Multi-Modal Human Detection, Tracking and Analysis for Robots in Crowded Environments

Timm Linder

Technische Fakultät
Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Dissertation zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften

Betreuer: Dr. Kai O. Arras

Multi-Modal Human Detection, Tracking and Analysis for Robots in Crowded Environments

Timm Linder

Dissertation zur Erlangung des akademischen Grades Doktor der Ingenieurwissenschaften
Technische Fakultät, Albert-Ludwigs-Universität Freiburg im Breisgau

Dekan:	Prof. Dr. Rolf Backofen
Erstgutachter:	Dr. Kai O. Arras (Robert Bosch GmbH)
Zweitgutachter:	Prof. Dr. Wolfram Burgard (Albert-Ludwigs-Universität Freiburg)
Tag der Disputation:	24.04.2020

Abstract

The ability to perceive humans in their surroundings is a key ingredient for robots that operate in environments shared with humans, for example in consumer, industrial and automotive applications – such as a service robot for person guidance, an autonomous forklift in a warehouse, or a self-driving vehicle. This thesis deals with the problem of robustly detecting and tracking humans and recognizing their attributes in challenging environments in real-time, from the egocentric perspective of a computationally constrained mobile robot equipped with multiple sensing modalities. To address this problem, we examine both classical, model-based approaches and deep learning-based methods, and evaluate them on novel datasets as well as during real-world deployments on different mobile robot platforms in populated indoor scenarios.

We start this thesis with the question if complex data association methods are suitable for tracking groups of people in general, and in crowded environments in particular. To this end, we address the problem of joint individual-group tracking using learned pairwise social relations in RGB-D by extending an existing multi-model multi-hypothesis tracking method with a mechanism to maintain consistent group identities. In qualitative experiments on a novel dataset from a pedestrian zone, we achieve good real-time tracking performance for varying group sizes with few identifier switches. We apply the method to socially-aware navigation use-cases and present further experiments on simulated data in a more crowded environment, where we examine limitations of the hypothesis-oriented MHT approach under real-time constraints.

We then take a step back from group tracking and investigate the problem of tracking individual humans in crowded scenes using a mobile platform with a multi-modal sensor setup. Here, we first introduce a computationally very efficient tracking baseline: Using a relatively cheap set of extensions from the target tracking community to systematically tackle shortcomings of current systems, we attempt to improve robustness without resorting to more complex data association methods. After automated hyperparameter optimization, we compare our method systematically under different detector combinations to a hypothesis-oriented MHT, a track-oriented MDL tracker, and different NN variants on two novel datasets. We find that our efficient baseline method outperforms all other evaluated methods on the MOTA metric across all settings. Our key finding is that detector performance is the single, most influential factor affecting tracking performance which goes far beyond the impact of the chosen tracking algorithm.

Therefore, we focus our subsequent research on the detection task. One insight we gain from initial experiments is that recent CNN-based detectors perform well on 2D image-based detection, but this does not easily translate into robust localization in 3D world space. To deal with this, we develop a fast CNN-based one-stage detector that benefits from complementary RGB and depth image data and regresses 3D human centroids in an end-to-end fashion. We show that we can efficiently learn their 3D localization from a highly randomized RGB-D dataset that has been synthetically generated using a modern game engine, while exploiting existing real-world 2D object detection datasets to pretrain the detection task. The resulting method outperforms several state-of-the-art baselines, including a 3D articulated human pose estimation approach.

For 2D laser-based leg detection, we examine several classical model-based detection approaches as well as a CNN-based method that can be improved by observing human leg movement over a sequence of frames, while conducting experiments on a large-scale dataset from an elderly care facility. We then consider also methods for human detection in 3D lidar and RGB-D, and quantitatively compare detection performance across all three sensor modalities on two novel sequences in a challenging intralogistics scenario. This provides us with interesting insights on their strengths, weaknesses and generalization capabilities: In particular, we learn that the 3D lidar methods, which have been trained on available autonomous driving datasets, do not seem to transfer well to our application domain, where large-scale training datasets are not available; we observe problems especially in narrow and cluttered spaces. This indicates the need for more large-scale, domain-specific datasets and benchmarks in robotics, as well as methods that can generalize better with limited amounts of training data.

We finally take a closer look at humans in order to recognize their individual attributes. To this end, we extend an efficient tessellation-boosting method to recognize human attributes from RGB-D point clouds. The method achieves over 300 Hz without GPU, and can compete with computationally more complex deep learning-based methods on our novel attributes dataset.

Throughout this thesis, we acquired, annotated and analyzed several novel datasets in challenging environments, like a pedestrian zone, a crowded airport terminal, and intralogistics warehouses. The presented methods have been extensively validated “in the wild” to show their general applicability. To combine the methods, we propose a unified, multi-modal, ROS-based human detection and tracking framework that facilitates their deployment and evaluation. Due to its modular design with reusable interfaces and software components, we were able to deploy it on close to a dozen different robot platforms.

In particular, we gathered experiences with a socially-aware mobile service robot for person guidance that we deployed inside a crowded airport terminal. Here, system contributions have been made that go beyond human detection, tracking and analysis and touch the topics of sensor calibration, human-robot interaction, distributed software architecture and practical safety considerations. We share previously unpublished lessons learned during this ambitious project, which we hope will benefit future research in this area.

Kurzfassung

Die Fähigkeit, Menschen wahrnehmen zu können, ist essenziell für Roboter, die mit Menschen in einer gemeinsamen Umgebung operieren, zum Beispiel für Anwendungen im Consumer-, Automotive- und industriellen Bereich – wie etwa ein Service-Roboter, der Personen zu ihrem Ziel führt, ein selbstfahrendes Fahrzeug oder ein autonomer Gabelstapler in einem Lagerhaus. Diese Doktorarbeit beschäftigt sich mit dem Problem der robusten Detektion und Verfolgung von Menschen sowie der Erkennung ihrer Attribute in anspruchsvollen Umgebungen in Echtzeit und aus der egozentrischen Perspektive eines Roboters, der in seinen Rechenressourcen beschränkt ist und über multiple Sensormodalitäten verfügt. Wir setzen uns damit auseinander, indem wir sowohl klassische, modellbasierte Ansätze wie auch Deep Learning-basierte Methoden untersuchen und sie auf neuen Datensätzen und in Einsätzen in der realen Welt auf verschiedenen mobilen Roboter-Plattformen in von Menschen bevölkerten Innenräumen evaluieren.

Wir beginnen die Arbeit mit der Fragestellung, ob komplexe Datenassoziationsmethoden geeignet sind, um Gruppen von Menschen im Allgemeinen zu erkennen, und insbesondere in größeren Menschenansammlungen. Dazu befassen wir uns mit dem Problem der gemeinsamen Verfolgung von Individuen und Gruppen mittels gelernter, paarweiser sozialer Relationen in RGB-D, indem wir einen existierenden Ansatz zum Multi-Model Multi-Hypothesis Tracking mit einem Mechanismus zur Bewahrung konsistenter Gruppenidentitäten erweitern. In qualitativen Experimenten auf einem neuen Datensatz aus einer Fußgängerzone erreichen wir gute Tracking-Ergebnisse für variierende Gruppengrößen mit wenigen Wechseln der Gruppenidentität. Wir applizieren die Methode außerdem in Anwendungsfällen zur sozialen Navigation und präsentieren weitere Experimente auf simulierten Daten in einer Umgebung mit dichten Menschenmengen, wo wir die Grenzen des hypothesen-orientierten MHT-Ansatzes unter Echtzeitbedingungen untersuchen.

Wir machen nun einen Schritt zurück von der Betrachtung von Gruppen und untersuchen als nächstes das Problem, einzelne Individuen inmitten von Menschenmengen unter Zuhilfenahme einer mit multimodaler Sensorik ausgestatteten mobilen Plattform zu verfolgen. Hierzu führen wir zunächst eine hinsichtlich ihrer Rechenzeit sehr effiziente Referenzmethode ein: Mittels einer relativ einfachen Menge an Erweiterungen aus dem Forschungsbereich des Target Trackings versuchen wir systematisch die Defizite bestehender Systeme zu reduzieren, um deren Robustheit zu erhöhen ohne dabei auf komplexere Datenassoziationsmethoden zurückgreifen zu müssen. Nach automatischer Hyperparameter-Optimierung vergleichen wir unsere Methode auf zwei neuen Datensätzen systematisch unter verschiedenen Kombinationen von Detektoren mit einem hypothesen-orientierten MHT, einem track-orientierten MDL-Tracker, und verschiedenen NN-Varianten. Dabei finden wir heraus, dass die von uns vorgeschlagene effiziente Referenzmethode alle anderen evaluierten Methoden in der MOTA-Metrik in allen untersuchten Szenarien schlägt. Unsere wichtigste Erkenntnis ist dabei, dass die Performanz der Detektoren der eine, entscheidende Faktor für die Performanz des Trackings ist, dessen Einfluss weit hinausgeht über den des gewählten Tracking-Verfahrens.

Daher fokussieren wir unsere weitere Forschung auf die Detektion. Aus initialen Experimenten lernen wir, dass CNN-basierte Detektoren gute Leistungen im Bereich der bildbasierten 2D-Detektion liefern, sich dies aber nicht einfach übertragen lässt in eine robuste Lokalisierung im 3D-Raum der Welt. Um dieses Problem zu lösen, entwickeln wir einen schnellen, CNN-basierten einstufigen Detektor, der von komplementären RGB- und Tiefenbilddaten profitiert und eine durchgehende Regression von menschlichen 3D-Zentroiden lernt. Wir zeigen, dass wir die 3D-Lokalisierung effizient mittels eines hoch randomisierten RGB-D-Datensatzes lernen können, den wir mittels einer modernen Spiele-Engine synthetisch erzeugt haben, wobei wir existierende 2D-Objektdetektions-Datensätze aus der realen Welt zum Vorlernen nutzen. Die sich ergebende Methode übertrifft mehrere aktuelle Referenzmethoden in der Erkennungsleistung, darunter auch ein 3D-Ansatz zur Schätzung der artikulierten menschlichen Pose.

Für die Detektion von Beinen in 2D-Laser untersuchen wir mehrere klassische, modellbasierte Detektionsmethoden sowie einen CNN-basierten Ansatz, der verbessert werden kann, indem die menschliche Beinbewegung über eine Sequenz von Einzelbildern beobachtet wird. Dabei führen wir Experimente auf einem großen Datensatz aus einem Altenpflegeheim durch. Anschließend berücksichtigen wir auch Methoden zur Detektion von Menschen in 3D-LiDAR und RGB-D, und vergleichen deren Erkennungsleistung quantitativ über alle drei Sensor-Modalitäten hinweg auf zwei neuen Sequenzen aus einem schwierigen Intralogistik-Szenario. Dies bietet uns interessante Einsichten hinsichtlich ihrer Stärken, Schwächen und Fähigkeiten zur Generalisierung. Insbesondere erfahren wir, dass 3D-LiDAR-Methoden, welche auf verfügbaren Datensätzen aus dem Automotive-Bereich trainiert wurden, nicht gut in unsere Anwendungsdomäne transferieren, in der es keine großen Trainingsdatensätze gibt; wir beobachten Probleme insbesondere in engen, dicht gedrängten Räumen. Dies deutet darauf hin, dass weitere umfangreiche und domänenspezifische Datensätze und Benchmarks in der Robotik benötigt werden, ebenso wie Methoden die besser in der Lage sind zu generalisieren, wenn nur wenige Trainingsdaten verfügbar sind.

Schließlich werfen wir einen genaueren Blick auf Menschen, um ihre individuellen Attribute zu erkennen. Dazu erweitern wir einen effizienten Tessellation-Boosting-Ansatz, um menschliche Attribute aus RGB-D-Punktwolken zu erkennen. Die Methode erreicht über 300 Hz ohne Grafikbeschleunigung, und kann auf unserem neuen Attribut-Datensatz mit Deep Learning-Ansätzen mithalten, welche mehr Rechenleistung benötigen.

Im Rahmen der vorliegenden Arbeit haben wir mehrere neue Datensätze in anspruchsvollen Umgebungen, wie zum Beispiel einer Fußgängerzone, einem beengten Flughafen-Terminal, und Lagerhäusern in der Intralogistik aufgezeichnet, annotiert und analysiert. Die dargestellten Methoden werden ausgiebig “in freier Wildbahn” validiert, um ihre generelle Anwendbarkeit zu zeigen. Um die Methoden zu kombinieren, schlagen wir ferner ein vereinheitlichtes, multimodales, ROS-basiertes Framework zur Detektion und zum Tracking von Menschen vor, welches den Praxiseinsatz und die Evaluation solcher Methoden erleichtert. Dank seines modularen Aufbaus mit wiederverwendbaren Schnittstellen und Softwarekomponenten waren wir in der Lage, es auf fast einem Dutzend verschiedener Roboter-Plattformen einzusetzen.

Insbesondere haben wir praktische Erfahrungen mit einem mobilen Service-Roboter gesammelt, welcher Personen inmitten von dichten Menschenmengen durch ein Flughafen-Terminal führt und sich dabei sozialer Aspekte bewusst ist. Hier haben wir Systembeiträge geleistet, die über die Detektion, das Tracking und die Analyse von Menschen hinaus gehen und Themenfelder wie Sensorkalibrierung, Mensch-Roboter-Interaktion, verteilter Softwarearchitekturen, und praktischer Sicherheitsfragestellungen berühren. Wir möchten diese bisher unveröffentlichten Lektionen, die wir während dieses ambitionierten Projektes gelernt haben, teilen in der Hoffnung, dass sie der weiteren Forschung in diesem Gebiet zuträglich sind.

Acknowledgements (Danksagung)

This doctoral dissertation is the product of countless long days and nights during my PhD years that I spent in the lab, office, an airport terminal and industrial warehouses. The research in this thesis would not have been possible without the support of many people:

First, I would like to thank my PhD advisor, Kai O. Arras, for his helpful advice, guidance and inspirations to my research; my second reviewer, Prof. Wolfram Burgard, and the other members of my dissertation committee, Prof. Bernhard Nebel and Prof. Joschka Bödecker, for their time, their helpful insights and valuable recommendations.

Successful research in a fast-paced, data-intensive field like robotic perception relies on scientific exchange and collaboration. I would like to thank my collaborators Lucas Beyer, Stefan Breuers, Alexander Hermans and my Social Robotics Lab colleagues Luigi Palmieri, Billy Okal and Matthias Luber for the many lessons that I learned from them. I am grateful for the intense and insightful discussions I had with my students Sven Wehner, Fabian Girrbach, Dennis Grießer and Michael Hernandez, our research assistants and interns Markus Schwenk, Gregor Richter and Kilian Pfeiffer, and lab neighbors Armin Hornung, Daniel Maier, Felix Burget and Luciano Spinello.

I would like to thank my collaborators in the ILIAD project, Manuel Fernandez, Marc Hanheide, and Martin Magnusson especially for their help during the data recordings; many thanks go to Tomasz Kucner, Rudolph Triebel, Lucas Beyer and Stefan Breuers for the adventures we experienced together with SPENCER during nightly mapping runs at the airport, seeing our robot seek physical contact with lab walls and occasionally losing its head, and re-soldering parts of our data recording platform in a hotel room in the middle of the night.

I am deeply grateful to my family, my parents, grandparents, uncles and brother, for supporting me in my education and my research endeavors. Many, many thanks go to all of my friends for their invaluable moral support and their patience, especially in the final year.

Lastly, I thankfully would like to acknowledge that the research in this thesis has been financially supported by the European Commission under contract number FP7-ICT-600877 (SPENCER) and H2020-ICT-2016-732737 (ILIAD).

Contents

Part I

Fundamentals

CHAPTER 1

Introduction

The ability to perceive humans in their surroundings is a key ingredient for robots that operate in uncontrolled human-populated indoor and outdoor environments, *i. e.* outside of a safety cage. For example, as depicted in Figure 1.1, a service robot that autonomously navigates through a crowded airport terminal to guide passengers to their departure gate needs not only to keep track of the group of passengers it is guiding, but also be aware of dozens of other people in the vicinity. At an airport, travelers from various cultures and age groups may be traveling individually or in groups to destinations with different weather conditions. Therefore, their clothing and appearance, including luggage they are carrying, may vary as much as the social rules and conventions that they are used to adhere to. Also human dynamics can vary spatially and temporally: At a gate, the robot might encounter people that are queuing up to board their plane; in a shopping area, there might be a dense crowd or it could be mostly empty, depending on the time of the day; and people might be standing or walking on horizontal escalators.

Not only the human subjects, but also the environment can pose its own challenges: Lighting can vary significantly from one second to another, for instance when the robot is turning, and challenges arise especially under the presence of direct sunlight or reflections. In outdoor scenarios, precipitation, fog and other vehicles can pose additional sensing issues and make detection of humans difficult.

Figure 1.1: A service robot is guiding passengers to their departure gate at Amsterdam Schiphol airport, while navigating through dense crowds of people which it perceives via its multi-modal human detection and tracking pipeline. (Photo credit: KLM / SPENCER project, used with permission)

1.1 Problem statement

This thesis deals with the problem of robustly detecting and tracking people and recognizing human attributes in such challenging environments in real-time, from the egocentric perspective of a computationally constrained mobile robot equipped with multiple sensing modalities.

In comparison to computer vision systems that can operate on large-scale GPU clusters, mobile robotic systems are usually far more constrained in terms of computational power, but have higher demands with respect to online real-time processing, and robustness: Because perception of the environment is one of the three essential, commonly accepted primitives of robotics – *Sense*, *Plan* and *Act* –, any errors from the sensing stage can propagate into the resulting plans and actions. In the best case, a wrong or delayed detection of a human might only lead to an incorrect world model. In the worst case, however, with heavy service robots or self-driving vehicles, a collision with serious injuries might be the result.

To improve robustness and learn more about people, multi-modal sensor setups can be used to combine advantages and mitigate weaknesses of different sensors, such as monocular cameras, RGB-D sensors, 2D safety laser range finders, or 3D lidars. When equipping the environment with external sensors is not an option, they are usually mounted on the mobile platform itself. What distinguishes such robots from, for instance, surveillance systems is that they perceive people and the environment from a non-stationary ego-centric perspective, where high levels of occlusion become a major challenge once the scene becomes crowded.

Previous work on human detection and tracking by Luber (2014) has explored strategies to make tracking of persons from a robot-centric perspective more robust by incorporating different forms of social context into a multi-hypothesis tracking system using detections from either a 2D laser-based person detector or an RGB-D-based detector as input. The system as such has mostly been evaluated offline on recorded datasets with a static sensor, or been deployed on a robot in small-scale lab experiments. Other related approaches focus on detecting or tracking humans by employing single 2D or 3D lidar sensors (Spinello, Luber, et al., 2011; Leigh et al., 2015) or RGB-D cameras (Hosseini Jafari et al., 2014; Munaro and Menegatti, 2014; Munaro, Lewis, et al., 2016; Wojke, Memmesheimer, et al., 2017; Kollmitz et al., 2019).

Methods that utilize heterogeneous multi-sensor setups for human detection and tracking have mostly been validated in urban autonomous driving scenarios (*e. g.* Spinello et al., 2008; Ku et al., 2018; Qi et al., 2018). Surprisingly, scenarios using such setups on a mobile platform in indoor environments have only received little attention over recent years (*e. g.* Martin et al., 2006; Bellotto et al., 2009; Volkhardt et al., 2013; Wengefeld et al., 2016; Wengefeld, Mueller, et al., 2019) – especially not in environments as crowded and complex as an airport terminal. Only very recently, the topic of multi-modal human tracking in crowded environments from an egocentric perspective has been gaining traction in the computer vision community[1].

[1] See the *ICCV 2019 Workshop on Visual Perception for Navigation in Human Environments* and the accompanying benchmark dataset, which features “crowded sequences”, a “novel [egocentric] perspective”, “multi-modal sensor streams” and aims to be the ”first large dataset with annotated indoor scenes”.

Within this context, this thesis makes contributions to the following key questions:

1. How can we robustly and efficiently track individuals and groups of people in crowded environments from a robot-centric perspective?
2. How large are the relative impacts of the chosen detection and tracking algorithms and sensor modalities on tracking performance?
3. How can we accurately detect and localize persons not only in 2D image space, but also in metric 3D coordinates – which is essential for robotics applications?
4. How can we train state-of-the-art deep learning-based detection methods using limited amounts of labeled real-world 3D data? How can synthetic data help in this context?
5. Can we learn more about humans by looking at their point clouds, *e. g.* human attributes?
6. How can we make a multi-modal human detection and tracking pipeline more modular and reusable across different robot platforms and application scenarios? Which additional engineering aspects are relevant when deploying such a system on a real robot that is operating at fast human walking speeds in a crowded airport environment?

1.2 Proposed approach and outline of this thesis

We approach these questions as follows: After giving a brief overview of the most important techniques for online multi-object tracking, we start this thesis by extending the work of Luber et al. (2013) on multi-model multi-hypothesis group tracking in 2D laser range data with the goal of gaining first research insights. We incorporate a mechanism to maintain robust group identities across merges and splits and reformulate the likelihood for continuation events. We then apply the method to a novel in-the-wild multi-sensor RGB-D person dataset as well as simulated data from a highly crowded scenario. In particular, we are motivated by the research question if complex data association methods such as hypothesis-oriented MHT are applicable also to crowded environments – the main topic of this thesis.

To further address this question, we shift focus from group-level to individual person-level tracking and systematically compare alternative data association methods, including a track-oriented MDL tracker and an efficient nearest-neighbor method that we extend by incorporating logic-based track initiation and deletion, IMM, and prior knowledge from an occupancy grid. To better understand the importance of detector performance, we conduct experiments on two novel multi-modal datasets, including a crowded airport scenario, and examine different combinations of 2D laser, RGB and depth-based detectors.

As we will find out, in such complex environments, detector performance matters much more than the choice of data association algorithm. Therefore, we shift research focus towards human detection. For person detection in RGB-D, despite recent advances in 2D image-based object detection using deep learning techniques, we show on a novel dataset from an intralogistics use-case that accurate 3D localization under occlusion is an unresolved issue; our earlier results indicated that this can be very harmful to tracking performance. To address this, we propose

an efficient and robust one-stage detector based upon the YOLO v3 architecture that exploits complementary information from both RGB and depth, and regresses 3D centroids end-to-end. One key challenge we encounter is the lack of labeled large-scale 3D datasets required by such deep learning methods. To deal with this, we propose novel techniques for data augmentation and synthesis of accurately labeled RGB-D data to reduce the manual annotation effort.

Especially human detection in 2D laser suffers from comparatively low detection performance. Existing methods that do not consider temporal information are not very robust, and model-based leg-tracking approaches that make hard model assumptions fail when people wear clothing under which legs are indistinguishable. We therefore contribute to a novel CNN-based person detector that learns deep feature representations from a large-scale dataset, while incorporating temporal information from multiple successive frames. We compare it to classical, model-based detection and leg tracking approaches that we retrain and tune using automated hyperparameter optimization on the same dataset. We then include further state-of-the-art 3D lidar and RGB-D baseline methods to perform a cross-modal evaluation and understand how well these methods generalize to a novel dataset from the intralogistics domain.

Now we have achieved state-of-the-art human tracking performance and can estimate human trajectories and motion states in real-time. However, for many applications *e. g.* in human-robot interaction, we want to gain a richer understanding of the robot's social environment. Therefore, we now take a closer look at humans, and try to recognize fine-grained attributes – for instance related to clothing, or gender – using a very efficient geometric approach operating on RGB-D point cloud data. We conduct extensive experiments on a novel RGB-D dataset for human attribute recognition and gain additional insights in-the-wild.

We then focus on the engineering aspects of deploying such systems in real-world applications. Integrating the components of a multi-modal human detection and tracking pipeline can be time-consuming, as often the pipeline needs to be tailored to a particular robot's sensory setup. To accelerate and simplify this process, we propose a robot- and application-agnostic ROS-based people tracking framework that can process input from multiple sensor modalities such as 2D laser, monocular cameras, RGB-D or 3D lidar in real-time. We demonstrate its modularity, reusability and extensibility by deploying it on different robots with distinct sensory setups.

Finally, around one year of research and development conducted by the author went into enabling the deployment of a 250 kg first-of-its-kind socially-aware autonomous guidance service robot in a crowded airport terminal for the final demonstration of the research project SPENCER. For a successful demonstration, this entailed not only devising of a multi-modal sensor setup and its calibration; but also implementing low-level safeguards to protect against failure of human detection and tracking; the design of user interfaces for human-robot interaction, remote operation, and diagnostics; as well as handling limitations of sensor technologies.

1.3 Scientific contributions

In Section 1.1, we outlined a number of interesting research questions. To address these challenges, this thesis makes the following scientific contributions:

- **Chapter 3** extends the work of Luber et al. (2013) on multi-model multi-hypothesis tracking of groups of people by an improved data-driven likelihood term for continuation events, a mechanism for retaining more stable group IDs across merges and splits, and by integrating it with a ROS-based multi-sensor RGB-D detector setup using the ComboHOD detector of Spinello and Arras (2011). The method obtains qualitatively good tracking performance on a novel dataset from a moderately crowded pedestrian zone. In simulation, we gain further insights on how the method scales to more crowded scenarios under real-time constraints. Our findings indicate that tracking of group formation processes in such crowded scenarios suffers from the real-time constraints imposed onto the MHT by a resource-constrained mobile robot, because not enough and sufficiently diverse global hypotheses can be generated within the available computational budget.

- **Chapter 4** compares several multi-object tracking algorithms, including a hypothesis-oriented MHT, a track-oriented MDL tracker and two nearest-neighbor variants, on two novel and challenging multi-modal datasets; one of them recorded at an airport and containing dense crowds of up to 30 individuals in close range at a time. This is an important contribution because a systematic comparison of different data association methods in such challenging human tracking scenarios from a robot-centric perspective had so far been missing. We study the impact of the examined methods on seven performance metrics, in both uni-modal and multi-modal detection setups. Our results indicate that computationally more complex data association methods do not necessarily lead to better results. On the contrary, we find that detector performance is the single, most influential factor impacting tracking performance that goes far beyond the impact of the chosen tracking method. We show that less resource-demanding tracking methods can obtain state-of-the-art performance by incorporating a well-tuned track initiation and deletion logic. To this end, we demonstrate how automated hyperparameter optimization can be used to tune such a tracking system without requiring extensive expert knowledge.

- **Chapter 5** revisits the human detection task in RGB-D. Recent advances with CNN-based single-stage object detectors such as YOLO v3 (Redmon et al., 2018), which are suitable for real-time application on a robot, have led to impressive results in 2D bounding box detection. However, our initial experiments reveal that accurate localization of person centroids in metric 3D space is not straightforward, even if additional registered depth images are available. This is especially true under the presence of background clutter and foreground occluder objects, in which case 2D bounding boxes are a sub-optimal intermediate representation for the 3D detection task. We therefore examine how we can robustly regress 3D centroids in an end-to-end fashion, without relying on such intermediate representation. We achieve this by extending YOLO v3 with a 3D centroid output and by fusing the RGB and depth modalities already in the feature extraction stage to exploit complementary information. We also find that standard 2D crop augmentations cannot be applied as-is, because they result in a loss of depth- and scale-awareness.

  Learning 3D coordinate regression end-to-end comes at the cost that a sufficiently large and diverse dataset with accurate 3D groundtruth is required. Unfortunately, such 3D labels are expensive to annotate by hand, and we were aware of no previously existing RGB-D datasets that are applicable to our scenario. We therefore show that we can learn

the 3D real-world detection task from a highly randomized and diverse synthetic RGB-D dataset, generated using Unreal Engine 4.

Combined with a proper transfer learning strategy that benefits from both existing large-scale real-world RGB datasets with 2D annotations, and our synthetic RGB-D dataset with 3D labels, our proposed image-based YOLO v3 human detector with RGB-D fusion outperforms several state-of-the-art baselines on a novel, challenging intralogistics dataset. Our baseline methods include an approach that first performs geometric clustering in the RGB-D point cloud to obtain region proposals, a concurrently developed Faster R-CNN detector with end-to-end 3D regression, and a computationally more complex articulated 3D human pose estimation method.

- **Chapter 6** shows that a recent deep learning-based wheelchair and walker detector for 2D range data by Beyer et al. (2016) also generalizes to detecting legs of people. By providing a fixed-size temporal window of past frames as input to the CNN, performance gains are obtained under a mid-level temporal fusion scheme. On an existing large-scale 2D laser dataset recorded in an elderly care facility, we present an extensive comparison against several classical methods that operate on handcrafted features and explicitly model leg movements using a Kalman filter. We carefully tuned all methods on the same data using automatic hyperparameter optimization for a fair comparison, and find that the proposed method outperforms all baselines, which also show specific strengths and weaknesses. However, when applying the methods to our application domain of intralogistics, we see that a classical Kalman-filter based tracking approach comes surprisingly close to the deep learning method, given the lack of additional training data. When considering further state-of-the-art methods for detection in 3D lidar that have been trained on existing autonomous driving datasets, we observe significant failure modes especially in cluttered and narrow environments in our cross-modal evaluation. This indicates a need for more domain-specific datasets in robotics, or methods that generalize better with less data.

- **Chapter 7** takes a closer look at humans, by performing fine-grained recognition of human attributes from (RGB-)D point clouds. Based upon earlier work by Spinello, Luber, et al. (2011) on human detection, we propose a tessellation-boosting approach for human attribute classification and train it on a large and novel RGB-D dataset annotated with several human attributes. The method uses Adaboost to select the most informative geometric features and 3D spatial volumes in which to compute these features for a particular human attribute at hand. Later, we also incorporate color cues that boost performance on certain attributes. The resulting classifier is highly efficient, and we demonstrate that it outperforms a HOG-SVM and two early deep learning-based baselines, while being suitable for robots without GPU acceleration. We present novel insights from in-the-wild experiments and discuss possible future extensions to the method.

- **Chapter 8** introduces a modular, ROS-based architecture for a multi-modal human detection and tracking framework. While past systems have been monolithic to a large extent, we define message interfaces and implement a common set of tooling to make components reusable across different mobile robots and sensor setups. The usefulness of this modular architecture for research in robot perception has been demonstrated by

deploying the tracking framework on several mobile robots across different application domains, including service robotics, social robotics, intralogistics and autonomous driving use-cases. The framework has also enabled further research by other researchers.

- **Chapter 9** documents further important system contributions by the author to a successful final demonstration of the SPENCER project at Amsterdam-Schiphol airport, including a low-level collision avoidance module for increased safety, a graphical user interface for human-robot interaction, a simple method for extrinsic sensor calibration, and significant parts of the overall software architecture. We compare SPENCER to related robots that are human- or socially-aware and share important lessons learned during the project.
- **Chapter 10** draws overall conclusions from the research presented within this thesis, and highlights remaining open issues and possible future research directions.
- **Appendix A** provides an overview of the novel datasets for multi-modal human detection, tracking and human attribute recognition that have been introduced this thesis.

1.4 EU Projects

The research in this thesis was conducted within – and aligned with – two publicly funded research projects. To provide further context to the research described in this thesis, we will briefly outline both projects in the following.

