UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Robotic Perception of Pedestrians in Crowded Environments

by Alexander Joseph Virgona

Thesis submitted in fulfilment of the requirements for the degree of

Master of Engineering (Research)

under the supervision of Alen Alempijevic and Teresa Vidal-Calleja

University of Technology Sydney
Faculty of Engineering and Information Technology

December 2022

Certificate of Original Authorship

I, Alexander Joseph Virgona declare that this thesis, is submitted in fulfilment of the requirements for the award of Master of Engineering (Research), in the Robotics Institute, 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.

Signed: Production Note:
Signature removed prior to publication.

Date: 16/10/2023

UNIVERSITY OF TECHNOLOGY SYDNEY

Abstract

Faculty of Engineering and Information Technology
Robotics Institute

Master of Engineering (Research)

by Alexander Joseph Virgona

Robots are no longer science fiction. As their capabilities and affordability have grown, they've greatly impacted industries such as manufacturing, mining, and logistics, increasing the productivity of these industries by taking over many dangerous, dirty or dull tasks and freeing humans to focus on more interesting work. For the most part however, these robots are deployed in environments where they are isolated from humans; robots work best with other robots and machines. There remains an untapped potential for robotic technologies to enhance our daily lives and work collaboratively with us, but to do this safely and effectively they must be able to perceive humans in their environment. This is a challenging problem as humans can vary wildly in their appearance and as human environments are often dynamic and cluttered, a long way from the precisely controlled environment of the production line.

The work presented in this thesis aims to enable robots and intelligent systems to better perceive humans through contributions to the core capabilities of detection and tracking. Considering that many human-robot interactions are likely to involve sharing walking space, this thesis considers these perception problems at the level of pedestrian interactions. A novel method for detecting the location and orientation of pedestrians from point-cloud data is presented which is able to handle occlusions of the lower body by virtue of focusing on the head and shoulders. Building on this detection capability, a tracking algorithm is proposed which leverages interpersonal distance constraints and assumptions

about relationship between shoulder alignment and walking direction, to maintain robust estimates of the pose of all pedestrians in a crowded scene. The accuracy of the pedestrian pose detection algorithm is quantitatively evaluated by comparison with precise pose estimates from an optical motion tracking system. The outputs from the detection front-end are tracked using the proposed algorithm which is evaluated based on the CLEAR-MOT tracking metrics. Tracking performance is compared to a state-of-the-art tracking algorithm fed with the same detection inputs, showing improved performance under heavy crowding.

Finally, a field study evaluates the tracking performance on real depth data captured in a busy inner city train station. The application of the technology has a patent, has been developed into a commercial product and is being trialled by a local government in Sydney, Australia as a congestion management tool. This showcases the applicability of this technology to enable the smart infrastructure of the future, able to perceive and therefore respond to human behaviour and better manage public space in our crowded cities.

Acknowledgements

I would like to sincerely thank my academic supervisors Alen Alempijevic and Teresa Vidal-Calleja for their unwavering support and encouragement. It has a been a pleasure to work with both of you and I hope to have many more opportunities to work with you in future. Thanks also to my former supervisor Nathan Kirchner for encouraging me to get into research in the first place.

I would like to express my deep thanks and appreciation to my wife Kimberley for her incredible patience, love and support throughout this degree. To my fellow students and colleagues: Julien, Phil, Laki, Cedric, Raphael, Stefan, Kara, Tamsin, Tom, Kat and too many more to mention. Thanks for all the chats, coffee, cake, climbs and good times. Thanks to my colleagues at UTS Rapido, in particular Stuart Warren who was an excellent mentor to me throughout the Dwell Track project. Thanks to Downer Rail and the Rail Manufacturing CRC for their support of this research project and to Queensland Rail and Sydney Trains for their cooperation and collaboration.

Finally a big thanks to my parents who have encouraged me and supported me in everything I've done.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

2D	two-dimensional
3D	three-dimensional
CAS	Centre for Autonomous Systems
CCTV	closed-circuit television
CNN	convolutional neural network
DOF	degree-of-freedom
FOV	field-of-view
GAN	generative adversarial network
GPU	graphical processing unit
HOG	histograms of oriented gradients
IMU	inertial measurement unit
LIDAR	light detection and ranging
LSTM	long short-term memory
MOTA	multiple object tracking accuracy
MOTP	multiple object tracking precision
POE	power-over-ethernet
QR	Queensland Rail
RANSAC	random sample consensus
RGB	red, green and blue colour

RGBD colour and depth

RMCRC Rail Manufacturing Cooperative Research Centre

SVM support vector machine

Chapter 1

Introduction

1.1 Why Do We Need Robotic Perception of Pedestrians?

The growing maturity of robotic technology has seen robots greatly impact many industries, from mining and manufacture to logistics. In many of these industrial settings the most effective way to deploy robotics to date, has been to exclude humans from robot workspaces. The main reasons for this are safety and efficiency. Industrial robots are often heavy, powerful, fast, or all of the above and can pose a great risk to the safety of humans in their workspace. Consider for instance the Kalmar Autostrad pictured in Figure 1.1 which autonomously moves shipping containers around in a yard isolated from humans. These industrial robots work best in precisely controlled, predictable environments and collaborate best with other robots, rather than humans who are by comparison approximate and unpredictable.

However as the sophistication of robotic technology increases there is growing desire and opportunity to deploy robots into environments populated by people. These *social* robots need to be capable of safely and efficiently sharing space and collaborating with humans. Examples of these robots have already begun to emerge but we are yet to see them become ubiquitous in society due to the challenges that social robotics poses.

One of the key challenges for social robots is the perception of humans in their environment. In order for robots to share space and cooperate with humans they must be able to perceive

FIGURE 1.1: The Kalmar Autostrad autonomously moves shipping containers around a shipping terminal devoid of humans

them. But what does it mean for a robot to *perceive* humans? Let's begin our discussion by unpacking the title of this thesis and defining some of the key terms.

1.2 What is Robotic Perception of Pedestrians in Crowded Environments?

1.2.1 Robotic Perception

When the general public imagine a robot they likely picture a bipedal humanoid and while such robots do exist, the definition of a robot is much more inclusive than this. Broadly speaking, a robot is any machine capable of interacting with the world around it autonomously. The capabilities required to achieve this feat can be grouped into three main categories: sensing, cognition, and actuation.

Sensing is how a robot collects information about the world around it, and involves a vast myriad of sensor devices such as: cameras, depth sensors, microphones, force sensors, and encoders. Actuation describes any technology which allows a robot to act upon the world

around it including: motors, speakers, lights, and robotic arms. Cognition is how a robot processes the information from it's sensors to determine what actions to take. This takes the form of algorithms implemented in software and run on computers. Perception sits somewhere in the overlap between sensing and cognition. Given raw data from sensors, perception algorithms are used to extract higher levels of information that may be used by cognitive algorithms to make decisions about how to act.

Perception algorithms are well studied; researchers in the field of computer vision have been posing and solving perception problems now for over 50 years. While the study of *robotic perception* overlaps significantly with computer vision it also implies a particular relevance to problems faced by robots. This distinction can be characterised by two main ideas. The first is that robots need spatial information.

In order to make decisions about how to move though space robots must have knowledge of the location and movement of people and objects in that space. Deep learning based, computer vision algorithms such a YOLO [3] (and its derivative works [1]) pictured in Figure 1.2 achieve state-of-the-art results in person detection, however they do not directly provide spatial information. Whilst it is possible to use such approaches in combination with data from an RGB-D camera to recover this spatial information, this has limitations as discussed further in Chapter 2. Rather than solve person detection in colour images and translate this to spatial information, the approach presented in Chapter 3 of this thesis solves person detection directly in the spatial domain by processing three-dimensional (3D) point-clouds to extract pedestrian poses. This yields not only the 3D position but also the shoulder orientation of pedestrians which is particularly relevant to the task of motion prediction discussed later in this thesis.

The second idea that distinguishes *robotic perception* from adjacent fields is that of uncertainty; robots need to know what they do not know. It is not enough in robotic perception to assign an approximate position to all people in the environment. Such a position undoubtedly contains some degree of error and in order to make safe decisions a robot needs a model of the magnitude and shape of this error. This type of approach, referred to as probabilistic robotics, involves making estimations and characterising their uncertainty with probabilistic distributions. The work presented in Chapter 4 of this thesis applies a

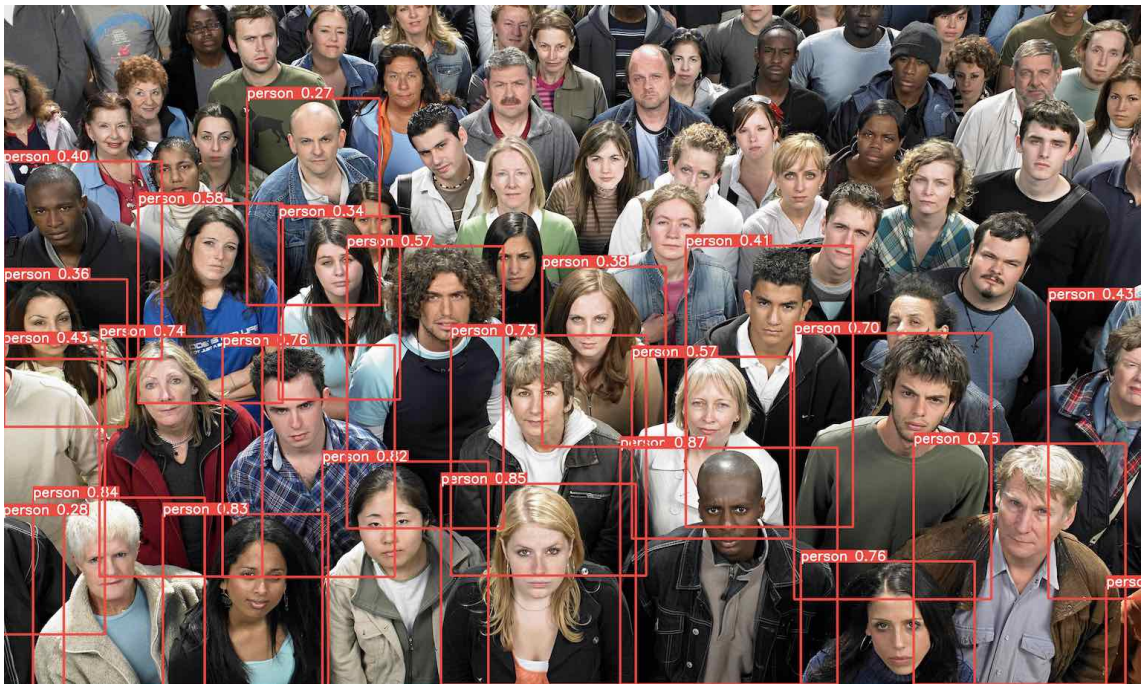


FIGURE 1.2: Person detection results from YOLOv3 [1]

common probabilistic framework called recursive Bayesian estimation to the problem of tracking pedestrians, yielding probabilistic estimates of their position, velocity and shoulder orientation.

1.2.2 Pedestrians

The word pedestrian describes any person travelling by foot, but in the context of this thesis it also implies a particular level of detail. The work presented in Chapter 3 targets a 4 degree-of-freedom (DOF) pedestrian pose comprised of the 3D position, and orientation of the shoulders about the vertical axis. The pedestrian tracking algorithm described in Chapter 4 operates in a two-dimensional (2D) plane, disregarding the shoulder height to estimate a 3 DOF pedestrian pose however the height signal is still useful information which could be used, for instance to differentiate between individuals. The inclusion of the shoulder orientation is informative for social robotics as it conveys information about how people are interacting with each other and with their environment. Furthermore orientation estimates can provide insights into the likely movement of a person, an idea leveraged in the tracking approach presented in Chapter 4 to improve velocity estimation.

While some social robots may need to perceive people in greater detail than this, such robots will still benefit from perception at this level of detail in situations where more detailed information is not available. Furthermore for many social robots, such as service robots or autonomous vehicles, the 3 DOF pedestrian pose is sufficient to make decisions about how to move through human occupied spaces.

1.2.3 Crowded Environments

In the context of this thesis crowded environments are those which are dominated by pedestrians. Examples include train stations, footpaths, shopping centres and airports. These environments are very challenging for robotic perception algorithms as the sheer number of people and chaotic dynamics lead to frequent occlusion of people and objects of interest. Furthermore robust solutions are required to ensure that robots are able to operate in such close proximity to humans without compromising their safety. However with these challenges comes some opportunity also; humans are social creatures and perception of the behaviour of one pedestrian can provide clues as to the likely behaviour of another. The tracking approach presented in Chapter 4 takes advantage of this idea to improve tracking performance in crowded environments.

1.2.4 Pedestrian Detection

Detecting pedestrians is the first step towards higher levels of perception and is the topic of Chapter 3. In the scope of this work it refers to the use of algorithms to identify the presence, and extract the pose, of one or more pedestrians in a frame of data. Pedestrian detections are valuable information for a robot which operates in human environments. This level of information, for instance, could be used to halt or slow the movement of a robot in the presence of pedestrians, or initiate an interaction when a pedestrian is detected in front of a robot. However detections alone are not enough to enable more sophisticated tasks, such as navigating amongst pedestrians or collaborating with a pedestrian to complete a shared task. These more sophisticated behaviours require a persistent awareness of pedestrians so that the robot can make sense of sequences of pedestrian behaviour associated with particular individuals. To achieve this we require tracking.

1.2.5 Pedestrian Tracking

Tracking is the process of associating multiple detections over time with persistent targets. It typically takes advantage of assumptions about the movement, or other processes affecting the state of an object over time, to determine which target each observation belongs to. This allows for a richer understanding of these targets. In the case of pedestrians it allows the robot to determine the velocity of pedestrians which may be crucial for navigating amongst them. Furthermore it enables the collection of past walking trajectories which could be used to extract social cues and other insights from their behaviour or even to predict their future actions.

1.3 Contributions

The work in this thesis aims to enable intelligent systems that can:

- see pedestrians in their environment
- interpret the movement of pedestrians to extract insights such as social cues, interpersonal interactions, and interaction with the environment.
- operate robustly in crowded environments

In pursuit of these aims the technical contributions of this thesis are as follows.

- A novel algorithm for detecting the 4 DOF pose (x, y, z, θ_z) of pedestrians from point-cloud data in crowded environments.
- A novel algorithm for robustly tracking the 3 DOF pose (x, y, θ_z) of pedestrians in crowded environments.
- A field study on the use of these algorithms in a prototype device for passenger congestion management in busy train stations.

1.4 Research Outputs

1.4.1 Academic Papers

Academic papers published during the course of this thesis are listed below.

N. Kirchner, A. Alempijevic, A. Virgona, X. Dai, P. G. Pl, and R. K. Venkat. A robust people detection, tracking, and counting system. In *Australasian Conference on Robotics and Automation*, pages 2–4, 2014

A. Virgona, N. Kirchner, and A. Alempijevic. Sensing and perception technology to enable real time monitoring of passenger movement behaviours through congested rail stations. *Australasian Transport Research Forum*, (October):1–14, 2015

A. Virgona, A. Alempijevic, and T. Vidal-Calleja. Socially constrained tracking in crowded environments using shoulder pose estimates. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4555–4562, 2018. ISSN 10504729. doi: 10.1109/ICRA.2018.8461030

1.4.2 Patent

The work of this thesis is covered by patent:

A. Alempijevic, A. Virgona, and T. Vidal-Calleja. Monitoring systems, and computer implemented methods for processing data in monitoring systems, programmed to enable identification and tracking of human targets in crowded environments, Issued to University of Technology Sydney and Downer EDI Rail Pty Ltd. Patent No. WO2019109142A1 / AU2018379393A1, 2018

1.4.3 Awards

The research project has received the following awards:

UTS Vice-Chancellor's Award for Research Excellence through Collaboration, 2018,

Awarded to UTS Responsive Passenger Information Systems Research Team

CRC Association's Excellence in Innovation Award, 2019,

Awarded to Downer and UTS for the Dwell Track project

1.5 Structure of this Thesis

Following this introduction, Chapter 2 discusses existing research work relevant to the problems of pedestrian detection and tracking; and situates the contributions of this thesis in relation to the state-of-the art. Chapter 3 describes my novel pedestrian detection algorithm and evaluates the algorithm on a point-cloud dataset with precise ground truth obtained from a commercial motion capture system. Chapter 4 describes a pedestrian tracking algorithm which leverages shoulder pose estimates and social constraints to improve performance in crowded environments. To explore the applicability of these algorithms to real world problems Chapter 5 discusses field trials and the development of a prototype system for passenger congestion management in busy train stations. Finally Chapter 6 draws conclusions from this work and proposes directions for future research.

Chapter 2

Related Work

With the broad aim of enabling intelligent systems to perceive pedestrians this thesis engages with the research problems of: person detection, human pose estimation, and person tracking. These problems are strongly related with many examples from the literature addressing several of them at once. As such, discussion of relevant related work is grouped under this chapter to give the reader a better understanding of how these topics relate to one another but divided into sections for easier reference when digesting the later chapters of the thesis. This chapter begins by discussing person detection and pose estimation in Section 2.1 which provides the background for Chapter 3. An understanding of person detection and pose estimation sets the scene for a overview of person tracking research in Section 2.2, in particular this section focuses on socially informed tracking, to prepare the reader for Chapter 4.

