for people-tracking," in *Proc. of the Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain*, 2004. [pp. 71, 73]

[57] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, pp. II–786, IEEE, 2001. [pp. 71]

[58] L. Beyer, A. Hermans, T. Linder, K. O. Arras, and B. Leibe, "Deep person detection in two-dimensional range data," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2726–2733, 2018. [pp. 71]

[59] K. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *ICRA 2008. IEEE International Conference on Robotics and Automation, 2008.*, pp. 1710–1715, IEEE, 2008. [pp. 71]

[60] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Multi-modal tracking of people using laser scanners and video camera," *Image and vision Computing*, vol. 26, no. 2, pp. 240–252, 2008. [pp. 71]

[61] S. Chan, R. E. Warburton, G. Gariepy, J. Leach, and D. Faccio, "Non-line-of-sight tracking of people at long range," *Optic Express*, vol. 25, no. 9, pp. 10109–10117, 2017. [pp. 71]

[62] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, MA, 2006. [pp. 71, 74, 75]

[63] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009. [pp. 71, 74, 80, 81]

[64] J. Ko and D. Fox, "Learning gp-bayesfilters via gaussian process latent variable models," *Autonomous Robots*, vol. 30, no. 1, pp. 3–23, 2011. [pp. 71]

[65] C. Plagemann, D. Fox, and W. Burgard, "Efficient failure detection on mobile robots using particle filters with gaussian process proposals," in *Proc. of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2007. [pp. 71]

[66] D. Avots, E. Lim, R. Thibaux, and S. Thrun, "A probabilistic technique for simultaneous localization and door state estimation with mobile robots in dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, vol. 1, pp. 521–526, IEEE, 2002. [pp. 72]

[67] L. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986. [pp. 72]

[68] T. L. Dean and M. S. Boddy, "An analysis of time-dependent planning.," in *AAAI'88: Proceedings of the Seventh AAAI National Conference on Artificial Intelligence*, vol. 88, pp. 49–54, 1988. [pp. 72]

[69] P. S. Maybeck, *Stochastic models, estimation, and control*, vol. 3. Academic press, 1982. [pp. 72]

[70] S. Bhaumik and P. Date, *The Kalman filter and the extended Kalman filter*, ch. Book: Nonlinear Estimation, pp. 27–50. Chapman and Hall/CRC, 2019. [pp.]

[71] A. Roux, S. Changey, J. Weber, and J.-P. Lauffenburger, "Projectile trajectory estimation: performance analysis of an extended kalman filter and an imperfect invariant extended kalman filter," in *2021 9th International Conference on Systems and Control (ICSC)*, 2021. [pp. 72]

[72] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*, pp. 182–193, International Society for Optics and Photonics, 1997. [pp. 72]

[73] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: Location estimation using sparse and noisy sensor data," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 723–728, IEEE, 2003. [pp. 73]

[74] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005. [pp. 73]

[75] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," *Pedestrian and evacuation dynamics*, vol. 21, no. 2, pp. 21–58, 2002. [pp. 73]

[76] T. Beckers, "An introduction to gaussian process models," *arXiv, Cornell University https://arxiv.org/abs/2102.05497*, 2021. [pp. 74]

[77] C. Guestrin, A. Krause, and A. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, pp. 265–272, ACM, 2005. [pp. 78]

[78] G. Schay, *Introduction to probability with statistical applications*. Birkhäuser, 2007. [pp. 79]

[79] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002. [pp. 79]

[80] G. Kitagawa and S. Sato, "Monte carlo smoothing and self-organising state-space model," in *Sequential Monte Carlo methods in practice*, pp. 177–195, Springer, 2001. [pp.]

[81] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009. [pp.]

[82] O. Frank, J. Nieto, J. Guivant, and S. Scheding, "Multiple target tracking using sequential monte carlo methods and statistical data association," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3, pp. 2718–2723, IEEE, 2003. [pp. 79]

[83] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting," in *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]* (S. Becker, S. Thrun, and K. Obermayer, eds.), pp. 529–536, MIT Press, 2002. [pp. 80]