1.4.1 SPENCER

The aim of the EU FP7 research project SPENCER (2013–2016), short for "social situation-aware perception and action for cognitive robots", was to develop algorithms for service robots that could guide groups of people through highly dynamic and crowded pedestrian environments, such as airports or shopping malls, while behaving in a socially compliant manner, for example by not crossing in between groups of people that belong together. Possible situations involving humans that such a robot might encounter are visualized in Figures 1.2 and 1.3. To this end, robust and computationally efficient components for the perception of humans in the robot's surroundings needed to be developed.

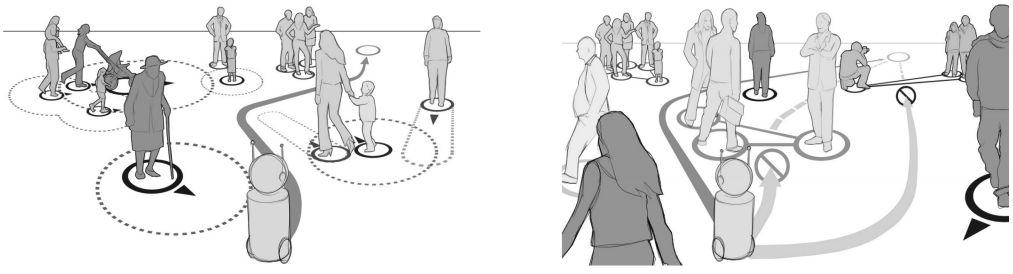


Figure 1.2: Typical situations encountered by a mobile service robot in crowded pedestrian environments, such as shopping malls or airports. To be able to behave in a socially compliant way while driving, by for instance not crossing through a group, the robot needs to gain a precise understanding of the persons in its environment.

(a) Keeping track of a group of passengers / Adapting driving speed

(b) Interacting with groups of users after detecting their presence

(c) Respecting social conventions / Not moving against the flow

Figure 1.3: Possible socially-compliant behaviors of a human-aware service robot. (Photo credit: KLM / SPENCER project consortium)

An international terminal at Amsterdam's Schiphol airport was chosen by the project consortium as an examplary use-case for data collection and final demonstration of SPENCER. Important challenges encountered in this environment have already been outlined at the beginning of this chapter. A detailed report of the robot deployment at Schiphol airport, as well as lessons learned, can be found in Chapter 9.

The multi-modal perception system of SPENCER comprises human detection and tracking modules (see Chapter 4 and Chapter 8), group tracking (Chapter 3), human attribute recognition (Chapter 7), head pose estimation (Beyer et al., 2015), and low-level obstacle detection (Chapter 9). These modules serve as an input to several other software components developed within the project, including human-aware motion planning (Palmieri et al., 2017), learning of socially normative behaviors (Okal and Arras, 2016), human-robot interaction via adaptation of the

robot's head direction (Khambhaita et al., 2016; Joosse, 2017) and via a graphical user interface on the touchscreen (Chapter 9), as well as situation assessment and person guidance (Fiore et al., 2015). An overview of the entire system is provided by Triebel et al. (2016).

1.4.2 ILIAD

The ongoing EU Horizon 2020 project ILIAD ("Intra-Logistics with Integrated Automatic Deployment: Safe and Scalable Fleets in Shared Spaces"; 2017–2021) is focused on the intralogistics domain, where shorter product life cycles and quickly changing market trends lead to demands for more flexible, highly reliable, self-optimizing, quickly deployable and safe yet efficient systems in environments shared with humans. The project aims at developing robotic solutions that integrate with existing warehouse facilities, extending the state-of-the-art to achieve[2]:

- Self-deploying fleets of heterogeneous robots in multiple-actor systems;
- Life-long self-optimisation;
- Manipulation from a mobile platform;
- Efficient and safe operation in environments shared with humans; and
- Efficient fleet management with formal guarantees.

The particular focus of the author of this thesis, as work package leader for *WP3: Human-aware AGV Fleets*, was on implementing and deploying a multi-modal human detection and tracking pipeline on a heterogenous fleet of autonomous guided vehicles (AGVs), comprising four smaller pallet trucks and two larger forklifts depicted in Figure 1.4.

As shown in Figure 1.5, experiments and demonstrations have been conducted in storage areas of two food factory environments in the UK and Sweden. While these environments are under normal circumstances less crowded by humans than in SPENCER, they pose some additional, unique challenges:

- Unusual and potentially look-alike appearance of warehouse workers (wearing safety vests or protective clothing in the food industry), making it hard to distinguish, segment, detect and track them in a robust fashion in case of occlusion,
- A heterogeneous *fleet* of robots with different sensor setups and compute power,
- Higher potential safety risks due to heavy weight and large footprint of the platform (especially when carrying a pallet), while operating at larger velocities (up to 8m/s for a large forklift, compared to 1.5m/s for SPENCER).

In ILIAD, the human detection and tracking framework from this thesis provides necessary inputs to a safety module (Mansfeld et al., 2018), for qualitative representations of human-robot spatial interaction (Fernandez-Carmona et al., 2019) that enable human-aware motion planning, and for the creation of long-term spatio-temporal flow maps that capture environment dynamics (Swaminathan et al., 2018; Molina et al., 2019).

[2]This synposis originates from the project website, http://iliad-project.eu (visited in 07/2019).

Figure 1.4: Two different types of robots used in the ILIAD project. *Left:* Smaller Linde Cititruck pallet trucks, equipped with an autonomous navigation system by Kollmorgen, a Kinect v2 RGB-D camera, a 2D safety laser, and a Velodyne VLP-16 lidar. *Right:* The Toyota BT truck platform, equipped with a Kinect v2 and a Velodyne HDL-32E lidar in addition to a 2D safety laser.

(a) Milestone 2 stakeholder meeting at UK site

(b) Milestone 3 stakeholder meeting at Swedish site (human tracking demo)

Figure 1.5: Live demonstrations of the integrated human detection and tracking system during different milestones in ILIAD, running in real-time on two different types of robots. On the right side of (b), we show that the big forklift in the background is able to robustly detect and track a human lying on the floor. (Photo credit: ILIAD consortium, Martin Magnusson, Tomasz Kucner)

1.5 Publications

1.5.1 Peer-reviewed conference proceedings

The work presented in this thesis led to the following peer-reviewed publications that have been (co-)authored by the author of this thesis:

- **Multi-Model Hypothesis Tracking of Groups of People in RGB-D Data**
  Timm Linder and Kai Oliver Arras.
  International Conference on Information Fusion (FUSION), 2014.
- **Real-Time Full-Body Human Gender Recognition in (RGB)-D Data**
  Timm Linder, Sven Wehner and Kai Oliver Arras.
  IEEE International Conference on Robotics and Automation (ICRA), 2015.
- **Real-Time Full-Body Human Attribute Classification in RGB-D Using a Tessellation Boosting Approach**
  Timm Linder and Kai Oliver Arras.
  IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.
- **SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports**
  Rudolph Triebel, Kai Oliver Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, Hayley Hung, Omar A. Islas Ramírez, Michiel Joosse, Harmish Khambhaita, Tomasz Kucner, Bastian Leibe, Achim J. Lilienthal, Timm Linder, Manja Lohse, Martin Magnusson, Billy Okal, Luigi Palmieri, Umer Rafi, Marieke van Rooij, Lu Zhang.
  Conference on Field and Service Robotics (FSR), 2015.
- **On Multi-Modal People Tracking from Mobile Platforms in Very Crowded and Dynamic Environments**
  Timm Linder & Stefan Breuers (equal contrib.), Bastian Leibe and Kai Oliver Arras.
  IEEE International Conference on Robotics and Automation (ICRA), 2016.
- **Accurate Detection and 3D Localization of Humans using a Novel YOLO-based RGB-D Fusion Approach and Synthetic Training Data**
  Timm Linder, Kilian Y. Pfeiffer, Narunas Vaskevicius, Robert Schirmer, Kai Oliver Arras.
  IEEE International Conference on Robotics and Automation (ICRA), 2020.

While working on this thesis, the author received a Best Reviewer Award at the IEEE International Conference on Robotics and Automation (ICRA) 2019 in Montréal, Canada.

This thesis does *not* report on the following collaboration, which is outside of its scope:

- **Metric-Scale Truncation-Robust Heatmaps for 3D Human Pose Estimation**
  István Sárándi, Timm Linder, Kai Oliver Arras and Bastian Leibe.
  IEEE International Conference on Automatic Face & Gesture Recognition (FG), 2020.

1.5.2 Peer-reviewed journal articles

The following co-authored journal article includes research conducted within the scope of this thesis (see Chapter 6):

- **Deep Person Detection in Two-Dimensional Range Data**
  Lucas Beyer, Alexander Hermans, Timm Linder, Kai Oliver Arras and Bastian Leibe.
  IEEE Robotics and Automation Letters (RA-L), 2018 with presentation at *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018*.

1.5.3 Peer-reviewed workshop proceedings

The research in this thesis led to the following peer-reviewed workshop papers:

- **Towards a Robust People Tracking Framework for Service Robots in Crowded, Dynamic Environments**
  Timm Linder, Fabian Girrbach and Kai Oliver Arras.
  Assistance and Service Robotics (ASROB-15) Workshop at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.
- **Towards Accurate 3D Person Detection and Localization from RGB-D in Cluttered Environments**
  Timm Linder, Dennis Grießer, Narunas Vaskevicius and Kai Oliver Arras.
  Robotics for Logistics in Warehouses and Environments Shared with Humans Workshop at the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS), 2018.
- **Towards Training Person Detectors for Mobile Robots using Synthetically Generated RGB-D Data**
  Timm Linder, Michael Johan Hernandez Leon, Narunas Vaskevicius and Kai Oliver Arras.
  3D Scene Generation Workshop at the IEEE Conf. on Computer Vision and Pattern Recognition (CPVR), 2019.

This thesis does *not* report on the following co-authored workshop papers:

- **Synthetic Occlusion Augmentation for 3D Human Pose Estimation with Volumetric Heatmaps** (ECCV 2018 PoseTrack 3D Challenge Winner)
  István Sárándi, Timm Linder, Kai Oliver Arras and Bastian Leibe.
  PoseTrack Challenge Workshop at the European Conference on Computer Vision (ECCV), 2018.
- **How Robust is 3D Human Pose Estimation to Occlusion?**
  István Sárándi, Timm Linder, Kai Oliver Arras and Bastian Leibe.
  Robotic Co-workers 4.0: Human Safety and Comfort in Human-Robot Interactive Social Environments. Workshop at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.

1.5.4 Peer-reviewed book chapters

The multi-modal human detection and tracking framework (Chapter 8) is described in the following book chapter:

- **People Detection, Tracking and Visualization using ROS on a Mobile Service Robot**
  Timm Linder and Kai Oliver Arras.
  Robot Operating System (ROS): The Complete Reference (Vol. 1). Springer Studies in Systems, Decision and Control, 2016.

1.5.5 Further media dissemination

As part of the dissemination of the research to the general public, the author's contributions were featured in several German television broadcasts, including:

- **KLM marketing video** (April 2016): This video used in social media marketing highlighted the benefits of a socially-aware robot from the viewpoint of a potential end-customer, with a particular focus on human-robot interaction. The graphical user interface of the SPENCER robot developed by the author was shown prominently.
- **SWR Landesschau** (April 22, 2016): As part of a news broadcast entitled "Ein Roboter mit Sozialkompetenz" (a robot with social expertise), the SPENCER robot was introduced and navigating through a crowd of students in building 101 at the University of Freiburg.
- **3sat nano** (April 29, 2016): In a longer documentary based upon the same material, entitled "Roboter unter Menschen – SPENCER weist den Weg" (robots among humans – SPENCER guides the way), challenges of the complex environment inside an airport terminal related to person detection and tracking were highlighted.
- **Tigerentenclub** (October 30, 2016): As part of this German children's television programme, the thesis author and the project coordinator had been invited to introduce the SPENCER project. As shown in Figure 1.6, the robot autonomously guided children through a TV studio which it had never been trained upon, showcasing its human detection, tracking and navigation capabilities as well as the graphical user interface.

The multi-modal human detection and tracking system for AGV fleets developed in ILIAD has also been presented by the author to interested participants from the logistics and food industry sectors during two stakeholder meetings (shown in Figure 1.5).

1.5.6 Open-source software

The following open-source software packages have been released on GitHub:

- **SPENCER people tracking framework** (Linder et al., 2016): The multi-modal human detection, tracking and group tracking framework that was deployed on the SPENCER robot and is described in Chapter 8. Forked 241 and starred 387 times as of 12/2019. Initially developed for ROS Hydro, this has subsequently been updated to ROS Indigo, Kinetic, Melodic and Noetic with the help of contributions from the open-source community.

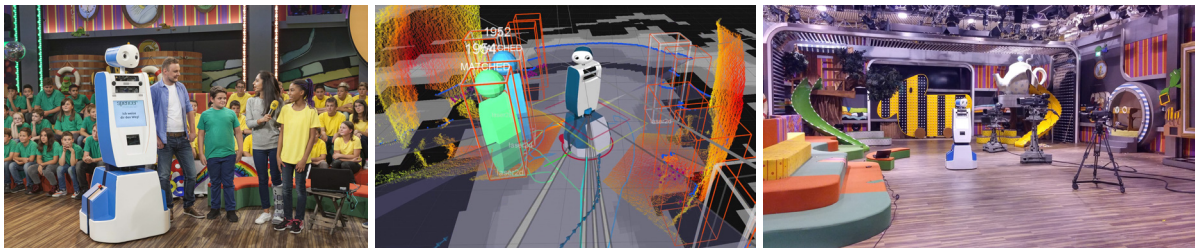


Figure 1.6: TV coverage of the SPENCER project, featuring a visualization of the human tracking framework from Chapter 8. (Source: SWR Presse/Fotoredaktion, own material)

- **SPENCER human attribute recognition** (Linder, 2015): Implementation of the tessellation-boosting classifier described in Chapter 7. Forked 22 and starred 29 times.

Further, smaller contributions have been made to various open-source ROS components, such as the resolution of bugs and implementation of new features in `rosmaster`, `rviz`, `robot_model` and `laser_geometry`.

1.5.7 Published datasets

In Appendix A, several novel datasets are introduced that have been used for the research throughout this thesis. Overall, the author performed or supervised the recording and partial annotation of several weeks worth of multi-modal raw sensor data, in total accounting to around 2.5 TB when camera images are stored in a compressed format.

For reproducibility of results and to advance the state of the art more quickly, it is of high interest for such datasets to be made available to the scientific community. In robotics, this is unfortunately not always possible when datasets are recorded outside of a lab environment, especially when operating on foreign property like an airport or in a commercially operated warehouse. In such cases, concerns related to security and protection of intellectual property need to be taken into account; when humans are the subject, as is the case in this thesis, privacy and works council regulations play an even bigger role. If at all, it is often only possible to publish recorded image data in a strongly anonymized form, which can significantly affect the performance of algorithms (for instance, due to person detectors overfitting to blurry regions).

The following datasets are publicly available upon request. The real-world datasets have been recorded in controlled environments, where subjects have expressed their explicit consent for the datasets to be published for scientific research purposes:

- **SRL Human Attributes Dataset** (Linder, Wehner, et al., 2015b): Access has been granted so far to around 20 different research groups.
- **Toulouse Motion Capture Dataset** (Linder et al., 2016): No statistics available regarding number of accesses (hosted by RWTH Aachen University).
- **Synthetic PedSim dataset** (Linder, Girrbach, et al., 2015)

See Appendix A for a more detailed description of these datasets.

1.6 Collaborations

Data-driven in its nature, research in robot vision is a time-intensive team effort. It often involves complex sub-tasks such as data collection and annotation, implementation and evaluation of baseline methods, integration with other modules such as motion and task planning, as well as deployment on real robots for evaluation and demonstration purposes. The research in this thesis would not have been possible without the following valuable contributions and collaborations:

- Research in **Chapter 4** on multi-modal human tracking in crowded environments is joint work with Stefan Breuers, who contributed equally to the ICRA'16 paper and focused on improving and tuning the MDL tracking algorithm as well as the upper-body and groundHOG detectors. The nearest-neighbor tracker, the partial re-implementation and ROS integration of the HO-MHT and the 2D laser detector, the detection-to-detection fusion pipeline, and the trajectory-based 3D annotation tool were implemented by me. Both first authors were involved in the data recording at Schiphol airport to equal parts. My master student Fabian Girrbach helped with the implementation of the IMM, track initiation logic, and hyperparameter optimization via SMAC.
- My master student Michael Johan Hernandez Léon helped in setting up the initial six scenes of the synthetic RGB-D dataset in Unreal Engine 4, described at the beginning of **Chapter 5**, and prepared the export scripts for 2D experiments in Linder et al. (2019). Underlying 3D assets were previously obtained by me from the Unreal Marketplace. Michael also implemented the Kinect v2 time-of-flight sensor noise modeling. The later extensions encompassing further scenes, human models, animations and foreground occluder objects, per-joint 3D groundtruth, and stronger domain randomization were implemented by me. I conducted all YOLO v3 2D experiments on the GPU cluster.

  The open research challenge of robustly detecting and localizing humans in 3D under occlusion was identified by me in initial experiments (Linder et al., 2018). I proposed to learn 3D localization from synthetic RGB-D data with accurate 3D groundtruth. Several concepts for YOLO v3 RGB-D fusion and weights transfer were originally developed in collaboration with my master student Dennis Griesser and my colleague Narunas Vaskevicius from Bosch Corporate Research. All three of us annotated the image-based real-world intralogistics dataset with 2D bounding boxes, while 3D centroid annotations were added by me after extending the trajectory annotation tool for that purpose. Experiments on real-world data and integration of existing baselines were performed to equal parts by Narunas and me. Our intern Kilian Yutaka Pfeiffer contributed to the novel 3D RGB-D human detector based upon YOLO v3. Narunas and he jointly came up with the depth-aware augmentation scheme. Under supervision by Narunas and me, Kilian implemented the mid-level fusion scheme, extended the loss function for 3D centroid regression, and designed and conducted ablation studies within a Python-based experimentation framework that he devised.
- In the joint RA-L article of Beyer et al. (2018), which constitutes a part of **Chapter 6**, Lucas Beyer and Alexander Hermans focused on improving the DROW algorithm and extending it to the person class, while I focused on the integration, evaluation and hyperparameter optimization of all other considered baseline methods. The concept to incorporate temporal information into DROW by exploiting multiple subsequent frames to improve detection

performance, as well as parts of the experimental design have been jointly developed. My colleague Robert Schirmer from Bosch Corporate Research helped with labeling of 3D groundtruth and training of the 3D lidar detectors for the cross-modal comparison.

- My master students Sven Wehner and our student assistant Marta Timón helped in the acquisition of the Human Attributes Dataset from **Chapter 7**. Sven Wehner also evaluated the two baseline methods that were used in Linder, Wehner, et al. (2015a).
- The recording of the real-world intralogistics datasets listed in **Appendix A** would not have been possible without the help of the ILIAD consortium partners from Örebro University, University of Lincoln, NCFM and Orkla. Special thanks go to Martin Magnusson, Chittaranjan Swaminathan and Manuel Fernandez-Carmona for setting up the robots. Daniel Adolfsson helped with extrinsic calibration of the 3D lidar on all AGVs, which allowed me to calibrate the remaining 2D and RGB-D sensors.

Finally, the approaches presented in this thesis have also been utilized in the following related SPENCER publications:

- An ICRA paper by Okal and Arras (2016) on learning socially normative robot navigation behaviors, in which the group tracking system presented in Chapter 3 was used for real-world experiments on the DARYL robot.
- A RA-L article by Palmieri et al. (2017) on finding diverse paths for robot navigation, in which the human detection and tracking framework described in Chapter 8 was used for experiments with the DARYL robot.
- In the PhD thesis by Joosse (2017), several user studies on the user acceptance of the SPENCER robot are presented. In one study on the robot's head turning behavior (pp. 125–138), conducted at the University of Freiburg, inputs from human tracking were used to compute distances and bearings to tracked group members, and to adapt the head turning behavior accordingly. The graphical user interface, task planning, collision avoidance, and the integration of human tracking with motion planning and group guidance were also used in these experiments. These components, which are described in Chapter 9, have also been used in the final SPENCER user study at the airport (Joosse, 2017, pp. 139–153).

CHAPTER 2

Online multi-object tracking: A brief overview

In this chapter, we introduce the different sub-problems of multi-object tracking, namely data association, state estimation and choice of object representation. We give a brief overview of the most important families of *online* methods to solve the tracking problem, and highlight which ones are most relevant for this thesis. Lastly, we explain and discuss important evaluation metrics that will be used in the following chapters.

For further details on the described methods, we point out relevant text books, theses and articles, like the books by Bar-Shalom et al. (1995) and Bar-Shalom et al. (2011). They provide a general overview of many classical, model-based techniques for multi-target tracking and sensor fusion, most of which originate from radar-based tracking of airborne targets. Pulford (2005) provide a taxonomy of such methods, while surveys by Yilmaz et al. (2006) and Luo et al. (2014) focus on video-based object detection and tracking. The benchmark by Leal-Taixé et al. (2017) and recent survey articles by Ciaparrone et al. (2019), Xu et al. (2019), and Fiaz et al. (2019) include also deep learning-based approaches. The book chapter by Bellotto et al. (2018) on human detection and tracking for robotics discusses also other sensors like RGB-D and lidar. Like this thesis, they put an emphasis on methods suitable for real-time robotics applications.

2.1 Tracking paradigms

The goal of multi-target tracking is to obtain robust estimates of the trajectories of multiple targets over time, while maintaining consistent identities of these targets.

The two prevalent paradigms in multi-object tracking, shown in Figure 2.1, are *tracking-by-detection* and *model-free tracking*. Our focus in this thesis is on the first approach, where one or multiple object class-specific detectors first aim to recognize the presence of all class instances in a given scene, usually on a frame-by-frame basis. In the subsequent tracking step, detected objects of the target classes are tracked over time. Bellotto et al. (2018) and Section 4.2 provide many robotics examples for this type of approach. Some authors propose to additionally feed back information from tracking to detection, *e. g.* to learn online instance-specific detectors (Luber, Spinello, et al., 2011).

Instead, more generic model-free tracking approaches (*e. g.* Luber et al., 2009; Teichman et al., 2011; Ošep et al., 2018) first segment the scene, for instance based upon spatial clustering criteria (*e. g.* Bogoslavskyi et al., 2017) or motion-based features (*e. g.* Dewan et al., 2016).

(a) Tracking-by-detection **(b)** Model-free tracking

Figure 2.1: Two basic tracking paradigms

Then, they track all kinds of objects that result from these segmentation criteria, without being aware of their specific object type. In the final step, where also historic track information from previous frames can be taken into account, tracked objects can get classified. We will briefly discuss such approaches in the final chapter. They have not been a focus of this thesis because our goal is to detect, track and analyze specifically humans to help make robots socially aware.

A third, more recent paradigm is *(deep) end-to-end tracking*, which combines detection and tracking into a single network that is learned end-to-end. So far, only few approaches follow this paradigm (*e. g.* Ondruska et al., 2016; Dequaire et al., 2018).

2.2 Object representations

In their survey on visual object tracking, Yilmaz et al. (2006) distinguish between five general types of object shape representations that might be used for tracking purposes: Points, primitive geometric shapes, object silhouettes/contours, part-based shape models, and skeletal models. While this thesis considers methods from multiple sensor modalities and does not focus on image data in particular, their categorization is still useful to us. The general goal in this thesis is to track objects of the first, *point*-based kind: For many robotics applications, knowledge of the *centroid* position of a human is sufficient, *e. g.* for socially-aware motion planning, or particular human-robot-interaction applications that we describe in Section 9.6.

Instead, typical examples of *primitive geometric shapes* are 2D and 3D bounding boxes, which we use at several places for detection purposes in Chapter 5 and 6. Works that jointly track and segment humans (Milan et al., 2015; Voigtlaender et al., 2019) output human *silhouettes* in the form of instance segmentation masks. While we do not track such masks, we show single-frame instance segmentation failure cases in Section 5.4. We briefly discuss *part-based models* in the form of *poselets*, which were originally proposed for human detection (Bourdev et al., 2010), as a related method for human attribute recognition in Chapter 7. *Skeletal models* are not the focus of this work, though we compare one of our detection methods to a skeleton-based approach in Section 5.7.1 and leverage synthetic skeletal groundtruth in the same chapter. We discuss the relationship between human detection and articulated human pose estimation in Section 10.2.

On a higher level, Granström et al. (2017) distinguish between *point object tracking*, *extended object tracking*, and *group tracking*. According to their definition, a point object only produces a single measurement (detection) per time step. In extended object tracking, each tracked object is still a single entity – for instance, a human –, but generates multiple measurements (*e. g.* detected body joints), and we might be interested in tracking its shape. Instead, in group tracking, each tracked object is a collection of individual sub-objects that share common dynamics, while allowing for additional intra-group dynamics. An example are groups of pedestrians, which we discuss in the next chapter.

2.3 Data association

Data association describes the problem of associating uncertain measurements, *i. e.* detections, to known tracks. It thus addresses the uncertainty of the origin of the measurements. In multi-target tracking, this is challenging because of false alarms and missing measurements, noisy measurement localization, and ambiguities that can only be resolved over time. Data association may involve track life-cycle management, *i. e.* track initiation and termination, where the challenge is to decide if a measurement originates from an existing track, a new one, or a false alarm; and likewise, if a tracked target has left the scene, or is just temporarily occluded.

There are many different ways of categorizing data association methods. In general, we can distinguish between *single-scan* methods that do not consider past measurements for decision-making, and *multi-scan* methods that may revisit past measurements and thus revise earlier decisions on the basis of new evidence (Pulford, 2005; Oh et al., 2009).

We now briefly introduce the most common families of approaches in the target tracking community (Vo et al., 2015; Granström et al., 2017) – JPDA, MHT and RFS – and further methods that are frequently or increasingly being used in robotics, namely nearest-neighbor methods and deep learning approaches.

The *nearest-neighbor (NN) standard filter*, and the *global nearest-neighbor (GNN)* approach are among the simplest, computationally least complex, single-scan methods. The first is a greedy non-deterministic, heuristic approach, while the second computes a maximum-likelihood solution by finding a unique, joint assignment which minimizes the total cost of measurement-to-track association. Both make "hard" assignment decisions and consider only measurements from the current time step. We will use a GNN method for experiments in Chapters 3 and 4.

The *joint-probabilistic data association filter (JPDAF)* (Bar-Shalom, 1987) is a Bayesian single-scan approach. At every time step, it computes a single joint state hypothesis by weighting all measurements by their association probabilities, which are computed by exhaustively enumerating all possible associations. Thereby, it makes soft assignments. However, the computation of all possible association probabilities is NP-hard, such that heuristic approximations are used in practice. The basic variant of JPDAF assumes the number of targets to be known and performs no track life-cycle management. Rezatofighi et al. (2015) re-visited the technique and achieved competitive, real-time performance using approximation techniques based upon recent developments in integer linear programming, by computing only the k strongest association hypotheses. They also extended the approach to a multi-scan method.

Multi-hypothesis tracking (MHT) (Reid, 1979) instead tracks all possible association hypotheses over time and integrates track management into a Bayesian framework. Being a multi-scan method, it defers final decisions to a later point in time at which more information is available. The method maintains a hypothesis tree, which keeps track of association history; in every tracking cycle, parent hypotheses give rise to a number of child hypotheses that represent alternative interpretations of the current set of measurements. The leaf hypothesis with the highest posterior is returned as the current best solution. As stated by Cox and Hingorani (1996), the principal disadvantage of the approach is its computational complexity. It arises from the combinatorial explosion of possible associations with a large number of targets.

In *hypothesis-oriented MHT (HO-MHT)*[1] as per Reid (1979), every single generated hypothesis represents a global, joint association of all measurements and tracks over a given number of scans. Therefore, several approximations are required in practice (Cox and Hingorani, 1996) which make the method lose its theoretical optimality in a maximum-a-posteriori sense. We use HO-MHT in Chapters 3 and 4.

The *track-oriented MHT (TO-MHT)* by Kurien (1990) re-builds a track hypothesis tree for each track individually at every tracking cycle and is recently being used more often because it is easier to implement and requires fewer hypotheses (Papageorgiou et al., 2009). A modern variant of this approach that integrates discriminative appearance models to make track hypotheses more distinctive achieved competitive results in computer vision benchmarks (Kim et al., 2015). In Chapter 4, we consider a baseline method that follows similar principles as TO-MHT.

Related to this, researchers have explored sampling-based approximation techniques based upon *Markov chain Monte Carlo data association (MCMCDA)*. Oh et al. (2009) propose a method whose single-scan variant approximates JPDAF with a fixed number of targets, while its multi-scan variant incorporates track life-cycle management and approximates an optimal Bayesian filter. They show that it can outperform HO-MHT if using weak detectors in dense environments.

Random finite set (RFS) methods based upon the theory of *finite set statistics (FISST)* (Mahler, 2007; Mahler, 2014) are gaining increasing popularity and are today among the three most popular tracking approaches. Different from the previous approaches, they do not perform explicit data association. Instead, they circumvent the combinatorial problem by only propagating density measures derived from the measurements. The *probability hypothesis density (PHD) filter* (Mahler, 2003), for example, approximates the Bayesian approach to multi-target tracking by recursively propagating only the first-order moment of the full multi-target posterior density. As it does not perform explicit data association, it can only provide likely object positions without assigning target identities. To deal with this issue, different extensions and alternatives such as the Labeled Multi-Bernoulli Filter (Reuter et al., 2014) have been proposed.

Deep learning approaches are the most recent family of methods. This field of research is still in its infancy and there is not yet a commonly accepted formulation of data association and the overall tracking problem. The end-to-end approach of Ondruska et al. (2016) does not perform explicit data association; instead, it operates on occupancy grids and predicts future occupancy without tracking individual target identities using a recurrent neural network (RNN). Milan et al. (2017) are one of the first to model an explicit data association step for multi-object tracking within a deep neural network. They combine an RNN for state estimation with a long short-term memory unit (LSTM) for data association, and achieve performance comparable to a simple nearest-neighbor approach. Xu et al. (2020) propose a differentiable framework for deep multi-object tracking, including a Deep Hungarian Net to solve the assignment problem, and differentiable loss functions inspired by multi-object tracking metrics (see Section 2.5.3).

[1] Bar-Shalom et al. (1995) and Pulford (2005) use the term *measurement-oriented* MHT.

		Groundtruth class	
		Human	Non-Human
Predicted class	Human	True Positive	False Positive
	Non-Human	False Negative	True Negative

Figure 2.2: Confusion matrix for binary classification

In general, many further methods developed in the computer vision literature step away from the recursive filtering approach, and instead perform global optimization over all sets of measurements of an entire sequence. Since such batch methods do not work in an online fashion, they are not suitable for real-time robotics applications and thus not considered here.