2.1 Person Detection

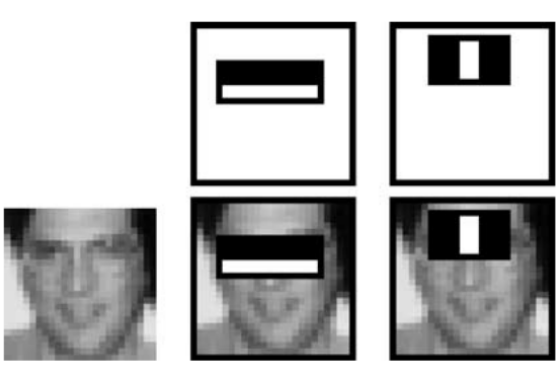
Chapter 3 of this thesis presents a novel approach to detecting and estimating the pose of people in crowded social scenes from 3D point-cloud data. Person detection and human pose estimation are mature research problems in the overlapping fields of computer vision and robotic perception. The person detection and pose estimation problems are highly related to one another and can be thought of as two ends of a spectrum. At one end, person

detection in its simplest form aims to answer the question, "Is there a person present in a given frame of sensor data?". At the other end of the spectrum human pose estimation can be as complex as recovering the full body skeletal pose of all people in the field of view of a sensor [8]. Most work in this area, including the work presented in Chapter 3 sits somewhere between these two extremes. The remainder of this chapter will use the term *person detection* to refer broadly to this spectrum of problems. Many publications in the area of person detection and human pose estimation also deal with person tracking but discussion of tracking techniques will be deferred until Section 2.2.

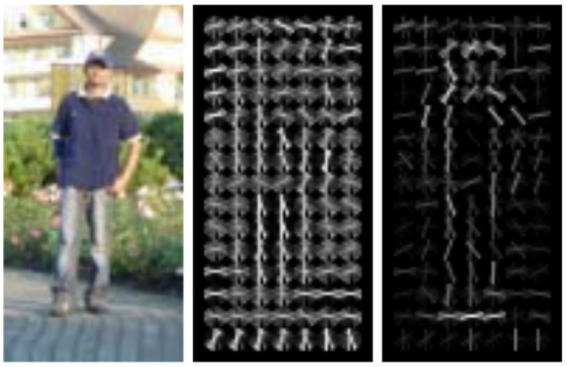
The problem of person detection has a long history in the field of *computer vision* with published work on visual analysis of human motion as far back as 1980 [9]. Work in this field has commonly, although not always, considered the problem in the context of static, monocular vision [10–13], often modelling human pose in the image space as bounding boxes [10] silhouettes [14, 15], or pixel coordinates of individual features such as the face. Conversely work from the robotic perception community often considers the problem in the context of a mobile robot with a suite of sensors of various modalities including: monocular vision, stereo vision, 2D laser range-finders, depth cameras and LIDAR. Pose estimation in the context of robotics is generally considered in 3D space as this yields more utility in robot decision making. Naturally there is a great deal of overlap and cross pollination between person detection work in the computer vision and robotics communities, the boundaries between which are becoming increasingly blurred. While a detailed review of person detection literature is beyond the scope of this thesis, this section will touch on some of the different sensing modalities that have been applied to this problem and their strengths and weaknesses. For a more thorough review the reader is directed to one of the many published surveys of the field [16–19].

2.1.1 Monocular Vision Based Person Detection

Monocular vision refers to image data from a single camera and is one of the earliest and most common sensing modalities to be applied to this problem. While vision-based solutions to the person detection problem have been around along time, early works of the 1980s and 90s, relied on strong assumptions about the motion or appearance of people to



(A) A figure from [12] illustrating the top two features selected by AdaBoost for detecting the face (top row), overlaid on a typical face (bottom row) where they align with the contrast between the eyes and cheeks, and the eyes and nose respectively.



(B) A figure from [13] showing an image of a person (left), the associated HOG descriptor (middle), the HOG descriptor weighted by SVM weights for a person

FIGURE 2.1: Examples of early works in monocular person detection

segment them from the image [14, 20, 21]. Later work in the late 90s and early 2000s improved upon this by extracting manually devised features from images and using machine learning models to perform classification. This marked a shift away from perception algorithms based on explicit human insights and towards more data driven approaches able to learn models from training data, albeit using manually devised features. Noteworthy work of this era includes Viola-Jones face detection [11, 12] which used a cascade of boosted classifiers trained on "Haar-like" image features to efficiently detect human faces. Another influential work from this time was histograms of oriented gradients (HOG) person detection [13] which extracted gradient based descriptors from a grid of sub-regions over an image and used a support vector machine (SVM) to determine if a person was present.

In the mid 2000s, driven by increasing power and affordability of the graphical processing unit (GPU), deep neural networks and in particular convolutional neural network (CNN)s began to show impressive results in many perception tasks including person detection. In comparison to previous machine learning based person detection algorithms which were able to learn the relationship between a set of features and a desired detection output, these deep learning based approaches are able to learn the relationship directly from the input image to the desired output, effectively learning to extract whatever intermediate features are best suited to the task. An advantage of deep learning based perception algorithms is that a general network architecture can be used for visual object detection

and, given adequate training data, learn to detect diverse and complex classes of object. The limitation of such data driven algorithms is the large amount of labelled data needed to train robust models and the difficulty of ensuring that models will generalise to unseen data.

Nonetheless deep learning based algorithms have come to dominate many perception tasks with CNN based algorithms [1, 3] able to efficiently detect the location of multiple objects in a scene in terms of 2D bounding boxes in the image space. More recently transformer networks have been applied to object detection problems (in combination with CNNs) [22] doing away with hand crafted elements such as anchor boxes and non-max suppression, albeit at the cost of higher convergence times.

While these results are very impressive, it is desirable in a robotics context to convert such detections into 3D coordinates to aid in spatial decision making tasks such as navigation. While it may be possible to combine 2D bounding boxes with depth data from colour and depth (RGBD) sensors to obtain 3D positions such a process is likely to be error prone in crowded environments where occlusions are common due to the coarseness of the bounding box representation.

Work such as DeepPose [23] improves on this situation, using a deep learning based approach to extract the skeletal pose of individuals from red, green and blue colour (RGB) images but ultimately these skeletal poses, while more detailed, are still in image coordinates and must be combined with depth information to obtain 3D, a task refered to in the literature as 2D-3D lifting. Furthermore a 2018 study [24] into the effect of occlusions on state-of-art deep-learning based human pose estimation found that the performance of such methods drops significantly when occlusions are introduced rendering them unsuitable for the crowded scenarios target by this thesis.

2.1.2 Person Detection in Three-Dimensional Data

As discussed above, in the context of robotics it is desirable to obtain 3D human poses to aid in tasks such as social navigation. Several distance sensing technologies can be applied to this problem in order to obtain pose estimates in spatial coordinates. The simplest

among these are laser rangefinders which measure the distance to the nearest surface within a 2D scanning plane using a spinning infrared light detection and ranging (LIDAR). Methods have been proposed to detect people in laser rangefinder data based on both the legs [25, 26] and the upper body [27]. Unfortunately the planar perspective of laser rangefinders renders them fundamentally more susceptible to occlusion issues which are already common in the crowded scenes targeted by this thesis.

Depth cameras overcome this issue by providing the same field-of-view as monocular cameras but measuring the distance to surfaces rather than light intensity and color. There are multiple types of depth cameras each with their own advantages including time-of-flight, structured light, and even stereoscopic cameras. What these technologies have in common is they output a depth image which can optionally be converted to a 3D point-cloud given a model of the cameras optics. As such there is a great deal of work in robotic perception dedicated to interpreting both depth images and point-clouds particularly since the release the Microsoft Kinect V1 in 2010 made depth cameras extremely affordable.

Researchers from Microsoft [28] propose a method for real-time full body pose estimation from single frames of depth data that uses randomised decision forests trained on large synthetic depth image datasets to classify pixels with body part labels. From these pixel labels body part modes are extracted to generate a set of confidence weighted 3D joint proposals. This approach allows full skeletal models to be extracted in real-time but relies on observing 31 separate body parts which is not feasible in crowded scenes.

To overcome problems associated with partial occlusion [29] maintain a 3D occupancy grid of the environment which they use to segment foreground voxels and determine weither they are observable or not. Voxels in the foreground are clustered based on connectivity and used as observations to inform a particle filter to estimate a 25 DOF body pose. Unfortunately the complexity of this pipeline is such that even with GPU acceleration their implementation only runs at 4Hz while tracking a single target and is unlikely to scale to robustly track multiple targets in a crowded scene. At any rate such a detailed skeletal model is not required for our pedestrian tracking application.

More recently [30] use voxel feature encoders in an end-to-end deep-learning approach trained on the KITTI dataset to perform 3D person detection in LIDAR data. Their

approach reportedly deals well with occlusions however results are quantified in terms of Average Precision, a metric used to summarise the success rate of a classification task suggesting that position accuracy was not a priority. Furthermore the output of the method is 3D bounding boxes only, with no estimate of orientation the importance of which is discussed further in the context of tracking in Chapter 4.

2.2 Person Tracking

2.2.1 Fundamental Approaches to Tracking

Tracking is a fundamental problem in robotics and a key component of the tracking problem is state estimation. The vast majority of tracking algorithms used over the past several decades have addressed this problem using Recursive Bayesian Estimation (or equivalently Bayesian filtering). This estimation framework maintains a probabilistic belief over the state space of a tracked target and recursively updates this belief to incorporate information from motion models and observations over time according to Bayes rule. Probabilistic frameworks are a valuable tool in robotics because they allow robots to reason about their uncertainty of the world when making decisions. The two most commonly used types of Bayesian filter are Kalman filters and particle filters (a.k.a. Sequential Monte Carlo Methods).

The Kalman filter, originally formulated in 1960 [31] offers an efficient solution to state estimation problems where the distribution of possible states and measurement model are Gaussian and the motion model is linear. Despite its age this technique is still relevant and used in recent state of the art tracking work [2, 32, 33]. The Extended Kalman Filter (EKF) [34] extends this method to cover non-linear models by taking a linear approximation of the model at the mean of the distribution.

Particle filters are a flexible framework for probabilistic state estimation involving non-Gaussian distributions and non-linear models of motion and observation processes. This flexibility combined with a sharp increase in computational power since the 1990s has

made particle filters a popular tool for many applications in robotics, and indeed they form the basis of the socially constrained tracking approach proposed in Chapter 4.

Apart from the state estimation problem addressed by Bayesian filters, multiple target tracking also requires solving the data association problem of which observations should be used to update which filters. Popular solutions to this problem include: greedy nearest-neighbour which repeatedly matches the two nearest candidates until there are none available; and the Hungarian Algorithm [35] which finds the lowest cost assignment between two sets (targets and observations) given their pairwise association costs. Despite their age these fundamental techniques are still relevant in modern tracking approaches [2, 32, 33]. Rather than commit to an explicit data association per frame the Joint Probabilistic Data Association Filter (JPDAF) [36] jointly reasons over the data association problem and filtering problem within a single framework. A drawback of this approach is that the state of targets that are close together can converge due to the sharing of observations.

While the approaches discussed in above (Section 2.2.1) form the foundations of many modern tracking approaches they do not take advantage of any insights specific to tracking people. In the context of these fundamental approaches there are two potential avenues for improving tracking of pedestrian: using knowledge of pedestrian motion to improve predictions between frames; or using features of pedestrians to improve data association.

2.2.2 Modelling Pedestrian Motion

A popular approach to modelling pedestrian motion has been the social force model (SFM) [37] which models pedestrians as particles in space under the influence of attractive forces towards their goal, and repulsive forces away from other pedestrians. These ideas have been popular in robotics with numerous works in pedestrian motion tracking and prediction based on SFM [38–40]. Unfortunately the SFM framework requires knowledge of the intended destination of pedestrians to define the attractive force, which is a challenging task in its own right. Furthermore in our own experiments, even with reasonable assumptions about intended destination, the close proximity of targets and frequency of occlusions in crowded environments caused instability in the state estimates.

Many modern approaches to modelling pedestrian motion use deep-learning architectures trained on pedestrian motion datasets to predict pedestrian trajectories. Authors of [41] use a recurrent long short-term memory (LSTM) module per target for trajectory prediction with a social pooling layer which shares the hidden states between nearby targets. In their take on this problem [42] aims to learn the relative importance of neighbouring pedestrians rather than rely on proximity as a proxy to this. Citing the multi-modal nature of the trajectory prediction problem [43] use a generative adversarial network (GAN) based architecture to learn to propose "socially plausible" trajectories while encouraging variety with a novel loss term.

While these learning based pedestrian trajectory prediction algorithms are capable of producing state-of-the-art results on horizons up to 4.8 seconds they do not offer significant improvements over a constant velocity model on a single time step basis (i.e. 33ms) as needed to benefit recursive Bayesian estimation [44]. Additionally these methods are trained and evaluated on publicly available pedestrian datasets [45, 46] where the subjects typically follow smooth trajectories. By contrast the work of this thesis is evaluated on a dataset designed to simulate conditions on a crowded commuter rail platform and hence contains complex pedestrian interactions and changes of direction.

2.2.3 Improving Data Association

Apart from using better motion models another way to enhance the performance of a tracking algorithm is to improve data association. The discriminative correlation filter (DCF) and derivative approaches [47, 48] learn a visual filter of their target which they use to search subsequent images for matches. The filter is updated each frame allowing the approach to adapt to gradually changing appearance.

In a similar vein the Deep SORT tracker [33] uses a pre-trained CNN to extract deep appearance descriptors of tracked targets. At each step the cosine distance between descriptors of tracks and observations is used in combination with the spatial metric from SORT [32] to determine associations. These approaches present impressive results in feature based tracking however they rely on sufficient variation between targets in a scene for the sake of discriminating them. In the crowded human environments targeted in this

thesis it is common for many subjects to have similar appearance, for business persons wearing dark wearing dark suits. Furthermore these approaches can generally only be applied to colour images and are not well suited to depth or 3D data as they rely on rich visual information to recognise tracked targets in subsequent frames.

The authors of TesseTrack [49] improve data association by learning to perform 3D pose estimation from multiple viewpoints, spatio-temporal tracking and matching all in an end-to-end differentiable framework. They show state-of-the-art results in multi-view person tracking but clearly benefit from multiple camera views with regard to occlusions. In the crowded scenarios targeted by this thesis not only are multiple views not available but occlusions are more severe due to higher person density.

2.3 Summary

While learning based approaches dominate person detection and tracking in colour images, recovering the 3DOF pose of pedestrians from these approaches is error prone in crowded environments. Meanwhile 3D point-clouds can be collected using a variety of sensors, provide natural spatial separation between targets, and contain valuable information about the physical pose of pedestrians.

State-of-the-art approaches to tracking pedestrians in robotics applications generally rely on fundamental approaches based on recursive Bayesian estimation because of their versatility and ability to model uncertainty. While more complex approaches exist comparative studies [2] still find that simple approaches to work well and are more computationally efficient. Nonetheless there remain opportunities to augment these fundamental approaches to better track pedestrians in crowds.

Chapter 3

Detecting People in Crowds

3.1 Introduction

The first step towards perception of people is to detect them. The task of detecting people and extracting their pose is challenging, as human environments are typically dynamic and unstructured, and people come in a variety of shapes, sizes, and appearances. The difficulty of this task is further increased in crowded environments due to frequent visual occlusions, and close proximity of people to one another.

Person detection and human pose extraction are both mature research topics with researchers applying a variety of sensing modalities, such as cameras, laser range finders and depth sensing cameras, to detect and estimate the pose of people in various scenarios. Amongst these approaches, those based on monocular vision are perhaps the most mature. The nature of monocular vision however, means that these techniques typically provide pose estimates in image coordinates and hence rely on multiple viewpoints, estimation of homographic transforms, or registration with accompanying depth images in order to convert these poses into 3D, metric coordinates. For this reason many robotics researchers have turned their attention to depth sensing cameras to directly estimate body pose in 3D

Whilst work on full body pose estimation using depth cameras [28, 29, 50, 51] has shown impressive results, the density of people in crowded environments and the frequency of occlusions makes reliably observing the whole body very difficult. This difficulty has

caused several authors [52, 53] to focus on the parts of the body that are most visible in crowded environments, namely the head and shoulders. Although the pose of the head and shoulders is less informative than a full skeletal pose, it still provides rich information about the social behaviour and intentions of people. For instance: whilst walking, people align their shoulders with their direction of travel, and when interacting with other people, they usually align their shoulders with those people.

This chapter presents an algorithmic framework for detecting and extracting the shoulder pose of multiple people in a crowded environment from 3D point-clouds, in real-time. Shoulder pose is defined here as a 3D position (x, y, z), and an angular orientation about the vertical axis θ_z. Pose extraction is achieved using a novel approach based on efficiently fitting ellipsoids to clusters of 3D points.

Following this introduction, Section 3.2 will give an overview of the pose detection framework and flow of data between each of the modules. Section 3.3 describes how the 3D point-cloud of the entire scene is segmented into *proposal clusters*. Section 3.4 provides a detailed explanation of the pose extraction algorithm applied to each cluster. Section 3.5 describes the collection of a set of depth image sequences with ground truth provided by an optical motion capture system and Section 3.6 presents an empirical evaluation of the pose detection framework on this data set. Finally, Section 3.7 summarises the contributions of the chapter and how they relate to the aims of this thesis.

3.2 The Person Detection Framework

The person detection framework takes as input a stream of upright, 3D point-clouds and outputs a collection of 4 DOF shoulder poses (x, y, z, θ_z) per frame of data. A 3D point-cloud is a collection of points in 3D space, and the concept of an *upright* point-cloud here refers to the alignment of a point-cloud such that the z-axis is perpendicular to the floor. 3D point-clouds can be obtained from various sources including LIDAR, stereoscopic vision, time-of-flight cameras and structured light depth cameras. The framework presented here has been designed for, and evaluated with, a structured light depth sensor but is applicable to any point-cloud with a density of approximately 6×10^3 points/m^2 on target

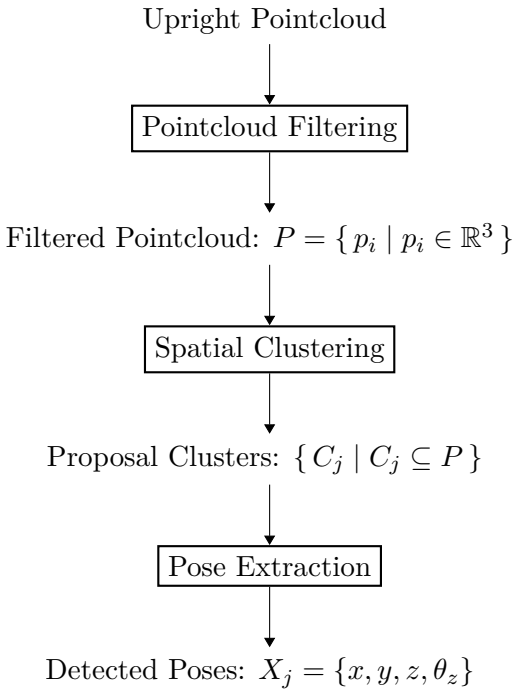


FIGURE 3.1: Overview of the Person Detection Framework

surfaces. Depth images from a depth sensing camera can be converted to 3D point-clouds given a model of the optics of the camera. Alignment with gravity can be achieved either by directly measuring the direction of gravity relative to the sensor using an inertial measurement unit (IMU) or by algorithmically determining the required transformation from features in the data. In the experiments described in Section 3.6, alignment was achieved by fitting a plane to data representing the floor and projecting the point-cloud such that the z-axis is perpendicular to this plane. This process is described in greater detail, in the context of a field study at a busy public train station, in Chapter 5.

The person detection framework is organised into several modules, as depicted in Figure 3.1, which sequentially process each frame of data. The first of these is the *point-cloud filtering* module which uses a voxel grid filter to downsample the point-cloud. Voxel grid downsampling reduces the number of points for subsequent modules to process which reduces the computational load and time taken to process each frame. It also imposes uniform point density which ensures that the geometric surface fitting used in the *pose extraction* module produces consistent results. Additionally the voxel grid filter can remove spurious data points by imposing a minimum number of points per voxel.

The *spatial clustering* module takes the filtered point-cloud and segments it into a collection of *proposal clusters* potentially representing people in the scene. Section 3.3 describes the clustering algorithm in detail. These *proposal clusters* are then processed by the *pose extraction* module which attempts to detect the head and shoulders of a person in the point cluster and if successful uses this model to determine the 4 DOF shoulder pose of the person. The algorithm used by the *pose extraction* module is described in greater detail in Section 3.4. The *detected shoulder poses* can be used in a variety of ways by subsequent perception algorithms. A useful next step which is common in robotics, is to use a tracking algorithm to establish persistent tracks for each observed individual, and indeed this is the topic of Chapter 4.

3.3 Spatial Clustering

In this module each *filtered point-cloud* is segmented into human sized clusters based on the following assumptions:

1. People are standing upright
2. The tallest point on a persons body is their head.
3. Peoples heads are spatially separated from one another.

The segmentation algorithm sorts the point-cloud in descending height order, then iterates through each point p_i, comparing the horizontal distance d_{ij} between each point p_i and each cluster C_j to a fixed separation distance threshold d_0. If $d_{ij} \leq d_0$ the point is added to the nearest cluster and the mean of the cluster is updated, otherwise a new cluster is created containing only p_i.

After clustering there are often cases where a person is split into multiple clusters due to data points at a person's horizontal extremities, such as their shoulders for which $d_{ij} > d_0$. To deal with this occurrence a final cluster joining step checks the distance between cluster means and joins those with a distance less than d_0. Cluster merging is performed in rounds. In each round as many merges as possible are performed provided each cluster

is only merged with one other cluster, with the nearest merges taking precedence over those further away. After each round in which any merges are executed, cluster means are reevaluated and another round of merges takes place if necessary. Finally any clusters with a small number of points or representing a small surface area are removed, this step is performed after merging to give small clusters a chance to be joined with a nearby larger cluster which often avoids discarding the extremities of detected people.

3.4 Pose Extraction

The task of the *pose extraction* module is to determine if each proposal cluster contains a person, and if so, to extract the 4 DOF pose of the person. In fact the algorithm works in the reverse order, by first attempting to fit a model of the visible surface of the head and shoulders to point-cloud data of the upper body, and second, judging whether the model parameters reflect those expected of a head and shoulders. The surface model selected for this purpose should:

1. be capable of representing the shape of the human head and shoulders,
2. allow extraction of a stable shoulder position and orientation,
3. be flexible enough to encompass the variety of shapes and sizes within the population,
4. be robust to relative motion between the head and shoulders, and
5. be computationally efficient to fit

With these requirements in mind a pair of ellipsoids was selected as a suitable surface model: one fitted to the head, and one fitted to the shoulders, as shown in Figure 3.2. While the complexity of the head and shoulder surface is not fully captured by the two ellipsoid model it meets the above requirments, providing a good compromise between fitting the data closely and simplicity of the model. Section 3.4.1 below describes the overall pose extraction algorithm and Section 3.4.2 describes the method used for fitting ellipsoids to 3D point-cloud data.

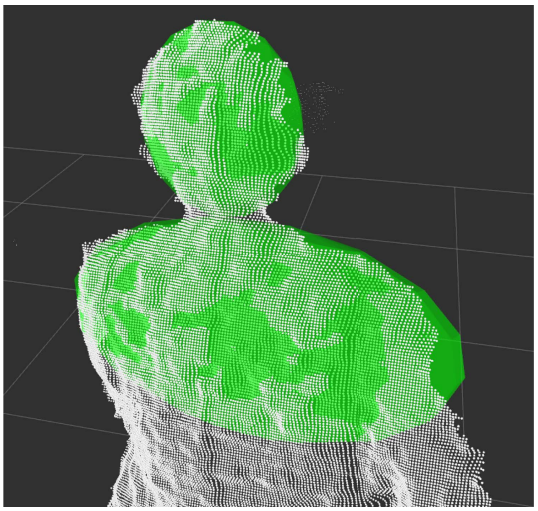


FIGURE 3.2: Head and shoulder ellipsoids (green) fitted to a point-cloud of a person (white).

3.4.1 Detecting the Head and Shoulders

In order to fit ellipsoids specifically to the head and shoulders, candidate points must be selected from the proposal clusters which are likely to represent these parts of the body. This task requires making some assumptions about the size and shape of the head and shoulders of a person, and is made challenging by the wide range of sizes and shapes within the population. To guide these assumptions we have used statistical data taken from a 2012 Anthropometric Survey Of U.S. Army Personnel [54] to set physical selection criteria where needed. The surveyed personnel consisted of men and women from a broad range of occupations, not only those on the front line. We also fit the ellipsoids sequentially to leverage parameters of the head ellipsoid in selecting candidate points for the shoulder fit, hence adapting our physical criteria to the individual and reducing the sensitivity of the method to the chosen parameters.

First the head ellipsoid is fitted to a vertical window of points with fixed height extending downward from the top of the point-cloud. A window size of 21cm is used based on the 10th percentile measurement from the top of head to the cervicale [54] to capture most of the points on the head while minimising the chance of including the neck or shoulders. The shoulder ellipsoid is similarly fitted to a fixed vertical window of points, extending 21cm downward (90th percentile neck to scye length [54]) from the centre of the head

ellipsoid. However, to ensure that the shoulder ellipsoid fits the breadth of the shoulders rather than the neck area, a dilated version of the head ellipsoid is used to remove the neck and collar region from the points to be fit. This ensures that the fit is dominated by the shoulder tips, improving the quality of orientation estimates obtained.

Algorithm 1: Pose extraction algorithm

Input: Point Cluster $C_j = \{\, c_k \mid c_k = \{c_k^x, c_k^y, c_k^z\} \,\}$ in descending height order
Output: Pose $X_j = \{\, x, y, z, \theta_z \,\}$
Data: Head ellipsoid $E_{head} = \{e, r\}$
Parameters: $h_{head}, h_{shoulder}$
$z_{max} \leftarrow$ `MaxHeight(`C_j`)` ;
$z_{min} \leftarrow z_{max} - h_{head}$;
$P_{head} \leftarrow \{c_k \in C_j \mid z_{min} < c_{kz}\}$;
$E_{head} \leftarrow$ `FitEllipsoidToPoints(`P_{head}`)` ;
valid $\leftarrow$ `CheckHeadValidity(`E_{head}`)` ;
if valid **then**
  $\{x, y\} \leftarrow$ `GetEllipsoidCentreXY(`E_{head}`)` ;
else
  return *FAIL* ;
end
$z_{max} \leftarrow$ `GetEllipsoidCentreHeight(`E_{head}`)` ;
$P_{shoulder} \leftarrow \{c_k \in C_j \mid c_{kz} \leq z_{max}\}$;
$E_{collar} \leftarrow$ `IncreaseEllipsoidRadii(`E_{head} , r`)` ;
$P_{collar} \leftarrow$ `PointsContainedByEllipsoid(`$P_{shoulder}$, E_{large}`)` ;
$P_{shoulder} \leftarrow P_{shoulder} - P_{collar}$;
$z_{max} \leftarrow$ `MaxHeight(`$P_{shoulder}$`)` ;
$z_{min} \leftarrow z_{max} - h_{shoulder}$;
$P_{shoulder} \leftarrow \{c_k \in C_j \mid z_{min} < c_{kz}\}$;
$E_{shoulder} \leftarrow$ `FitEllipsoidToPoints(`$P_{shoulder}$`)` ;
valid $\leftarrow$ `CheckHeadAndShoulderValidity(`E_{head}, $E_{shoulder}$`)` ;
if valid **then**
  $z \leftarrow$ `EllipsoidTop(`$E_{shoulder}$`)` ;
  $\theta_z \leftarrow$ `MajorAxisAngle(`$E_{shoulder}$`)` + 90°;
  return $\{x, y, z, \theta_z\}$;
else
  return $\{x, y\}$;
end

Once the head and shoulder ellipsoids have been fitted they are used to extract a shoulder pose consisting of a 3D position and angle of orientation about the vertical axis. The horizontal components of the pose are taken directly from the centre of the shoulder ellipsoid as this position is more stable than that of the head. However the vertical component of the shoulder ellipsoid is less stable due to its high dependence on the vertical

window used to select points for the fit. For this reason the vertical component of the pose is based on the top surface of the shoulder ellipsoid as it is more indicative of the true height of the persons shoulders. It is calculated by the intersection between a vertical line passing through the ellipsoid centre and the upper surface of the ellipsoid.

Finally the orientation of the shoulders is obtained by projecting the major axis of the shoulder ellipsoid into the horizontal plane and taking the angle of the resulting line. This angle is offset by 90° to obtain the facing direction of the person rather than the line of their shoulders, however the forwards direction is ambiguous based on the axis of the shoulders alone. To resolve this ambiguity we make the assumption that the head is forward of the shoulders. The horizontal location of the head ellipsoid centre relative to the shoulder ellipsoid major axis is used to determine the forwards facing direction and set the orientation angle accordingly.

3.4.2 Ellipsoid Fitting

In the crowded scenarios targeted by this work, the number of people in the field-of-view (FOV) of the sensor at any time can be upwards of 20. With 2 ellipsoids to be fitted per person and 30 frames of depth data per second this could mean the fitting of as many as 1200 ellipsoids per second. In order to process all data in real-time it was therefore a priority to use an efficient method for ellipsoid fitting.

The ellipse fitting method used, proposed by Li et al. [55], finds the least squares fit of a quadric surface of the form

$$ax^2 + by^2 + cz^2 + 2fyz + 2gxz + 2hxy + 2px + 2qy + 2rz + d = 0$$

to a set of 3D points subject to the constraint $4J - I2 > 0$

where:

$$I = a + b + c\,,$$

$$J = ab + bc + ac - f^2 - g^2 - h^2\,.$$

TABLE 3.1: Depth image sequences were captured representing a range of scenarios with accurate ground truth provided by an optical motion capture syste

Dataset	No. ppl	Duration (seconds)	Descripion
Wandering 1	3	120	
Wandering 2	4	123	Participants casually moving and stopping within the
Wandering 3	8	47	FOV of the depth camera
Wandering 4	8	98	
Alighting 1	6	24	Participants simulating situations where 4 train
Alighting 2	6	21	passengers wait to board a service while 2 passengers
Alighting 3	6	16	alight
Walkthrough	8	142	4 participants stand still while 4 others repeatedly cross the FOV weaving between stationary participants
Passing	8	131	All participants repeatedly crossing the FOV weaving past one another (pictured in Figure 3.3)

Li et al. [55] show that this constraint is sufficient to guarantee that the quadratic surface fit is an ellipsoid, and the problem can be efficiently solved by formulating it as an eigensystem.

3.5 Data collection

In order to quantify the precision and accuracy of our approach, a dataset was captured consisting of 9 depth image sequences of people moving in different ways through the depth sensor FOV, with accompanying ground truth measured using an optical motion capture system. The dataset was captured in the UTS Data Arena, a circular cinema room, with an Optitrack motion capture system comprising of high frame rate cameras with infrared illumination. Each participant had a rigid infra-red marker card attached to their back using a velcro strap (pictured in Figure 3.3), used to accurately track the position and rotation of their upper body. A brief description of the different depth sequences is provided below.

FIGURE 3.3: *Left:* The sensor platform was mounted on a fixed pole aimed at the centre of the room and the movement of all participants was tracked using optical motion capture with infrared marker cards. *Right*: A sample depth image taken from the *Passing* sequence.

3.6 Experimental Results

In order to evaluate the shoulder pose estimation algorithm presented, depth image sequences from the lab dataset were processed using our framework and the results of pose extraction were compared with the pose ground truth obtained from the motion capture system. Table 3.2 summarises the results of this comparison in terms of precision, horizontally, vertically and in orientation angle.

To account for the unknown offset between infra-red markers attached to participants and the centre-of-shoulder position extracted by our algorithm, a single 3D offset has been applied to the ground truth data in the local frame of each marker card based on the mean 3D position error. The results presented here therefore do not capture any positional bias in the extracted poses but do capture the consistency of the extracted poses which is the focus of this analysis. The starting orientation of the marker cards is also arbitrary and a similar offset has been applied to each card orientation prior to error computation. Orientation errors are wrapped between $\pm\frac{\pi}{2}$ to better characterise errors by separating orientation inaccuracies from errors due to the ambiguity between the forwards and backwards direction. For clarity the percentage of extracted poses which correctly estimated the forwards direction (and hence did not require wrapping) are also given.

TABLE 3.2: Mean Absolute Errors of Pose Extraction Against Ground Truth From an Optical Tracking System

Dataset	Horizontal (cm)	Vertical (cm)	Orientation (°)	Forwards (%)
Wandering 1	8.37	3.34	13.99	70.59
Wandering 2	7.36	3.10	10.40	72.20
Wandering 3	9.40	3.63	14.29	84.74
Wandering 4	9.02	3.44	12.37	82.12
Alighting 1	7.82	5.37	10.00	77.59
Alighting 2	8.97	3.95	9.10	71.37
Alighting 3	16.80	5.93	18.55	74.74
Walkthrough	8.37	3.99	12.03	77.88
Passing	12.93	5.67	14.85	80.26

3.7 Summary

To address the challenges of detecting and extracting the pose of people in crowded environments, a novel method has been developed for pedestrian pose extraction in 3D point-clouds. The method leverages the high observability of the head and shoulder region in crowds and the spatial separation typically maintained between individuals' heads, even when other parts of their body may be in contact, to successfully segment them. The ellipsoid based model used to fit the head and shoulders allows rejection of false positives and accurate estimation of 3D position (x, y, z) and angular orientation θ_z and can be efficiently implemented to run at typical sensor frame-rates. An evaluation has been conducted on a dataset with accurate ground truth demonstrating the precision of the pose extraction technique.