2.4 State estimation

Before being able to perform data association, we need to predict track states from the previous time step into the current. After data association, track states need to be updated with the associated new measurements. The most common techniques to solve this state estimation problem, given a sequence of measurements, are recursive estimators in the form of Kalman filters, particle filters, and interacting multiple-model (IMM) approaches. Bar-Shalom et al. (2001) give an overview of the estimation topic with applications to tracking.

In many practical use-cases, human walking behavior can reasonably well be explained through a linear, *constant velocity (CV)* model on a short time scale. Because in our tracking framework, we transform all incoming detections into a common, fixed world frame, the robot's own motion does not need to be modeled if odometry is available at a high frequency. Therefore, a standard Kalman filter with sufficient amount of process noise to account for dynamics is often sufficient. To model more complex motions, the IMM approach allows to combine multiple weighted models into a single Bayesian framework in which all models are updated simultaneously. Switches of the currently active model occur on the basis of pre-defined transition probabilities.

2.5 Evaluation metrics

We now introduce several measures that are used in experiments throughout this thesis for the quantitative evaluation of classification, detection and tracking methods.

2.5.1 Binary classification metrics

The confusion matrix in Figure 2.2 shows the four different outcomes when evaluating a binary classifier – in this case for the human class – on a single sample. From the total number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) across all samples, we can derive aggregate measures

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \quad \text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \tag{2.1}$$

in the range $[0.0; 1.0]$, where larger is better. *Accuracy* is only a sensible measure when the classes in the dataset are well-balanced[2]. The balanced *F-measure* is defined as the harmonic mean of equally-weighted precision and recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.2}$$

Many binary classifiers output a probability for a sample being of the positive class. By varying the decision threshold, we can obtain *precision-recall curves* that describe the classifier's behavior at different set points, that each represent a specific trade-off between precision and recall. At *equal-error rate (EER)*, precision equals recall, and the curve intersects the 45° diagonal. *Peak-F_1* is the maximum F_1 measure obtained when considering all possible decision thresholds. As another overall quality measure, we can compute the *area under the curve (AUC)*.

2.5.2 Object detection metrics

The two most popular 2D object detection benchmarks in computer vision in the last decade were the Pascal VOC challenge (Everingham et al., 2010) and the COCO challenge (T.-Y. Lin et al., 2014). Both use *average precision (AP)* as their main evaluation metric, which approximates the area under the precision-recall curve. To compute the integral, the latest version of VOC AP samples the curve at every unique recall value, whereas COCO AP interpolates over 101 uniformly distributed recall thresholds.

In addition to correctly classifying objects, a good object detector should also *localize* them accurately in the scene. Correctly evaluating detection is more difficult than pure classification, because we have to solve a many-to-many assignment between detected and groundtruth objects before we can count correctly and incorrectly assigned class labels (see Figure 2.2) and compute precision-recall curves. For objects represented as 2D bounding boxes, we therefore first compute the *intersection-over-union (IoU)* for all possible pairings and only retain those that exceed a given IoU threshold. IoU measures the relative overlap between two boxes. We then iterate over detections in descending order of confidences, and find the groundtruth with the highest IoU.

VOC AP uses a fixed IoU threshold of 0.5 and is thus not very sensitive to localization inaccuracies. Instead, COCO AP is computed and averaged over ten equally-spaced IoU thresholds in the interval $[0.5; 0.95]$. At higher IoU levels, already small discrepancies in location or extents of the bounding box can prevent a matching, and thus reduce precision and recall. Therefore, COCO AP is more discriminative than VOC AP if accurate localization is important. One weakness is that without comparing AP at individual IoU thresholds, one can still not make a qualified statement about localization accuracy, because classification and localization errors are intertwined in a single scalar measure. More recently proposed object detection metrics (Oksuz et al., 2018) introduce a distinct measure for localization accuracy, but have not yet been widely adopted.

[2]Therefore we only use it to evaluate binary human attributes after balancing in Chapter 7.

If objects are represented by centroids in metric space, IoU is not well-defined. Instead, we compute the Euclidean distance between groundtruth and detection centroids, and compute AP at multiple metric distance thresholds (*e. g.* 0.25 m, 0.50 m, 0.75 m). For association, we negate the values because smaller distances imply better matching.

2.5.3 Multi-object tracking metrics

The most commonly used measure for the evaluation of multi-object tracking performance are the CLEAR-MOT metrics by Bernardin et al. (2008). By solving a linear assignment problem between predicted and groundtruth tracks, they count for each time step k the number of false positives (FP), false negatives (FN) and track identity switches (IDS). Then, they define an aggregate error measure, MOT Accuracy (MOTA):

$$\mathrm{MOTA} = 1 - \frac{\sum_k (\mathrm{FP}_k + \mathrm{FN}_k + \mathrm{IDS}_k)}{\sum_k \mathrm{GT}_k}. \tag{2.3}$$

The maximum attainable MOTA score is 1.0. MOTA can reach negative values if the tracker makes more errors than there are ground truth objects GT over the entire duration of the dataset. In this work, we do not consider MOT precision (MOTP), which gives a metric estimate of the track localization error, and only makes sense to compute if highly accurate groundtruth is available, for instance from a motion-capture system.

Further trajectory-based measures are *mostly tracked (MT)* and *mostly lost (ML)* tracks. As per Y. Li et al. (2009), they specify the number of groundtruth tracks that have been tracked for more than 80%, or less than 20% of their total length. In user studies, Leal-Taixé et al. (2017) found that, among several tracking measures, MOTA has got the highest correlation with human visual assessment, followed by MT. They also found that differences of five percentage points or less in MOTA or MT are indistinguishable by humans, and the compared tracking methods should thus be considered equivalent.

Issues of CLEAR-MOT

Milan et al. (2013) note that MOTA scores can vary between implementations and are highly dependent on meta-parameters such as the matching distance threshold and the way in which track hypotheses are assigned to groundtruth objects.

A further downside of above definition is that the number of identity switches (IDS) has very low influence on the overall MOTA score, because the absolute false positive and false negative counts are often significantly higher per frame. This problem is discussed by Leal-Taixé et al. (2015), who propose to compute the *relative number of ID switches* (rIDS) as product of IDS and the inverse of the recall over all frames:

$$\mathrm{rIDS} = \mathrm{IDS} \cdot \frac{\sum_k \mathrm{GT}_k}{\sum_k \mathrm{TP}_k}. \tag{2.4}$$

Part II

Tracking people in crowded environments

CHAPTER 3

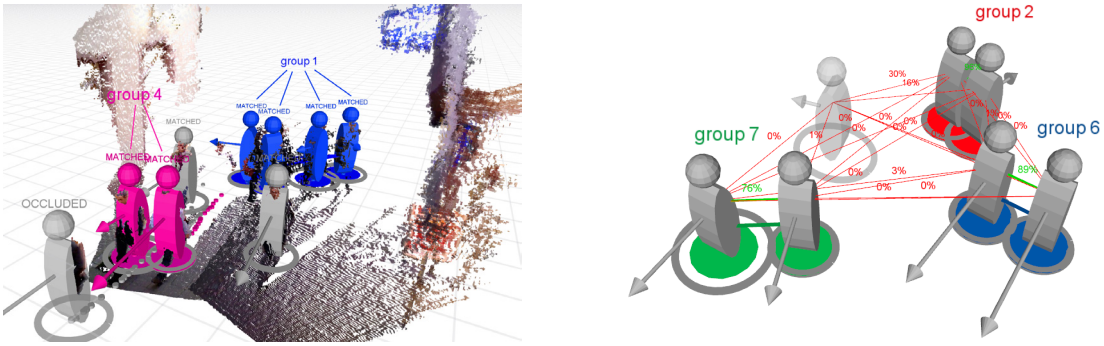
Tracking groups of people in crowded environments

We start this thesis with the question if complex data association methods are suitable for tracking groups of people in general, and in crowded environments in particular. Concretely, in this chapter, we address the problem of joint individual-group tracking using learned pairwise social relations, for example based upon coherent motion indicator features, from a mobile sensor platform in a challenging egocentric perspective. For this, we extend an existing multi-model multi-hypothesis tracking method by Luber et al. (2013), that tracks and reasons about multiple social grouping hypotheses in a recursive way, with a mechanism to maintain consistent group identities across group merges and splits. In qualitative experiments on a novel multi-sensor RGB-D dataset from a moderately crowded pedestrian zone, we achieve good real-time tracking performance for varying group sizes with few identifier switches. We then conduct experiments on simulated data in a more crowded environment, where we examine limitations of the hypothesis-oriented MHT approach under real-time constraints. We also demonstrate the applicability of the method for socially-aware navigation. To facilitate further research, we fully integrated the system with the ROS middleware.

Parts of this chapter have been previously published in the paper "Multi-Model Hypothesis Tracking of Groups of People in RGB-D Data" by T. Linder, K. O. Arras in the Proceedings of the 17th International Conference on Information Fusion (FUSION) 2014 in Salamanca, Spain.

3.1 Introduction

Tracking groups of people is an important skill for surveillance systems, intelligent vehicles and robots that operate in populated environments. Empirical research has found that up to 70% of pedestrians walk in groups (Moussaïd et al., 2010). Knowledge about groups, their position, size, motion state, and social activities can enable systems to gain a deeper understanding of human environments and to provide better services to users. Examples include multi-party human-computer interaction or socially compliant robot navigation among groups of people. The goal in this chapter is to track groups of people from a mobile sensor in a first-person perspective. Unlike stationary overhead cameras in surveillance applications, this is a challenging scenario



(a) Our goal **(b)** Social relationship graph

Figure 3.1: *Left:* Groups of four (blue) and two (pink) persons being tracked by our multi-model hypothesis tracker in RGB-D data from a pedestrian zone. *Right:* A social relationship graph between individual person tracks. Strong social relations (indicating a positive group affiliation) are shown in green, weak relations in red. The relation probabilities are *e. g.* determined by a probabilistic SVM trained on coherent motion indicator features, and incorporated into the group model probability in Equation (3.2). The semi-transparent person is temporarily occluded but still considered in the graph.

because with sensors at human eye-level, people in groups are occluded more frequently and are harder to detect and track reliably as individual targets. In addition to gaining possible insights on social relations between people, group tracking also allows to improve person-level tracking by feeding back the grouping information, for example, to better deal with lengthy occlusions of individual targets.

The problem of tracking groups of people has been addressed using image data as well as 2D range data from on-board laser scanners. To our knowledge, this work is the first to use RGB-D data to this end. We extend an approach by Lau et al. (2009) and Luber et al. (2013) based on a 2D laser-based multi-model multi-hypothesis group tracker with a mechanism to maintain consistent group IDs across multiple group splits and merges and conduct experiments on a novel, unscripted, real-world multi-sensor RGB-D dataset from a mobile platform, as well as in simulated, highly crowded scenarios. As shown in Figure 3.1 (a), the approach jointly tracks both individual persons and groups, while explicitly tracking group formation processes using multiple, alternative social grouping hypotheses in parallel. We have successfully used it in socially-aware navigation tasks. To facilitate further research, we have completely refactored the original code base by Luber et al. (2013) to use the ROS middleware.

Outline This chapter is organized as follows: After the discussion of related work, we describe in Section 3.3 how we detect groups based upon social relations. In Section 3.4 we introduce the multi-model multi-hypothesis tracker and our proposed extensions. Experimental results are given in Section 3.5, and Section 3.6 concludes the chapter.

3.2 Related work

For group detection and tracking, we can distinguish three lines of work, which we will describe in the following.

The first one, typically carried out in the social computing community, is concerned with the understanding of (static) *social situations*. Using interpersonal distance and relative body orientation, Groh et al. (2010) study social situation recognition of standing people from static cameras. Similarly, Cristani et al. (2011) address the problem of social relation recognition in conversation situations. Using interpersonal distance only, they estimate pairwise stable spatial arrangements called *F-formations*. Vascon et al. (2016) present a game-theoretic approach to detect static F-formations based upon human positions and body orientations.

A second group of works addresses *social relation recognition* in still images and video. G. Wang et al. (2010) extract social relations from photographs. They use the knowledge that social relations between people in photographs influence their appearance and relative image position. From the learned models, they are able to predict relationships in previously unseen images. Social relations between film actors in video are estimated by Ding et al. (2011). A social network graph with temporal smoothing is learned using actor occurrence patterns. The approach also allows for changes in social relations over time. Choi et al. (2012) recognize atomic activities of individuals, interaction activities of pairs, and collective activities of groups, jointly, using an energy maximization framework. Q. Sun et al. (2017) leverage findings from domain-based theory in social psychology (Bugental, 2000), and successfully exploit semantic human attributes, like those used in Chapter 7, to recognize 16 types of social relations and 5 different domains of social life on a novel image dataset. They combine visual features extracted from a CNN and low-dimensional attribute-based features. J. Li et al. (2017) present a novel, large-scale image dataset of people in social context, and define a hierarchy of social relationship categories. Their proposed dual-glance model also consider contextual cues, similar to the deep knowledge graph approach that Z. Wang et al. (2018) propose. Goel et al. (2019) propose an end-to-end trainable network that generates social relationship graphs from still images by jointly predicting attributes related to age, gender and clothing along with social relations and domains in a multi-task learning framework.

A third line of works, most related to our context, is concerned with *detecting and tracking groups* from image or range data. In their book chapter, Vascon et al. (2017) provide a broad overview of common definitions and features from sociological sciences for detection and tracking of interacting groups of people. Yu et al. (2009) address the problem of discovery and analysis of social networks from individuals tracked in surveillance videos. A social network graph is built over time from observations of interacting individuals. Social relations between persons in overhead video data are recognized by Pellegrini et al. (2010). They use approximate inference on a third-order graphical model to jointly reason about correct person trajectories and group memberships. Based on learned statistical models on people's behavior in groups, they also perform group-constraint prediction of motion. Leal-Taixé et al. (2011) model social and grouping behavior from tracked individuals in video data using a minimum-cost network flow formulation. Qin et al. (2012) improve tracking of individuals by considering social grouping in a tracklet

linking approach. Using large numbers of hypothetical partitionings of people into groups, solutions are evaluated based on the geometrical similarity of trajectories of individuals with the hypothesized group. Similar to our work, Bazzani et al. (2015) perform joint individual-group tracking. They leverage frame-wise information in a decentralized particle filtering framework, where groups are detected through an a-priori classifier or an online learning approach. Solera et al. (2016) detect and track social groups in crowds from an overhead perspective by performing correlation clustering on individual person trajectories. To learn a socially meaningful clustering rule and obtain group member affinities, they use a structural SVM framework on top of features based upon proxemics, motion causality, trajectory-shape similarity, and shared goals. Setti et al. (2019) recently introduced a novel set of metrics for evaluation of group detection performance taking group cardinalities into account.

With regard to robotics applications, Lau et al. (2009) track groups of people in 2D range data from a mobile robot. A multi-model hypothesis tracking approach is developed to estimate the formation of tracks into groups that split and merge. Groups are collapsed into single states loosing the individual person tracks. Because this reduces data association complexity, runtime performance is improved compared to the standard person-level MHT. Instead, Luber et al. (2013) adapt their approach to joint individual-group tracking; they hypothesize and track social groupings as collections of individual person tracks with group affiliation estimates. This allows to improve person-level tracking by adapting per-target occlusion probabilities and predicting the motion of occluded group members through a constrained particle filter. Experiments are conducted on 2D laser datasets and evaluate the quantitative impact on person-level tracking.

We now describe an improved variant of the joint individual-group tracking approach by Luber et al. (2013), that we use for group tracking experiments on a novel RGB-D dataset and further experiments on a highly crowded scene from a pedestrian simulator.

3.3 Group detection and modeling

3.3.1 What defines a group?

In the field of social psychology, there exist dozens of possible definitions of “what is a group”. According to Forsyth (2019), a group can be defined as “two or more individuals who are connected by and within social relationships”. These social relationships can include family relationships, professional task-related interdependencies, or common interests and goals. They can be long-lasting or short-lived (ad-hoc) in nature. In this work, we adopt this basic definition, and represent groups through a social relationship graph[1]. The edges of the graph denote pairwise social relations, and their weights the probabilities of such relations. Figure 3.1b shows an example social relationship graph that we obtained with our method from real-world data, where estimated strong relations are visualized by green edges.

[1] This was referred to as “social network graph” by Luber et al. (2013), but we adopt here the terminology used in more recent works.

3.3.2 Estimating relations via coherent motion indicator features

Now the question is how to estimate such social relations. Large-scale empirical experiments in crowd behavior analysis and social science (Moussaïd et al., 2010) have shown that so-called *coherent motion indicator features* can indicate group affiliation between people. Concretely, as described by Luber et al. (2013), they are comprised of relative spatial distance, difference in velocity and difference in orientation of two given person tracks i and j:

$$\mathcal{F}^{i,j} = \left(d^{i,j}, |\phi^i - \phi^j|, |v^i - v^j|\right) \tag{3.1}$$

where $d^{i,j}$ is the current Euclidean distance between both person tracks, $v = \sqrt{\dot{x}^2 + \dot{y}^2}$ a track's linear velocity and $\phi = \mathrm{atan2}(\dot{y}, \dot{x})$ the orientation obtained from the current motion direction.

For the purpose of this work, we utilize these low-level motion-based features because they can easily be applied across different sensor modalities and, for instance, do not rely on the availability of high-resolution image data.

Like Luber et al., we derive social relation probabilities $\mathcal{R}^{i,j}$ between individual person tracks i and j using a probabilistic support vector machine (SVM) classifier, obtained by scaling its output scores using the method of Platt (1999), that we train on pairwise features $\mathcal{F}^{i,j}$. As a supervisory signal, pairs of person tracks are labeled as either "socially related" or "not socially related" on a large-scale dataset by human annotators with access to image data for annotation purposes.

Clearly, using coherent motion indicators to predict group affiliations implies that groups are considered simply as collectives of individuals that are spatially close and share a common motion goal. However, note that this approach scales with more available cues such as age, gender, clothing, body pose or other human attributes which we will aim to recognize later in Chapter 7, and that may indicate membership of the same group[2].

3.3.3 Group detection through a social relationship graph

We can now detect groups by constructing and pruning a weighted social relationship graph (as shown in Figure 3.1b) over all current individual person tracks. In particular, after building a graph from the social relation probabilities $\mathcal{R}^{i,j}$ between individual person tracks i and j, we discard all graph edges below a threshold of 0.5, which has been chosen because it retains all pairwise social relation candidates above chance level. We then consider all remaining connected components as socially related, and infer that they should form a group. The key question now is how to robustly track such group formation processes over time. In the next section, we describe how the presented mechanism for group detection can be integrated into a multi-model multi-hypothesis tracker, as proposed by Lau et al. (2009) and Luber et al. (2013).

[2]For instance, more recent work by Goel et al. (2019) exploits visual image data and learns an end-to-end network to generate (unweighted) social relationship graphs which incorporate several of the aforementioned semantic human attributes.

3.4 Tracking groups of people using a multi-model hypothesis-oriented multi-hypothesis tracker

3.4.1 Probabilistic group model

People undergo complex group formations and our goal is to track those formation processes over time. As is common in the literature, we use *merge*, *split* and *continuation* events to model the dynamic nature of group formations. As proposed by Lau et al. (2009), the events are treated as binary operations in the sense that in a single time step, a group may split into only two groups and only two groups may merge into one group. This is a weak assumption even in the case when an entire group enters the sensor field of view at once: single-person groups will be initialized from the new tracks and after their Kalman filters have reached steady state after 4–5 cycles, the single-person groups will correctly merge into one group.

A group $\mathcal{G}_j$ becomes part of the set of *merge* candidates $\mathcal{M}_i$ of group $\mathcal{G}_i$ when the corresponding two components in the social relationship graph become connected (above the probability threshold of 0.5), *i. e.* a group has been detected according to our previous definition. Likewise, when two formerly connected components become disconnected (a group is dissolved), the pair of sub-groups $\mathcal{G}_i', \mathcal{G}_i''$ is added to the set of *split* candidates $\mathcal{S}_i$ belonging to group $\mathcal{G}_i$. Note that, although the chosen threshold is just above chance, this will not cause oscillations between merge and split because the social relation probabilities (edge weights) also influence the group model data likelihood term that will be discussed shortly. Finally, a group belongs to the set of *continuation* candidates $\mathcal{C}$ if it continues as is, without being involved in any split or merge event. $\mathcal{C}$ and all $\mathcal{M}_i$, $\mathcal{S}_i$ are re-computed every cycle for *each* single parent data association hypothesis.

A *group model* $M(t)$ at time t represents the current group formation state, which has evolved over time through continuation, merge and split events of all groups in the scene. Formally, $M(t)$ is a partitioning of the set of all tracks at time t into groups.

3.4.2 Group model probability

We derive the probability of a group model $M(t)$ at time t, conditioned on a parent data association hypothesis Ω^{t-1}, from the probabilities of continuation, merge and split events of its groups. We compute those from prior and data-driven probabilities based upon the social relationship graph. The prior probabilities of the group formation events *continuation* p_C, *split* p_S, and *merge* p_M, can be learned from annotated real-world data sets – as in Lau et al. (2009) and Luber et al. (2013). Note that similar to other priors in the MHT framework, they could be place-dependent (Luber, Tipaldi, et al., 2011b).

Concretely, the probability of group model $M(t)$ conditioned on its parent hypothesis Ω^{t-1} is

$$p(M(t) \mid \Omega^{t-1}) = \underbrace{\prod_{\mathcal{G}_i \in \mathcal{C}} p_C \, p_C^{\mathcal{G}_i}}_{\text{continuations}} \cdot \underbrace{\prod_{\substack{\mathcal{G}_i \in \overline{\mathcal{C}} \\ \mathcal{G}_i', \mathcal{G}_i'' \in \mathcal{S}_i}} p_S \, p_S^{\mathcal{G}_i' \mathcal{G}_i''}}_{\text{splits}} \cdot \underbrace{\prod_{\substack{\mathcal{G}_i \in \overline{\mathcal{C}} \\ \mathcal{G}_j \in \mathcal{M}_i}} p_M \, p_M^{\mathcal{G}_i \mathcal{G}_j}}_{\text{merges}} \tag{3.2}$$

with the data-driven terms

$$\begin{aligned} p_S^{\mathcal{G}_i'\mathcal{G}_i''} &= 1 - \mathcal{R}_{max}^{\mathcal{G}_i'\mathcal{G}_i''} \\ p_M^{\mathcal{G}_i\mathcal{G}_j} &= \mathcal{R}_{max}^{\mathcal{G}_i\mathcal{G}_j} \\ p_C^{\mathcal{G}_i} &= 1 - \max_{\substack{\mathcal{G}_j \in \mathcal{M}_i \\ \mathcal{G}_i'\mathcal{G}_i'' \in \mathcal{S}_i}} \{p_S^{\mathcal{G}_i'\mathcal{G}_i''}, p_M^{\mathcal{G}_i\mathcal{G}_j}\} \end{aligned} \tag{3.3}$$

where $\overline{\mathcal{C}}$ is the complement of the set $\mathcal{C}$ of all continued groups for the current parent data association Ω^{t-1}.

As can be seen in Eq. 3.3, the probability for a split of group $\mathcal{G}_i$ into sub-groups $\mathcal{G}_i'$ and $\mathcal{G}_i''$ depends on the strongest social relation between these groups, $\mathcal{R}_{max}^{\mathcal{G}_i'\mathcal{G}_i''}$. Likewise, the probability for a merge between two groups $\mathcal{G}_i$ and $\mathcal{G}_j$ is equal to the highest probability for a person-to-person relation between these groups $\mathcal{R}_{max}^{\mathcal{G}_i\mathcal{G}_j}$. The probabilities $\mathcal{R}_{max}^{\mathcal{G}_i'\mathcal{G}_i''}$ and $\mathcal{R}_{max}^{\mathcal{G}_i\mathcal{G}_j}$ are readily available from the SVM output in the social relationship graph.

Here, we extend the approach of Luber et al. (2013) by the data-driven probability for continuation events $p_C^{\mathcal{G}_i}$. The continuation probability scales inversely with the highest probability for a non-continuation of that group (if, for instance, $p_S^{\mathcal{G}_i'\mathcal{G}_i''} = 1.0$, then $p_C^{\mathcal{G}_i}$ must be 0.0). Without this term, continuation events are overly biased which causes the tracker to not split up group tracks as intended[3].

3.4.3 Maintaining stable group identities

One of the goals of tracking is to maintain correct track identities despite misdetections, occlusions or measurement origin uncertainty. To achieve this goal in the case of group tracking, we extend the original concept by Luber et al. (2013) and define a set of maintenance rules for group track identifiers that are robust against identifier switches on the individual person track level. To motivate the need for this, Figure 3.2 shows a synthetically generated example that visualizes how group identifiers switch and stay constant without and with our proposed extension. Stable group IDs can be important *e. g.* for group guidance applications, as described in Section 9.6.

When groups undergo merge or split operations, their IDs need to be consolidated. Concretely, for merge events, we continue with the ID of the larger group – following a "merge-into" policy. If the groups are of the same size, we continue with the ID of the older group. This helps to maintain group ID consistency across identity switches of its member tracks, as long as not all person tracks switch IDs at the same moment.

[3] The reason is that Equation (3.2) favors few large groups, rather than many small ones: For a joint large group, we only count a single continuation event, and thus only end up with a single factor in the equation. Instead, after a split into two groups, subsequent model hypotheses contain two of these factors that make this hypothesis branch overall less likely since $p_C \cdot p_C < p_C$. This was not an issue in the original method by Lau et al. (2009) because data association occurred on collapsed group states, not individual person states. There, after a split that is backed by the measurements, the subsequent data association hypotheses with two separate groups would eventually make this branch of the hypothesis tree more likely.

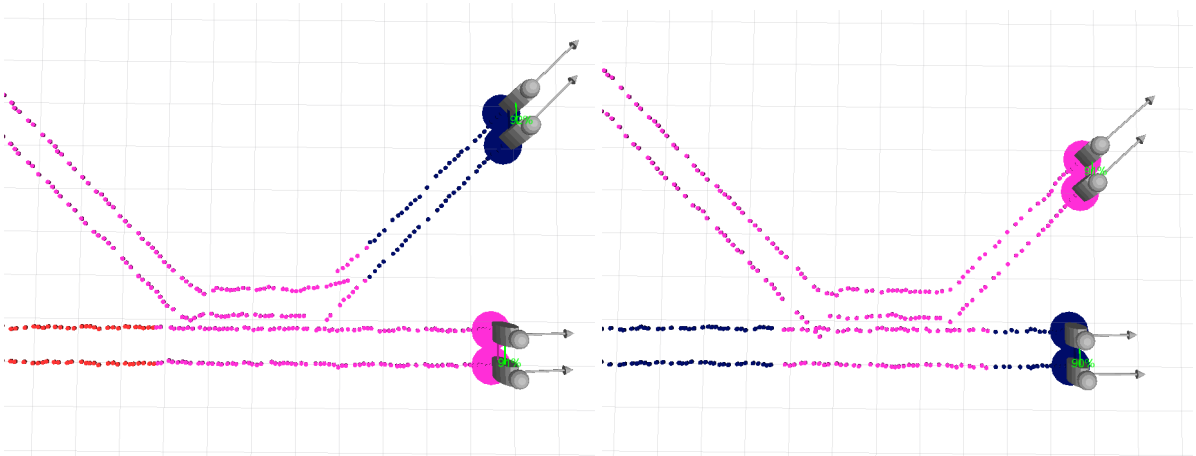


Figure 3.2: *Left:* Two groups that (wrongly) merge into one group, and then split again. Without the group identifier lookup described in Section 3.4.3, one of the groups undergoes an ID switch during the split (shown as different trace color). *Right:* With our extension, the previous IDs are restored.

For split events, we maintain the previous social groupings in a memory including their assigned group IDs. This allows to reassign the correct ID when a group merges into another and then splits off again. The strategy is useful to reidentify sub-groups that wrongly merged with a different group, for example when groups come temporarily close in a narrow passage. The memory is implemented as a map with circular buffer, keeping the last n group ID assignments, where n can be a large number.

3.4.4 Integration of group models into the hypothesis-oriented MHT

Tracking of individual persons

Individual persons are tracked using a hypothesis-oriented multiple hypothesis tracker (HOMHT) due to Reid (1979) that we introduced in Section 2.3. The approach generates hypotheses about the state of the world by taking into account all statistically feasible assignments between measurements and tracks, as well as all possible interpretations of measurements as false alarms or new tracks, and of tracks as matched, occluded or deleted. At time step t, a hypothesis Ω_i^t represents one possible set of such assignments, and measurement and track interpretation labels. We call $Z(t)$ the set of detected persons at step t, $\psi_i(t)$ the predicted track-to-measurement assignments and Z^t the aggregated set of all measurements up to t. Given a parent hypothesis $\Omega_{l(i)}^{t-1}$ with index $l(i)$ to accommodate for pruning, and new incoming measurements $Z(t)$, the MHT creates new assignment sets $\psi_i(t)$, each of which gives rise to a new child hypothesis branching off from its parent. To prune the resulting exponentially growing hypotheses tree, a probability is assigned to each hypothesis that is calculated in a recursive fashion using a normalizer η, the measurement likelihood, the assignment set probability and the probability of the parent hypothesis (Reid, 1979):

$$p(\Omega_i^t \mid Z^t) = \eta \cdot p(Z(t) \mid \psi_i(t), \Omega_{l(i)}^{t-1}) \cdot p(\psi_i(t) \mid \Omega_{l(i)}^{t-1}, Z^{t-1}) \cdot p(\Omega_{l(i)}^{t-1} \mid Z^{t-1}) \tag{3.4}$$

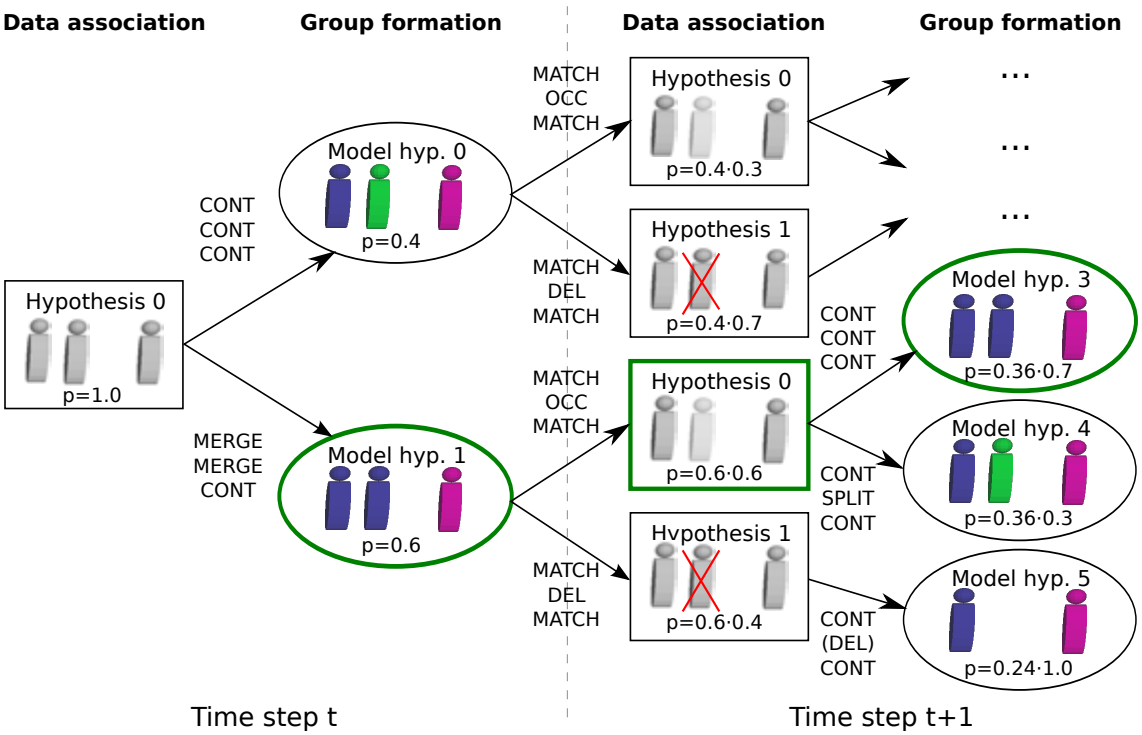


Figure 3.3: An example hypothesis tree with an intermediate tree level that corresponds to a group model hypothesis step. For each of the k person-level data association hypotheses in every step, the l most probable group model hypotheses are generated that postulate different group continuation, merge and split events. The green borders indicate the maximum probability hypotheses, persons with the same color are in the same group.

For pruning, we use multi-parent k-best branching according to Murty (*cf.* Cox et al., 1995) and N-scanback pruning (Cox and Hingorani, 1996). Unlike Lau et al. (2009), we do not use ratio pruning. A standard Kalman filter with a constant-velocity motion model is used to predict the state of person tracks that are not part of a group.

Integration of group models

To enhance the MHT with the ability to reason about group models, Lau et al. (2009) suggested to extend the hypothesis tree by an intermediate tree level to hypothesize about possible group formation processes. As shown in Figure 3.3, multiple group model hypotheses spring off from regular measurement-to-track data association hypotheses in each step, reflecting different possible evolutions of the parent group model $M(t-1)$ that are feasible given the social relationship graph at time t. Each group model hypothesis, in turn, will give rise to a number of child data association hypotheses conditioned on that particular group model.

Since for each data association hypothesis, a number of different group models is generated, modeling different splits, merges and continuations of all groups in the scene, there are in

total more model than data association hypotheses. To limit growth of the tree, we perform multi-parent k-best branching and restrict the number of possible group models to the $l < k$ most probable models.

The group model probability from Equation (3.2) can be incorporated as an extra factor and conditioning variable into the recursive update rule from Equation (3.4) for the probability of a hypothesis Ω_i^t (Lau et al., 2009; Luber et al., 2013):

$$\begin{aligned} p(\Omega_i^t \mid Z^t) \;=\; & \eta \cdot p(Z(t) \mid \psi_i(t), \underline{M(t)}, \Omega_{l(i)}^{t-1}) \\ & \cdot\, p(\psi_i(t) \mid \underline{M(t)}, \Omega_{l(i)}^{t-1}, Z^{t-1}) \\ & \cdot\, \underbrace{p(M(t) \mid \Omega_{l(i)}^{t-1})}_{\text{Equation (3.2)}} \cdot p(\Omega_{l(i)}^{t-1} \mid Z^{t-1}). \end{aligned} \tag{3.5}$$

Group formation feedback into person-level tracking

The principal advantage of the joint individual-group tracking approach by Luber et al. (2013) is that, due to the interleaving of data association and group model hypotheses, group tracking can inform person-level tracking and thereby make it more robust. The information about tracked groups can be used to adapt the per-track occlusion probabilities of group members, via a slight reformulation of the MHT proposed by Arras et al. (2008) that allows the MHT to not only reason about the interpretation of tracks to be *detected* or *deleted*, but also to be *occluded*. The reformulation generalizes track interpretation to an arbitrary number of labels using a multinomial distribution. Furthermore, we also adapt the motion model of occluded tracks inside groups, by switching from the constant-velocity model to a curvilinear model (Best et al., 1997) for motion prediction that is informed by geometric relations to non-occluded group members. This is realized through a particle filter, see Luber et al. (2013) for details.

Note that the feedback of group-level information to the person-level tracker is possible because the multi-model MHT generates group formation hypotheses for each new data association hypothesis. This means that when a best hypothesis switch occurs because new evidence has made the current branch unlikely, the group tracker will instantly have the best group model available which has already evolved over time with all measurements up to time t. This would not be possible if only a global, single-hypothesis group model had been generated.

Luber et al. (2013) show on large-scale datasets that the feedback from group tracking can effectively improve tracking performance by reducing the number identifier switches of person tracks (which, in turn, can make group tracking more robust). Our focus here is on the group tracking aspect and how it scales to more crowded environments, thus we refer the reader to their work for further implementation details and experiments.

3.5 Experiments

We now present qualitative experiments which we conducted on a novel in-the-wild RGB-D dataset from a pedestrian zone. Afterwards, we focus on more crowded scenarios with help of a pedestrian simulator.

Implementation details

The multi-model MHT is implemented completely in C++. While our implementation is based upon the one by Luber et al. (2013), the generation of group model hypotheses and construction of the social relationship graph has been completely re-implemented. To facilitate use of different detectors and multi-sensor setups, we have completely refactored the original CARMEN-based codebase[4] to use the more recent ROS middleware[5]. Each module (one detector per sensor, the filter that merges detections from multiple sensors, the multi-model MHT) is executed in a separate ROS node and process, such that they can benefit from parallelization on multi-core CPUs. Communication between modules occurs via ROS messages, which are described in more detail in Chapter 8. An overview of the resulting architecture is shown in Figure 3.4 (left). We run the group tracker including dual-sensor GPU-accelerated RGB-D person detection on a single Intel Core i7-2600 quad-core PC at 3.4 GHz with a GeForce GTX480 graphics card.

Experimental setup and tracking parameters

For the multi-model MHT, we use a scanback depth of $N = 6$ with a maximum of $k = 100$ hypotheses with up to $l = 6$ different group model branches. To further bound the space of possible group model transitions, in our implementation we allow only a single binary group merge or split per model hypothesis. This becomes relevant in our experiments in the crowded scenario, where it is possible that multiple groups are likely to split or merge in parallel. We adapt the prior probabilities for group continuations, merges and splits from Lau et al. (2009) as $p_{\mathrm{C}} = 0.63$, $p_{\mathrm{M}} = 0.21$, $p_{\mathrm{S}} = 0.16$. For person-level detection, occlusion and deletion probabilities, we use $p_{\mathrm{det}} = 0.55$, $p_{\mathrm{occ}} = 0.4$, $p_{\mathrm{del}} = 0.05$ outside of groups and $p_{\mathrm{det|G}} = 0.45$, $p_{\mathrm{occ|G}} = 0.5$, $p_{\mathrm{del|G}} = 0.05$ within groups. The Poisson rates for false alarms and new tracks are $\lambda_{\mathrm{new}} = 0.001$ and $\lambda_{\mathrm{fal}} = 0.009$. These hand-tuned parameters yielded qualitatively better results in our scenarios than the values learned by Luber et al. (2013) on a different dataset, and could indicate that detection is more challenging in our scenarios. Like them, we use 200 particles per occluded group track for the curvilinear motion model.

3.5.1 Experiments on a novel multi-sensor RGB-D dataset

Rathausgasse RGB-D dataset

For our experiments, we collected a sequence of 9 minutes, covering 300 meters of distance in an urban pedestrian zone in the city center of Freiburg, Germany. The data has been collected

[4] Due to the historically very large codebase (>100,000 lines), this process took several months. While doing so, we cleaned up the code considerably and made it more modular. An added benefit is that we can now exploit powerful ROS-based visualization tools like Rviz.

[5] Robot Operating System – http://www.ros.org

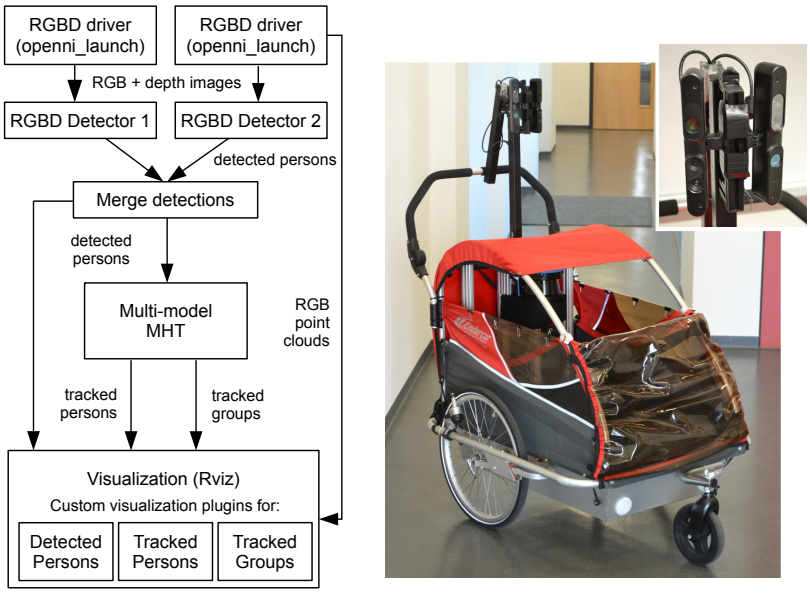


Figure 3.4: *Left:* The ROS-based architecture of our tracking setup consists of two separate RGB-D detectors, a module which merges their detection hypotheses, the multi-model MHT for person and group tracking, and visualization in Rviz. *Right:* Our mobile data capture platform that was used to capture the Rathausgasse dataset in a pedestrian area. The two vertical Asus Xtion Pro RGB-D sensors are shown enlarged.

using the mobile platform shown in Figure 3.4, which we equipped with two Asus Xtion Pro Live RGB-D sensors, see Appendix A for further details. The RGB-D sensors were mounted vertically at a height of 1.4 m with about $10°$ of overlap, yielding a field of view of $76°$ in horizontal and $57°$ in vertically direction.

Data has been recorded shortly before sunset to reduce IR interference (which can render the depth sensors inoperable) at the cost of slightly longer exposure times. The sequence contains 298 individual person tracks in 204 groups in total, of which 130 (64%) are individuals (single-person groups), 65 (32%) two-person groups, and 9 (4%) groups of at least three persons. The largest group within sensor range has 9 persons. The average group size is 1.5, or 2.3 if only groups with at least two people are considered.

Multi-sensor person detection in RGB-D

For human detection in RGB-D from a first-person perspective, we use the Combo-HOD detector as presented in Spinello and Arras (2011), a GPU-accelerated combination of HOG and HOD (histogram of oriented depths). HOD locally encodes the direction of depth changes and follows the same principle as HOG, while operating on the depth image. After subdividing the search window into cells, a descriptor is computed for each cell, the oriented depth gradients are collected into 1D histograms, and cells are grouped into blocks of four which then get normalized. The resulting HOD features are used to train a soft linear SVM. A depth-informed scale-space search is used to accelerate the sliding-window-based detection process.

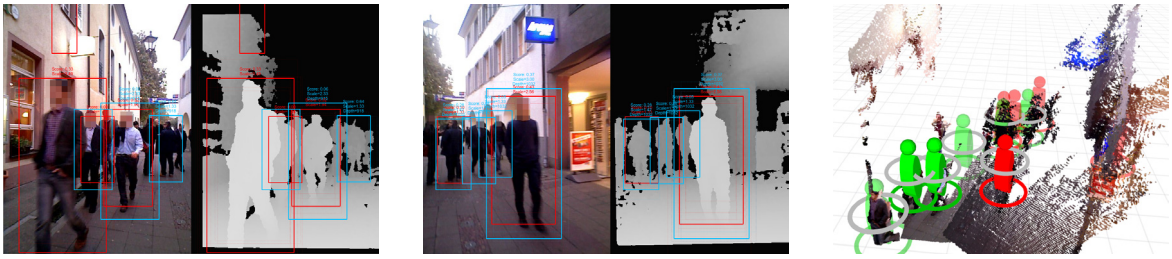


Figure 3.5: *Left and middle:* RGB and depth images of left and right RGB-D sensor with HOG detections shown in red and HOD detections in blue. *Right:* Point cloud with persons detected by the left (green) and right (red) sensor. Detections where only one of the two detectors has fired are semitransparent. Overlapping detections at the common sensor boundary are merged using non-maxima suppression. Grey circles denote fused detection candidates.

For the sake of generality and to increase the observable field of view, we consider a sensor setup with more than one RGB-D sensor (similar to setups of Spinello and Arras, 2011; Munaro et al., 2012). To fuse information from multiple sensors, we deploy an individual person detector instance per sensor, but combine the output of both detectors before tracking as shown in Figure 3.5. This has the advantage that the existing single-image detector can be readily used without having to fuse the RGB-D raw data which would raise the issues of image mosaicing or point cloud registration. Furthermore, a practical advantage is that each detector can be run on a single CPU core or graphics card allowing for easy parallelization. In case of a small overlap of the field of views between adjacent sensors (as in the case of our setup), we can detect people right at the center between both sensors. In a non-maxima suppression step, we finally consolidate duplicate detection hypotheses that are seen by both sensors.

Qualitative results on the Rathausgasse dataset

Figures 3.6 and 3.7 show some exemplary situations which the tracker is able to handle. By qualitatively inspecting the full real-world sequence, we were able to validate that our group tracking algorithm is able to track groups of people with varying sizes over relatively long distances (up to around 70 m if our sensor platform was following behind) with a low number of group ID switches. This is partly achieved due to the robustness of the person-level tracking, which benefits from the incorporated group information. Also, the probabilistic SVM based upon motion indicators allows our group detection stage to discriminate between groups passing close by each other at different velocities or in different directions. In our real-world dataset, over a duration of 9 minutes (9535 frames), we observed 11 group ID switches in total for the 74 groups of size two or larger.

Failure cases

There are two main failure cases: Firstly, group cardinality gets underestimated when individual persons are not detected due to partial occlusion (and thus no person tracks initiated). The second failure case is the incorrect merging of groups especially when persons are standing still. In such cases, the orientation feature that we use for group detection is not very reliable, because

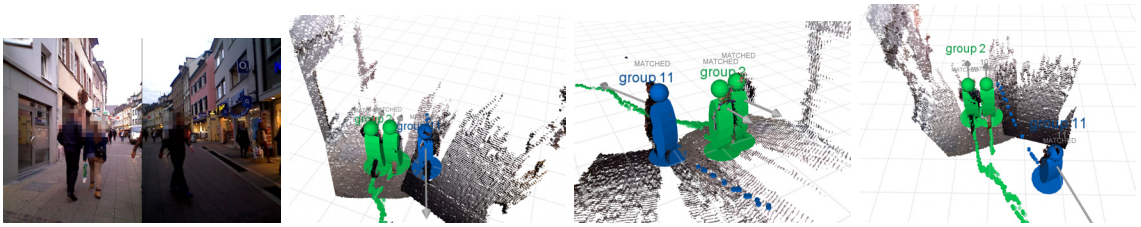


Figure 3.6: Two groups – a two-person group (green) and a single-person group (blue) – are passing close-by each other, and are *correctly* not merged by our tracker. The SVM-based group detection using coherent motion indicators identifies them as two distinct groups of persons.

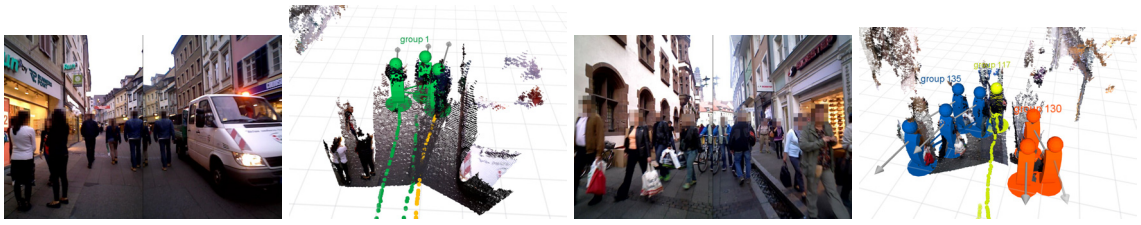


Figure 3.7: *Left two pictures:* A man in blue yeans (yellow) merges into the two-person group walking in front (green). *Right two pictures:* Our approach is able to track complex multi-person formations from a first-person perspective.

we currently do not estimate body orientations during detection, and instead derive it from the person's linear velocity.

Runtime performance

The person-level MHT tracker without the RGB-D person detector reaches an average processing rate of 59 Hz on a single CPU core with 100 parallel hypotheses and around 10 groundtruth tracks visible at a time. This rate decreases to 24 Hz for the multi-model hypothesis group tracker, where most of the overhead is caused by the prediction of pairwise social relations using the linear SVM classifier (7% of total runtime cost) and the update of the particle filters for the curvilinear motion model (about 6% total) used for motion prediction of occluded group members. The memory consumption is around 2 GB. The entire system (detection and joint individual-group tracking) reaches 20 Hz.

3.5.2 Experiments in more crowded environments

Our next goal is to learn about the behavior and possible limitations of the method when scaling to more crowded environments, as one could expect them *e. g.* inside a busy airport terminal. Since real data from the SPENCER project (see Chapter 9) was not yet available, we instead utilize a simulated dataset.

PedSim dataset

To simulate a crowded environment with plausible pedestrian dynamics, we leverage *PedSim*, a pedestrian simulator that is based upon the social force model by Helbing et al. (1995). As described in Section 8.9, we have integrated the simulator with ROS and the physics-based

simulator Gazebo. We use Gazebo to model a virtual 2D laser scanner via raytracing to simulate partial observability under occlusion. Figure 3.8 shows the resulting scenario, which depicts a 30×30 meter square area in which pedestrians move in fast-moving flows, and a slow-moving queue. There are also static groups and persons moving as individuals. From this simulation, we generate a fixed 120-second sequence consisting of 4,800 frames. The total number of individuals is constant at 90, of which around 50 can be observed at a time with a simulated detection range of 20 m. This is a significantly more crowded scenario than *e. g.* the City Center and Main Station datasets used by Luber et al. (2013), which on average contain 10 and 16 visible tracks at a time, and the sequences of Lau et al. (2009) with 6 and 9 tracks on average.

To decouple detection and tracking performance, we assume a perfect detector without false positives or negatives for these experiments, but we model occlusions by passing only those groundtruth pedestrian positions as detections to the tracker which produce at least one 2D laser return. While we cannot expect such good performance in practice, we operate under the assumption that future detectors – as proposed *e. g.* in the upcoming Chapters 5 and 6 – will become more robust.

A simple but efficient group tracking baseline

As a baseline for comparison, we further implemented a simple group tracking baseline that first tracks persons and then groups in a sequential, non-joint fashion. The approach is similar to the baseline used by Lau et al. (2009), but with memory. For person tracking, we use the efficient nearest-neighbor tracker that we will describe in more detail in Section 4.3. To estimate social relations, we use the same probabilistic SVM trained on coherent motion indicator features as before. We then detect groups through single-linkage clustering on the full set of social relations $\{\mathcal{R}^{i,j}\}$ with a threshold of 0.5. To track group affiliations over time, we keep a memory of previous group ID assignments per set of clustered track IDs. For each detected track cluster, we iterate through the group ID memory and find the ID of the largest (or oldest) previously existing group that most strongly intersects with the current set of track IDs. This mimics a similar behavior as our extension to the multi-model MHT in Section 3.4.3.

Qualitative results

Our qualitative results on the crowded simulated scenario are visualized in Figure 3.8. We make the following observations:

- At real-time simulation speed **(b)**, the multi-model MHT with $k = 100$ hypotheses does not behave well in this crowded scenario with approx. 60 simultaneous measurements. With a cycle time of around 5 seconds (0.2 Hz), it is not able to cope with the dynamics of the scene. Many tracks are either not initiated, or their covariances "explode" because the tracker drops entire cycles of measurements which it cannot process fast enough. No groups are formed. If we reduce the number of data association hypotheses, tracking becomes more responsive, but even less tracks get initiated.
- If we reduce the playback rate by a factor of 20 **(c)**, tracks are initiated as expected and tracked more smoothly. We also see that groups slowly start to form. Person tracks lag

behind their groundtruth by around 1 meter, which is due to us visualizing the best root hypothesis (at a scanback depth of 6). This could be alleviated by using the (less informed) best leaf hypothesis.

- The NN group tracking baseline tracks all individual persons smoothly at significantly lower CPU usage (<10% on a single core). It deals relatively well with momentary occlusions (note that we do not simulate false alarms or position measurement noise). Groups are formed more quickly, but also more volatile. The baseline yields comparatively larger groups than the multi-model MHT. Both methods switch IDs of larger groups every few seconds when they undergo repeated splits and subsequent merges.
- In general, in such dense flows, without additional visual features it is even for a human observer difficult to distinguish and track distinct groups.

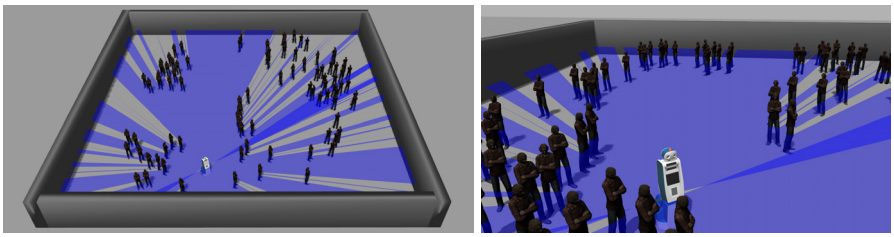
Figure 3.9 shows a real-world group tracking result obtained using the baseline method in an airport scenario with the SPENCER robot (for details on the multi-modal sensor setup and the dataset, see the next chapter). Here, the robot is following a larger flow of people. Again, it is difficult to decide for a human observer (even from a sequence of frames) if *e. g.* the yellow group #5 should be considered as one group, or not, if we take only spatial proximity and motion into account.

Discussion of runtime and tracking performance in crowded scenarios

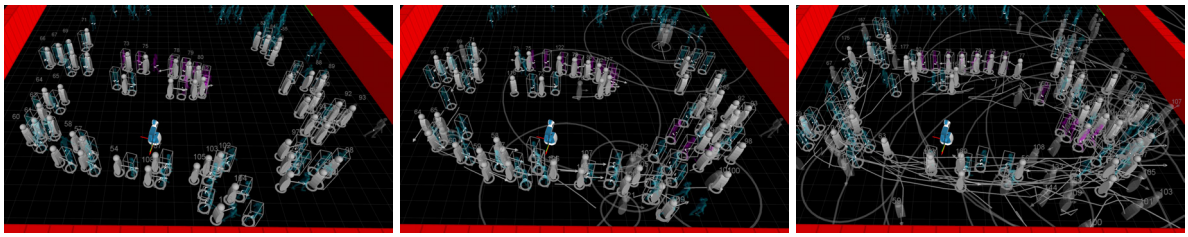
With regard to real-time performance, it is not a novel insight that the hypothesis-oriented MHT approach after Reid (1979), even without the additional interleaving of group model hypotheses, is computationally very complex and starts to break down in terms of either speed, or tracking performance, with more than 20–30 tracks. For instance, Oh et al. (2009) have shown that both MCMCDA and a greedy NN tracker outperform HO-MHT in such cases, while the MHT approach is superior with fewer tracks and at higher detection probabilities. They state that even though using ten times as many hypotheses as we do, MHT performance still deteriorates due to pruning. Our qualitative observations support their quantitative findings.

One strategy to reduce the computational effort, which has already been proposed earlier (Reid, 1979; Cox and Hingorani, 1996) but is not part of our implementation, is to spatially subdivide the scene into smaller, more tractable data association problems. However, such a spatial clustering might be difficult to achieve in very dense environments. Another practical solution could be to pass only the m closest detections to tracking, at cost of a reduced tracking range.

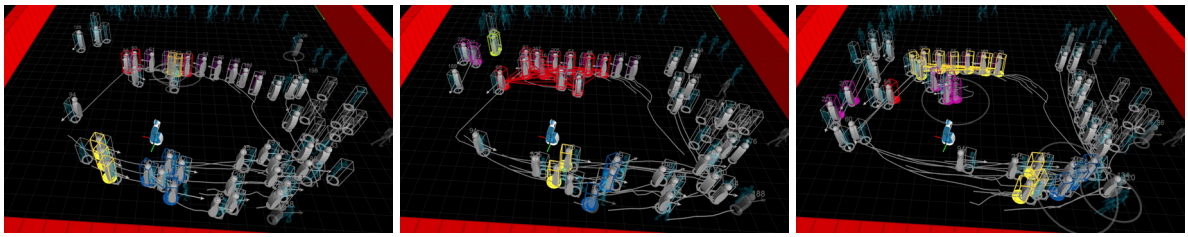
The joint-individual group tracking approach that Luber et al. (2013) and we follow here is computationally even more complex due to the additional group model hypothesis generation step, especially because we need to estimate social relations using the SVM for every single data association hypothesis; this could easily be parallelized. Instead, Lau et al. (2009) were able to *improve* runtime performance through tracking of groups, by collapsing the states and not tracking individuals, which simplifies data association. However, if tracking of individuals is required, the joint individual-group tracking approach allows to feed back information from group- into person tracking, which can make tracking of persons in groups much more robust as demonstrated by Luber et al. (2013).



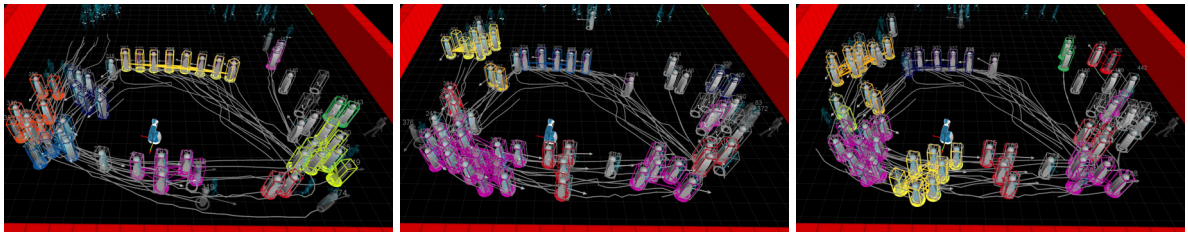
(a) Simulation of partial observability in 2D laser via raytracing in the Gazebo simulator



(b) Multi-model MHT at 1.0× real-time speed – High latency in person tracking, no group formation



(c) Multi-model MHT at 0.05× real-time speed – Groups slowly begin to form



(d) NN person tracker followed by group tracking without feedback, 1.0× speed

Figure 3.8: Simulation of very crowded scenarios (50–60 parallel detections) using the PedSim simulator. All images show different, but representative moments of the simulation. Grey boxes are person detections, cylinders person tracks, colored boxes indicate groups. Groundtruth is visualized by magenta and cyan person meshes that symbolize queuing and flow behaviors.

We observed that the groups formed by the multi-model MHT are not as large as those of the baseline. Possible reasons for this could be the fact that our implementation of the multi-model MHT only allows a single group to merge or split in a binary fashion per model hypothesis, or that the prior probabilities for group merges, split and continuations from Lau et al. do not generalize well to our scenario.

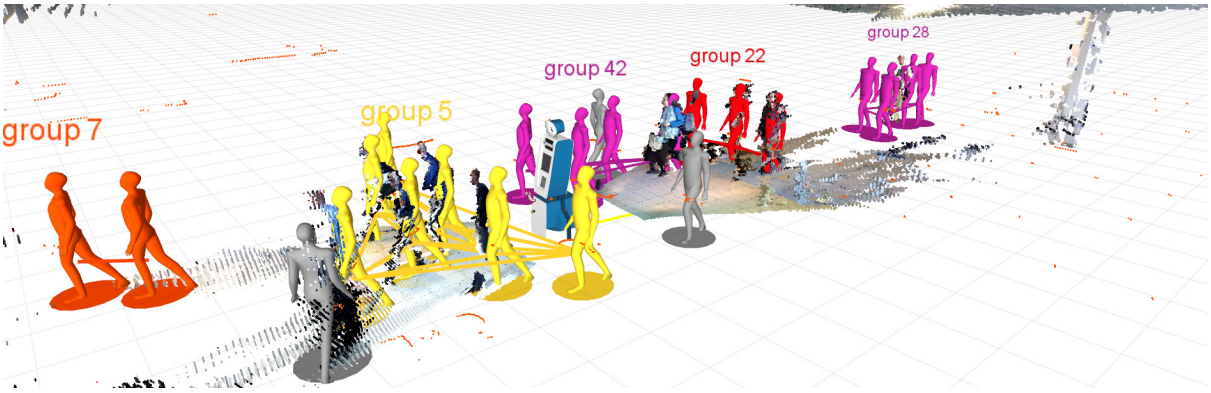


Figure 3.9: Group tracking within a pedestrian flow in the airport environment

3.5.3 Experiments on socially-aware navigation

Lastly, we apply group tracking for socially-aware navigation. Figure 3.10 shows different experiments that we conducted with the social robots DARYL and SPENCER, where we tried out both the proposed multi-model MHT approach and the baseline method.

Experimental setup

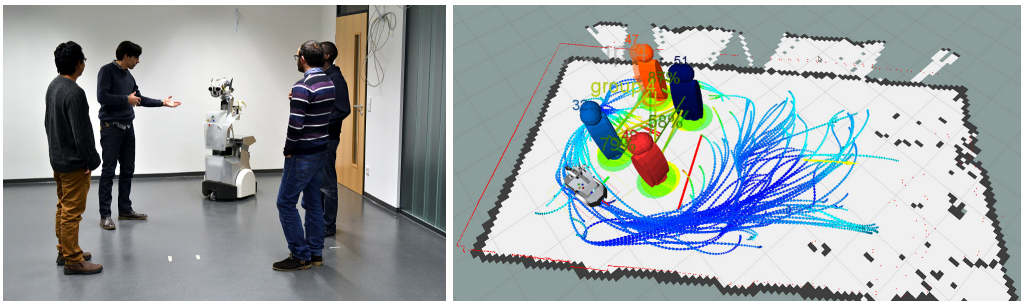
In the first row **(a)**, we see an example of a socially normative behavior that has been learned from large-scale simulations in PedSim using a Bayesian inverse reinforcement learning approach by Okal and Arras (2016). The different learned behaviors (polite, sociable and rude) lead to different social costmap representations, on which the robot performs path planning using an A* or RRT* planner (right picture) as described by Palmieri et al. (2014). In the shown example, the robot is supposed to navigate *politely* around the tracked group, instead of *rudely* passing through them. Here, persons are detected through two back-to-back 2D laser scanners and the detector by Arras et al. (2007). Further details and results can be found in the paper by Okal and Arras (2016).

In the second row **(b)**, we perform a similar experiment with the SPENCER robot in an airport environment. Here, we use a multi-modal detector setup comprising two RGB-D and two 2D laser detectors that we describe in the next chapter.