A limitation of this approach is poorer orientation estimation at long range (>4m). This is due to two main factors: insufficient point-cloud density at longer ranges due to the limited angular resolution of the sensor; and artefacts of the depth estimation process which present as coarse steps in measured depth values, causing bias in orientation estimates. However, these are largely limitations of depth sensing cameras rather than the algorithm and could be addressed by deploying multiple sensors to obtain a more complete point-cloud. Another limitation is the need for manually selected thresholds on the head and shoulder ellipsoid parameters for rejection of false positives. Such manual thresholds are

sometimes ineffective in the presence of broad hats and large backpacks causing higher false negative rates.

A worthwhile direction for future work would be to empirically characterise errors in pose extraction and their correlation with observable factors such as range and observation angle, with a view to correct bias in pose estimates and provide a measure for uncertainty in each of the estimated dimensions. Additional future work could investigate the use of learning based classification of ellipsoid parameters to replace manually set thresholds in regard to rejection of false positives. Beyond this binary classification task one could investigate the potential for the ellipsoid parameters to be used as descriptive features for re-identification across multiple sensors.

Accurate fast person detection and pose estimation is a critical step towards perception of pedestrians and will form the foundation of further perception algorithms presented in this thesis. Chapter 4 describes an approach to tracking people in crowded scenarios which makes use of such pose estimation results, in particular leveraging the shoulder orientation to inform its motion model and improve frame-to-frame predictions. Beyond such improvements to tracking, these pose estimates can be used to infer the attention and intentions of people, for instance in indicating their participation in group interactions or signifying interaction with features of the environment such as signage, regions of shelf space in a retail context, or ticket machines in a public transit context.

Chapter 4

Pedestrian Tracking with Social Constraints

4.1 Introduction

In order to create robots and autonomous systems which can respond intelligently to humans it is critical that these robots are able, not only to detect people, but to observe sequences of movement behaviour. Consider the case of a mobile robot which detects that there is a person three metres in front of it. With no other contextual information it is hard to determine what response this robot should have. At best you might say that this represents a safety hazard and the robot should stop moving, but a robot that stops moving any time there is a person three metres in front of it will be of little use in an environment populated by people. Now consider the same scenario except that the robot now has knowledge of the sequence of poses leading up to the current moment. Suddenly the robot is empowered to make much better decisions. If the person has been walking away from the robot at a steady pace it may be quite safe to move forwards at a similar pace, however if the person is walking towards the robot then it may indeed be best to stop or actively avoid the person. This is only a simple example but pose trajectories can additionally enable much more sophisticated perception, such as intention inference, or modelling of social interactions and grouping. Consider a busy urban train platform, a

person who has been standing waiting on the platform for some time is likely to board the next train, whereas a person who has just stepped off a train is likely to exit the platform.

Constructing sequences of poses from a stream of unassociated pose observations, such as those produced by the work in Chapter 3, is referred to as *tracking-by-detection*. This type of tracking algorithm maintains a set of persistent tracks and given each new frame of observations attempts to associate each observation with one of these tracks. Tracking is a mature topic in both the computer vision and robotics research communities and relevant related work has been discussed in 2.2. A common approach to this problem in the robotics literature is to maintain a probabilistic estimate of the state of each person using a framework called Recursive Bayesian Estimation and to associate observations with each track based on whether they are consistent with the estimated distribution. The work presented in this chapter uses a variant of Recursive Bayesian Estimation called a *particle filter* to maintain a probabilistic estimate of the 3 DOF pose of each person. A particle filter based approach has been chosen here, as opposed to a Kalman filter, due to its ability to model the non-Gaussian distributions resulting from the inclusion of social constraints.

One of the major benefits of this type of probabilistic tracking framework is its ability to deal with occlusions and noisy observations. If observations of all individuals in the scene were available with high accuracy at a high frame-rate the task of tracking them would be rather trivial, but this is rarely the case. In most real world scenarios, poses can only be approximated within some a degree of error, or uncertainty. Furthermore it is common in a robotics setting, where the robot observes the scene from a single point-of-view, that human targets will at times be occluded, either by objects in the environment or by other people. Trackers based on Recursive Bayesian Estimation are able to overcome these problems by estimating the state of each track, every frame, before attempting to associate observations with them. The ability of a tracker to maintain accurate tracks, therefore depends upon its ability to predict the movement of people.

Crowded social environments present many challenges to tracking systems. In these environments occlusions become very common and may be quite prolonged, partial occlusions

lead to inaccurate observations, and the high number of targets and close proximity between them makes the task of data association more error prone. These compounding factors lead to the failure of many tracking algorithms in such circumstances, and yet humans are quite capable of tracking the movement of others through a crowd. A potential explanation for this lies in the ability of humans to interpret social cues and context and leverage them to make better predictions about the movements of others. One such social cue is the orientation of a persons shoulders. People typically align their shoulders with their intended direction of travel while walking, rendering shoulder orientation a valuable piece of information in predicting human motion. Another useful social insight is the tendency of people to maintain a social distance between themselves and others. In a crowded situation this limits the set of likely trajectories to those that do not violate these social constraints.

This chapter presents a tracking approach which builds on the detection method discussed in Chapter 3 and leverages social insights to achieve robust pedestrian tracking in crowded environments. Following this introduction, Section 4.2 describes the tracking algorithm as a whole and the role each step of the tracking loop. This overview is followed in Section 4.3 by more detailed discussion of the novel aspects of the tracking approach beginning with a method of track prediction which enforces interpersonal distance constraints to improve prediction in crowds. This is followed in Section 4.4 with a description of the data association and track management steps. Next, Section 4.5 describes the observation update step which incorporates shoulder pose observations into the track state estimate with variable confidence in the observed orientation. Section 4.6 introduces a novel shoulder alignment *pseudo-observation* step which updates track state estimates to favour alignment between shoulders and movement direction of people. An empirical evaluation of the tracking method is provided in Section 4.7 which compares the work with another state-of-the-art person tracking algorithm. Finally Section 4.8 summarises the contributions of this chapter relating them back to the aims of the thesis.

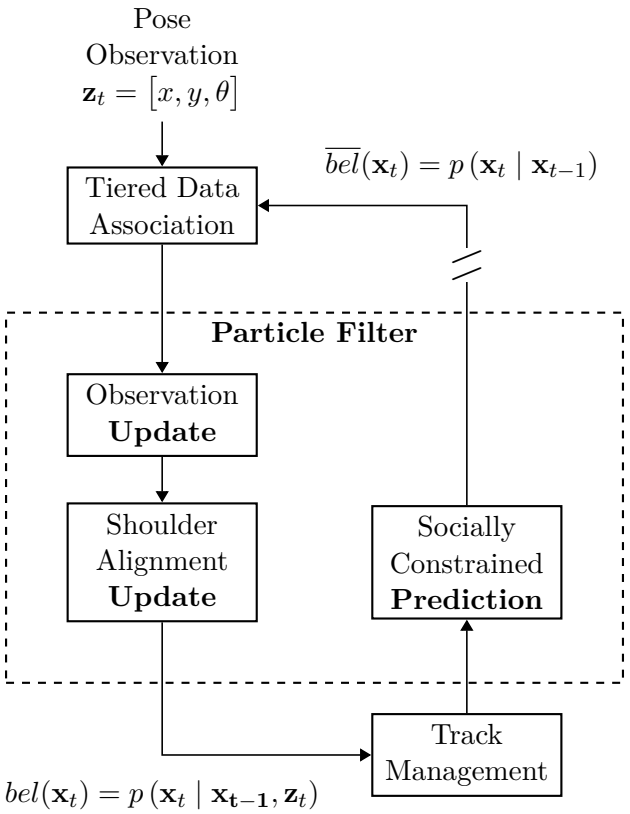


FIGURE 4.1: A diagram of the main steps of the tracking loop. At each iteration t, Each observation $\mathbf{z}_t$ is associated with an existing track if possible. Associated observations and alignment of shoulders to walking direction are used to update the state belief of each track $bel(\mathbf{x}_t)$. Tracks may be disregarded in the *Track Management* step if their state belief becomes too uncertain. Finally the next state belief of each track is predicted based on a socially constrained motion model and the loop starts again.

4.2 A Framework for Pedestrian Tracking

The tracking algorithm presented in this chapter takes as input a stream of 3 DOF pedestrian pose observations $\mathbf{z}_t$ each comprised of a 2D position (x, y) and an orientation about the vertical axis θ. For each individual the algorithm establishes a track, and maintains a probabilistic belief $bel(\mathbf{x_t})$ over possible values of the pose at each time step t using a particle filter.

Each filter has a set of particle states $\mathcal{X}_t = \left\{\mathbf{x}_t^{[1]}, \cdots, \mathbf{x}_t^{[M]}\right\}$ and a corresponding set of weights $\mathcal{W}_t = \left\{w_t^{[1]} \cdots, w_t^{[M]}\right\}$ which together represent the belief distribution over possible states of a person at time t. Each particle $\mathbf{x}_t^{[m]} = \left[x, y, \dot{x}, \dot{y}, \theta\right]^\top$ represents a possible state in terms of position x, y, velocity $\dot{x}, \dot{y}$ and orientation about the z-axis θ.

The number of particles per filter M is selected as a trade-off between computational cost and better expression of the underlying distribution ($M_0 = 500$ in our experiments). Although the pose extraction method described in Chapter 3 is capable of extracting the 3D position of the shoulders, the tracker only operates in the the horizontal plane, ignoring height, as this is sufficient for the pedestrian perception applications targeted by this thesis. The tracker works in an iterative fashion and is made up of several steps which form a tracking loop. Figure 4.1 shows an overview of the tracking loop and the steps involved.

At each iteration, triggered by a new frame of sensor data at 30Hz in our experiments, the tracker performs the following steps:

1. **Socially Constrained Prediction** – The state belief $\overline{bel}(\mathbf{x}_a)$ of each track is predicted based on its previous state, the assumed motion model, and social distance constraints. See Section 4.3 for details.

2. **Tiered Data Association** – Pose observations are associated to tracks based on their consistency with the predicted state distributions with priority given to *confirmed* tracks. See Section 4.4 for details.

3. **Observation Update** – Particle weights of observed tracks are updated based on their likelihood given the associated observations, taking into account a measure of confidence in the orientation estimate. See Section 4.5 for details.

4. **Shoulder Alignment Update** – Particle weights of all tracks are updated based on the alignment between shoulder orientation and velocity direction, gated by the velocity magnitude. See Section 4.6 for details.

5. **Track Management** – *Tentative* tracks are created for unassociated observations, and the status of existing *tentative* tracks may be upgraded to *confirmed* if they have been consistently observable. Tracks may also be declared *inactive* based on high uncertainty in the state belief or low observability. See Section 4.4 for details.

6. **Particle Resampling** – The particles of each filter are resampled to represent the weighted particle distributions as equivalent uniformly weighted particle distributions. The resampling method used is systematic resampling. This process ensures

that the density of particles continues to reflect the target distribution and is a standard step in a particle filter.

4.3 Socially Constrained Prediction

Prediction is a critical step in Recursive Bayesian Estimation as it incorporates knowledge of the state transition process $p(x_t \mid u_t, x_{t-1})$ into the belief distribution $\overline{bel}(x_t)$. Furthermore in particle filters this step serves to perturb the particle states with noise, driving the algorithm to explore the state space. In the context of a pedestrian tracking algorithm the state transition process does not consider control inputs u_t, as is common in robot localisation problems. This is because the intended control actions of the people being tracked are unknown. Rather a motion model is used which computes new states as a function of the previous states only $p(x_t \mid x_{t-1})$.

While such a model will do a reasonable job of predicting the motion of a lone individual, it may fail in crowded situations as pedestrians alter their paths to accommodate one another and maintain a social distance buffer between themselves and others. This work seeks to leverage this social insight in the prediction step by limiting the set of allowable predictions to only those which adhere to interpersonal distance constraints. Section 4.3.1 below describes the motion model used in this work to predict pedestrian movement. It is followed by Section 4.3.2 which describes a novel method for enforcing interpersonal distance constraints in the prediction step.

4.3.1 Pedestrian Motion Model

At each time step t, the position x, y and velocity $\dot{x}, \dot{y}$ of each particle m are propagated according to a *continuous white noise acceleration model* [56]. Additionally the shoulder orientation θ is propagated independently of the rest of the state based on a *continuous white noise angular velocity model*. The shoulder orientation is assumed to be independent of the position and velocity in the motion model based on the fact that pedestrian movement is holonomic, that is pedestrians can walk sideways and even backwards. Despite this fact it is clear that people show a strong preference for walking forwards, that is, in the

direction their upper body is facing. While this preference is certainly a valuable insight into human motion it does not belong in the kinematic model but rather is imposed in the *Shoulder Alignment Update* step described in detail in Section 4.6.

The updated state $\mathbf{x}_t^{[m]}$ of each particle m at each time step t is computed as follows.

$$\mathbf{x}_t^{[m]} := \mathbf{F}\mathbf{x}_{t-1}^{[m]} + \boldsymbol{\nu}_t \qquad \boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

The new state is the sum of two terms. The first represents the deterministic component of the motion, characterised by the transition matrix $\mathbf{F}$. The second term $\boldsymbol{\nu}_t$ adds correlated Gaussian noise to the state vector characterised by covariance matrix $\mathbf{Q}$. These matrices are as follows, where T is the time elapsed in seconds since the previous state estimate.

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \frac{a}{3}T^3 & 0 & \frac{a}{2}T^2 & 0 & 0 \\ 0 & \frac{a}{3}T^3 & 0 & \frac{a}{2}T^2 & 0 \\ \frac{a}{2}T^2 & 0 & aT & 0 & 0 \\ 0 & \frac{a}{2}T^2 & 0 & aT & 0 \\ 0 & 0 & 0 & 0 & \omega T \end{bmatrix}.$$

The model has two design parameters a and ω which are chosen empirically. The parameter a scales the variance of the *continuous white noise acceleration model* while ω scales the variance of the *continuous white noise angular velocity model*. Note that the choice of a motion model driven by a *continuous-time white noise process* rather than *piece-wise constant white sequence* allows for the sampling period T to change without severely affecting the process variance [56].

4.3.2 Interpersonal Distance Constraint

Crowded environments present two major challenges to the tracking algorithm. The first is that pedestrians alter their walking motion in the presence of others to avoid collisions and maintain comfortable interpersonal distances. This causes pedestrian motion behaviour that violates the assumptions of the motion model and leads to increased error between the predicted and true pedestrian states. The second challenge is that the high number

of people and their close proximity to one another greatly increase the difficulty of the data association problem leading to false associations between observations and filters. This problem typically occurs when the separation between two or more targets is small compared to the observation error and is compounded by the aforementioned increase in prediction error. To improve the robustness of the tracking algorithm in crowded scenes, social constraints are introduced based on the study of proxemics [57] which describes people's inclination to maintain comfortable interpersonal distances from one another, even in crowded situations. Algorithm 2 describes the method used to impose these social constraints.

Algorithm 2: Socially Constrained Prediction

Input: T, $\{\mathcal{X}_t^{[1]} \cdots, \mathcal{X}_t^{[N]}\}$, $\{c^{[1]} \cdots, c^{[N]}\}$
Output: $\{\mathcal{X}_t^{[1]} \cdots, \mathcal{X}_t^{[N]}\}$
Parameters: r
Data: $\{\bar{\mathbf{x}}_t^{[j]}, \cdots, \bar{\mathbf{x}}_t^{[N]}\}$

```
1  for i ∈ {1, ··· , N} do
2      for x_t^[m] ∈ X_t^[i] do
3          x_t^[m] ← PredictMotion(x_t^[m], T);
4      end
5      x̄_t^[i] ← ComputeMean(X_t^[i]);
6  end
7  for i ∈ {1, ··· , N} do
8      for x_t^[m] ∈ X_t^[i] do
9          for j ∈ {1, ··· , N} do
10             if i ≠ j and c^[j] = true then
11                 if |x̄_t^[j] − x_t^[m]| ≤ r then
12                     X_t^[i] ← X_t^[i] − {x_t^[m]};
13                     break;
14                 end
15             end
16         end
           // breaks to here
17     end
18 end
```

The inputs to the algorithm are the time elapsed since the previous update T, the particle states of each filter $\left\{\mathcal{X}_t^{[i]}\right\}_{i=1}^{N}$, and the boolean values $\left\{c^{[i]}\right\}_{i=1}^{N}$ indicating whether each track is *confirmed* (true) or *tentative* (false). In lines 1-4 each particle state is propagated

without social constraints according to the motion model described in Section 4.3.1. Once all particles of a filter have been propagated the mean of the new particle states is computed (on line 5) for the purpose of checking interpersonal distances. Note that in the general case the state mean $\bar{\mathbf{x}}_t^{[i]}$ should be a computed as a weighted mean considering the particle weights $\mathcal{W}_t^{[i]}$. However given that the prediction step occurs immediately after the resampling step, the weights $\mathcal{W}_t^{[i]}$ will be uniform and hence the unweighted mean is equivalent. This initial prediction, although not socially constrained provides a reasonable estimate of the updated position of each person for the sake of applying interpersonal distance constraints. Errors in this initial unconstrained prediction depend on the time between updates which is suitably short in our experiments at approximately 33ms.