In the third row **(c)**, we perform group *guidance*, as opposed to avoidance: SPENCER's task is to guide a group of persons over 25 m distance through a laboratory environment while avoiding other pedestrians. Tracked groups are relayed from the group tracker to a guidance supervision module (see Section 9.6).

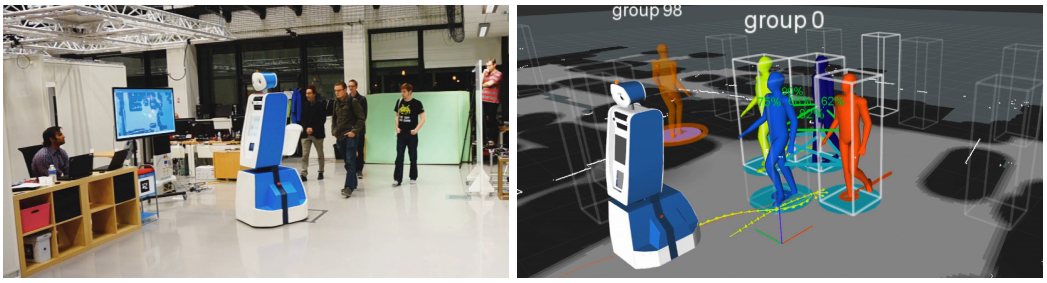
Qualitative insights

In the depicted, relatively simple open-space scenarios where no crowds are present, both the multi-model MHT and the baseline exhibit similar group tracking performance. They mainly differ in how well they track individual persons. While the baseline quickly loses tracks because group members occlude one another, the MHT can keep them alive for longer because it is aware of their group affiliations and can switch to the curvilinear motion model that is informed by the motion of other group members.



(a) Group-aware social navigation *(polite behavior)* with the robot DARYL

(b) Group avoidance experiment with the SPENCER robot at the airport



(c) Group guidance experiment with the SPENCER robot

Figure 3.10: Experiments with real-world applications of group tracking. Colored circles below visualized person tracks indicate group affiliations.

In the group avoidance scenarios, both methods often split up groups too quickly when members make space for the robot to pass through. This is due to the coherent motion indicators' reliance on spatial proximity. As we discussed before, proper estimation of body orientations for still-standing persons could improve results. As an interim solution, we temporally smooth the social relations to delay split events and obtain "long-term relations", with a method similar to the Bayes filter used by Luber et al. (2013).

3.6 Conclusions

In this chapter we addressed the problem of detecting and tracking groups of people in RGB-D data. Groups are detected from predicted social relation probabilities between individuals and tracked using an extension of the hypothesis-oriented MHT that incorporates a group model hypothesis step. This approach allows to recursively reason about regular measurement-to-track data associations and group formation processes at the same time and in the same probabilistic

framework. We extended this approach, which was initially introduced by Lau et al. (2009) and Luber et al. (2013), with new expressions for group model probabilities and a book keeping logic to maintain stable group track identifiers which are more robust to sporadic identifier switches of the underlying person tracks.

Our experiments qualitatively demonstrated the viability of the approach on a real-world, unscripted, novel, outdoor multi-sensor RGB-D dataset collected with a mobile platform in a busy urban pedestrian zone. We have integrated the implementation with the ROS middleware and have shown on two different robot platforms that the approach can be practically used for group-aware social navigation. One benefit of the joint individual-group tracking approach, that has been demonstrated experimentally by Luber et al. on 2D laser range data, is that group tracking can inform person-level tracking and thereby make it more robust, for instance by adapting occlusion probabilities and motion models within groups.

Limitations and open issues

In this chapter, we did not conduct extensive quantitative experiments. The reasons for this are threefold: First, our main goal was to gain initial qualitative insights into the most prominent research challenges related to multi-modal human and group detection and tracking in crowded environments, in order to motivate subsequent research. Second, there were no established metrics for the evaluation of group tracking performance: Some previous works had been evaluated using the CLEAR-MOT metrics, either on individual person level (see Luber et al., 2013), or by associating group centroids and computing centroid localization error or bounding box intersection. However, this was not the focus of our research. Our interest was in group detection and tracking of group formation processes. The lack of relevant group tracking metrics has recently been addressed by Vascon et al. (2017), who introduced GDSR (group detection success rate), and Setti et al. (2019), who propose the GRODE (group detection) metrics which take cardinalities into account. Third, to our best knowledge, no suitable multi-modal, multi-sensor annotation tool existed to label groundtruth on our RGB-D dataset. In Section 8.8, we will present our trajectory-based annotation tool that closes this gap.

Qualitatively, we observed the following limitations of the method:

- Partially occluded group members are often not detected and thus not merged into the group. This is best addressed at the detection stage, see Chapter 5 and 6.
- Static persons are sometimes wrongfully merged or split off, due to the lack of body orientation estimates. This topic is not addressed in this thesis, but there exist real-time methods for estimation of rough body orientations or 3D articulated human pose from RGB-D (Wengefeld, Lewandowski, et al., 2019; Zimmermann et al., 2018).
- Coherent motion indicators do not work very well for group detection in static scenes (where there is no motion), and in very dense and crowded environments (where there is no spatial separation). For the former case, presumably methods dedicated to detecting *F-formations* are more suited. In crowded scenes, additional features should be incorporated to make group tracking more robust. These could *e. g.* be human attributes, like the ones we recognize in Chapter 7 from RGB-D point clouds; recent work by Goel et al. (2019) includes additional visual attributes such as age to generate more informed social

relationship graphs. An open question is how to apply this to multi-modal sensor setups that do not offer 360-degree camera coverage. To this end, Solera et al. (2016) focus on tracking groups in crowds, and introduce further trajectory-based features that *e. g.* account for trajectory shape similarity. However, the principal challenge in our case is the much shorter observation timespan compared to overhead surveillance scenarios.

- The most obvious limitation of the approach – when real-time constraints play a role – is its computational complexity, especially in crowded environments: In a simulated scenario with 50–60 simultaneous tracks, we found that the method is not able to cope with the dynamics at real-time simulation speed on a high-end PC. If we reduce the number of generated data association hypotheses, tracking performance degrades. Already the regular HO-MHT, without group models, is known to be computationally very complex (Cox and Hingorani, 1996). The described behavior is consistent with observations by others (Oh et al., 2009). The multi-model extension with feedback into person-level tracking further increases complexity, also because the more efficient shallow track tree implementation by Kurien (1990) that shares information between hypotheses is not applicable anymore. There are several possible approaches to improving real-time performance:

  - *Increasing efficiency.* The computation of social relations and the generation of group models could be parallelized across all data association hypotheses. Also, diversity of measurement-to-track hypotheses could be increased such that less of them would need to be generated at every time step: X. He et al. (2013) observe that in HO-MHT, data associations for confirmed tracks vary only slightly in the top K assignments generated by Murty's algorithm, which degrades tracking performance. They modify Murty's method to guarantee diversity of assignments for a designated set of tracks.

  - *Further approximation.* It might be sufficient to re-compute social relations and group models only every few frames, if taken into account during data association.

  - *Studying less complex approaches.* From the experiments of Luber et al. (2013), it is clear that feedback from group-level tracking can make person-level tracking more robust. It would be interesting to see if, how, and how successfully this concept could be integrated into simpler and computationally more efficient methods such as TO-MHT, MCMCDA or GNN approaches.

- In many crowded and highly dynamic scenarios, a human would most likely track and re-identify groups through the unique identities and appearances of their members. To this end, group-based re-identification is an interesting and unsolved recently studied topic in computer vision (*e. g.* W. Lin et al., 2019).

CHAPTER 4

Multi-modal human tracking in crowded and dynamic environments

We now take a step back from tracking of groups and investigate the problem of tracking individual humans with a multi-modal sensor setup on a mobile platform: In this chapter, we take the human tracking challenge to new extremes by aiming to track people in a 360-degree field around the robot in crowded and dynamic environments like a busy airport terminal. While prior research has examined various approaches to make human tracking more robust by incorporating human-specific models and priors, the impact of data association methods of varying complexity has not been studied extensively and systematically, especially in such difficult scenarios and observed from a robot-centric perspective.

We make two main contributions: First, we present a computationally efficient tracking baseline that extends a nearest-neighbor tracker in several ways in order to make it more robust in such scenarios. In initial experiments, we show that simple track initiation and deletion logic can help to significantly reduce the number of ID switches. We then compare our method systematically under different detector combinations to a hypothesis-oriented MHT, a track-oriented MDL tracker, and different NN variants on two novel annotated datasets. We find that our efficient baseline outperforms all other evaluated methods on the MOTA metric across all settings. We also gain interesting insights that motivate our later research: Most importantly, we learn that detector performance is the single, most influential factor affecting tracking performance which goes far beyond the impact of the chosen tracking algorithm.

Parts of this chapter have been previously published in the paper "On Multi-Modal People Tracking from Mobile Platforms in Very Crowded and Dynamic Environments " by T. Linder and S. Breuers (who contributed equally), B. Leibe and K. O. Arras in the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2016 in Stockholm, Sweden.

Further parts have been previously published in the workshop paper "Towards a Robust People Tracking Framework for Service Robots in Crowded, Dynamic Environments" by T. Linder, F. Girrbach and K. O. Arras at the Assistance and Service Robotics (ASROB-15) Workshop of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS) 2015 in Hamburg, Germany.

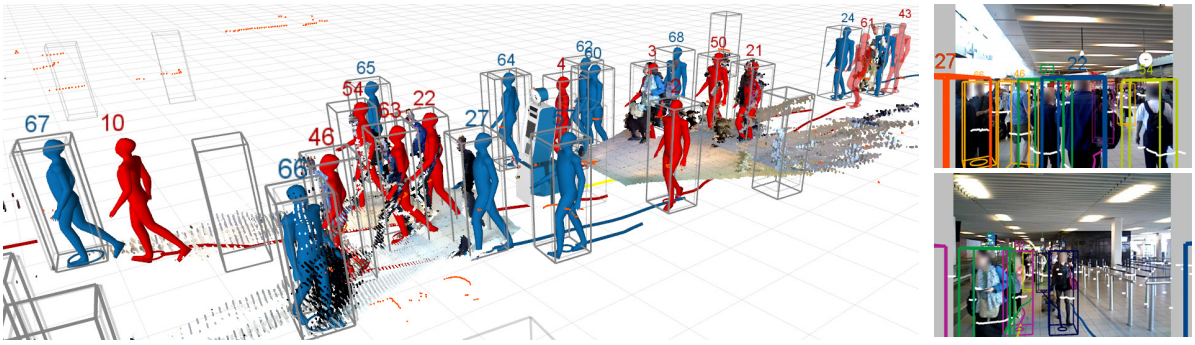


Figure 4.1: Typical example of a crowded, highly dynamic situation in an airport terminal where we want to robustly and efficiently track persons. Boxes indicate multi-modal detections and person meshes are tracked persons; red ones have been confirmed visually, blue ones were detected only using 2D laser.

4.1 Introduction

The problem of tracking people from a mobile platform in a first-person perspective has been studied in the robotics and computer vision communities for over two decades, and provides important functionality for assistance and service robots operating in human environments. It can lay the foundations to higher-level social processing such as group detection and tracking (see previous chapter), human-aware and socially-aware navigation (Kruse et al., 2013; Charalampous et al., 2017), social activity detection (Okal et al., 2014), or general human-robot interaction, and is crucial in person following and guidance tasks (Leigh et al., 2015; Wojke, Memmesheimer, et al., 2017; Kollmitz et al., 2019; Wengefeld, Mueller, et al., 2019). The tracking problem can be considered as mostly solved in scenarios with only few persons walking in front of 2D laser scanners (Schulz et al., 2003; Arras et al., 2008; Leigh et al., 2015) or RGB-D sensors (Munaro and Menegatti, 2014; Kollmitz et al., 2019), and good progress has been made in semi-crowded scenarios with 10–15 people simultaneously visible (Ess et al., 2009; Luber et al., 2010; Luber, Spinello, et al., 2011; Hosseini Jafari et al., 2014; Wojke et al., 2016). Some approaches that cover either of these scenarios combine multiple sensing modalities (Martin et al., 2006; Bellotto et al., 2009; Spinello et al., 2010; Volkhardt et al., 2013; Dondrup et al., 2015; Wengefeld et al., 2016; Wengefeld, Mueller, et al., 2019).

However, even larger numbers of persons may need to be tracked when the entire 360-degree field of view around the robot is covered by an array of sensors and the robot is operating in very crowded environments. So far, few methods have been evaluated in complex and highly dynamic scenarios where over 30 persons can be present at the same time, such as in a crowded terminal of a busy airport where our person guidance robot was to be deployed (Figure 4.1).

In this chapter, we therefore want to go one step further and examine how well tracking methods of different complexity perform in such challenging, highly crowded and dynamic scenarios. Inspired by trends in the computer vision community towards a standardized benchmark for multi-object tracking methods (Milan et al., 2013; Leal-Taixé et al., 2015), and to enable a fair comparison of different tracking methods, we integrate several publicly available tracking approaches into a unified tracking framework and provide them with a common set of detections

as input, to learn more about their relative strengths and weaknesses. With this, we are one of the first to systematically examine human tracking using a multi-modal sensor setup in such challenging, first-person perspective scenarios – a topic that only recently has been gaining more attention in computer vision and robotics (Martín-Martín et al., 2019; Wengefeld, Mueller, et al., 2019).

Outline Concretely, our contributions are the following: After an extensive discussion of related work, we develop a highly efficient baseline method for tracking in crowded environments. For this, we compare different greedy and global nearest neighbor methods with a hypothesis-oriented MHT approach, and show in two different scenarios that the simpler methods can reach similar or higher performance when combined with a deterministic track initiation and deletion logic. We learn a well-performing set of tracking hyperparameters using automated hyperparameter optimization by maximizing MOT accuracy.

We then conduct experiments on two challenging new datasets collected from a first-person perspective, one of them containing very dense crowds of people with up to 30 individuals within close range at the same time. We consider four different tracking approaches of varying complexity and study their impact on seven different performance metrics, in both single and multi-modal settings. In contrast to many works in computer vision, we evaluate tracking performance in world space coordinates. We thoroughly discuss strengths and weaknesses of the examined methods, provide practical guidelines and derive possible future research directions.

4.2 Related work

Human tracking from a robot-centric perspective

Most human tracking approaches in robotics follow the tracking-by-detection paradigm which we introduced in Chapter 2, and which is our focus in this thesis. An overview of the topic, which also references the work that we present in this chapter, is provided in a recent book chapter by Bellotto et al. (2018). The most commonly used sensor modalities for human detection are RGB-D (or stereo) cameras, 2D laser range finders and 3D lidar. They will be examined in the following Chapters 5 and 6, therefore we focus our discussion here on related work that investigates the tracking problem.

In the literature, various online methods for tracking of people from a robot-centric perspective have been studied. Out of the generic multi-object tracking approaches explained in Chapter 2, the most common ones in robotics applications are based upon nearest-neighbor (NN) data association or multi-hypothesis tracking (MHT). More recently, random finite set (RFS) approaches have also started attracting attention.

Nearest-neighbor approaches

Bellotto et al. (2009) and Bellotto et al. (2010) detect people using a 2D laser-based leg and a vision-based face detector, and track them using NN data association. They experiment with different forms of Kalman filters and a particle filter, and find that an Unscented Kalman Filter

(UKF) leads to significantly fewer identity switches than an EKF[1] while being computationally more efficient than the particle filter solution. Z. Yan et al. (2017) use the tracking framework of Bellotto et al. (2015) for online learning of a lidar-based human detector, whereas Dondrup et al. (2015) generate qualitative representations of human-robot spatial interactions and use an HMM to classify human-robot encounters. They integrate the upper-body detector of Mitzel et al. (2012) that we also utilize. We use their implementation as a baseline in our experiments.

Martin et al. (2006) use a probabilistic fusion scheme based upon covariance intersection to combine detections from different sensors. Volkhardt et al. (2013) use the same framework, but different detectors. Wengefeld, Mueller, et al. (2019) incorporate detector-specific detection probabilities. They include recent detectors like OpenPose (Cao et al., 2017) and conduct experiments also on our motion capture sequence. They highlight that not only 2D detection accuracy, but also accurate 3D coordinates are important for robust tracking in world space, a problem we reported earlier (Linder, Breuers, et al., 2016; Linder et al., 2018) and aim to solve in Chapter 5.

Munaro and Menegatti (2014) track people in RGB-D point clouds using GNN data association with a three-term joint association likelihood that incorporates a histogram-based online appearance model. They use Euclidean head sub-clustering and a HOG-SVM for detection.

Kollmitz et al. (2019) track people with walking aids in RGB or depth data using multi-assignment NN data association. During detection with an image-based Faster R-CNN approach (Ren et al., 2017) they distinguish between five different classes and later smooth the class estimates using a hidden Markov model. Like in their earlier work (Vasquez et al., 2017), they track positions in a class-agnostic fashion in world space, but this time model measurement noise in image space and thus rely on an EKF for the non-linearities that arise from the measurement model.

Instead, Gritti et al. (2014) focus on small-footprint robots that can only observe legs of humans in RGB-D. They first track legs independently using an NN approach, and then associate leg tracks with human tracks in a second data association step. Depending on the distance between tracks and measurements, they either perform a direct Kalman filter update, or a joint update using all surrounding measurements as in PDAF. They achieve around 15% higher precision in RGB-D than when detecting legs in 2D laser with the method of Arras et al. (2007).

Leigh et al. (2015) propose a joint leg tracker that tracks human centroids in 2D laser by detecting individual leg segments and then directly associating them with person tracks using a single GNN data association; to accomplish this, they create a temporary copy of each track to allow association of both legs. In addition, they build and maintain a dynamic occupancy grid to filter false positive detections within static obstacles. They point out that although complex data association methods, such as JPDAF or MHT, have been shown to deliver better performance in application areas with high clutter like radar tracking (Blackman et al., 1999), no systematic comparison between simpler and more complex data association methods has been performed for human tracking, where false positive detections occur systematically rather than randomly.

[1] Their system is non-linear because they use a range-bearing measurement model and parameterize the motion model based on body orientation, assuming that persons only move in forward direction.

Multi-hypothesis approaches

Arras et al. (2008) apply the HO-MHT framework by Cox and Hingorani (1996) to the same problem of tracking legs of people in 2D laser. They extend the original formulation to reason also about occluded tracks by generalizing track interpretation to an arbitrary number of labels using a multinomial distribution.

Luber (2014) studied different extensions to the HO-MHT framework which introduce human-specific priors and social constraints. Most experiments have been conducted in 2D laser with a static sensor setup. Specifically, they include the integration of a social force model for motion prediction (Luber et al., 2010), spatial priors for people tracking (Luber, Tipaldi, et al., 2011b), and adaptation of motion models and occlusion probabilities within groups (Luber et al., 2013). Luber, Spinello, et al. (2011) integrate the ComboHOD RGB-D detector by Spinello and Arras (2011) into the HO-MHT framework, combine it with a target-specific online detector based on RGB-D appearance features, and evaluate the approach in a multi-sensor setup.

Hosseini Jafari et al. (2014) use a vision-based MDL tracker with bi-directional EKF that is loosely based upon the principles of TO-MHT, but formulated as a quadratic pseudo-boolean optimization problem in a minimum description length (MDL) framework to find compatible track hypotheses (Ess et al., 2009). They combine a monocular HOG detector for far-range vision with an upper-body depth template-based detector for near range on a mobile platform.

Random finite set approaches

Wojke et al. (2016) present an RFS multi-target tracker which provides instantaneous state estimation with delayed decisions on data association, by combining a PHD filter with a min-cost flow network (L. Zhang et al., 2008). They evaluate their method on the RGB-D dataset by Luber, Spinello, et al. (2011) using the PCL detector by Munaro et al. (2012) and achieve slightly better tracking performance than the Munaro GNN method, but slightly worse performance than the HO-MHT of Luber, Spinello, et al. (2011) with the detector by Spinello[2]. Interesting for our work is their insight that their SMC-PHD approach suffers especially from temporary occlusions in the robot-centric perspective. They state that these are not easy to model in the PHD framework, and instead mitigate this through global optimization with the min-cost flow network. In contrast to the other methods, they thereby obtain significantly fewer ID switches at cost of latency.

Wojke, Memmesheimer, et al. (2017) combine two RFS methods, a Bernoulli and a PHD filter, to reliably track a single person through a crowded environment. The Bernoulli filter is used to track the robot operator, while the PHD filter models the statistics of other pedestrians in the scene. They additionally learn in an online fashion a deep appearance-based classifier to re-identify the operator after lengthy occlusions.

[2]It is thus not clear if the performance difference results from the tracking method, or the detector. It is noteworthy that they adapt clutter intensity along walls based upon an occupancy grid map in order to reduce false positives. This is similar to the filtering of detections we propose in Section 4.3.5.

Closely related to our work is the experimental comparison of different tracking approaches by Correa et al. (2013). They compare a PHD filter to JPDA[3] and NN data association in a crowded environment that is observed by a 270-degree 2D laser range finder. The NN implementation by Bellotto et al. (2009) that they compare against is also the basis for the tracker by Dondrup et al. (2015) which we include in our experiments. In the experiments by Correa et al., their proposed PHD approach outperforms all other baselines in their scenario. Unfortunately, their implementation is, to our best knowledge, not publicly available and not trivial to re-implement. In contrast to them, we use a moving platform with multiple sensor modalities and consider multiple NN and two MHT variants in our experiments.

Datasets and benchmarks for human tracking

In the computer vision community, Leal-Taixé et al. (2015) and Milan et al. (2016) introduced the MOTChallenge, a benchmark for the evaluation of multi-object tracking methods in the context of people tracking. It contains crowd-sourced video sequences from both overhead and ego-centric perspectives – some of the latter recorded with a mobile sensor – and a set of standard detections to put a stronger focus on the tracking aspect. Leal-Taixé et al. (2017) describes interesting insights gained through the challenge, which is split into a 2D part for tracking in image space, and a 3D part for tracking in world coordinates.

Most existing datasets for human tracking in robotics focus on a single sensor modality, such as the 2D laser datasets by Luber, Tipaldi, et al. (2011b), Leigh et al. (2015), and Álvarez-Aparicio et al. (2018), or the RGB-D datasets by Luber, Spinello, et al. (2011), Munaro and Menegatti (2014), and Vasquez et al. (2017). There are only few publicly available multi-modal datasets for this purpose. Autonomous driving datasets such as KITTI (Geiger et al., 2012) or nuScenes (Caesar et al., 2020) focus on outdoor use-cases where the pedestrians are usually not as close to the ego-vehicle as in our robotics scenarios. The multi-modal dataset by Z. Yan et al. (2018) covers a similar indoor robotics scenario as ours, but is only sparsely annotated and does not include full trajectories. Recently, Wengefeld, Mueller, et al. (2019) presented a novel multi-sensor dataset for indoor person guidance with a setup similar to ours. Though their annotated sequences are longer than ours (11.5 vs. 5 min), the average track count is significantly lower than in our crowded airport sequence (approx. 5 tracks vs. 25).

Most recently, Martín-Martín et al. (2019) introduced the JackRabbot social navigation dataset and benchmark (JRDB). Similar to us and Wengefeld, Mueller, et al. (2019), they acknowledge the need for a systematic benchmark for multi-modal human tracking from a robot-centric perspective, with a more diverse sensor setup than in the vision-based MOTChallenge. In contrast to our dataset, they provide a large-scale publicly available training and testing set with over 64 minutes of annotated data from 2D laser, 3D lidar, 360-degree fisheye camera and 360-degree cylindrical stereo recorded on a university campus in both indoor and outdoor settings. While we have made our motion capture sequence from a lab environment publicly

[3] A sampling-based variant of JPDA was used for people tracking by Schulz et al. (2001) and Topp et al. (2005). In robotics, JPDA appears not to be used recently anymore, though Rezatofighi et al. (2015) revived the technique for pedestrian tracking in computer vision by computing only the k strongest hypotheses using integer linear programming to make it tractable.

available, it was not possible to publish the crowded sequences from the airport environment for privacy and security reasons.

4.3 Our approach

In this section, we describe a new tracking system developed with robustness and computational efficiency in mind specifically for the deployment on resource-constrained mobile robots in crowded environments. Using a relatively cheap set of extensions from the target tracking community to systematically tackle shortcomings of current systems in such scenarios, we want to improve robustness without having to resort to multi-hypothesis tracking methods that are orders of magnitudes more complex in terms of implementation and computational requirements, and might break down under real-time constraints in very crowded environments, as we have seen in the previous chapter and others have found in similar experiments (Oh et al., 2009).

4.3.1 Data association

Correctly associating new detections with existing tracks is crucial for good tracking performance. On the other hand, related research by Correa et al. (2013) gives a first indication that for human tracking from a robot-centric perspective, with increasing environment complexity the difference in performance between tracking approaches based upon NN data association, JPDA and PHD filter approaches might diminish. Considering the fact that we want to track humans in highly crowded environments, we propose here to use a less complex single-hypothesis approach for data association and experiment with different nearest-neighbor variants.

We assume that detections arrive in their sensor-specific coordinate frame and are instantaneously transformed into a locally fixed frame (based upon robot odometry) that does not move with the robot. This ensures that the motion prediction of tracked persons is independent from the robot's ego-motion. In the resulting set of measurements $Z = \{\mathbf{z}_1, ..., \mathbf{z}_n\} \subset \mathbb{R}^2$, we have dropped the z coordinate as we currently only track in 2D world coordinates over the ground plane.

When new detections arrive, we first perform a standard χ^2 gating test of possible detection-to-track assignments. We then construct a rectangular assignment cost matrix for all validated pairings based upon the Mahalanobis distance between measurements predicted from track states $\mathbf{x} \in T$ and new measurements Z:

$$C = \begin{bmatrix} d_{11} & \cdots & d_{1j} \\ \vdots & \ddots & \vdots \\ d_{i1} & \cdots & d_{ij} \end{bmatrix} \quad i < |T|, j < |Z| \quad \text{with} \quad d_{ij} = \sqrt{(H\hat{\mathbf{x}}_i - \mathbf{z}_j)^{\mathrm{T}} S_{ij}^{-1} (H\hat{\mathbf{x}}_i - \mathbf{z}_j)} \tag{4.1}$$

with S_{ij}^{-1} being the associated innovation covariance matrix, $\hat{\mathbf{x}}_i$ the state prediction and H a measurement model that yields x, y positions in world space.

To implement global nearest-neighbor (GNN) data association, we compute the optimal solution to the resulting linear assignment problem using the method by Jonker and Volgenant (1987). We also implement a faster, greedy NN variant with unique assignments that instead searches for minima in the cost matrix, deletes corresponding rows and columns, and then proceeds with

subsequent minima. Out of curiosity, we further include a greedy variant that allows multiple associations per measurement by simply assigning to each track the closest measurement.

4.3.2 State estimation

Before data association, we predict the target motion by maintaining a Kalman filter for each tracked person. At a detection rate of around 30 Hz, human motion can in most scenarios be assumed to be locally linear, leading to the Nearly Constant Velocity model with state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]^T$ and transition matrix

$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

to predict a future state $\hat{\mathbf{x}} = F\mathbf{x}$, where Δt is the length of the tracking cycle. We further predict the track covariance using error propagation law and incorporate additive process noise with covariance Q to account for small changes in the velocity of the human motion:

$$Q = \begin{bmatrix} \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 & 0 & 0 \\ \frac{1}{2}\Delta t^2 & \Delta t & 0 & 0 \\ 0 & 0 & \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} q_{\mathrm{L}}. \tag{4.3}$$

The process noise level q_{L} varies with the dynamics of the application scenario, and in practice controls the filter's robustness to unexpected maneuvers.

Better motion prediction in dynamic scenarios[4]

We additionally integrate a bank of first- and second order motion models to cover non-linear aspects of human motion. This can be important especially in crowded environments, where people are forced to change their motion abruptly according to dynamics of the environment. In addition to Nearly Constant Velocity (CV), we model Brownian Motion (BM), Nearly Constant Acceleration (CA) and Nearly Coordinated Turn (CT) with additive white Gaussian noise.

In a master's thesis by Girrbach (2015), different combinations of these models have been systematically compared and combined in an interacting multiple models (IMM) filter. IMM is a suboptimal hybrid filter, which according to Mazor et al. (1998) achieves an excellent compromise between performance and complexity and can during maneuvers perform dynamic "soft switching" between different models, represented for instance by individual EKFs. In experiments on both synthetic and real data, Girrbach (2015) found that the exact combination of above single motion models (CV, BM, CA, CT) is not so important for tracking of people, as long as different levels of process noise are covered, which we adopt by combining two CV models with different process noise levels q_L.

4.3.3 Track initiation logic[4]

Especially in sparse 2D laser range data, false positive detections are frequent with current detectors and sensor technology. To reduce the resulting number of falsely initiated "ghost" tracks, we propose to use a rule-based approach to confirm track creation. It is based upon the track initiation logic described by Bar-Shalom (1987) which has been demonstrated to perform well in a statistical analysis (Hu et al., 1997) and practical evaluation for radar-based tracking (Leung et al., 1996). In comparison to other initiation techniques such as the track score approach from Blackman et al. (1999), which uses the innovation from the Kalman filter to recursively compute a track score, the logic-based approach is relatively easy to parameterize. We thus adapt it for the purpose of human tracking and describe the working principle of our method (Algorithm 1) in the following paragraph. A similar initiation logic, but with different constraints, was previously used for human tracking by Bellotto et al. (2010).

Algorithm 1: Cascaded logic for track initiation

Data: unmatched detections Z^u, initiation candidates C
Params: required match count n_{init}, velocity thresholds
Result: new tracks T_{new} to be initiated
foreach *existing initiation candidate* $\mathbf{c}_i \in C$ **do**
  foreach $\mathbf{z}_j \in Z^u$ **do**
    Predict next measurement from state of $\mathbf{c}_i$
    Check distance between prediction and $\mathbf{z}_j$
    Check velocity between $\forall \mathbf{z} \in Z_{c_i}$ and $\mathbf{z}_j$
    if *checks passed* **then**
      $Z^u = Z^u \setminus \{\mathbf{z}_j\}$
      if $|Z_{c_i}| \geq n_{\text{init}}$ **then**
        $C = C \setminus \{\mathbf{c}_i\}$
        Add $\mathbf{c}_i$ to T_{new}
      else
        Add $\mathbf{z}_j$ to $Z_{\mathbf{c}_i}$
        $n_{\text{miss},\mathbf{c}_i} = 0$
  if $\mathbf{c}_i$ *has no matching detection* $\mathbf{z} \in Z^u$ **then**
    $n_{\text{miss},\mathbf{c}_i} = n_{\text{miss},\mathbf{c}_i} + 1$
    if $n_{\text{miss},\mathbf{c}_i} > n_{\text{miss}}$ **then**
      Delete initiation candidate: $C = C \setminus \{\mathbf{c}_i\}$
foreach $\mathbf{z}_j \in Z^u$ **do**
  $C = C \cup$ new candidate $\mathbf{c}_k$ with $Z_{\mathbf{c}_k} := \{\mathbf{z}_j\}$

In each tracking cycle, every new measurement that has not been paired with a track during regular data association is passed on to the initiation logic. There, it either leads to the creation of a new *track candidate* and an associated Kalman filter, or it is greedily associated with an existing track candidate if it passes two gating tests. For a track candidate to be confirmed eventually, it must repeatedly be assigned new measurements that pass those gating tests.