Having computed estimates for the predicted state of each filter *without* social constraints, the second part of the algorithm, described in lines 7-18, discards particles which violate the interpersonal distance constraint. The distance between each particle state $\mathbf{x}_t^{[m]}$ and the mean of every *other* filter $\bar{\mathbf{x}}_t^{[j]}$ is computed on line 11. Line 10 checks that particles are not compared with their own filter's estimate and, importantly, that they are only measured against *confirmed* tracks. This ensures that the creation of *tentative* tracks for unassociated observations does not disrupt the prediction step. The track confirmation process and notion of *tentative* vs *confirmed* tracks is explained in more detail in Section 4.4. If the distance between a particle and a track estimate is less than or equal to the interpersonal radius parameter r the particle is discarded from the filter (line 12) and the algorithm immediately moves to the next particle. This is equivalent to applying a likelihood update in which the weight $w_t^{[m]}$ of particles that violate the interpersonal distance constraint is set to zero, however it avoids performing any further calculations on these particles in the current iteration of the tracker. While it is theoretically possible that predicted particles could pass through the interpersonal constraint region to the other side in a single update and not be penalised for it, in reality the effect of this is mitigated by the frequency (30Hz) of tracking updates.

Discarding of particles will cause the particle count M to drop below the nominal particle count M_0 reducing the expressiveness of the filter in the subsequent measurement updates. However the nominal particle count M_0 will be restored each iteration of the filter in the resampling step, preventing a progressive decay of M. In order to address the reduction in

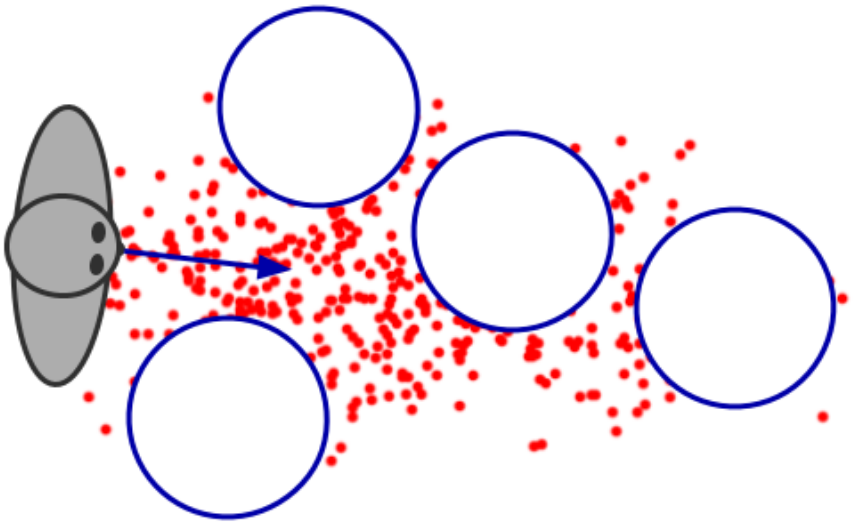


FIGURE 4.2: An illustration of socially constrained prediction. Red dots represent the particles of a filter tracking the grey person. The blue circles show the area around four other people in the scene within which particles are removed. Note that the spread of particles here is exaggerated for the sake of illustration and in reality depends on the update period which in our experiments is approximately 33ms.

filter expressiveness additional particles could be drawn based on the previous states and checked against the interpersonal distance constraint until $M = M_0$. This would however add complexity to the algorithm and in our experiments was not found to be necessary.

When people are in close proximity to one another the effect of the algorithm is to shape the state belief $\overline{bel}(\mathbf{x}_t)$ of each filter to consider only the spaces in-between people. This allows the tracking algorithm to maintain low uncertainty estimates of a persons state even when they have a low observation frequency as it leverages it's knowledge of nearby people to limit its predictions. Figure 4.2 illustrates the concept of socially constrained tracking.

4.4 Tiered Data Association

Following the socially constrained prediction step is the tiered data association step. This step attempts to match each observation with the appropriate track so that the state of the filters can be updated. Following this association process there may be some tracks

to which no observations were associated, which will be referred to as *unobserved trac ks*. Conversely there may be observations which were not associated with any track referred to as *unassociated observations*. Naively we might choose to initiate a new track for every such unassociated observation, however in a crowded environment new tracks created due to false positive observations are likely to "steal" observations from legitimate existing tracks in subsequent data association steps and adversely affect the prediction step with regard to the social constraints described in Section 4.3.2.

To avoid this issue a two tiered approach to data association was devised. When a new track is created it is considered *tentative* until has been observed reliably enough to become *confirmed*. In order to evaluate track observability each track is assigned an observation counter which is initialised to zero. In the data association step of the tracking loop the observation counter of observed tracks is incremented, while the observation counter of unobserved tracks is decremented.

In the track management step of the tracking loop, if the observation counter reaches the negatively-valued, track deletion threshold, the track is deleted for lack of observations. If the observation counter reaches the positively-valued, track validation threshold, the track is permanently promoted to *confirmed*. The effect of this observation counter is that, if a track is observed in more than 50% of frames it's counter will progress towards *confirmed* status. Conversely if the track is observed in fewer than 50% of frames it will progress towards deletion. With the track validation count of 15 used in our experiments a consistently observed track with be confirmed in 0.5 seconds, meanwhile the track deletion threshold of -25 used in our experiments will cause new tracks to be deleted after 0.83 seconds without observation.

Given these two tiers of track confirmation the data association is performed in two stages. First the set of confirmed tracks are each matched with the nearest observation that is statistically consistent with the filter's distribution over positions represented by its particles, with 95% confidence according to the Chi-squared test. This matching is done in a greedy fashion whereby the nearest statistically consistent pair is matched at each iteration and removed from further consideration until there are no consistent pairs remaining. This same process is then followed for the remaining unassociated observations and the set of

tentative tracks. In this way, confirmed tracks are given precedence over newly created tracks in the data association, and are therefore less likely to be adversely affected by false positive observations. Future work could replace the greedy track matching algorithm with a globally optimal assignment algorithm such as the Hungarian method [35].

4.5 Observation Update

Following the data association step all observed tracks are subject to the observation update step. The role of the observation update in recursive Bayesian estimation is to incorporate information from the observations $\mathbf{z}_t$ into the state belief $bel(\mathbf{x}_t)$. In a particle filter this is achieved by assigning an importance weight to each particle proportional to likelihood of the particle state $\mathbf{x}_t^{[m]}$ given the associated observation $\mathbf{z}_t^{[j]}$, or equivalently the probability of the observation $\mathbf{z}_t^{[j]}$ given the state $\mathbf{x}_t^{[m]}$. The weight is updated as follows where η is a normalising term such that $\sum_{m=1}^{M} w_t^{[m]} = 1$.

$$w_t^{[m]} = \eta p(\mathbf{z}_t \mid \mathbf{x}_t^{[m]})$$

The likelihood function is the product of two terms: one concerning the 2D position L_{xy} and the other concerning the orientation L_θ

$$p(\mathbf{z}_t \mid \mathbf{x}_t^{[m]}) = L_{xy} L_\theta \,.$$

The term L_{xy} is the probability of observing the Euclidean position error δ_{xy} between the particle state $\mathbf{x}_t^{[m]}$ and the observation $\mathbf{z}_t$ assuming a Gaussian sensor model with zero bias and variance σ_{xy}^2

$$L_{xy} = p(\delta_{xy} \mid 0, \sigma_{xy}^2) \qquad \delta_{xy} = \|\mathbf{z}_{xy} - \mathbf{x}_{xy}^{[m]}\|_2$$

The shoulder poses obtained by the algorithm presented in Chapter 3 have some ambiguity in their orientation, between forwards and backwards, as discussed in Section 3.4.1. To deal with this ambiguity, the term L_θ is a sum of two components: one relating to the angular error of the detected orientation δ_θ, and the other relating to the angular error of

the opposite orientation $\delta_{\theta+\pi}$

$$L_\theta = \beta p(\delta_\theta \mid 0, \gamma\sigma_\theta^2) + (1-\beta)p(\delta_{\theta+\pi} \mid 0, \gamma\sigma_\theta^2)\,.$$

Both terms are modelled as Gaussian distributions with zero mean and variance $\gamma\sigma_\theta^2$. The balance between components is controlled by the ambiguity ratio β which represents the proportion of pose observations expected to have the correct facing direction. Based on the empirical evaluation of the pose extraction algorithm in Section 3.6, the value $\beta = 0.7$ is used in the evaluation of the tracking algorithm.

Additionally the shape of the ellipsoid fit to the shoulders in the pose extraction algorithm (Section 3.4.2) gives an indication of the quality of the extracted orientation. If the ellipsoid fit is spherical, the extracted orientation is completely arbitrary and therefore uninformative, however if the ellipsoid is narrow it is likely to provide a more reliable orientation measurement. To reflect this a variable noise sensor model is used to calculate L_θ.

The orientation confidence measure γ is computed based on the eccentricity of the shoulder ellipsoid $\mathcal{E}$ and used to scale the variance σ_θ^2 of the orientation observation model. The eccentricity is defined as the ratio of the shortest radius of the ellipse over the longest and can therefore have values in the interval $(0, 1]$. The orientation confidence is given by the following formula.

$$\gamma = \max\left(\frac{1-\mathcal{E}_0}{1-\mathcal{E}}, 1\right)$$

Where $\mathcal{E}_0$ is a parameter representing a typical shoulder eccentricity. The effect of this is that for an ellipsoid which matches the typical eccentricity, $\gamma = 1$ and the variance of the measurement model is σ^2. However as the ellipsoid fit approaches a sphere the uncertainty in the orientation variance approaches infinity representing a highly uncertain orientation estimate. Note that in the implementation of this algorithm γ is capped to avoid numerical issues.

4.6 Shoulder Alignment Update

To enforce the social insight that people tend to align their shoulders with their walking direction, a shoulder alignment pseudo-observation update is applied on each iteration of the tracker. This update is similar to the observation update described in Section 4.5 in that the weights of particles are updated based on a likelihood function, however it differs in that it is not based on any actual observation and is applied to all filters regardless of whether they have any associated observations. As the particle weights may have already been updated in the observation update the new weights are computed as product of their current weight and the likelihood function $L_{\theta v}$, and normalised by η such that the weights of each filter sum to one.

$$w_t^{[m]} \leftarrow \eta w_t^{[m]} L_{\theta v} \qquad L_{\theta v} = p(\delta_{\theta v} \mid 0, \sigma_{\theta v}^2)\,,$$

$$\delta_{\theta v} = \begin{cases} \theta - \operatorname{atan2}(\dot{y}, \dot{x}) & \text{if } v > v_0 \\ 0 & \text{otherwise} \end{cases},$$

Where v is the magnitude of the velocity in the particle state $\mathbf{x}_t^{[m]}$ and v_0 is velocity threshold above which velocity direction and shoulder orientation are to be considered correlated. For particles where $v > v_0$ this has the effect of assigning a lower weight when their velocity direction is not aligned with their shoulder orientation.

This pseudo-observation has the effect of correlating shoulder orientation and walking direction in the particle distribution of each filter. This allows the walking direction of the person to refine the estimated orientation of each person when they are moving with sufficient velocity, particularly with regard to the ambiguity of shoulder orientation estimates between forwards and backwards facing directions. Additionally when people transition from stationary to moving, which is often a challenge for tracking systems with a single motion model, the estimated orientation allows the tracker to better predict the direction the person will move in.

TABLE 4.1: Comparison between Socially Constrained Tracker and [2] on the CLEAR-MOT [58] performance metrics

		MOTP (mm)		MOTA (%)	
Sequence	No. ppl	Socially Constrained	Linder et al. [2]	Socially Constrained	Linder et al. [2]
Wandering 1	3	**705.1**	719.3	**98.90**	97.77
Wandering 2	4	**691.5**	709.0	**98.77**	97.39
Wandering 3	8	**669.9**	680.4	92.60	**94.24**
Wandering 4	8	**690.7**	703.8	90.67	**91.86**
Alighting 1	6	**684.7**	686.3	**59.04**	52.39
Alighting 2	6	**688.3**	707.6	**56.37**	46.65
Alighting 3	6	633.9	**619.6**	**53.94**	46.74
Walkthrough	8	657.8	**643.2**	**63.63**	57.43
Passing	8	682.4	**642.9**	**35.72**	26.45

4.7 Experimental Results

This section presents an empirical evaluation of the tracking algorithm described throughout this chapter and benchmarks it against the tracking algorithm proposed by Linder et al. [2]. The evaluation is conducted using the same dataset as the person detection algorithm in Chapter 3. This dataset consists of 10 depth image sequences with varying lengths, person counts and complexity. Each depth image sequence is accompanied by precise ground truth captured using a commercial motion capture system. For more information on the dataset see Section 3.5.

In order to make a fair comparison between the two tracking algorithms the person detection algorithm described in Chapter 3 was used to extract a sequence of pose observations for each depth image sequence. The resulting pose observation sequences were then processed independently by both the author's tracking algorithm and the state of the art algorithm proposed by Linder et al. [2], for which source code was available. The results of the comparison are presented in Table 4.1 in terms of the CLEAR-MOT metrics [58].

The CLEAR-MOT metrics [58] were devised to enable intuitive and fair benchmarking of multiple object tracking systems, providing a systematic approach to compute two complimentary performance measures. multiple object tracking precision (MOTP) is a measure of how close positions estimated by the tracker are to the truth and is computed in terms of the average position error in physical units, where a lower number indicates

TABLE 4.2: Breakdown of association errors produced by the Socially Constrained Tracking Algorithm

Sequence	No. Ppl	MOTA (%)	FPR (%)	FNR (%)	MME
Wandering 1	3	70.51	0.27	0.79	4
Wandering 2	4	69.15	0.46	0.74	4
Wandering 3	8	66.99	1.32	6.02	7
Wandering 4	8	69.07	3.43	5.89	4
Alighting 1	6	68.47	0.93	40.03	0
Alighting 2	6	68.83	4.41	39.16	2
Alighting 3	6	63.39	5.09	40.97	0
Walkthrough	8	65.78	4.42	31.84	41
Passing	8	68.24	4.74	59.26	89

better performance.

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}$$

Multiple object tracking accuracy (MOTA) is a measure of how correct the data association decisions of the tracker are, and is based on the sum of three distinct types of association error:

- Misses m_t aka. false negatives: Ground truth object occurrences for which there was no associated track frame
- False positives fp_t: Track frames for which there was no associated ground truth occurrence, and
- Miss-match errors mme_t: Tracking frames where the track identify was erroneously switched

MOTA is given as the percentage of associations that are correct, where 100% is the best possible score and is computed as follows.

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

Table 4.2 provides a breakdown of the association errors produced by the Socially Constrained Tracker in each sequence. False positive rate (FPR) is computed as $\frac{\sum_t fp_t}{\sum_t g_t}$, false negative rate (FNR) is $\frac{\sum_t m_t}{\sum_t g_t}$, and miss-match errors (MME) is $\sum_t mme_t$.

The socially constrained tracking algorithm performed similarly well to the tracker from Linder et al. [2] in the *Wandering* sequences, which is unsurprising as both trackers are provided with the same pose detections and use similar motion models. Interestingly this author's tracker performed better in terms of MOTA for the *Alighting*, *Walkthrough* and *Passing* sequences all of which involve movement of people through a densely crowded area in close proximity to others. This improvement can likely be attributed to the addition of social constraints in track prediction which significantly narrow the spread of particle states in crowded scenarios by avoiding predictions in close proximity to others. Note that MOTA scores of both trackers are low on some sequences due to time participants spent outside the depth sensor FOV but still visible to the optical tracking system. However the comparison between the trackers remains fair.

Poorer MOTP scores are likely due to the pose detections rather than either tracker. Two major sources of error exist which have not been accounted for in these results: (1) the arbitrary offset between the rigid marker placed on the back of subjects and their shoulder-centre, (2) significant scale errors in the depth values reported by the depth camera. The first of these is simply the result of the ground truth data collection method and cannot be eliminated, the second could be addressed by calibrating for depth scaling using one of several published methods [59, 60].

4.8 Summary

This chapter presented a method for robustly tracking pedestrian movements in crowded environments by leveraging insights into social behaviour. Specifically the method imposes an interpersonal distance constraint in the prediction step and a shoulder alignment pseudo-observation in the likelihood update of a particle filter based tracking algorithm to improve robustness to crowding. Additionally a measure of confidence in the orientation estimates provided by the detection algorithm from Chapter 3 is used to adjust the variance of the orientation measurement model allowing the tracking algorithm to give more consideration to better observations. Finally the tiered approach to data association allows the tracking algorithm to pay attention to unassociated observations which may

represent new targets, without letting intermittent false positive observations adversely affect tracking confirmed targets.

The algorithm is evaluated in terms of precision and accuracy on a dataset of depth image sequences with accompanying ground truth and compared to a state-of-the-art tracking algorithm. The improved tracking accuracy achieved by the socially constrained tracker in densely crowded scenarios points to the benefits of using a socially informed model in such settings.

In order to create robots that can operate in human environments it is vital to build perception algorithms which can leverage an understanding of social behaviour. Human environments present many challenges for perception algorithms as they are unstructured, cluttered, and dynamic. Additional challenges introduced in densely crowded environments include: frequent occlusions which increase the difficulty of detecting and persistently tracking individuals; and close proximity of people to one another which make data association difficult. The upshot of such crowded environments comes from the fact that people are social, and as such their behaviour is influenced by that of the people around them. By modelling these social influences the perception algorithms presented here are able to take advantage of the social information inherent in crowds to overcome the challenges they introduce.

The techniques introduced in this chapter for modelling social interaction are simple but effective. Clearly the social behaviour of humans is more complex than merely looking where they are going and maintaining interpersonal distances, but attempting to manually develop models of more complex behaviours is prone to over-fitting. Simple models are useful because they are easy to implement and tend to generalise better than more complex models. Furthermore simple models make efficient use of computational resources which makes them suitable for use embedded applications such the passenger monitoring system discussed in Chapter 5.

Chapter 5

Field Study: Managing Congestion on Train Platforms

5.1 Why Monitor Train Passengers?

"A well-functioning transport system is vital to the productivity of all economies…", stated a 2014 report from Price Waterhouse Coopers on the productivity benefits of public transport [61]. Passenger rail services are a critical component of public transport systems in major cities all over the world, however passenger crowding in peak travel times poses significant challenges for rail operators. As the populations of these cities continue to rise transport operators must find ways to maximise the capacity of existing transport infrastructure. One factor greatly affecting the capacity of a passenger rail network is train *dwell time*.

Dwell time is the time a train spends stopped at a station. When scheduling train services rail operators must estimate and account for dwell times, trading off between maximising throughput and risking disruptions to the schedule. When train dwell times exceed the allocated time, referred to as *over-dwell*, it causes delays which can snowball and affect the entire network. Dwell time is influenced by a number of factors including the numbers of boarding and alighting passengers, amount of passenger congestion on the train platform, and the profile of passengers (eg. regular commuters vs tourists).

In an effort to reduce dwell times and control the risk of over-dwell, rail operators often deploy staff in peak travel times to monitor and direct passenger movements. This task however, is challenging and highly dependant on the ability of staff to determine, communicate and influence passenger density and behaviour along the entire platform in a timely fashion. Better tools are needed to assist with such operations by providing rail staff with clear and up-to-date information about passenger distribution and movement along the platform before and during train dwell time.

The work presented in this thesis provides the algorithmic foundations for a powerful tool capable of providing live, accurate data of passenger movements in this crowded environment. Furthermore the data gathered by such a tool could be collected and used to analyse passenger movements in greater detail than ever before. This data could give train operators a greater understanding of passenger behaviour, as well as a powerful diagnostic tool for evaluating crowd management strategies, and even train station design.

This chapter discusses my work in collaboration with Centre for Autonomous Systems (CAS) and UTS Rapido, supported by Downer Rail and Rail Manufacturing Cooperative Research Centre (RMCRC) towards building a system for real-time monitoring of passenger movements on a busy train platform. The system, dubbed ”Dwell Track” took the algorithms developed in my research at CAS and, over a two year period, created a commercial grade prototype for use by Sydney Trains. The work culminated in April 2019 with an operational trial at Wynyard Station where it was used by Sydney Trains’ “Fast Track” teams to improve their dwell management operations. In this chapter I will discuss the hardware and software used in the system; present the data collected by the system and the types of desktop analysis it enables; and finally summarise the Wynyard Station field trial.

5.2 System Design

5.2.1 Hardware

The hardware used to realise this system went through several iterations as a research prototype before the latest version devised by UTS Rapido and trialled by Sydney Trains, however the core components of all versions have remained the same:

- A depth-sensing camera, to capture 3D data
- An embedded computer, to interface with the camera and run the perception algorithms
- A storage device, for recording the outputs of the algorithms
- An enclosure with a power source

The specific hardware components used in the final Dwell Track system are described in further detail below.

5.2.1.1 Depth Camera

The most important hardware component in the Dwell Track system is the depth camera. There are a wide variety of sensor types available that are capable of generating the 3D point-clouds required by the detection algorithm proposed in Chapter 3. The Orbbec Astra, pictured in Figure 5.1, is a structured-light based depth camera and was selected from a list of candidate sensors for use in the Dwell Track system. This type of sensor uses a projected infrared light pattern and infrared camera to measure the scene and provide a sequence of depth images, where each pixel records the distance of a visible surface in the scene from the camera baseline. Given a model of the camera optics these depth images can be converted into 3D point-clouds for use by the detection algorithm.

An advantage of structured-light depth cameras is that they are self-illuminating allowing them to work consistently in environments with lighting conditions ranging from complete

FIGURE 5.1: Orbbec Astra depth-camera used in the Dwell Track system

darkness up to typical bright artificial lighting. A disadvantage of these sensors is that they do not work well in sunlight due the high magnitude of infrared light from sunlight which overexposes the infrared camera, obscuring the structured light projection. Given the majority of trial sites were indoors or undercover this trade-off was suitable. To adapt the system to work on outdoor train platforms a stereoscopic sensor would likely be a better choice due to their superior performance in sunlight.

The relevant technical specifications of the Orbbec Astra are given in Table 5.1. Importantly the 60° horizontal field-of-view (FOV) and stated 8m sensing range provide a coverage zone large enough to monitor passenger movements around a typical set of train doors. In practice however we found the usable range of this camera to be closer to 6m. While the depth image resolution was high enough to provide sufficient point density at higher ranges, increasingly coarse steps in depth values negatively impacted pedestrian detection performance beyond this range.

TABLE 5.1: Technical specifications of Orbbec Astra Depth Sensor

Sensing range	0.6 - 8.0m
Horizontal FOV	60.0°
Vertical FOV	49.5°
Depth image resolution	640W × 480H
Frame-rate	30Hz
Depth accuracy	±3mm @ 1m

5.2.1.2 Embedded Computer

The main criteria for selection of the embedded computer was form factor. The selected computer needed to be compact enough to fit into a housing around the size of a typical closed-circuit television (CCTV) camera whilst still running the pedestrian perception pipeline as close as possible to the sensor frame-rate of 30Hz. A fanless pico-ITX motherboard with an 4 Core Intel Celeron CPU (shown in Figure 5.2) was selected for Dwell Track for its compact size and the fact that it out performed the ARM based alternatives in initial testing. This initial testing also revealed that the while the computer ran the pedestrian tracking pipeline at approximately 24Hz with 5 people present, this performance dropped to around 13Hz with 10 people. This highlighted a need for optimisation of the pedestrian perception pipeline software in order for it to perform acceptably on the embedded computer.

5.2.1.3 System Enclosure

The aim of the Dwell Track prototype was to be as close as possible to a commercially viable version of the system and as such the enclosure required a higher standard of engineering than the research prototypes preceding it. The main engineering considerations were:

- Dust and moisture ingress protection
- Power supply
- Heat dissipation
- Network connectivity
- Vandalism protection

Engineers at UTS Rapido designed an enclosure to meet these needs, pictured in Figure 5.2. It is made from machined aluminium with powder coated finish and assembled with Torx fasteners to provide some tampering protection. To allow for heat dissipation the pico-ITX computer is thermally coupled, via a heat spreader to the aluminium enclosure which has

FIGURE 5.2: Images of the Dwell Track System designed and built by UTS Rapido

external fins to increase its surface area. The enclosure includes a power-over-ethernet (POE) board so that both power and network connectivity can be provided via a single RJ45 connector.

5.2.2 Pedestrian Perception Software

The key components of the software framework used in the Dwell Track system are the pedestrian detection and pedestrian tracking algorithms that have already been described in chapters 3 and 4 respectively. These core algorithms however did require additional optimisation work discussed below to run at the sensor frame-rate. In addition to these key components a number of components were developed for the transport use case to allow each Dwell Track unit to automatically calibrate itself after installation as shown in Figure 5.3. The two main tasks performed in the calibration are static background subtraction and ground-plane alignment each of which are discussed in more detail below.

5.2.2.1 Perception Algorithm Optimisation

As inventor of the core algorithms, one of my key contributions to the Rapido project was to optimise the core perception algorithms to maximise the frame-rate of the perception pipeline and thus facilitate more robust operation in crowded environments. The most obvious change was to multi-thread the detection algorithm such that consecutive frames

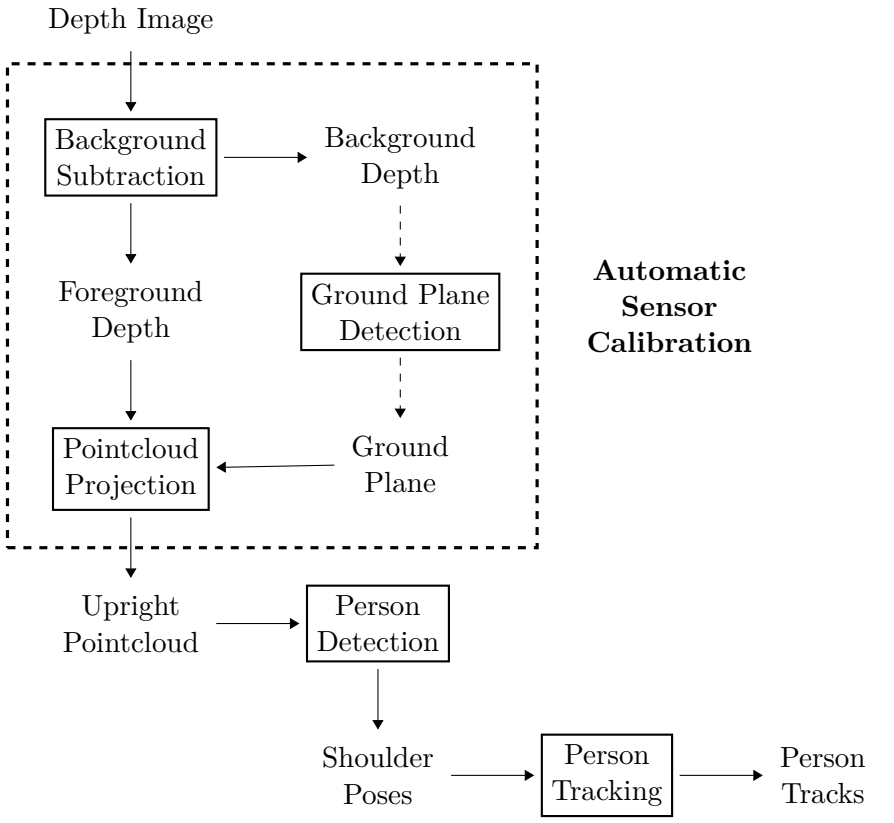


FIGURE 5.3: Overview of the software framework used in the Dwell Track system

could be processed in parallel. Implementing this change allowed greater CPU utilisation and greatly improved the frame-rate. Further to this, profiling of the software revealed that the main bottleneck of the algorithm was in the point-cloud clustering step of the person detection algorithm, when searching for the nearest cluster to each point. I was able to increase the efficiency of this algorithm by organising point clusters into a spatial grid, hence limiting the search space for each new point to a nearby subset of clusters. Additionally the use of background subtraction, discussed below, helped to lower the number of points input to the point-cloud clustering algorithm.

Figure 5.4 shows the frame-rate of the perception pipeline before and after these optimisations based on playback and of previously collected data from train stations at double the original rate i.e. 60Hz. The negative correlation seen here between frame-rate and the number of people in the scene is likely due to the positive relationship between the number of people in the scene and the number of pixels input to the point-cloud cluster algorithm. This is a result of the background subtraction only passing through points in

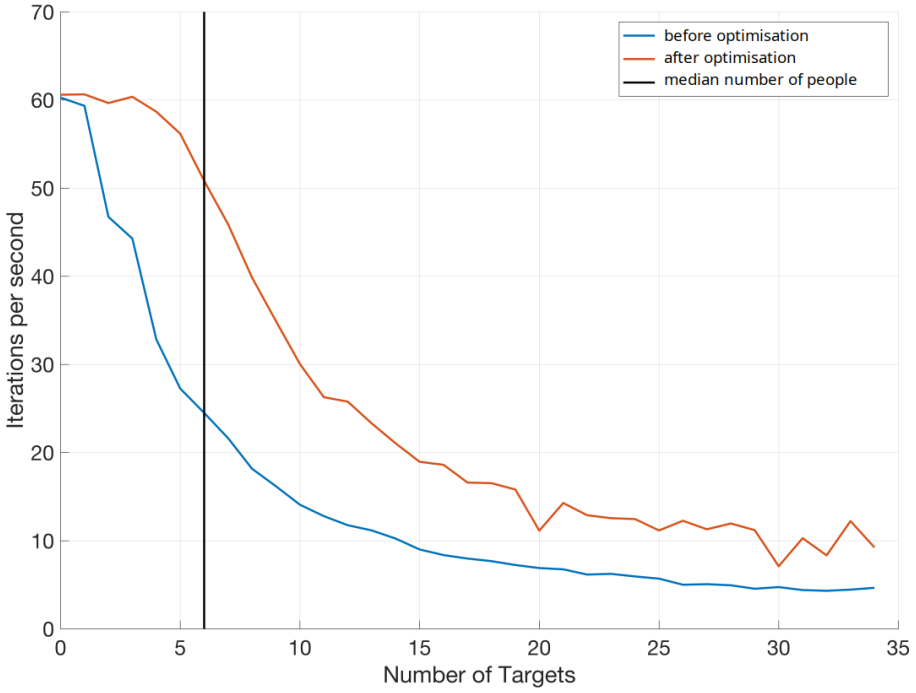


FIGURE 5.4: Frame-rate of the perception pipeline compared to the number of people visible, before and after optimisation efforts

the foreground, most of which represent people in the train platform context. As shown in the figure the aforementioned optimisation efforts raised the performance significantly with the full sensor frame-rate of 30Hz achieved in scenes of up to 10 people, up from 4 people prior. The frame-rate does drop off with higher numbers of people but seems to plateau around 9Hz compared to 4Hz previously.

5.2.2.2 Background Subtraction

Background subtraction segments parts of the depth image potentially describing people from those representing the static environment. A model of the static background is learned incrementally from the depth image and used to mask out pixels of each depth image consistent with the model, leaving only those considered to describe the foreground as illustrated in Figure 5.5. This reduces the amount of downstream processing required which improves the efficiency of the software. Background subtraction also helps with

detection of pedestrians that are in physical contact with the static environment by eliminating the static environment prior to point-cloud clustering. Additionally the learned background model is used to align data to the ground plane, further discussed in Section 5.2.2.3.

The background is learned based on the ideas presented in [62], in which the background value of each pixel in a colour or greyscale scene is modelled as a mixture of Gaussians, but with two simplifying assumptions resulting from the inherent relevance of depth information to the task of background modelling. The first assumption is that the background is uni-modal, which is reasonable in the case of depth data due to the relative invariance of depth values obtained from a given surface in a static scene, compared with light intensity and colour which may change depending on lighting conditions and surface orientation. The second assumption is that the background will correspond with the highest valued mode (furthest away) of the overall depth pixel process. This holds true in the case of depth because, unlike light intensity and colour, the depth value directly relates to the notion of background and foreground. With these assumptions we arrive at a model for the background depth consisting of a uni-variate Gaussian distribution per pixel which is represented by mean image $\boldsymbol{\mu} = \left[\mu_1, \cdots, \mu_N\right]$ and sigma image $\boldsymbol{\sigma} = \left[\sigma_1, \cdots, \sigma_N\right]$ with resolution N, equal to that of the depth image.