The first gating test checks if the distance between a current measurement $\mathbf{z}$ and the measurement prediction $H\hat{\mathbf{x}}$ of a track candidate's Kalman filter is below a certain threshold. Initial experiments by Girrbach (2015) on an existing dataset had shown that the Euclidean distance is better suited to suppress FPs than the Mahalanobis distance, as visualized in Figure 4.2 (right).

[4] GNN, IMM and track initiation logic were implemented by my master student F. Girrbach under my supervision.

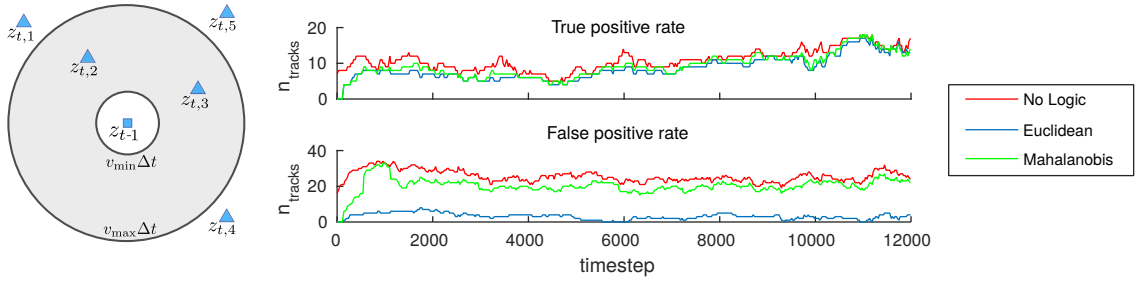


Figure 4.2: Logic-based track initiation. Velocity-based gating (left) and impact of Euclidean *vs.* Mahalanobis-based distance gating (right). (Figures adapted from Girrbach, 2015)

A second gating test ensures that linear velocities derived from relative measurement positions are within an interval of acceptable velocities $[v_{\min}, v_{\max}]$. Given the cycle time Δt, this translates into ring-shaped gates around previous measurement positions z_{t-1} as shown in Fig. 4.2 (left). If $v_{\min} > 0$, track initiation is effectively restricted to moving targets, which prevents tracks from being created when objects like trees or columns resemble humans in 2D laser. As an extension to the original method, in each tracking cycle we perform the velocity gating recursively for each observation with regard to all observations that have already been associated to the track candidate c_i. A track candidate must pass both gating tests at least n_{init} times before being confirmed. Confirmed candidates are transformed into regular tracks while retaining their filter states. If it misses measurements for more than n_{miss} times, the candidate gets deleted.

4.3.4 Track deletion logic

In the base version of our tracking system, we delete occluded tracks after a fixed number of tracking cycles n_{del} if no matching detection could be found. To prevent possible false alarm tracks from staying in the system for too long, we add a discrimination between 'young' and 'mature' tracks, where the latter is a track that over its lifetime has been matched against a measurement at least n^{mat} times. In case a young track is occluded, it is deleted after being without measurement for $n_{\text{del}}^{\text{yng}}$ cycles, whereas a mature track is deleted after a larger number of steps $n_{\text{del}}^{\text{mat}} > n_{\text{del}}^{\text{yng}}$ without associated detection.

4.3.5 Incorporating prior map knowledge

In our later experiments in Section 4.5, we want to examine how much we can benefit from incorporating knowledge about static structures from an existing occupancy grid map in order to suppress false positive detections. This is essentially a form of background subtraction that compensates for detector weaknesses. While such approaches have been used even before machine learning-based detectors became popular, an occupancy grid map is often available on a mobile robot for navigation purposes, and thus it makes sense to also exploit such prior knowledge. Unlike Leigh et al. (2015), we use an offline generated map to be more efficient.

To filter detections using a map, we rasterize a circle of fixed radius (*e. g.* 15 cm) around a given detection in the given occupancy grid map. If more than 10 percent of all covered grid cells are occupied, we consider the detection as a false positive and reject it.

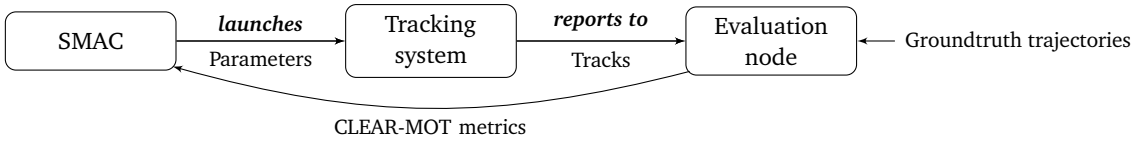


Figure 4.3: Integration of SMAC and tracking for hyperparameter optimization

4.3.6 Optimization of tracking hyperparameters

Even in a simple NN-based system, there can be over 20 inter-dependent parameters that can affect tracking performance and require expert knowledge when tuned manually. These parameters are often part of noise models which significantly abstract from the underlying models of reality, and can at times be even counter-intuitive to use. In particular, our experience is that 1) the process noise level q_L, 2) the measurement covariances R, 3) track initiation- and 4) track deletion thresholds can have a high performance impact and are difficult to estimate. Therefore, we propose to use automated hyperparameter optimization to ease the burden of finding good parameters and deploying our tracking framework in new scenarios.

To optimize the hyperparameters, we integrated pySMAC[5], a Python wrapper for the sequential model-based optimization library SMAC (Hutter et al., 2011), with our extended NN tracker. SMAC allows to define categorical, integer and float parameter ranges and boundary conditions to be met. It uses a given performance metric, in our case multi-object tracking accurancy (MOTA), to fit a surrogate model in the form of a random forest. The model is used for prediction of promising configurations, which are then optimized using a combination of Bayesian optimization and local search. As shown in Figure 4.3, we have integrated SMAC such that in each trial, it launches a new instance of the entire tracking system via `roslaunch`, while providing the current set of hyperparameters via the parameter server. A separate ROS node listens to the tracking output, computes evaluation results, and reports them back to SMAC.

4.4 Initial experiments using only 2D laser detectors[6]

We now present experiments that compare different variants of our method against a multi-hypothesis tracking approach on two datasets using a 2D laser-based detector.

4.4.1 Experimental setup

As tooling for the systematic evaluation of multi-modal people tracking algorithms under identical conditions, we used the ROS-based people detection and tracking framework which we describe in detail in Chapter 8.

[5] https://github.com/automl/pysmac

[6] The experiments concerning alternative NN variants, track initiation, IMM and hyperparameter optimization were part of a master's thesis and lab project by Girrbach (2015) which were supervised by the author of this thesis.

Figure 4.4: Exemplary groundtruth tracks of the Freiburg Main Station and PedSim datasets that we use in our experiments, and laser endpoints (in grey).

Datasets

We evaluate the proposed system on two datasets of different complexity. Exemplary screenshots are shown in Figure 4.4. The first one is a popular 2D laser dataset recorded with a stationary SICK LMS-291 laser scanner at Freiburg Main Station (Luber, Tipaldi, et al., 2011b). The second one has been generated synthetically using PedSim, and was introduced in Section 3.5.2.

Baseline tracking methods

Besides our own approach, we include the following baseline:

Multi-hypothesis tracker (Arras et al., 2008). This is a variant of the hypothesis-oriented multi-hypothesis tracker (HO-MHT) after Reid (1979) and Cox and Hingorani (1996) with explicit occlusions labels (Arras et al., 2008). This probabilistic method does not incorporate a track initiation logic; instead, new tracks are initiated following a Poisson process. Different from the original method, based upon qualitative results from the previous chapter we enforce track deletion after a fixed number of 10 cycles without association by setting p_{del} to 1.0 and all other priors to a very small number for such tracks before constructing the cost matrix.[7]. We use a constant velocity model and do not use any of the extensions by Luber (2014), as they are orthogonal to the data association method and could also be integrated into the other baselines. We manually tuned the hyperparameters of the method on the Main Station Dataset, yielding a scanback depth of $N = 6$, generating up to $k = 1000$ hypotheses with a time limit of 30 ms per cycle to ensure real-time performance. For evaluation, we always consider the best leaf hypothesis (=no smoothing) to reduce latencies.

Detector setup

For human detection, we first segment all 2D laser scans using agglomerative hierarchical clustering with a distance threshold of 0.4 m. Resulting segments are classified using an Adaboost

[7] In the crowded PedSim scenario, we observed exploding track covariances and tracks not getting deleted quickly enough. We believe this is due to $p_{\text{del}} << p_{\text{occ}} << p_{\text{det}}$, *i. e.* deletion is much less likely than for instance occlusion. In such crowded scenarios, the hypotheses which postulate a deletion might get pruned too quickly, or not even be among the k best global hypotheses.

	Main Station dataset					PedSim dataset				
Method	MOTA	ID	FP%	Miss%	Hz	MOTA	ID	FP%	Miss%	Hz
Global NN	71.9%	1280	8.4%	18.6%	6997	80.4%	4962	12.4%	5.7%	1323
Greedy NN, multi-association	70.7%	1439	9.8%	18.3%	6766	78.8%	5627	14.1%	5.5%	1403
Greedy NN	71.9%	1279	8.4%	18.6%	6976	80.5%	4968	12.4%	5.7%	1061
+Extended initiation logic	67.2%	781	2.5%	30.0%	7069	78.9%	3112	4.9%	15.3%	1260
+Deletion logic	66.7%	463	20.1%	12.7%	5732	71.1%	1842	25.0%	3.3%	747
+Base initiation logic	72.6%	274	6.6%	20.1%	6816	80.8%	1234	9.1%	9.6%	1025
+Extended initiation logic	73.3%	306	7.2%	19.3%	6472	82.2%	1315	9.4%	7.9%	935
+IMM (CV + CV)	73.3%	311	7.2%	19.3%	3433	82.2%	1272	9.4%	8.0%	606
MHT $t_{\max} = 0.03$s	72.2%	815	9.4%	17.4%	33	71.3%	3670	23.0%	4.8%	32

Table 4.1: Tracking results with a 2D laser detector

classifier with 500 decision stumps trained on a subset of the Freiburg Main Station dataset using the geometric features of Arras et al. (2007). We discard all segments with less than 3 points.

Evaluation metrics

We use the CLEAR-MOT metrics as described in Section 2.5.3. Milan et al. (2013) noted that MOTA scores can vary between implementations and are highly dependent on meta-parameters such as the matching distance threshold θ_d and the way in which track hypotheses are assigned to groundtruth objects. Therefore, we use the same implementation and parameters for all of the following evaluations.

We compute groundtruth correspondences using the Hungarian method, based upon Euclidean distances between object centroids in 2D world coordinates with a maximum gating distance of $\theta_d = 1$ m, similar to the 3D MOTChallenge. We ignore correspondences where the groundtruth track is physically occluded, which is determined by searching for associated laser points within a radius of 0.3 m of the annotated position, shifted towards the sensor origin by 0.2 m to take into account that the laser sensor only perceives the surface of the person.

4.4.2 Results

Table 4.1 shows tracking results of our extended NNT as well as the MHT baseline on the moderately crowded Freiburg Main Station and the very crowded PedSim dataset.

Impact of data association

The results for the different data association methods, namely NN with multiple associations per detection, Global NN and Greedy NN, show that the sub-optimal solution of the assignment problem of the greedy method yields nearly the same results as the optimal solution found by applying the Jonkers-Volgenant algorithm (Jonker and Volgenant, 1987). The NN variant that allows a detection to be assigned to multiple tracks yields higher results in MOTA, but also a higher number of ID switches. This can be explained by the fact that a track with a missing detection converges towards its nearest neighbor, which after a number of missed detections results in a track duplicate.

Impact of track initiation logic

The track initiation logic by itself drastically reduces the FP rate by around half, but also leads to an increased miss rate for three different reasons. First, by requiring at least n_{init}=6 matches in our case for a track confirmation, each track's appearance is delayed by the same number of tracking cycles. Second, for targets outside of our velocity boundaries of $[v_{\min}, v_{\max}] := [0.2, 2]$, such as static persons, no tracks are initiated[8]. Lastly, for targets that only infrequently trigger a detection, the number of consecutive allowed misses n_{miss} might be too low.

Impact of track deletion logic

Our track deletion logic has been tuned to delete young tracks after $n_{\text{del}}^{\text{yng}} = 20$ cycles and mature tracks after $n_{\text{del}}^{\text{yng}} = 50$ cycles. A track is being considered mature after at least $n^{\text{mat}} = 100$ matches. As can be seen from the results in Table 4.1, this initially leads to a worse MOTA score compared to the baseline NN tracker due to the increase in false positives, as certain tracks are now allowed to survive for a longer time compared to the case without deletion logic where they are already deleted after $n_{\text{del}} = 5$ cycles.

However, once deletion and initiation logic are combined, MOTA increases because the two extensions augment each other well. Combining the two, MOTA on the Main Station dataset rises from 71.9% to 73.3%, with an impressive reduction in ID switches from 1279 to 306. Similarly on the PedSim dataset, the number of ID switches is reduced by around 75%. Here, the extended version of the track initiation logic that recursively performs velocity gating against all previous detections that were already associated with the candidate, achieves 0.7–1.4% higher MOTA scores than the basic version which just performs velocity gating on the latest detection.

Impact of IMM

The IMM results in Table 4.1 were achieved using two CV models with different process noise levels q_{L_1}=0.035 and q_{L_2}=0.267. The IMM parameters and overall choice of motion models were found by automated hyperparameter optimization via SMAC. While MOTA did not improve by adding the IMM, the number of ID switches went down slightly by 3.6% on the more challenging PedSim dataset. Additional qualitative experiments in our lab, where multiple persons interacted with our robot in a narrow environment, showed a reduction in track losses and subsequent ID switches when using the IMM. We believe that in such human-robot interaction scenarios, the IMM can lead to more obvious improvements, because the human subjects often try to 'play' with the robot and trick the tracking system into errors by performing erratic maneuvers. This happens less often in our recorded datasets, which do not include explicit human-robot interactions and therefore contain fewer sharp turns and persons stopping abruptly.

[8] As discussed later, we can relax this constraint in the multi-modal scenario.

Comparison between NNT and hypothesis-oriented MHT

Compared to the baseline NN method, the hypothesis-oriented MHT – as expected – achieves better scores on the Main Station dataset[9]. This indicates that the manually tuned MHT hyperparameters provide a good configuration. On the highly challenging PedSim dataset, however, the MHT performs 11% worse in MOTA. We believe this to be a combination of two factors: First, the combinatorial explosion of possible data associations given the high track count[10] means that only few hypotheses can be generated within the given cycle time limit of 30 ms[11]. As discussed by X. He et al. (2013), a lot of these hypotheses might in fact be spent on reasoning about false alarms as opposed to actual measurement-to-track association[12]. Second, while the NNT maintains only a *single* hypothesis, it initiates and deletes tracks in a deterministic fashion. From the results in Table 4.1, we see that on the PedSim dataset MHT suffers especially from a high false positive rate; similarly, the NNT reached high FP numbers before we incorporated the initiation logic. Therefore, we speculate that MHT could also benefit from such a logic-based initiation mechanism, as opposed to the current probabilistic approach.

Track identity switches

Even with no initiation logic, the NNT produces only half the number of ID switches compared to the MHT. One reason for the higher number of ID switches by the MHT is likely a frequent switching of the global best leaf hypothesis; if among the alternative global hypotheses, tracks have evolved differently, they may bear different IDs. This is a well-known problem in multi-hypothesis tracking that is not straightforward to solve without *e. g.* an extra data association step at the output of the MHT.

Runtime performance

In the last column of Table 4.1, we show the median of the extrapolated processing rates based upon measured cycle times (without taking the detection stage into account). All implementations have been optimized for runtime performance. For an equal comparison, and with the application scenario of the service robot with limited on-board processing capabilities in mind, we only use a single CPU core.

It can be seen that the NNT, even with extensions, is extremely efficient due to its simplicity, being able to theoretically process over 5000 tracking cycles per second in moderately crowded scenarios (Main Station), and still over 600 in very crowded scenarios (PedSim). Looking at the cycle rates of the NNT on the PedSim dataset, which are in the order of around 1000 Hz, it easily becomes apparent why in such highly crowded scenarios with over 30 tracks, a hypothesis-oriented multi-hypothesis approach cannot succeed without massive parallelization.

[9] These results are 8% worse than the baseline results reported by Luber et al. (2013), because we track targets up to a max. range of 20 m (instead of 12 m) and use a different CLEAR-MOT implementation.
[10] Arras et al. (2008) generate more than 500 hypotheses for only four tracks.
[11] If we do not impose a cycle time limit, we obtain a negative MOTA of -22.6%.
[12] See also our discussion at the end of Sections 3.5.2 and 3.6.

Since in the MHT, the runtime complexity for the generation of k hypotheses[13] for n targets is in the order of $\mathcal{O}(kn^3)$, we cannot expect significantly more than 1000 Hz : 500 hyp. $\approx$ 2 Hz on a single CPU core assuming we want to generate at least 500 hypotheses. Even on a processor capable of executing 8 threads in parallel, the expected frame rate would drop below 20 Hz without yet running any detection or higher-level perception components.

4.5 Experiments using multi-modal detections on the airport and motion capture datasets

The previously described experiments have been conducted with a single sensor modality. In this section, we conduct experiments on two novel datasets with a multi-modal detector setup. We incorporate further tracking baselines in order to gain additional insights.

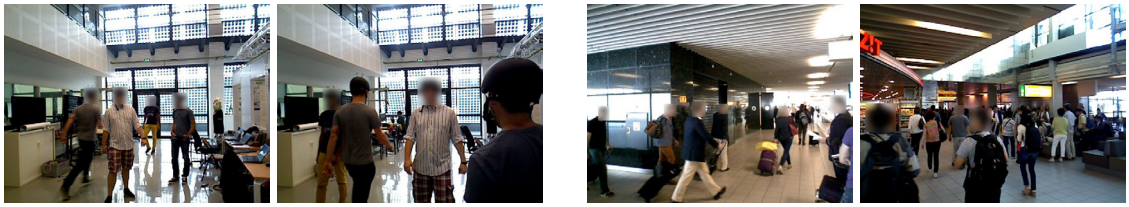
4.5.1 Experimental setup

Datasets

Figure 4.5 shows two novel multi-modal datasets that we have recorded. The *Motion Capture Sequence* has been acquired in a narrow lab environment in front of the SPENCER robot platform, which remains stationary throughout this sequence. The recorded sensor data includes a 190-degree frontal 2D laser scan from a SICK LMS 500 sensor, and the data from a front-facing Asus Xtion Pro Live RGB-D sensor. In this four-minute sequence which we made publicly available, four persons wear motion capture markers on their heads for groundtruth acquisition and walk around in highly dynamic and erratic patterns, such that they frequently occlude each other and accelerate or stop abruptly. This dataset is mainly intended to simulate human-robot interaction, in which case people often 'play' with the robot, or try to challenge its tracking abilities.

The second sequence, *Airport Sequence 03*, is part of a much larger dataset that was recorded at Amsterdam-Schiphol airport using our moving sensor platform that closely replicates the sensor setup on the SPENCER robot. The dataset includes 2D laser range data from two back-to-back SICK LMS 500 scanners at 70 cm height that cover a 360-degree field of view around the robot. It also includes RGB-D data from two Asus sensors mounted in horizontal orientation and facing into forward and rearward direction. In the first half of this sequence, the sensor platform remains stationary and observes a dense flow of passengers disembarking from an airplane. In the second half, the platform joins the flow of people towards a large, open area inside the terminal. During the entire 4-minute sequence, the platform is almost constantly surrounded by 20–30 persons that follow various motion patterns at different walking speeds and undergo many severe occlusions. In total, 172 ground truth tracks have been manually annotated using our trajectory-based annotation tool (see Section 8.8) up to a distance of 12.0 m, beyond which annotation becomes increasingly different due to extreme occlusions and increasing inaccuracy in sensor calibration. We ignore all tracks and groundtruths at further distances.

[13] This was shown by Luber (2014) for the implementation based upon Murty's method that we use.



(a) Motion Capture Sequence **(b)** Airport Sequence 03

Figure 4.5: Exemplary RGB frames (front sensor) from our new multi-modal datasets.

Baseline tracking methods

In addition to our extended nearest-neighbor tracker and the HO-MHT from Section 4.3, we include the following tracking approaches in our experimental comparison:

Nearest-neighbor tracker (Dondrup et al., 2015). This is a different NN implementation based upon the Bayes tracking library by Bellotto et al. (2015) which uses GNN in combination with an EKF with constant-velocity model and Cartesian measurement model. Tracks are initiated if a minimum number of detections occur within a small radius, and deletion takes place if the track covariance exceeds a certain limit. We use the recent implementation with ROS integration by Dondrup et al. (2015) with their provided set of parameters.

We were not able to include the alternative NN-JPDA implementation that they also provide, as it exhaustively computes all possible associations (without *e. g.* performing distance-based gating), which clearly exceeds the available computational resources and real-time budgets in the crowded scenario that we consider in our experiments.

Vision-based MDL tracker (Hosseini Jafari et al., 2014)[14]. This method is based on the work of Leibe et al. (2008) and uses the tracking framework of Ess et al. (2009) and Schindler et al. (2010) to build an overcomplete set of track hypotheses, similar to TO-MHT. Via bi-directional EKF new trajectories are generated for the current tracking cycle by following the motion model backwards in time, while existing trajectories are extended from the previous to the current cycle. Each track is then scored individually based upon the motion model, detection confidences and color-based appearance of inlier detections, adapted on the fly. The interaction cost between tracks takes physical overlap and shared detections into account. Selecting the best subset of hypotheses from the score matrix is then formulated as a quadratic binary problem. It is solved using a minimum description length (MDL) framework with the multi-branch method of Schindler et al. (2006).

Tracking hyperparameters

As highlighted by Milan et al. (2013), for evaluations of multi-person tracking it is important that parameters of the tracking algorithm are tuned on a separate validation dataset to verify its generalization capabilities and avoid overfitting. With this in mind, we carefully tuned all algorithms on separate, similar, but not identical datasets. Specifically, for tuning the NNT, our

[14]The experiments with and integration of the MDL tracker have been performed by S. Breuers.

extended NNT, and the MHT, we used the laser-based Freiburg Main Station dataset as well the synthetic PedSim dataset from the previous section. The extended NNT has been tuned automatically via SMAC. The vision-based MDL tracker has been manually tuned on the ETH dataset (Ess et al., 2008) that was recorded in a pedestrian zone.

Because the tracking methods that we consider in our evaluation require a large number of different hyperparameters, we do not report all of them here in detail. Instead, we provide the exact ROS launch files with the corresponding configurations that we used online in our Github repository[15] to allow replication of our experiments.

Detector setup

We now briefly describe the multi-modal detector setup that we will use in the following experiments. We will discuss more recent methods for human detection in RGB-D in Chapter 5, while in Chapter 6 we provide a systematic evaluation of detectors for 2D/3D lidar. For details on how all these detectors have been integrated, see Chapter 8.

2D laser. For human detection in 2D laser range data, we use a random forest classifier trained on the geometric features described by Arras et al. (2007) with 15 trees and maximum depth of 10. It has been trained on 2D laser range data that we recorded along with the Rathausgasse dataset (see Chapter 3) using a SICK LMS 500 laser scanner at a height of 70 cm and 0.25 degrees angular resolution. We use jump-distance clustering for segmentation of the laser scans.

Depth images. We include the depth template-based upper-body detector proposed by Mitzel et al. (2012), which runs in real-time at 20-30 Hz on the CPU. It uses the depth images from the Asus Xtion Pro Live RGB-D sensors. We use the provided model pre-trained on their dataset.

RGB images. We further include a GPU-accelerated person detector ("groundHOG") that relies on HOG features from the RGB image provided by the Asus Xtion Pro Live sensors. It uses the estimated person height over the ground plane to estimate depth, as described by Sudowe et al. (2011). We use their implementation and trained model.

Multi-modal detection fusion

Except for the MDL tracker, which is hardwired to the upper-body and groundHOG detectors, all other baselines natively support only a single source of measurements. We therefore incorporate a simple detection-to-detection fusion stage before the tracking stage to add support for our multi-modal multi-sensor setup. This ensures that the trackers are provided with the same set of fused detections, and avoids many changes that would be required within the tracking methods to implement detection-to-track fusion. In the fusion stage, using greedy NN association, we first fuse detections from sensors with overlapping fields of view (*e. g.* front laser, front RGB-D) and then aggregate the resulting sets of detections that do not overlap (*e. g.* front and rear detections). As association cost, we either use the Euclidean distance between individual detections, or –

[15] https://github.com/spencer-project/spencer_people_tracking, "icra16" branch, in the sub-folder: `tracking/people/srl_nearest_neighbor_tracker/launch/icra16_experiments`.

when the groundHOG detector is involved – a cost computed in polar coordinates that penalizes discrepancies in distance less strongly. Other methods, *e. g.* based upon covariance intersection (Martin et al., 2006), could easily be integrated.

Further details on the experimental setup

All of our experiments were conducted on a high-end gaming laptop equipped with a quad-core Intel Core i7-4700 MQ processor and 8 GB of RAM under Ubuntu 14.04 with ROS Indigo. Each single experiment has been run at least three times and metrics have been averaged to ensure stable results that are not negatively affected by the not fully deterministic message passing, synchronization and transform lookups in ROS. For the computationally more complex experiments on the airport dataset sequence, we have pre-recorded all detections to ensure that all tracking algorithms are always fed with the same input for a fair comparison. On the motion capture sequence, we limit the maximum evaluation distance such that persons in the background who are not wearing motion capture markers are ignored. We use the CLEAR-MOT metrics implemented as in Section 4.4, but report relative instead of absolute ID switches and in addition mostly tracked (MT) and mostly lost (ML) tracks as explained in Section 2.5.3.

4.5.2 Results using multi-modal detections

Quantitative results

In Table 4.2, we present quantitative results of the considered tracking methods for different combinations of sensor modalities on our test sequences. Always in the same order, we look at the following methods:

- Nearest-neighbor tracker (NNT) as described by Dondrup et al. (2015),
- Our extended nearest-neighbor tracker (NNT++) as proposed in Section 4.3,
- Hypothesis-oriented MHT based upon work of Arras et al. (2008),
- Track-oriented MDL tracker as described by Hosseini Jafari et al. (2014).

For the motion capture sequence, we additionally include results that have recently been published by Wengefeld, Mueller, et al. (2019) for their approach (W2019), and the method of Volkhardt et al. (2013) (V2013), using the same detector setup as in our experiment. Dashes indicate measures not reported in their paper. We do not report mostly lost tracks (ML) for this sequence as it was always zero for all our methods.

As the MDL tracker by Hosseini Jafari et al. (2014) relies on appearance information for data association and therefore only supports image-based detections from the upper-body and groundHOG detectors, we only use it in experiments with the front RGB-D sensor.

	Airport Sequence 03							Motion Capture Sequence					
Method	MOTA	rIDS	FP%	Miss%	MT	ML	Hz	MOTA	rIDS	FP%	Miss%	MT	Hz
NNT	27.7%	227	39.4%	**32.5**%	**92**	**47**	**13701**	60.7%	**131**	23.6%	**14.3**%	4	**20726**
NNT++	**44.4**%	**210**	13.1%	42.1%	63	60	4287	**69.8**%	151	7.8%	20.9%	4	5637
MHT	26.9%	338	39.4%	33.0%	87	51	28	57.9%	173	24.7%	15.6%	4	28
MDL	43.7%	428	**12.5**%	43.1%	36	59	53	60.7%	373	**4.8**%	31.3%	1	139

(a) Using only front RGB-D detectors (evaluation with small FOV)

	Airport Sequence 03							Motion Capture Sequence					
Method	MOTA	rIDS	FP%	Miss%	MT	ML	Hz	MOTA	rIDS	FP%	Miss%	MT	Hz
NNT	59.7%	**236**	20.8%	**19.3**%	**112**	27	**6184**	25.6%	**54**	70.4%	**3.3**%	4	**17968**
NNT++	**62.8**%	331	**3.4**%	33.5%	68	35	2307	**68.8**%	60	**25.3**%	5.2%	4	4988
MHT	58.9%	700	16.6%	23.9%	85	**26**	29	28.0%	85	67.3%	3.8%	4	28

(b) Using only the 2D laser detectors (evaluation with large FOV)

	Airport Sequence 03							Motion Capture Sequence					
Method	MOTA	rIDS	FP%	Miss%	MT	ML	Hz	MOTA	rIDS	FP%	Miss%	MT	Hz
NNT	45.7%	325	36.4%	**17.7**%	**123**	**19**	**4590**	14.8%	**55**	81.7%	**2.7**%	4	**15703**
NNT++	**62.1**%	**313**	**8.2**%	29.4%	96	26	2005	**74.9**%	58	**20.1**%	4.3%	4	4690
MHT	46.3%	692	34.9%	18.2%	117	22	31	8.6%	74	87.6%	2.9%	4	29

(c) Multi-modal detector setup (evaluation with large FOV)

	Airport Sequence 03							Motion Capture Sequence					
Method	MOTA	rIDS	FP%	Miss%	MT	ML	Hz	MOTA	rIDS	FP%	Miss%	MT	Hz
NNT	62.1%	**226**	18.7%	**19.0**%	**114**	27	**6100**	18.1%	**52**	77.6%	3.7%	4	**15857**
NNT++	**64.2**%	262	**3.3**%	32.4%	77	33	2222	**77.4**%	62	16.5%	5.4%	4	4744
MHT	60.2%	676	17.2%	22.0%	97	**24**	29	17.8%	76	77.7%	**3.6**%	4	28
V2013	–	–	–	–	–	–	–	61.0%	158	11.0%	26.5%	–	–
W2019	–	–	–	–	–	–	–	66.6%	59	**1.7**%	31.9%	–	>625

(d) Multi-modal without HOG (evaluation with large FOV)

Table 4.2: Tracking results with different detector setups

Comparison of the different tracking approaches

If we compare the results of different tracking approaches under identical conditions (sequence, modality), we note that the approaches based upon NN data association often obtain the highest MOTA scores. This might be due to a lower number of parameters, which could result in better generalization capabilities with regard to new scenarios.

Our proposed approach, the extended NNT, outperforms all other methods in the MOTA metric on all scenarios. This, first of all, is an indication that our automated hyperparameter optimization (with MOTA as objective to maximize) worked, and found a set of parameters that generalizes well to these new scenarios. We also see that NNT++ achieves very low FP%, most likely due to its track initiation and deletion logic. Especially on the *Motion Capture Sequence* with four groundtruth tracks, one or two ghost tracks are enough to cause bad FP scores for methods

(a) (b) (c)

Figure 4.6: HOG failure cases

Method	MOTA	rIDS	FP%	Miss%	Hz
	Motion Capture Sequence				
NNT	78.0%	**50**	18.5%	**2.9%**	**19050**
NNT++	**89.4%**	59	**5.3%**	4.6%	4926
MHT	73.8%	71	21.9%	3.4%	28

Table 4.3: Multi-modal + static map

without a sophisticated initiation logic, such as the basic NNT and MHT. Interestingly, both of these perform very similarly in most of the tested scenarios concering MOTA and FP%. On the other side, the extended NNT often obtains the highest miss ratio, which is likely caused by the delayed track initiation.

Both multi-hypothesis methods seem to suffer from frequent switching between hypotheses, leading to a high number of relative ID switches. We already discussed this problem in Section 4.4.2. The MDL-Tracker gives best FP% and thus a MOTA score comparable to the one of extended NNT on the front RGB-D only scenario (Table 4.2a). Note that the HO-MHT might generally obtain better results if priors and parameters such as new track rates were re-tuned on the test sequences, which we did not do here to test generalization capabilities; or if it were allowed to delay decision making in a fixed-lag smoothing sense, which however can be problematic for real-time motion planning applications due to the introduced delay, and leads to lower MOTA when comparing against the most recent groundtruth.

The simple NNT tracker dominates in the number of consistently tracked targets, *i. e.* higher MT and lower ML, due to a more straightforward initiation of tracks. The results reported by Wengefeld, Mueller, et al. (2019) indicate that their NN method and the earlier one by Volkhardt et al. (2013) perform significantly better in MOTA than our NN baseline, but cannot outperform our NNT++ with its initiation logic. In general, they trade in recall for a very low false positive rate. In their paper, they note that their method is able to initiate tracks also for static persons, and thus cannot show its full potential in the dynamic motion capture scene where every person is constantly moving. However, we want to note that the latest version of our system can override the minimum-velocity constraints during track initiation for specific detectors that are known to output high-confidence detections (*e. g.* in RGB-D). Therefore, our proposed approach can also track standing persons.

Laser-only vs. multi-modal detections

Next, we want to investigate the benefits of the multi-modal sensor platform and the use of both 2D laser and RGB-D sensors. On the airport sequence, incorporating vision-based detections from groundHOG and the upper-body detector increases the number of mostly tracked targets. This leads to a higher track recall and lower miss ratio for all approaches, at the cost of an increased FP%, ultimately resulting in a lower MOTA score (Table 4.2c). A more sophisticated fusion scheme of the different detector outputs might yield some improvement, however, a visual inspection of our naïve fusion scheme reveals no immediately apparent problems. Instead,

further experiments reveal that the HOG detector causes many false alarms (Figure 4.6) and provides imprecise depth estimates for distant persons, obtained by projecting image footpoints onto the estimated ground plane. Table 4.2d shows the multi-modal result without HOG, but still using upper-body detections from the RGB-D sensor. The FP% decreases, but unfortunately also MT goes down and miss rate increases. Anyhow, general tracking quality improves, which is reflected in the highest MOTA score for each tracking approach so far using this configuration.

On the *Motion Capture Sequence*, all methods struggle with an extremely high FP%, except for the extended NNT, whose extensive track initiation logic can again compensate for false alarms. The resulting discrepancy of the MOTA scores is huge (75% vs. 8-15%). It seems here that the laser detector – instead of HOG – is responsible for most of the false positives, often in chairs and other furniture: when using only front RGB-D (Table 4.2a), FP% is lower by around 50 percentage points.

Impact of filtering detections by an a-priori map

In Table 4.3, we show how filtering of fused detections by an occupancy grid map can improve tracking performance in comparison to Table 4.2c (right side). In our case, this map has been generated using an offline mapping solution. As we did not yet have a map available in the airport environment, we restricted this experiment to the *Motion Capture Sequence*, where it leads to a substantial increase in MOTA of 15–65 percentage points for the different tracking methods, highlighting that there is still significant room for improvement on the detection side.

Qualitative results

In Figure 4.7 and Figure 4.8, we show illustrative results of our tracking system on our airport test sequence. Qualitatively, we observe that at the detection stage, different detectors (2D laser, upper-body on depth images, groundHOG in RGB) complement each other well and significantly increase the field of view around the robot in which persons can be tracked. In certain scenarios, it might make sense to filter out tracks in a post-processing step which have not been visually confirmed (shown in blue in the last row of Figure 4.8), as especially the laser-based detector can sometimes trigger systematic false alarms when objects appear human-like in the laser scan. More qualitative results in the form of video sequences can be found online.[16].

Runtime performance

Our system with the entire multi-modal detector setup comprising two upper-body detectors, two groundHOG detectors and two 2D laser detectors for 360-degree coverage as well as the NNT++ runs in real-time at 20–25 Hz on two Intel Core i7-4700MQ gaming laptops with 8 GB RAM and nVidia GeForce GTX 765M graphics[17]. These laptops are dedicated to perception tasks on the robot, see Section 9.4 for details on the architecture. The computationally most expensive

[16] http://www.youtube.com/spencereuproject

[17] The GPUs are only used by the groundHOG person detectors. For the final deployment of the robot, we disabled those because they are detrimental to tracking performance, as discussed. For the combination of a single upper-body detector and a single 2D laser detector, one laptop is sufficient.

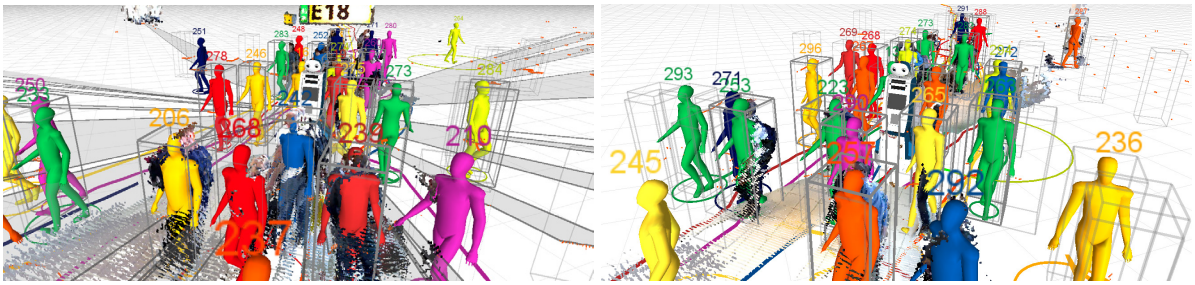


Figure 4.7: Qualitative tracking results within crowds at the airport

components are the person detectors, each requiring about one CPU core, with the tracker itself requiring only around 10% of a single CPU core when 20–30 tracks are being tracked.

4.6 Discussion

In the following, we discuss the most important conclusions that can be drawn from our experiments on the crowded airport dataset and the dynamic motion capture sequence.

Detector performance matters more than the choice of tracking algorithm

A major observation that we made during our experiments is that *detector performance is the single, most important factor influencing tracking performance* which goes far beyond the impact of the chosen tracking algorithm. In a nutshell, none of the examined tracking methods deal really well with high false positive rates. During initial experiments, we used a 2D laser detector trained on a different sensor model, and using a less restrictive selection of positive and negative training samples. This detector caused extremely bad MOTA scores between -3.3 and -1.3 due to enormous false positive rates ($> 200\%$). None of the examined methods could cope with this high number of false positives, which occur *systematically* and *repeatedly* at the same locations. Although the track initiation logic of the extended NNT was able to suppress a significant amount of the false positives, MOTA still did not exceed -1.3. Even though multi-hypothesis trackers have been shown to work well in (random) high clutter in radar tracking (Blackman et al., 1999), the worst MOTA score was obtained using the HO-MHT which does not possess any dedicated track initiation logic. After incorporating the static occupancy grid map to filter out false detections beforehand, MOTA scores of all examined approaches became positive, but were still significantly below the levels of Table 4.2.

Integrating 2D image-based detections

While a vision-based tracker can significantly benefit from 2D image-based detections (e. g. from HOG) that extend its maximum tracking distance beyond the useful working range of RGB-D sensors (around 6–7 m), their depth estimates are often very imprecise[18]. In a multi-modal setup where precise laser measurements are available ($\sigma \approx 3$cm), using image-based detections as a

[18] See for example the long association arrow in Figure 4.8, middle row.

Figure 4.8: Multi-modal human detection and tracking from a mobile robot in a crowded airport environment. *First row:* Color images of the upper rear and front RGB-D sensors, with projected bounding boxes of tracked persons and laser points. *Second row:* 2D laser detections (orange), upper-body (cyan) and groundHOG (yellow) along with RGB-D and 2D laser point clouds. Fused detections are shown as grey cylinders, pink arrows indicate associated detections. The area visible to the laser scanners is shaded in grey. *Third row:* Resulting person tracks. As opposed to blue persons which are only tracked in laser, red tracks have been confirmed by a visual detector.

direct input to data association in world space may therefore be detrimental. Wengefeld, Mueller, et al. (2019) make similar observations. Instead, more accurate depth estimates could be obtained by *e. g.* projecting lidar points into detection bounding boxes, or associating detections in image space (Spinello et al., 2010). More recently, detection approaches have been presented that operate jointly on images and point clouds and combine these representations in a more sophisticated way, for example the methods by Vasquez et al. (2017) and Qi et al. (2018). We will further discuss the issue in the next chapter.

The impact of tracking hyperparameters

Our experience shows that the correct choice of parameters significantly outweighs the choice of data association. Tracking approaches with only few parameters may generally be the preferrable choice, as they may generalize better towards new scenarios. Especially complex, probabilistic multi-hypothesis approaches often require re-learning or manual tuning of parameters by an expert. For example, new-track or deletion Poisson rates can vary depending on the given scenario and also location and time (e. g. when a new plane arrives at an airport and the passengers start disembarking). Automatic parameter learning approaches as we introduced them in Section 4.3 may help to simplify the process. To make our results easier to reproduce and allow researchers from other fields (e. g. HRI) to benefit from our findings, we have shared the parameter configurations used in our experiments online.

Tracking metrics trade-off between FPs, miss rate and ID switches

Another lesson we learn from our experiments is that the choice of parameters greatly depends on the desired application that relies on the people tracking system. There appears to be no universal set of parameters that fully accommodates all requirements, as a trade-off has to be made between attaining a low false positive count, a low miss rate, and a low number of ID switches. The first two can be important when using the people tracker output for socially aware navigation, since high false positive rates (i. e. ghost tracks) could freeze the robot, while missed tracks can cause the robot to behave impolitely or even endanger people. On the other hand, in person guidance scenarios, it is of utmost importance to maintain the ID of a tracked person as long as possible, while false positives might not be such a large issue.

As shown in our experiments, low false positive rates can be achieved by a dedicated track initiation logic, pre-filtering on a static map, and early track deletion. The first two options can cause the tracker to miss certain (e. g. static) tracks, while the last option may result in ID switches if the track suddenly reappears after an occlusion.

Importance of standardized tracking metrics

Even minor differences in tracking metrics implementation or its parameters can have significant influence on results. We agree with findings from the vision community (Milan et al., 2013) which underline the importance of using a standardized evaluation script and the same detection input for all tracking systems. Our tracking framework that we describe in Chapter 8 is, to our knowledge, the first that allows for *multi-modal* data annotation in RGB-D, 2D laser and 3D lidar,

and enables a systematic evaluation and comparison of different tracking methods and detectors in a unified framework.

Which tracking approach to choose?

Finally, we attempt to answer the question which of the examined tracking methods to choose for real-time people tracking from a mobile platform in very crowded and dynamic environments. Looking at the multi-modal results on the motion capture sequence (Table 4.2c, right), we see that the same, underlying NN data association method delivers an astonishing difference in MOTA performance of 60%, depending on the presence or lack of a dedicated track initiation logic. On the other hand, on the crowded airport sequence, we observe only a a 0.5% difference in MOTA between simple NN and sophisticated MHT data association. This is consistent with findings by Correa et al. (2013), who noted that with increasing scene complexity, the difference in performance between evaluated data association methods decreases.

Therefore, as already hinted at in the previous section, it appears that incorporating promising tracking extensions (e. g. Luber et al., 2010; Luber et al., 2013) into a simple data association scheme might be the way to go. The computation time which is saved by refraining from using a more complex multi-hypothesis data association method could instead be spent on higher-level perception, or to improve detector performance, which has a high impact as previously discussed. Both of the discussed NN-based approaches are relatively easy to configure and generalize well across test sequences, and run at low CPU usage (<10% on a single core) – which is crucial on a mobile service robot platform that also needs to localize itself, plan and navigate.

The hypothesis-oriented MHT after Reid (1979) and Cox and Hingorani (1996) may be at disadvantage in very crowded environments, as discussed in Section 3.6. Since the entire state of the scene is represented within each single hypothesis, a very large number of hypotheses may be needed to adequately represent all likely combinations of possible track states. Generating as many hypotheses as possible within a given time window, as in our experiments, ensures a certain minimum cycle rate to be met, but may result in only few hypotheses being generated.

Finally, scenarios where some delay in decision making can be tolerated, such as offline video analysis or static observation of people behavior, allow for a different mode of evaluation where the delayed selection of the best hypothesis can be taken into account, by deferring matching with the groundtruth by a certain number of tracking cycles. We believe that in these cases, the multi-hypothesis approaches can show their full potential and attain higher scores – although in such cases, global optimization approaches as typically used in computer vision might be an even more attractive choice. Wojke et al. (2016) show an application of this in a robotics context.

4.7 Conclusions

In this chapter, we have evaluated four different tracking approaches of varying complexity, in order to study them on two challenging new, multi-modal datasets – one of them recorded from a static platform in a highly dynamic HRI scenario, and another one from a moving platform inside a crowded airport terminal. We have carefully analyzed the performance of

these existing methods with regard to multiple tracking metrics under different multi-modal configurations, and identified and extensively discussed their strengths and weaknesses that may guide possible future directions of research. In particular, we have demonstrated that the choice of data association method matters less than expected, and more time should be invested into developing stronger detectors, and properly tuning the hyperparameters of the tracking algorithms.

For practical applications on resource-constrained service robots, simple data association methods combined with effective extensions like a track initiation logic can be a better choice than highly complex multi-hypothesis approaches. To demonstrate this, we presented a very efficient tracking approach that combines standard techniques, but is able to outperform much more sophisticated techniques in the MOTA metric in almost all experiments. A similar discovery was recently made by Weng and Kitani (2020). They propose a simple baseline for 3D object tracking which achieves top 3D MOT performance on the KITTI dataset, by combining a state-of-the-art 3D lidar detector for cars with simple NN association using the Hungarian method and a constant velocity model without appearance information. They integrate a simple initiation and deletion logic and propose to evaluate in world space coordinates like we did.

One of our – also for the further research in this thesis – most important findings was that detector performance is the single, most influential factor on tracking performance. After we reported these findings in Linder, Breuers, et al. (2016), also several other researchers in the computer vision community independently made similar observations: Wojke, Bewley, et al. (2017) report that their nearest-neighbor tracker with deep visual association metric and a state-of-the-art person detector ranks higher on the MOT challenge than an MHT approach with standard detections. They conclude that *"This not only underlines the influence of object detector performance on overall tracking results, but is also an important insight from a practitioners point of view"*. In their survey on video-based object tracking, Ciaparrone et al. (2019) find that the best-ranking methods on each dataset use their own detectors, *"confirming the fact that the detection quality dominates the overall performance of the tracker"*.

Based upon these observations, in the following two chapters we will focus our research on human detection in RGB-D, 2D laser and 3D lidar in order to make human tracking more robust in challenging environments.

Ideas for improvement

We believe the following points, which in prior work have been demonstrated to be helpful individually, could also benefit our tracking approach:

- In the multi-modal experiments, we performed detection-to-detection fusion to able to provide a single set of detections to every examined tracking method, in order to make results comparable. However, it is generally accepted that fusing detections inside the tracker yields better results due to more informed decision-making. In that case, measurements from different detectors could be fused asynchronously, and the tracker could run at a sensor-independent frame rate. Bellotto et al. (2009) and Wengefeld, Mueller, et al. (2019) choose such an approach.

- Detection scores should be considered in the tracking process, and differences between different detectors should be taken into account. Wengefeld, Mueller, et al. (2019) present one way of accomplishing this.
- Appearance-based cues, for instance through online learning (Luber, Spinello, et al., 2011; Munaro et al., 2012) or deep associative embeddings (Wojke, Bewley, et al., 2017), can certainly improve data association. A key question is how to handle this in a multi-modal setup, when a person might occasionally only be seen for instance by 2D laser. In case of 3D lidar, height-based cues could help.
- On the example of the RGB groundHOG detector, we saw that it can be difficult to correctly associate detections in world space when depth estimates from the detector are inaccurate at larger distances. While we will try to improve this on the detector side in the next chapter, it would also make sense to allow for larger depth estimate uncertainty during data association in the tracker. To this end, Bellotto et al. (2010) use a polar (range-bearing) measurement model in combination with an Unscented Kalman Filter (UKF) to deal with resulting non-linearities. A similar approach with an image-specific measurement model, but with an EKF, was chosen by Kollmitz et al. (2019). We use association in polar coordinates only during detection-fusion, where it does not help if there is no other (more accurate) detection of the same person.
- Also in the simpler NN approach, motion prediction under temporary occlusion could likely be improved by incorporating a social force model (Luber et al., 2010) and information from group tracking (Luber et al., 2013).
- During lengthy occlusions, positional uncertainty of tracks grows so quickly that they need to be deleted after 1–2 seconds. Unless appearance is incorporated into the data association likelihood and combined with delayed decision-making or global optimization, only person re-identification might help to recover track identities in such cases.

  We conducted initial experiments in this regard using the method by Hermans et al. (2017), but found three so far unresolved problems in this challenging open-world re-id case: 1) Varying lighting conditions lead to larger intra-person distances within the associative embedding than inter-person. 2) Color-based features are not very informative when people wear similar (*e. g.* professional) clothing, thus maybe skeletal information should be considered (Munaro, A. Basso, et al., 2014). 3) Detector failures "pollute" gallery images with background, or badly aligned foreground, even if combined with tracking (as they are systematic in nature). This is a point we want to improve upon in the following chapters.

Part III

Multi-modal human detection for mobile robots

CHAPTER 5

Training 3D person detectors using low amounts of real-world RGB-D data

While 2D image-based object detection has made significant progress, robustly localizing objects in 3D space under presence of occlusion is still an unresolved issue. As we saw in the previous chapter, inaccurate 3D localization can negatively influence performance of tracking in world space coordinates. Therefore, our focus in this chapter is on accurately detecting and localizing 3D human centroids in RGB-D data. We first examine how well current 2D detectors cope with challenging situations in intralogistics scenarios. We then propose a fast image-based detection approach which extends the YOLO v3 architecture with a 3D centroid loss and mid-level feature fusion to exploit complementary information from both modalities. We employ a transfer learning scheme which can benefit from existing large-scale 2D object detection datasets, while at the same time learning end-to-end 3D localization from our novel highly randomized, diverse synthetic RGB-D dataset with precise 3D groundtruth. We propose a depth-aware crop augmentation scheme for training on RGB-D data, which helps to improve 3D localization accuracy. In experiments on our challenging intralogistics dataset, we achieve state-of-the-art performance when learning 3D localization just from synthetic data.

Parts of this chapter have been submitted to the IEEE International Conference on Robotics and Automation (ICRA) 2020 in Paris, France. The submission "Accurate Detection and 3D Localization of Humans using a Novel YOLO-based RGB-D Fusion Approach and Synthetic Training Data" by T. Linder, K. Y. Pfeiffer, N. Vaskevicius, R. Schirmer and K. O. Arras is currently under review.

Parts of this chapter, in particular Section 5.3, have been previously presented in an extended abstract and a poster with the title "Towards Accurate 3D Person Detection and Localization from RGB-D in Cluttered Environments" by T. Linder, D. Grießer, N. Vaskevicius and K. O. Arras at the Workshop on Robotics for Logistics in Warehouses and Environments Shared with Humans during the IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS) 2018 in Madrid, Spain.

Parts of this chapter, in particular Section 5.5, have been presented in a workshop paper and poster "Towards Training Person Detectors for Mobile Robots using Synthetically Generated RGB-D Data" by T. Linder, M. J. Hernandez Leon, N. Vaskevicius and K. O. Arras at the 3D Scene Generation Workshop during the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2019 in Long Beach, CA.

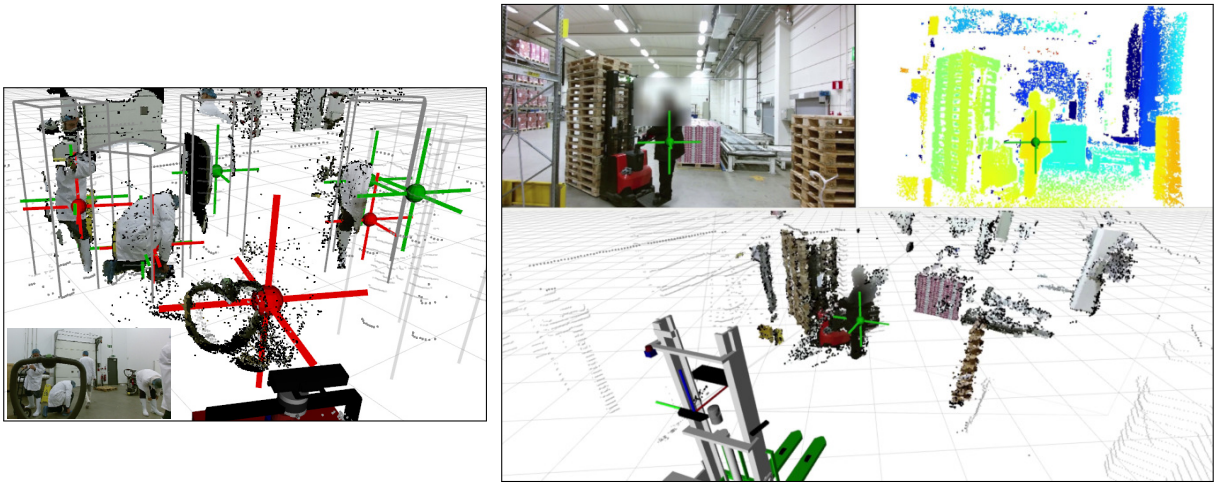


Figure 5.1: *Left:* Our method (green) localizes 3D person centroids more robustly than a baseline (red) on our intralogistics dataset. Grey boxes are persons. *Right:* Results of our RGB-D approach with end-to-end 3D centroid regression on a different scene.

5.1 Introduction

Detection of persons and objects in 3D space is an important capability for service, domestic and industrial robots that interact with their environment. In indoor scenarios, RGB-D sensors such as the Kinect v2 are often used for this purpose. However, while recent advances in computer vision have mostly solved the unimodal 2D detection problem on RGB images, it is not yet fully understood what is the best representation and strategy for approaching the 3D detection task, especially on multi-modal RGB-D data, where large-scale datasets are scarce and we want to benefit as much as possible from existing work on 2D detection.

In this chapter, we tackle the problem of learning to detect and *accurately localize 3D centroids* in RGB-D data in an end-to-end fashion, with an experimental focus on human detection in a challenging intralogistics context. We show that 3D localization can, to a large part, be learned from a diverse and *highly randomized synthetic RGB-D dataset* with perfect 3D groundtruth, and that for successful fine-tuning on real-world data, no manual 3D annotation is required. Our proposed real-time approach uses a strong image-based YOLO v3 single-stage detector as starting point, extends the RGB feature extractor with a separate depth stream via mid-level fusion, and utilizes a hardwired transfer learning strategy that can reuse existing pretrained weights from large-scale 2D object detection datasets. Thereby, incorporating the depth channel does not require training from scratch and thus does not lead to a loss in 2D detection performance. For 3D localization, we extend the resulting RGB-D YOLO v3 detector with a centroid regression output. Finally, we propose a depth-aware, scale-preserving variant of zoom-in/zoom-out training-time augmentation (W. Liu et al., 2016).

As opposed to the existing methods we compare against (Vasquez et al., 2017; Kollmitz et al., 2019; Zimmermann et al., 2018 and our own baseline), our end-to-end 3D regression can exploit complementary RGB and depth information by fusing modalities at the feature extractor stage,

and does not rely on any 3D point cloud representation. It is therefore robust to missing depth data and works well under partial occlusion.

Outline This chapter is structured as follows: First, we describe related work on 3D person detection, RGB-D datasets, and learning from synthetic data. Then, we present insights that we gained from initial experiments on the 2D bounding box detection task, using several state-of-the-art single- and two-stage detectors on our novel intralogistics dataset. We then describe a naïve baseline method to lift these 2D detections into 3D, and observe several interesting failure cases in preliminary experiments; this motivates the introduction of our novel synthetic RGB-D dataset, which provides accurate 3D groundtruth to enable end-to-end learning of 3D centroid regression. To leverage complementary information from RGB and depth, we then introduce our RGB-D fusion variant of YOLO v3 along with a training strategy that benefits from synthetic data. We extensive evaluate the method on the 3D human detection task both quantitatively and qualitatively on real-world data from the intralogistics domain.

5.2 Related work

3D person detection in RGB-D

There is a vast amount of literature on multi-modal (Feng et al., 2019) and RGB-D-based (Gao et al., 2019) object recognition. In Table 5.1 we list recent works that have been evaluated by their authors on the human detection task. Some fuse color and depth information, but only output 2D bounding boxes and do not tackle the issue of 3D localization (Mees et al., 2016; Guerry et al., 2017; Ophoff et al., 2019). Several approaches utilize a geometric 3D point cloud representation (Munaro and Menegatti, 2014; Vasquez et al., 2017; Zimmermann et al., 2018; Qi et al., 2018; Lewandowski et al., 2019), which comes with certain drawbacks – for example, the method by Qi et al. (2018) suffers from three weaknesses towards which our proposed method should be more robust: 1.) Their 3D stage fails to accurately localize objects in locally sparse point clouds. Here, our approach can leverage complementary RGB data as it does not rely on a point cloud representation. 2.) When multiple instances of a class share the same 3D frustum, only a single instance is detected. This scenario is frequent in our indoor environments, where humans often partially occlude each other. 3.) Their RGB-based 2D detector fails under difficult lighting conditions, where our method can exploit complementary depth data due to our mid-level fusion strategy. The methods by Munaro and Menegatti (2014) and Vasquez et al. (2017), which we include as baselines in our experiments, suffer from similar conceptual limitations.

More recent works therefore often leverage a 2D image-based representation (Mees et al., 2016; Guerry et al., 2017; Ali et al., 2018; Simon et al., 2019; Kollmitz et al., 2019; Ophoff et al., 2019) in order to exploit advances in 2D object detection. They are based upon single-stage detectors like YOLO v2 (Redmon et al., 2017) or the computationally more expensive two-stage R-CNN framework (Girshick, 2015; Ren et al., 2017). There are several extensive and recent surveys on such 2D object detectors (Z.-Q. Zhao et al., 2018; L. Liu et al., 2019; Zou et al., 2019).

Method	Modalities	Detector	Fusion strategy	Output	Dataset
Munaro and Menegatti (2014)	RGB+D	PCL+HOG-SVM	3D proposals → 2D classifier	3D boxes	KTP
Mees et al. (2016)	RGB+D+Flow	Fast R-CNN	2D late (adaptive gating)	2D boxes	InOutDoor (IO)
Vasquez et al. (2017)	RGB+D	Fast R-CNN	3D proposals → 2D detector	2.5D centroids	MobilityAids, IO
Guerry et al. (2017)	RGB+D	Faster R-CNN	2D early/mid/late	2D boxes	Onera, Mensa, IO
Ali et al. (2018)	Projected LiDAR	YOLO v2	–	3D boxes + orient.	KITTI
Simon et al. (2019)	Projected LiDAR	YOLO v2	–	3D boxes + orient.	KITTI
Qi et al. (2018)	RGB+D/LiDAR	FPN+Fast R-CNN	2D boxes → 3D frustums	3D boxes + orient.	KITTI, SUN
Zimmermann et al. (2018)	RGB+D	OpenPose	2D joints → 3D voxel grid	3D body joints	MKV-t, CAP-t
Lewandowski et al. (2019)	D	FPFH-SVM	–	3D boxes	Supermarket
Kollmitz et al. (2019)	RGB or D	Faster R-CNN	–	3D centroids	MobilityAids
Ophoff et al. (2019)	RGB+D	YOLO v2	2D mid-level features	2D boxes	KITTI, EPFL
Our approach	RGB+D	YOLO v3	2D mid-level features	3D centroids	Intralogistics

Table 5.1: Qualitative comparison of existing methods for 3D/RGB-D human detection

Most closely related to our work are the methods by Ophoff et al. (2019) and Kollmitz et al. (2019). With focus only on 2D detection, Ophoff et al. (2019) incorporates RGB+D fusion into the earlier YOLO v2 architecture. It is not as deep, uses no shortcut connections and does not include a feature pyramid, as is the case with the YOLO v3 (Redmon et al., 2018) architecture that our work is based upon. For our work, these shortcut connections impose extra constraints on where we can fuse features. Similar to our method, and contrary to their earlier work (Vasquez et al., 2017), Kollmitz et al. (2019) do not employ a 3D point cloud representation, and instead utilize an end-to-end 2D detector with 3D centroid output. Their two-stage approach is evaluated in a multi-class hospital scenario including persons with walking aids. They provide separate models for detection on either RGB or depth data. In contrast, our method follows an efficient one-stage approach, performs principled fusion of the RGB+D modalities to exploit complementary information, utilizes synthetic training data to learn 3D localization, and incorporates a depth-aware crop augmentation scheme for more accurate results. We evaluate our method and baselines on a novel, challenging dataset from the intralogistics domain.

RGB-D datasets

Generic large-scale benchmark datasets such as ImageNet (Russakovsky et al., 2015) and MS COCO (T.-Y. Lin et al., 2014) enabled a rapid evolution of deep learning methods in the RGB domain. Examples of person detection benchmarks in RGB images from urban environments include Caltech (Dollár et al., 2012) and CityPersons (S. Zhang et al., 2017; Cordts et al., 2016). However, for robotic applications usually 3D awareness is required. Therefore, additional sensing modalities such as depth are often included.

In the robotics community, several RGB-D datasets have been introduced for person detection, covering various depth sensing technologies such as structured light, time-of-flight, or stereo (Spinello and Arras, 2011; Hanten et al., 2018). Existing available Kinect v2 datasets for person detection have been recorded in indoor and outdoor environments at a university campus (Bagautdinov et al., 2015; Ophoff et al., 2019; Mees et al., 2016; Wojke, Memmesheimer, et al., 2017) or a hospital (Kollmitz et al., 2019). Some lack person annotations when depth is not available (Mees et al., 2016), or are destined for multi-class detection involving persons with walking aids (Kollmitz et al., 2019). In addition, these datasets are an order of a magnitude smaller and show less variation and scene diversity in comparison to existing RGB-only datasets.