Given the background model $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ a background threshold image $\boldsymbol{b} = \left[b_1, \cdots, b_N\right]$ is computed as follows for the purpose of classifying pixels of the live depth image $\boldsymbol{d} = \left[d_1, \cdots, d_N\right]$ as either foreground or background.

$$b_u = \begin{cases} \mu_u - \tau_s \sigma_u - \tau_0 & \text{if } \sigma_u < \sigma_0 \\ 0, & \text{otherwise} \end{cases}$$

where the threshold scale τ_s and threshold offset τ_0 parameters determine the window of depth values in front of the mean background depth μ_u to consider part of the background. Note that applying such a window on the far side of the mean is not required as all points further away than the mean are considered part of the background. The parameter σ_0 is used as a threshold to only populate the pixels of $\boldsymbol{b}$ where the background model is sufficiently confident of the background depth value. Zero is used as a special value in

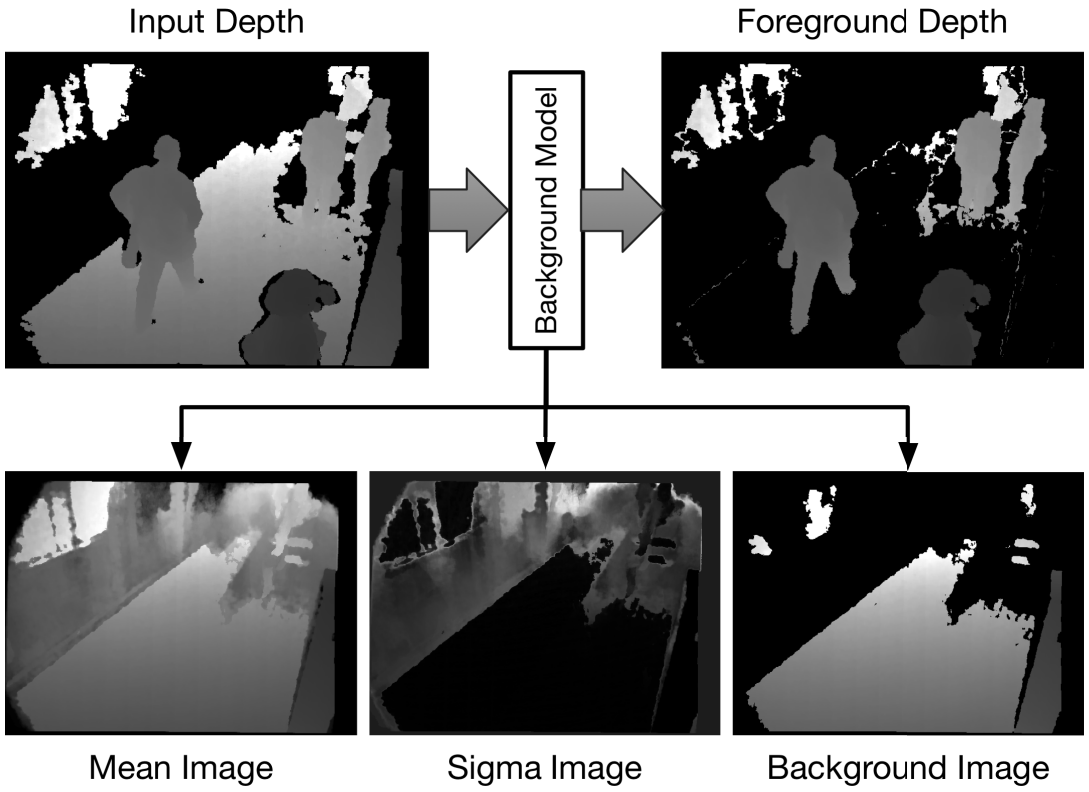


FIGURE 5.5: Each frame of input depth data is compared to the current background image in order to mask out background pixels and output the foreground depth image. At regular intervals the input depth image is used to update the background model.

$\boldsymbol{b}$ to indicate that the background will be ignored. To avoid unnecessary processing the background model $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is only updated every k frames with $k = 30$ our experiments. Examples of $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\boldsymbol{b}$ are given in Figure 5.5

Each data frame, each pixel d_u of the depth image is classified as belonging to either the foreground or the background by comparing it with the corresponding pixel of the background threshold image b_u. Pixels where $b_u \neq 0$ and $d_u \geq b_u$ are considered background pixels and masked out; the remaining pixels make up the foreground depth image. On background update frames the background pixels are used to update the background model as follows, based on the learning rate parameter α.

$$\mu_u \leftarrow (1-\alpha)\mu_u + \alpha d_u$$

$$\sigma_u \leftarrow \sqrt{(1-\alpha)\sigma_u^2 + \alpha(d_u - \mu_u)^2}$$

The new background model $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is used to update the background threshold image $\boldsymbol{b}$ as described above and the mean image $\boldsymbol{\mu}$ is used for ground plane alignment.

5.2.2.3 Ground-Plane Alignment

After allowing an initial *burn-in time* for the background model to be established, the mean image $\boldsymbol{\mu}$ is projected into a point-cloud representation and a plane is fit using random sample consensus (RANSAC) [63]. It is assumed here that the dominant plane represents the floor. In all subsequent frames the foreground depth image is projected into a point-cloud and transformed using the established floor plane such that the $z = 0$ plane is aligned with it. This process is performed on installation of the sensor or each time it's position is adjusted, eliminating the need for tedious manual calibration.

5.3 Data gathering

Over the course of the research project numerous field trials (listed in Table 5.2) were undertaken with the Australian rail transport providers, namely Queensland Rail (QR) and Sydney Trains. The goal of these field trials was two fold: 1) to collect depth image data from crowded train platforms for development of perception algorithms; and 2) to test each of the hardware iterations of the system in the field.

In the first two iterations of the device (2015 and mid 2016), QR provided access to a few of their inner city stations. Foremost interest was in the busiest station of their network Brisbane Central, while Roma St and Milton were also examined as feeder stations. While all three stations form part of the Central Business District in Brisbane, Central experienced extended dwell times in the peak hours of 15:00-17:30 on platforms 5 and 6. Roma St on the other hand has a train/bus interchange, Milton as a more suburban station experiences large crowds during sporting events.

In the last two iterations of the device (late 2016 and 2017), Sydney Trains provided access to two of their stations. The busiest station Town Hall had pressing issues of large congestion on platforms 5 and 6 which are used for interchange between several

Table 5.2: Field Trials undertaken for data collection

Date	Station	Platform Number(s)	No. Cams	Time of Day (24h)	Duration (hours)	Data (GB)
10/02/2015	Brisbane Central	5	3	10-12, 16-18	5	66
11/02/2015	Brisbane Central	5	3	07-10, 11-12	4	45
12/02/2015	Roma St	3	3	10-11, 15-18	4	63
10/05/2016	Brisbane Central	5, 6	3	15-18	3	78
11/05/2016	Brisbane Central	5, 6	3	17-18	3	60
11/05/2016	Milton	1, 4	1	07-08	1	2
12/05/2016	Brisbane Central	5, 6	3	17-18	1	60
12/05/2016	Milton	1, 2, 4	1	07-08, 15-16	2	3
13/12/2016	Town Hall	1, 2	4	05-11	6	436
16/12/2016	Town Hall	5, 6	4	04-12, 16-18	12	560
26/06/2017	Redfern	1, 4	4	14-23	7	496
27/06/2017	Redfern	1, 4	4	06-09, 14-23	10	560
28/06/2017	Redfern	1, 4	4	06-09, 14-23	10	579
29/06/2017	Redfern	1, 4	4	06-10, 14-23	10	718
30/06/2017	Redfern	1, 4	4	06-10, 14-23	10	775
01/07/2017	Redfern	1, 4	4	16-19	3	325
02/07/2017	Redfern	1, 4	4	16-19	3	355

lines, in particular the T1 (North Shore and Western Line) and T4 (Eastern Suburbs and Illawarra Line). Redfern station was also examined, as an alternative interchange station. These two stations had almost four times the volume of pedestrians compared to Brisbane Central station, pushing the envelope of detection and tracking of very dense pedestrian environments.

As an example of typical sensor positioning for these field trials Figure 5.6 shows the location of 4 systems used for data gathering on Town Hall platforms 5 and 6. Additionally the photos in Figure 5.7 show systems 2 and 4 of this layout in-situ. System 2 here covers an escalator where passengers will arrive to the platform, while systems 1, 3 and 4 cover the platform edge where passengers will board and alight from train services. By covering ingress and egress points with these systems the intention is to be able infer levels of platform occupancy as well as measure boarding and alighting behaviour.

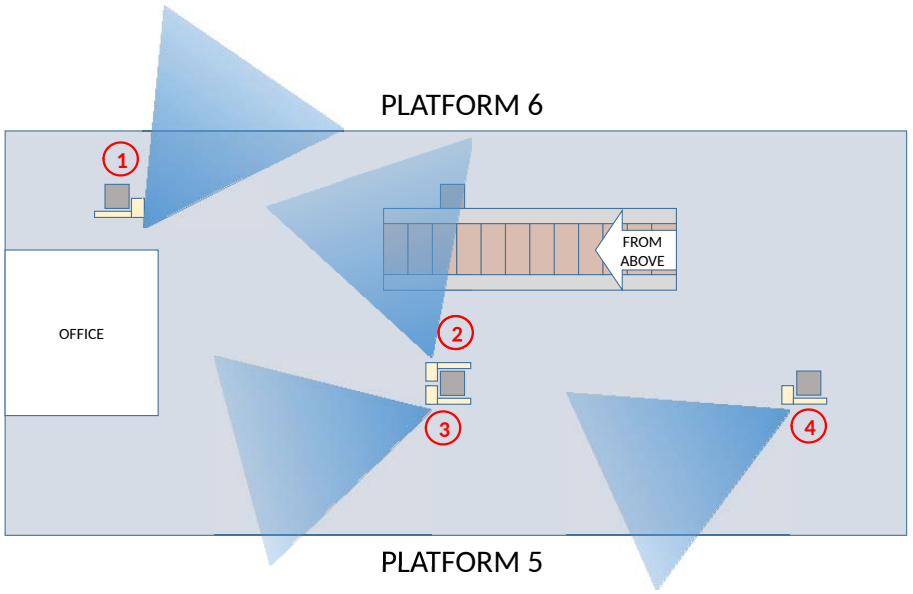


FIGURE 5.6: Sensor configuration on Town Hall platforms 5 and 6 covering passenger ingress from the escalator with system 2 and passenger exchange at the platform edge with systems 1, 3 and 4.

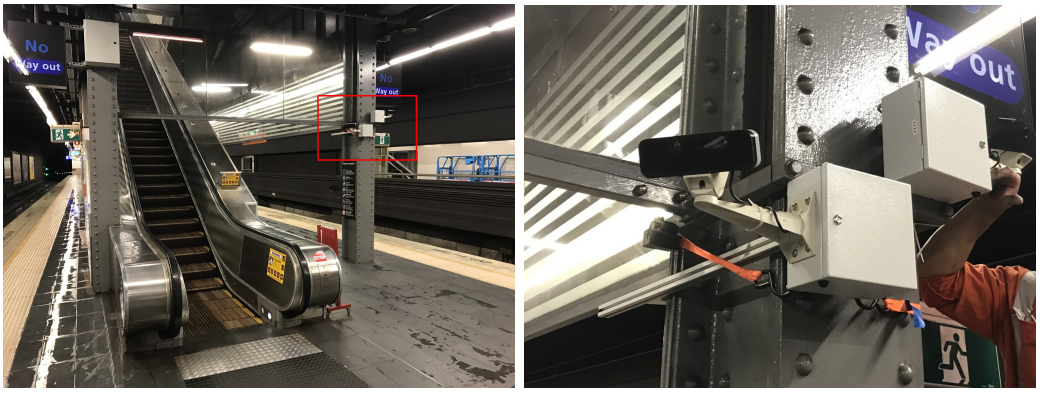
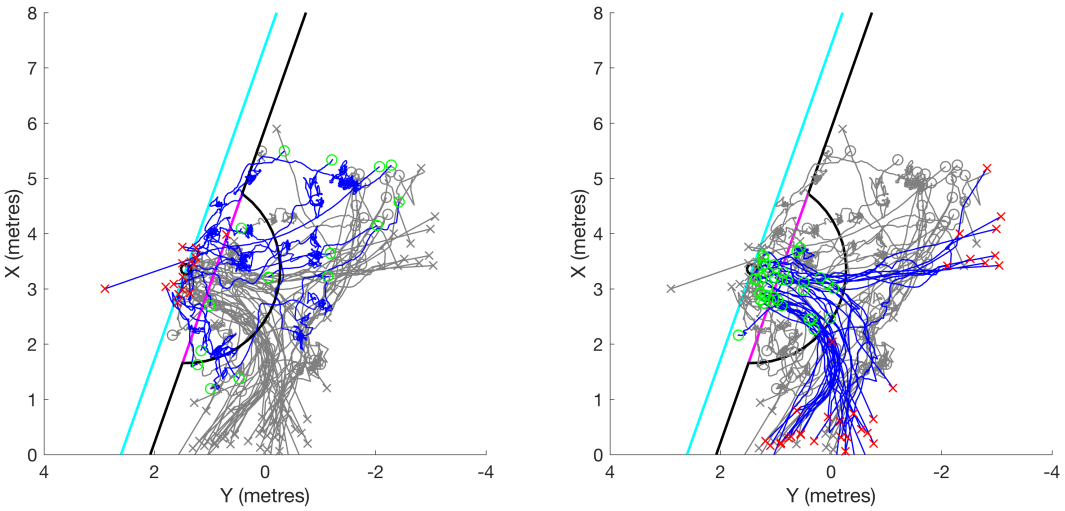


FIGURE 5.7: Early prototype systems installed on Town Hall platform 5/6 for data collection. Left image shows the location of system 2 and 3 from Figure 5.6. Right image shows a close up of system 2.

5.4 Passenger Behaviour Analysis

In consultation with rail operators data collected at these field trials were analysed to extract information relevant to congestion and dwell time management. Figure 5.8 shows the outputs of analysis software I developed for this purpose. Given tracking outputs from the pedestrian perception pipeline, the location of the train doors and the time of



(A) Boarding trajectories

(B) Alighting trajectories

FIGURE 5.8: Boarding and alighting trajectories for a single passenger exchange on Platform 5 of Town Hall station as measured by system 3 from Figure 5.6. Green circles show the starting position of each passenger when the train doors open, red crosses show their final position and dark blue lines show their trajectory. The pink line shows the train door threshold used by the analysis algorithm to count boarding passengers while the black arc shows the threshold used to count alighting passengers

doors opening and closing, this software is able annotate tracks of people who boarded (Figure 5.8a) and alighted (Figure 5.8b) from a train in a given passenger exchange.

This type of analysis allows operators to study the behaviour of passengers in greater detail than ever before and draw conclusions about passenger flows and use of space, for instance in Figure 5.8b where we clearly see two groups of passengers alighting from the train. The largest group of 22 passengers moves along platform 5 in the negative x direction (down) after exiting the train while separate group of 9 passengers appears to walk across to platform 6 (to the right) perhaps to wait for a connecting service.

Aside from such spatial insights the the same analysis can be used to examine passenger behaviour over time which is of particular relevance to the task of dwell time management. Figure 5.9 shows the same boarding and alighting data plotted as a histogram over time. Here we can observe some fairly expected behaviour where the majority of alighting passengers are able to exit the train prior to the majority of boarders entering. We see some overlap in these behaviours between the 25 and 35 second mark however, which may be

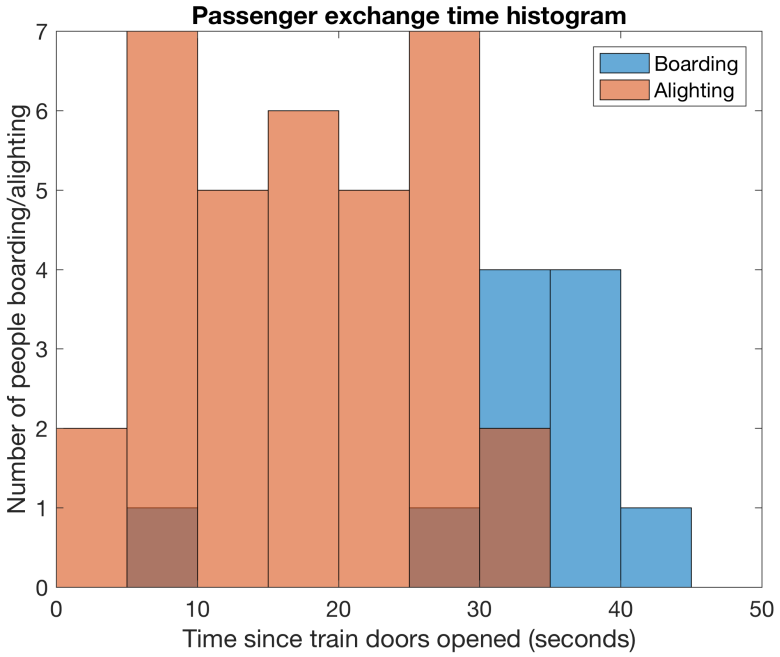


FIGURE 5.9: Passenger exchange histogram showing the numbers of passengers alighting (red) and boarding (blue) over time beginning when the train doors open.

considered counterproductive as boarding passengers potentially block alighting passengers. One ambitious passenger even boards between 5 and 10 seconds, likely obstructing the 7 others attempting to alight at that time. The entire exchange is over within 45 seconds with the doors closing 5 seconds later at the 50 second mark.

A small set of tracks were not counted as either boarding or alighting. In many cases such tracks represented passengers who were present on the platform prior to the exchange and remained on the platform, likely waiting for the next service. In some cases however these uncounted tracks are a result of tracks that were erroneously terminated by the tracking algorithm due to a lack of observations. Some failures of this kind are expected in such a crowded environment due to prolonged occlusions of passengers. Social constraints discussed in 4.3.2 go a long way to reducing the incidence of these failures however another pragmatic treatment to this issue is to place sensors higher and angled downward to the area of interest to reduce occlusions in the raw 3d data.

While the above insights may appear trivial at the level of one exchange at one set of doors, the same analysis can easily be applied to every train exchange, at every train door

FIGURE 5.10: One of 16 Dwell Track devices installed at Wynyard station as part of a 2019 trial undertaken by Sydney Trains

to build a very complete picture of boarding behaviour and how it might change service to service and throughout the day. This big picture understanding could be used to optimise dwell time estimates for scheduling purposes and inform strategies for influencing passenger behaviour to expedite passenger exchange. It also provides a baseline of passenger behaviour on which the effect of interventions and even platform design changes can be accurately measured.

5.5 Dwell Track Field Trial

In August 2019 Sydney Trains commenced a trial of the Dwell Track system on Platform 3 of Wynyard Station aimed at supporting the operations of their Fast Track teams in managing train dwell time during peak travel times. The trial was the culmination not only of the research presented in this thesis but of a two year engineering effort to build a commercial prototype suitable for application on train platforms. The project was supported by Downer Rail and the Rail Manufacturing CRC and undertaken by Rapido: a rapid prototyping and commercialisation group within the University of Technology Sydney.

FIGURE 5.11: Tablet application developed at UTS Rapido and used by Fast Track teams at Wynyard station to support dwell time management operations

A total of 16 Dwell Track devices (pictured in Figure 5.10) were installed along Platform 3 at Wynyard station covering the locations of every train door. The devices were connected via ethernet to a server, used to aggregate the data from all devices and serve it to a series of tablets used by the Fast Track team. The tablets ran a web-based application developed by engineers at Rapido displaying:

- Live summaries of passenger flow and congestion driven by Dwell Track devices
- Real-time service scheduling updates
- Carriage loading for the next incoming train
- Current, estimated and historical dwell times

The tablets (pictured in Figured 5.11) were used as a tool to support decision making by members of the Fast Track team by giving them real time insight into which areas of the platform were causing delays to dwell time at any given moment. In addition to the tablet application engineers as Rapido developed a set of dashboards to provide visual summaries of data from the Dwell Track devices.

Chapter 6

Conclusions

6.1 Contributions

This thesis contributes to the field of robotic perception by proposing and evaluating algorithms to detect and track the pose of pedestrians in crowded environments. Furthermore it documents the translation of these algorithms into a prototype system and the application of this prototype to solve real world problems. Chapters 3, 4 and 5 respectively detail the 3 main contributions of the thesis which we will summarise again here.

6.1.1 Pedestrian Pose Detection in Crowds

Chapter 3 presents an novel algorithm for detecting the 4DOF pose (x, y, z, θ_z) of pedestrians from 3D point-cloud data in crowded environments. The approach presented leverages the insight that the head and shoulders of pedestrians often remain visible even in very crowded scenes by efficiently fitting a pair of ellipsoids to regions of the point-cloud representing them. The parameters of these ellipsoids provide an estimate of the 3D position of the centre of the persons shoulders and the orientation of their shoulders about the vertical axis. The accuracy of poses extracted by the algorithm is evaluated in a variety of challenging scenes against ground truth obtained from an optical motion capture system resulting in an average error of 9.89cm horizontally, 4.27cm vertically and 12.84° in orientation which is sufficient to support robust tracking.

6.1.2 Socially Constrained Pedestrian Tracking in Crowds

Chapter 4 presents a novel algorithm for robustly tracking the 3DOF pose (x, y, θ_z) of pedestrians in crowded environments. The algorithm uses orientation estimates θ_z provided by the detection algorithm to inform velocity predictions based on the assumption that people align the front of their shoulders with their walking direction. The algorithm constrains its position predictions based the assumption that people will maintain a minimum interpersonal distance. These two ideas lead to improved predictions and thus more robust tracking in crowded environments. The tracking algorithm is compared to a state-of-the-art pedestrian tracking algorithm using the CLEAR-MOT tracking metrics and shows improved performance as crowding increases.

6.1.3 A System for Monitoring Passenger Crowding on Train Platforms

Chapter 5 discusses the application of the algorithms presented in Chapters 3 and 4 to real problems faced by transport operators in managing passenger behaviour. The chapter outlines the efforts of the author in adapting and improving the algorithms for use in this application and in the collection and analysis of field data to explore its potential value to transport operators. The success of these efforts is demonstrated by extension of the research project into a commercialisation project supported by industry partner Downer Rail resulting in a high quality prototype: Dwell Track. Furthermore an operational trial of 16 Dwell Track systems was undertaken by transport operator Sydney Trains at Wynyard station to support dwell management operations in peak travel times.

In press discussing the operational trial, Andrew Constance, New South Wales Minister for Transport and Roads was quoted saying:

> "This could be a technological solution to a very human problem... Precise mapping of crowd behaviour and what we call 'train dwell times' will help us improve systems to manage customers and make sure they get where they need to go."

while Tim Young, Executive General Manager, Downer's Rollingstock Services, said:

> "Dwell Track not only provides real-time data to aid decision making but will also provide longer-term insights into dwell management and platform operations. This technology is a key example of what collaborative partnerships between industry, university and our customers can achieve"

6.2 Future Work

Beyond detection and tracking of pedestrians, if we want to create systems that can safely work alongside us in human dominated environments we need methods to predict pedestrian trajectories ahead of time. Such a capability would allow, for instance, autonomous vehicles to anticipate the path of jaywalking pedestrians, avoiding accidents. The tracking algorithm presented in Chapter 4 uses a constant velocity model to predict pedestrian movements between detection frames, but to make accurate longer term predictions in crowds without prior knowledge of intended destinations requires a more sophisticated model.

As with many challenging problems in perception today deep-learning may offer answers. In 2016 researchers from Stanford published their seminal work on this problem: Social LSTM [41], which used a LSTM based neural architecture to learn to predict pedestrian trajectories by modelling their social interactions. This work inspired many subsequent papers in this field [42, 43, 64–66] all applying deep learning techniques to learn models for socially aware trajectory prediction from publicly available pedestrian datasets.

In my own preliminary experiments on this problem in 2018 I found that while I could achieve comparable results using similar LSTM based architectures trained on these datasets, ablation studies removing the social interaction mechanism suggested that its influence was minimal. In 2020 the work of [44] took this further by comparing a simple constant velocity model to state-of-the-art deep-learning based approaches to pedestrian motion prediction; to their surprise the constant velocity model outperformed them. The paper analyses in depth the assumptions underlying the deep-learning based approaches concluding that the social interactions observed in common pedestrian motion datasets are either less relevant than commonly believed or too complex to aid in prediction.

Clearly there is more work to be done to better understand and predict pedestrian motion. Perhaps rather than aiming for interaction aware models that generalise well across all environments, a fruitful research direction could be to lean in to the environmental bias present in these datasets and develop approaches which leverage this to infer pedestrian intentions based on the context.

Bibliography

[1] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. 4 2018. doi: 10.48550/arxiv.1804.02767.

[2] T. Linder, S. Breuers, B. Leibe, and K. O. Arras. On multi-modal people tracking from mobile platforms in very crowded and dynamic environments. pages 5512–5519. IEEE, 5 2016. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487766.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Cvpr 2016*, pages 779–788, 2016. ISSN 10636919. doi: 10.1016/j.nima.2015.05.028.

[4] N. Kirchner, A. Alempijevic, A. Virgona, X. Dai, P. G. Pl, and R. K. Venkat. A robust people detection, tracking, and counting system. In *Australasian Conference on Robotics and Automation*, pages 2–4, 2014.

[5] A. Virgona, N. Kirchner, and A. Alempijevic. Sensing and perception technology to enable real time monitoring of passenger movement behaviours through congested rail stations. *Australasian Transport Research Forum*, (October):1–14, 2015.

[6] A. Virgona, A. Alempijevic, and T. Vidal-Calleja. Socially constrained tracking in crowded environments using shoulder pose estimates. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4555–4562, 2018. ISSN 10504729. doi: 10.1109/ICRA.2018.8461030.

[7] A. Alempijevic, A. Virgona, and T. Vidal-Calleja. Monitoring systems, and computer implemented methods for processing data in monitoring systems, programmed

to enable identification and tracking of human targets in crowded environments, Issued to University of Technology Sydney and Downer EDI Rail Pty Ltd. Patent No. WO2019109142A1 / AU2018379393A1, 2018.

[8] U. Iqbal, A. Milan, and J. Gall. PoseTrack: Joint multi-person pose estimation and tracking. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:4654–4663, 2017. doi: 10.1109/CVPR.2017.495.

[9] J. O'Rourke and N. I. Badler. Model-Based Image Analysis of Human Motion Using Constraint Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(6):522–536, 1980. ISSN 01628828. doi: 10.1109/TPAMI.1980.6447699.

[10] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (May 2014):193–199, 1997. ISSN 10636919. doi: 10.1109/cvpr.1997.609319.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, 1:I—-511—-I—-518, 2001. ISSN 1063-6919. doi: 10.1109/CVPR.2001.990517.

[12] P. Viola and M. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000013087.49260.fb.

[13] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886–893, 2005. ISSN 1063-6919. doi: 10.1109/CVPR.2005.177.

[14] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: a real time system for detecting and tracking people. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 962, 1998. ISSN 10636919.

[15] M. Yamada, K. Ebihara, and J. Ohya. A new robust real-time method for extracting human silhouettes from color images. pages 528–533, 1998.

[16] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001. ISSN 10773142. doi: 10.1006/cviu.2000.0897.

[17] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3 SPEC. ISS.):90–126, 2006. ISSN 10773142. doi: 10.1016/j.cviu.2006.08.002.

[18] L. Chen, H. Wei, and J. Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34(15):1995–2006, 2013. ISSN 01678655. doi: 10.1016/j.patrec.2013.02.006.

[19] C. Zhang and Z. Zhang. A Survey of Recent Advances in Face Detection. (June), 2010.

[20] Y. Kameda and M. Minoh. A human motion estimation method using 3-successive video frames. pages 135–140, 1996.

[21] C. Bregler. Learning and recognizing human dynamics in video sequences. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 568–574, 1997. ISSN 10636919. doi: 10.1109/cvpr.1997.609382.

[22] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. *End-to-End Object Detection with Transformers*, pages 213–229. 2020. doi: 10.1007/978-3-030-58452-$8_1$3.

[23] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014. doi: 10.1109/CVPR.2014.214.

[24] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe. How Robust is 3D Human Pose Estimation to Occlusion? pages 1–5, 2018.

[25] T. Horiuchi, S. Thompson, S. Kagami, and Y. Ehara. Pedestrian tracking from a mobile robot using a laser range finder. pages 931–936. IEEE, 2007. ISBN 978-1-4244-0990-7. doi: 10.1109/ICSMC.2007.4413964.

[26] C. T. Chou, J.-Y. Li, M.-F. Chang, and L. C. Fu. Multi-robot cooperation based human tracking system using laser range finder. pages 532–537. IEEE, 5 2011. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980484.

[27] E.-J. Jung, J. H. Lee, B.-J. Yi, J. Park, S. Yuta, and S.-T. Noh. Development of a laser-range-finder-based human tracking and control algorithm for a marathoner service robot. *IEEE/ASME Transactions on Mechatronics*, 19:1963–1976, 12 2014. ISSN 1083-4435. doi: 10.1109/TMECH.2013.2294180.

[28] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time Human Pose Recognition in Parts from Single Depth Images. *Commun. ACM*, 56(1):116–124, 2013. ISSN 0001-0782. doi: 10.1145/2398356.2398381.

[29] A. Dib and F. Charpillet. Pose estimation for a partially observable human body from RGB-D cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, number SEPTEMBER 2015, pages 4915–4922. IEEE, sep 2015. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS.2015.7354068.

[30] M. Roth, D. Jargot, and D. M. Gavrila. Deep end-to-end 3d person detection from camera and lidar. pages 521–527. IEEE, 10 2019. ISBN 978-1-5386-7024-8. doi: 10.1109/ITSC.2019.8917366.

[31] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 3 1960. ISSN 0021-9223. doi: 10.1115/1.3662552.

[32] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. pages 3464–3468. IEEE, 9 2016. ISBN 978-1-4673-9961-6. doi: 10.1109/ICIP.2016.7533003.

[33] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. pages 3645–3649. IEEE, 9 2017. ISBN 978-1-5090-2175-8. doi: 10.1109/ICIP.2017.8296962.

[34] B. Anderson and J. B. Moore. Optimal filtering. *Prentice-Hall Information and System Sciences Series*, 1979.

[35] J. Munkres. ALGORITHMS FOR THE ASSIGNMENT AND TRANSPORTATION PROBLEMS. 5(1), 1957.

[36] Y. Bar-Shalom, F. Daum, and J. Huang. The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6):82–100, 2009. doi: 10.1109/MCS.2009.934469.

[37] D. Helbing and P. Molnar. Social force model for pedestrian dynamics, 1995. ISSN 1063651X.

[38] R. Gayle, W. Moss, M. C. Lin, and D. Manocha. Multi-robot coordination using generalized social potential fields. *2009 IEEE International Conference on Robotics and Automation*, pages 106–113, 2009. ISSN 1050-4729. doi: 10.1109/ROBOT.2009.5152765.

[39] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras. People tracking with human motion predictions from social forces. pages 464–469. IEEE, 5 2010. ISBN 978-1-4244-5038-1. doi: 10.1109/ROBOT.2010.5509779.

[40] G. Ferrer, A. Garrell, and A. Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. pages 1688–1694. IEEE, 11 2013. ISBN 978-1-4673-6358-7. doi: 10.1109/IROS.2013.6696576.

[41] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. ISSN 10636919. doi: 10.1109/CVPR.2016.110.

[42] A. Vemula, K. Muelling, and J. Oh. Social Attention: Modeling Attention in Human Crowds. 2017.

[43] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2255–2264. IEEE, jun 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00240.

[44] C. Scholler, V. Aravantinos, F. Lay, and A. Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5:1696–1703, 4 2020. ISSN 2377-3766. doi: 10.1109/LRA.2020.2969925.

[45] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum*, 26:655–664, 9 2007. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2007.01089.x.

[46] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. pages 261–268. IEEE, 9 2009. ISBN 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459260.

[47] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. pages 2544–2550. IEEE, 6 2010. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5539960.

[48] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. 11 2016. doi: 10.1007/s11263-017-1061-3.

[49] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath, and S. G. Narasimhan. Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15190–15200, 2021.

[50] A. Baak, M. Muller, G. Bharaj, H. P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1092–1099, 2011. ISSN 1550-5499. doi: 10.1109/ICCV.2011.6126356.

[51] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time Identification and Localization of Body parts from depth images. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3108–3113, 2010. ISSN 10504729. doi: 10.1109/ROBOT.2010.5509559.

[52] N. Kirchner, A. Alempijevic, and A. Virgona. Head-to-shoulder signature for person recognition. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1226–1231, 2012. ISBN 9781467314039. doi: 10.1109/ICRA.2012.6224901.

[53] D. Brščić, T. Kanda, T. Ikeda, and T. Miyashita. Person tracking in large public spaces using 3-D range sensors. *IEEE Transactions on Human-Machine Systems*, 43 (6):522–534, nov 2013. ISSN 21682291. doi: 10.1109/THMS.2013.2283945.

[54] C. C. Gordon, C. L. Blackwell, B. Bradtmiller, J. L. Parham, P. Barrientos, S. P. Paquette, B. D. Corner, J. M. Carson, J. C. Venezia, B. M. Rockwell, M. Mucher, and S. Kristensen. 2012 Anthropometric Survey Of U.S. Army Personnel: Methods And Summary Statistics. Technical report, 2012.

[55] Qingde Li and J. Griffiths. Least squares ellipsoid specific fitting. In *Geometric Modeling and Processing, 2004. Proceedings*, volume 2004, pages 335–340. IEEE, 2004. ISBN 0-7695-2078-2. doi: 10.1109/GMAP.2004.1290055.

[56] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, Inc., New York, USA, 2001. ISBN 047141655X. doi: 10.1002/0471221279.

[57] E. T. Hall. *The hidden dimension.* Doubleday & Co, 1966.

[58] K. Bernardin and R. Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008: 1–10, 2008. ISSN 1687-5176. doi: 10.1155/2008/246309.

[59] A. Teichman, S. Miller, and S. Thrun. Unsupervised Intrinsic Calibration of Depth Sensors via SLAM. In *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, jun 2013. ISBN 9789810739379. doi: 10.15607/RSS.2013.IX.027.

[60] M. Di Cicco, L. Iocchi, and G. Grisetti. Non-Parametric Calibration for Depth Sensors. *Robotics and Autonomous Systems*, 74:309–317, 2015. ISSN 09218890. doi: 10.1016/j.robot.2015.08.004.

[61] T. Zimmerman, B. Bost, M. Streeting, T. Zimmerman, B. Bost, M. Streeting, and R. Tyson. Better Public Transport, Better Productivity: The economic return on public transport investment. Technical report, Price Waterhouse Coopers, 2014.

[62] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*, 2(c):246–252, 1999. ISSN 10636919. doi: 10.1109/CVPR.1999.784637.

[63] M. A. Fischler and R. C. Bolles. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. ISSN 15577317. doi: 10.1145/358669.358692.

[64] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena. A Data-driven Model for Interaction-aware Pedestrian Motion Prediction in Object Cluttered Environments. In *IEEE International Conference on Robotics and Automation*, pages 5921–5928, 2018. ISBN 9781538630808.

[65] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. pages 1349–1358. IEEE, 6 2019. ISBN 978-1-7281-3293-8. doi: 10.1109/CVPR.2019.00144.

[66] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. pages 12077–12086. IEEE, 6 2019. ISBN 978-1-7281-3293-8. doi: 10.1109/CVPR.2019.01236.