Learning from synthetic RGB-D data

Learning from simulation and transfer into the real world are currently quickly evolving topics in computer vision and robotics. Most work so far has been focused on rigid objects. Synthetic RGB-D data generation for a variety of scene understanding tasks, including object detection, has been explored by McCormac et al. (2017) and Georgakis et al. (2017). Tobin et al. (2017) explore domain randomization for robotic manipulation: Using synthesized RGB images with random camera and object positions, lighting, and textures, they learn accurate 3D detectors for geometric primitives without pretraining on real images. Sundermeyer et al. (2018) learn to estimate 3D orientation of objects using synthetic RGB images of CAD models with domain randomization. Danielczuk et al. (2019) focus on category-agnostic 2D instance segmentation using synthetic depth also from CAD.

Humans vary greatly in shape and appearance, and are thus particularly challenging to simulate. The work by Shotton et al. (2011) on articulated human pose estimation focused on single-person scenarios using synthetic depth images, without simulating any 3D background. This also applies to the SURREAL dataset (Varol et al., 2017), which however contains additional modalities such as RGB. Richter et al. (2016) perform semantic segmentation on KITTI (Geiger et al., 2012) by training on synthetic RGB images and groundtruth masks from a commercial computer game. In contrast to these works, our synthetic dataset presented in Section 5.5 focuses on multi-person human detection in clutter and under occlusion. It contains up to several dozen person instances per frame, diverse 3D backgrounds and large amounts of foreground occluder objects with strong domain randomization (which is not used as extensively in the works of Gaidon et al., 2016; Ros et al., 2016; Fabbri et al., 2018). Our dataset is composed of synchronized RGB-D pairs, where we additionally model noise characteristics of the Kinect v2 time-of-flight sensor.

5.3 Initial insights on 2D detection performance

Because the 3D detector that we will present in Section 5.6 will be based upon a 2D method, we first evaluate existing 2D methods on our intralogistics dataset to be able to select a method that offers a reasonable compromise between detection accuracy and runtime performance. This is essential for the desired deployment on a mobile robot platform; according to the author's experience, for robust person tracking, a frame rate of at least 10 Hz is desirable to account for human dynamics and if multiple targets are in close proximity.

Experimental setup

For a fair comparison, we train all methods on MS COCO (T.-Y. Lin et al., 2014) and tested them on our complete real-world intralogistics dataset, consisting of 3.1k frames as described in Appendix A. Using standard COCO protocol, we report average precision (AP) for the person class over multiple IoU levels [0.5; . . . ; 0.95], and at IoU 0.5. All experiments were conducted on a desktop PC with GTX 1080 Ti GPU and 16-core AMD Threadripper 1950X CPU. Inference time at batch size 1 includes NMS, but no disk I/O.

Quantitative results

Table 5.2 shows the results of our initial experiments, using only the RGB modality.

Detector	Backbone	Implementation	Test resolution	AP[0.5; 0.95]	AP0.5	Inference time
YOLO v2	Darknet19	Darknet	416 × 416	0.357	0.706	7 ms
YOLO v3	Darknet 53			0.500	0.852	15 ms
			832 × 832	0.531	0.883	32 ms
			608 × 608	0.536	0.878	23 ms
YOLO v3-SPP				0.580	0.883	23 ms
YOLO v3		GluonCV/ Apache MxNet	416 × 416	0.617	0.876	29 ms
			740 × 416	0.680	0.912	⋆ 32 ms
			1024 × 576	0.698	0.914	46 ms
	MobileNet 1.0		740 × 416	0.587	0.868	24 ms
SSD	ResNet50		1024 × 576	0.604	0.877	32 ms
Faster R-CNN	ResNet50-FPN			0.644	0.889	35 ms
	ResNet101			0.669	0.907	84 ms
	ResNet101-FPN			0.700	0.912	294 ms
Mask R-CNN				0.697	0.918	100 ms
	ResNet18-FPN			0.696	0.923	277 ms
				0.640	0.890	97 ms

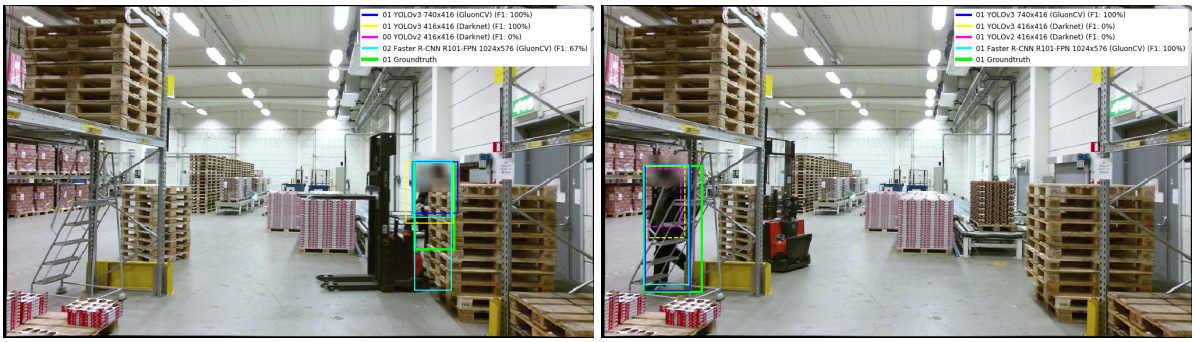
Table 5.2: Evaluation of different single- and two-stage 2D detectors on the intralogistics dataset (person class only), showing the speed/accuracy trade-off.

We note a big performance difference between different YOLO v3 implementations. With the same network architecture, the GluonCV implementation significantly outperforms the original by Redmon et al. (2018), due to a number of tricks applied at training time (see Z. Zhang et al., 2019) and due to its support for non-square input resolutions at test time. It also outperforms the standard single-shot multi-box detector, SSD (W. Liu et al., 2016), in our use-case. All methods except the older YOLO v2 achieve high $AP_{0.5}$, which indicates that they all offer good detection performance, but differ mostly in their 2D bounding box localization accuracy. We further see no big advantage of the computationally much more complex two-stage detectors Faster R-CNN (Ren et al., 2017) and Mask R-CNN (K. He et al., 2017) at identical input resolution. Therefore, we believe that the GluonCV variant of YOLO v3 with Darknet53 backbone ($\star$) is the most reasonable choice as a basis for our 3D RGB-D detector in Section 5.6.

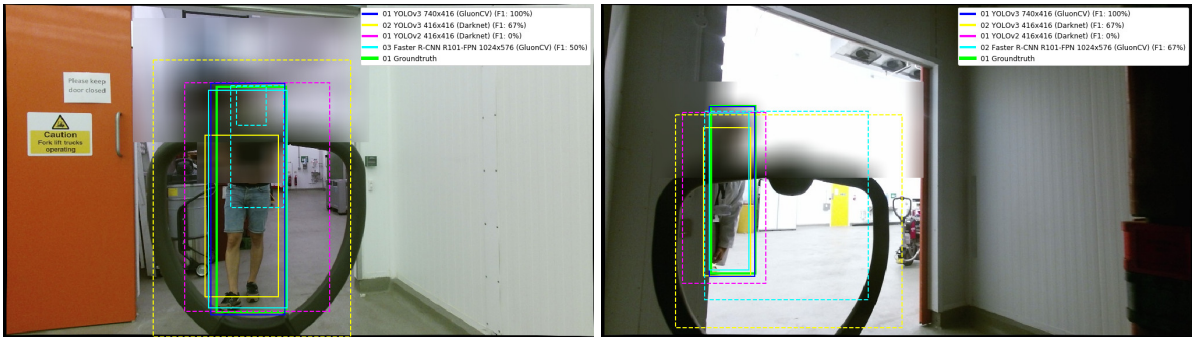
2D detection failure cases

Next, we want to understand qualitatively in which cases the examined 2D methods fail to detect persons on our intralogistics dataset, which introduces several domain-specific challenges such as protective clothing. Figure 5.2 shows a few interesting failure cases. For these visualizations, a fixed default detection threshold of 0.5 has been used for every method. Images have been blurred after detection to protect privacy. Dashed lines indicate false positive detections at a bounding box IoU level of 0.5.

First of all, we can see that even without any fine-tuning, the protective clothing in our dataset does *not* appear to negatively affect person detection accuracy. This indicates that the 2D training datasets (ImageNet and MS COCO) are sufficiently diverse. However, in cases of heavy occlusion through clutter, the separation of persons sometimes fails due to their very similar appearance in color space, making it hard to detect contours. In general, as shown in Figure 5.2b, all methods have problems when the handle bar grip of our ego-vehicle partially occludes the view,

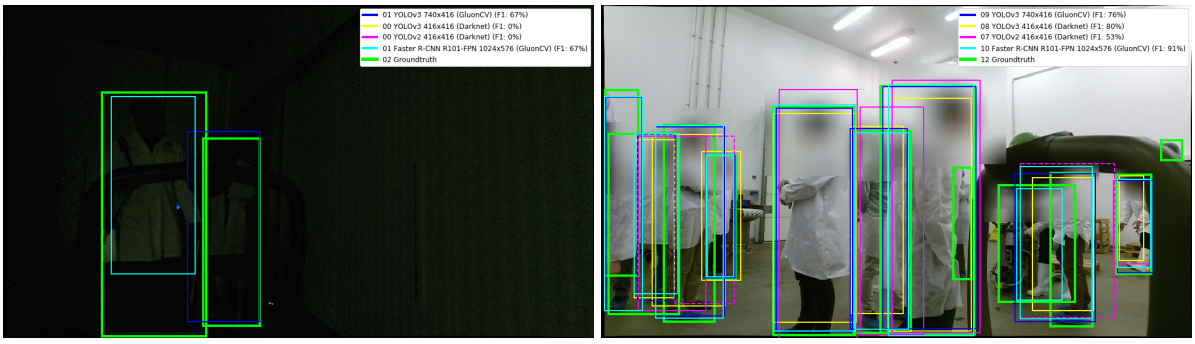


(a) Inaccuracies in 2D bounding box localization



(b) Larger inaccuracies and false positives due to partial occlusion by foreground objects

(c) False positives due to reflections in glass and metal



(d) False negatives due to lighting conditions and cluttered environments

Figure 5.2: Qualitative 2D human detection results on our intralogistics dataset. Dashed lines indicate false positive detections. Groundtruth, Faster R-CNN, YOLO v3 (GluonCV), YOLO v3 (Darknet), YOLO v2

Method	AP0.25m	AP0.5m	Hz
MobilityAids, 3D proposals + RGB classifier (Vasquez et al., 2017)	0.580	0.717	15
YOLO v3, Darknet implementation, *naïve depth*	0.718	0.809	50
+ instance masks (Mask R-CNN) to improve depth estimation	0.738	0.829	4

Table 5.3: Initial quantitative results for 3D person detection with the naïve baseline, and a variant using instance masks, on a short annotated test sequence.

leading to either over-segmentation and resulting false positive detections (because of failure in non-maximum suppression), or badly localized 2D bounding boxes; one question is whether this matters in practice if the ultimate goal is to estimate 3D centroids. As shown in Figure 5.2 (c) and (d), methods with higher recall at their default threshold, such as Faster R-CNN and GluonCV YOLO v3 , suffer from false positives due to reflections in glass and metal as well as shadows (not shown), but offer better performance in crowded scenes and under low lighting.

5.4 A naïve RGB-D baseline to localize humans in 3D space

Given the good performance of 2D image-based single-stage detectors, it appears reasonable to use them as a basis for building a detector with 3D centroid output. This approach differs from the one chosen by Vasquez et al. (2017), where at the beginning geometric clusters in the RGB-D point cloud are used to seed region proposals for the second stage of Fast R-CNN (Girshick, 2015); an approach that will lead to reduced recall when depth data is missing *e. g.* due to illumination interference. Our proposed image-based method does not suffer from this limitation. However, it poses the challenge of obtaining correct depth estimates for a given 2D bounding box in the image.

A naïve method to lift these 2D bounding boxes into 3D space would be to sample depth values from the registered depth image within the bounding box, for example by computing the median to be robust against outlier values. A more informed approach, based upon existing methods, would be to leverage 2D instance segmentation masks to sample foreground pixels from the depth map. Different efficient methods for real-time instance segmentation are currently under development (Uhrig et al., 2018; Milioto et al., 2019; Bolya et al., 2019).

Insights from initial experiments

We observed that in many cases, the naïve method will already yield useful results; in fact, on one of our test sequences, we obtained significantly better 3D results using this method than with the method by Vasquez et al. (2017), most likely due to our stronger 2D detector trained on the diverse, large-scale COCO dataset. However, in some cases, the majority of all valid depth pixels inside the 2D bounding box might belong to an entirely different foreground or background object, leading to a wrong depth estimate using this naïve method, as shown in Figure 5.3. For example, we saw this happening frequently with the autonomously moving handle bar grip of our pallet truck, which, like the RGB-D sensor, cannot easily be mounted differently.

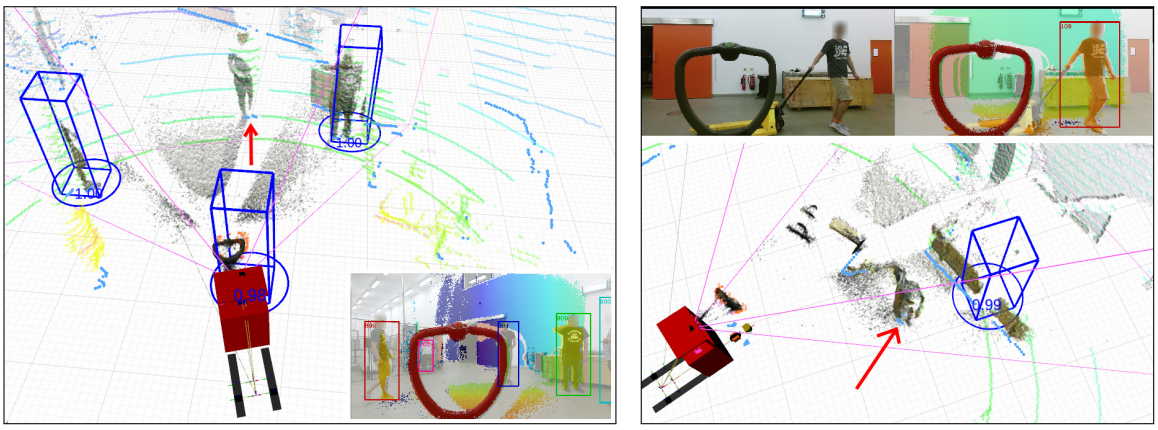


Figure 5.3: A naïve depth estimation method fails to correctly localize 3D person centroids under partial occlusion (left image) or when the 2D bounding box contains more background than foreground pixels (right image). The small images show an overlay of the jet-encoded depth map on the RGB image.

In Table 5.3, we see that using foreground instance segmentation masks could improve performance by around 2% in AP on our small test sequence, when generating masks offline using Mask R-CNN (K. He et al., 2017) and then associating them with YOLO v3 bounding boxes by maximizing IoU overlap. However, as we can see in several examples in Figure 5.4, also instance segmentation methods trained on MS COCO often fail in the cases that are relevant to us. While performance could potentially be improved by fine-tuning, this would require expensive per-pixel mask annotations. As this is not our final goal, we believe that regressing 3D centroid coordinates in an end-to-end fashion, without any intermediate 2D representation, is a better choice. The disadvantage is that this requires a large-scale dataset with accurate 3D groundtruth.

5.5 A heavily randomized synthetic RGB-D dataset

Manual 3D annotation, for instance of person centroids, is a difficult and time-consuming process. Another challenge encountered in robotics applications is that the sensor setup can be multi-modal, and vary significantly between robots. Working with different robots, or even a heterogeneous fleet of robots such as in the ILIAD project, might therefore repeatedly require large data recording and annotation efforts if the whole system is to be trained or evaluated in its entirety. Furthermore, the application domain may provide additional challenges which are underrepresented in commonly utilized real-world object detection datasets (for instance, persons wearing protective clothing and thus all looking alike; or unusual poses, such as persons lying on the floor). We were therefore wondering if, in combination with appropriate transfer learning techniques, synthetically generated data could improve results and significantly accelerate the deployment on a new robot, as in simulation a sensor setup can be changed very quickly and further scenarios and groundtruth (such as 3D body joint positions) can easily be added.

Using the Kinect v2 RGB-D sensor as an initial example, we explore in this section how we can leverage near photo-realistic game engines, such as Unreal Engine 4 (UE4), to generate synthetic

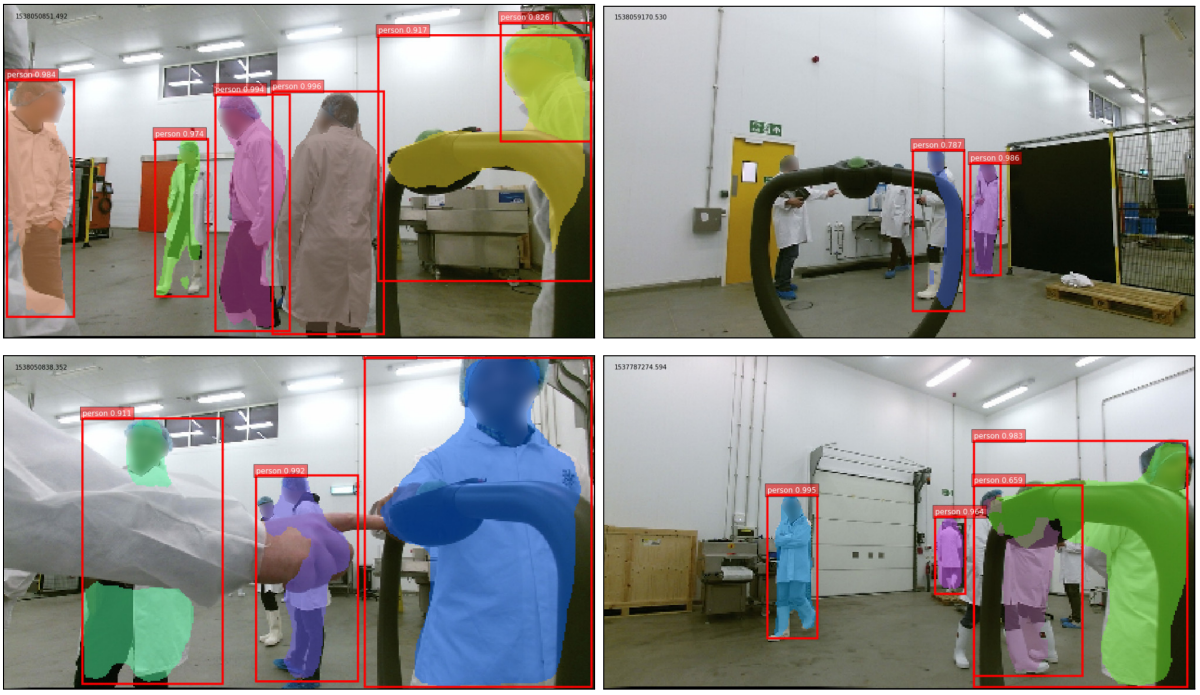


Figure 5.4: Instance segmentation methods (here Mask R-CNN with ResNet18 backbone) often fail when persons are partially occluded by foreground objects. Thus, they are also not able to fully prevent wrong depth estimates caused by occlusion.

datasets for training robust person detectors. Through extensive use of domain randomization, we synthesize a highly varied training set as observed from a mobile robot, comprising persons in confined and cluttered indoor spaces. Figure 5.5 shows examples of RGB and corresponding jet colormap-encoded depth frames from a single scene of our synthetic dataset.

Overview of the synthetic RGB-D dataset and scene randomization

Our simulation environment is composed of six different scenes for training, and two scenes for validation[1]. Example RGB frames from all scenes are shown in Figure 5.7. Scenes 1-3 have been modeled by hand from existing stock assets and contain various warehouse shelves and objects. The 2D background of these scenes is randomized at regular intervals from 25 publicly available HDR images. Instead, scenes 4-6 are modeled entirely in 3D, including their backgrounds, and are based upon publicly available environments representing a warehouse, a train station, and an outdoor factory environment. The validation set consists of two further scenes, representing an industrial assembly line and an office building with staircases, hallways and a reception area. As can be seen in Figure 5.8, all scenes have been enhanced with randomized light sources, where we vary intensities, light colors and positions at frequent intervals. All randomizations, including the following aspects, were implemented using UE4's Blueprint scripting system.

[1] My master student M. Hernandez helped in setting up the six training set scenes in UE4, and implementing the sensor model and export pipeline. Details are described in the master thesis by Hernandez Leon (2019), which was co-supervised by A. Csiszar and S. Fur at the University of Stuttgart.

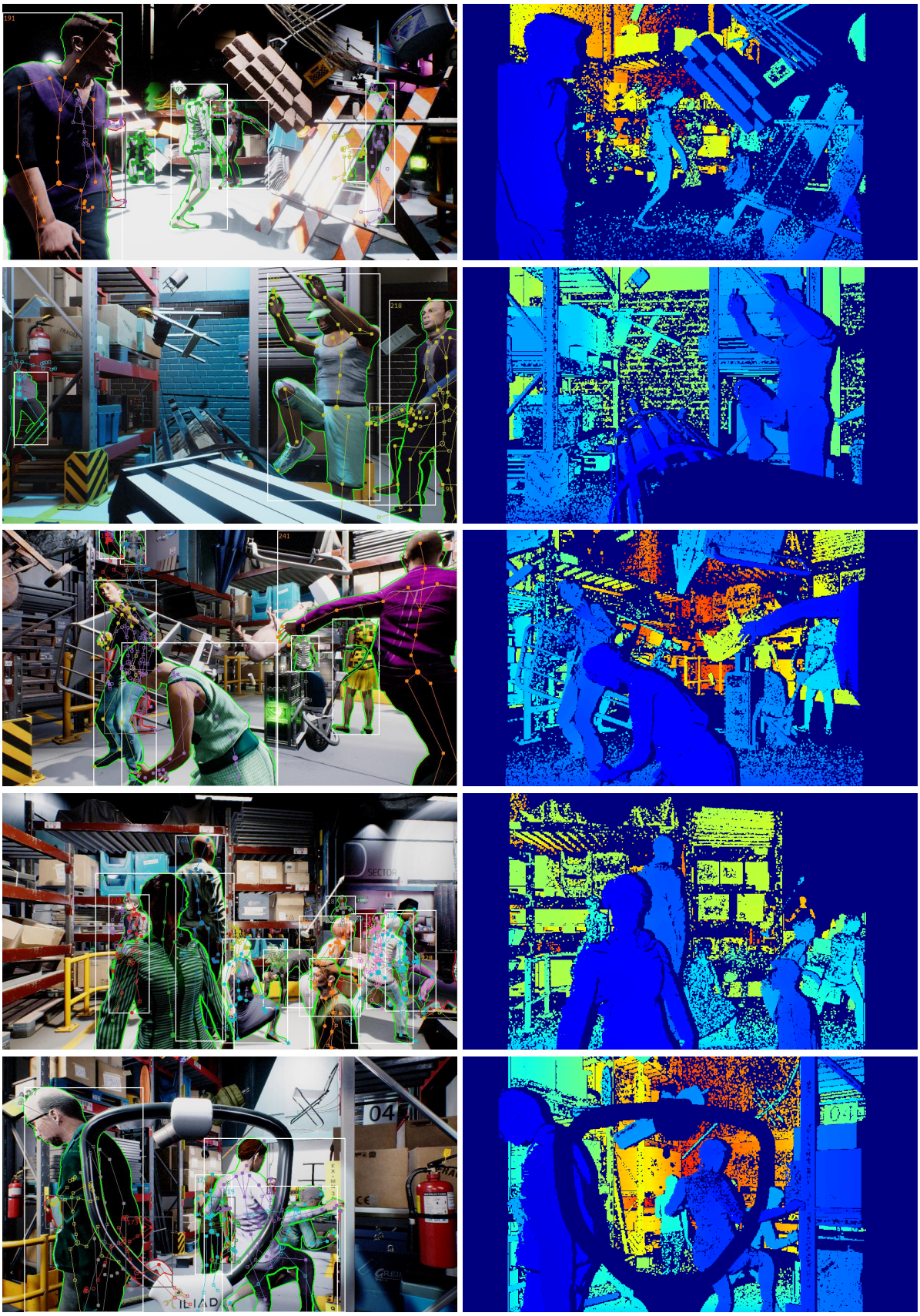


Figure 5.5: RGB-D frames from our highly randomized synthetic dataset

Randomized synthetic human 3D meshes

As shown in Figure 5.6, we use a set of 81 person meshes that were synthetically generated using a commercial character creation software[2], based upon a method initially proposed by Chaudhuri et al. (2011). Within UE4, we randomly switch between around 100 different motion capture-based animations every few seconds. The animations include several idling and walking styles, dancing, jumping, falling down, kneeing, sitting, lying on the floor or doing push-ups. Using the material masks of the human characters to separate clothing and skin texture, we randomly augment clothing colors by hue shifting.

Figure 5.6: 81 synthetically generated human models with random animations

Robot modeling

To replicate the extrinsic sensor setup and geometry of our autonomous pallet truck platform as closely as possible, we replicate the robot as an animated 3D mesh inside UE4 based upon real-world CAD drawings and photographs. Like on the real robot, the handle bar grip of our simulated pallet truck can rotate along two axes, and its position is randomized, so that it occasionally appears as a foreground occluder object.

Randomized navigation of human characters and robot

Our dataset is composed of a large number of short, continuous sequences. Every 15 seconds, the ego vehicle teleports itself to a random, reachable location in the map. At the same time, human characters are teleported into vicinity of the robot, while their density is varied randomly. During each sequence, we pan around the virtual camera at random speed. Both the robot and the simulated humans randomly navigate in the scene by individually performing A* pathfinding within the navigation mesh (see *e. g.* Toll et al., 2016, for an overview). The navigation mesh and pathfinding are provided by the open-source Recast and Detour libraries by Mononen (2009) that are built into the UE4 game engine. To build the navigation mesh, Recast discretizes the scene geometry into cubic voxels and then generates a polygon representation of navigable space. We use simplified capsule-shaped collision meshes to prevent collisions with dynamic objects. We do not explicitly model any social, group or flocking behaviors, as we did with the PedSim simulation in our experiments at the end of Chapter 3.

[2]The synthetic human meshes were created using Adobe Fuse and kindly provided by S. Aghaie.

(a) Training set scenes 1–3

(b) Training set scenes 4–6

(c) Validation set scenes 7–8

Figure 5.7: Our synthetic training set comprises 6 different scenes. The first three contain real 2D HDR images as backgrounds, whereas the latter three are completely composed of 3D objects. Lighting, foreground 3D objects, *etc* are highly randomized. The validation set with two scenes contains less randomization, and is used for ablation studies on 3D detection performance in Section 5.7.2. Testing always happens on real-world intralogistics data.

3D foreground and background augmentation

Inspired by the positive impact of augmentation with random 2D occluder objects for human pose estimation (Sárándi, Linder, et al., 2018a), we enrich all scenes with random flying 3D occluder objects, moving forklifts and pallet trucks. The around 700 different occluder objects contain various publicly available 3D meshes representing industrial, office and household objects, as well as vegetation. The motion of the flying objects is randomized by applying a randomly oriented force with random magnitude (within certain bounds) every few frames. We use the UE4 physics engine to prevent severe object overlap, by approximating their shapes with inscribed sphere colliders. Figure 5.5 and Figure 5.9 show examples of this form of 3D augmentation.

Modeling of time-of-flight sensor noise

Our real-world intralogistics dataset has been recorded with a Kinect v2 RGB-D sensor at 30 Hz, see Appendix A. The depth sensor operates under the time-of-flight principle with a maximum practical range of around 10 m and horizontal field of view of 70.6°. The color camera has a larger horizontal field of view of 84.1°. Further technical details can be found in the works by Sell et al. (2014) and Fankhauser et al. (2015).

For our synthetic RGB-D dataset, we approximately model the Kinect v2 sensor in simulation. The RGB and depth imagers are simulated by separate virtual cameras, with their respective real-world resolutions of 1920×1080 and 512×424 pixels. Based upon experiments by Lachat et al. (2015), Fankhauser et al. (2015), and Wasenmüller et al. (2017) and extensive comparison to our real-world data, we empirically model further acquisition-based and geometry-related errors to varying extents:

- *Depth distortion:* Per-pixel depth measurement noise is modeled using a zero-mean normal distribution with standard deviation that grows with increasing sensor distance, following results from Fankhauser et al. (2015). At distances for which no further measurements are available, we extrapolate using a linear interpolant. We do not model the depth-dependent oscillatory pattern ("wiggling effect").
- *Amplitude distortion:* Typically caused by inconsistent reflection of infrared light. Modeled using a look-up table (in form of a 2D texture) based upon measurements from Wasenmüller et al. (2017), under a model assumption of a flat target surface.
- *Axial noise:* Approximated as a function of incident angle, computed from objects' normal vectors and the camera's principal axis. We do not model lateral noise.
- *Illumination interference:* Overexposed areas in the high-dynamic range (HDR) image of the scene are used to simulate IR interference via a thresholding operation. An example is shown in the first row of Figure 5.5.
- *Material IR response:* We randomly drop depth measurements with increasing probability for darker materials, using the unlit scene color frame buffer provided by Unreal Engine's deferred renderer; see *e. g.* person in center of Fig. 5.5, last row
- *Multi-path interference (reflections):* For computational reasons, we do not model ray bounces. However, we simulate missing depth measurements on reflective surfaces using the per-pixel roughness material properties of scene objects.
- *Registration shadowing:* Via back-projection, an approximately 5 centimeter offset between infrared emitter and receiver is simulated, leading to registration shadowing effects especially at close-range (for instance visible in Fig. 5.9).

No integration-time related errors have been modeled. Further details on the implementation of these effects using GPU shaders and additional post-processing in Python are described in the master thesis by Hernandez Leon (2019). The right column of Figure 5.10 shows the result of simulating these effects, compared to the perfect, simulated raw depth image in the middle column. Real-world Kinect v2 depth images for comparison can be seen in Figure 5.11.

In initial experiments, described in detail in Linder et al. (2019), we found that modeling these effects can have a small, but positive impact on 2D bounding box detection performance, when training a YOLO v3 detector on synthetic jet-encoded depth images. Figure 5.11 illustrates qualitative 2D detection results on real depth frames. Figure 5.12 shows the results of the corresponding ablation studies; the diagram shows how performance degrades from the proposed default configuration if certain effects are selectively disabled. Removing simulation of sensor noise reduces performance by around -1.2% on our real-world 2D test set. These experiments were conducted on an earlier version of our synthetic dataset, with less foreground randomization and only 24 human characters, therefore we do not discuss them here in detail. Note that we do not model all possible noise sources, and registration shadowing was always enabled.

RGB-D registration

For each of the six scenes, we initially generate 5,000 RGB-D frames. Similar to our real-world dataset, the raw full-resolution RGB and depth images from UE4 are fed into the `iai_kinect2` ROS package (Wiedemeyer, 2014) to simulate registration. For the synthetic dataset, we assume a perfect extrinsic and intrinsic calibration, as we do not model any lens distortion effects. Registration yields RGB-D image pairs at QHD resolution of 960×540 pixels.

Generation of 2D and 3D groundtruth

To compute 2D groundtruth bounding boxes, we rely on instance segmentation masks that we simultaneously export from UE4 by exploiting the GPU's stencil buffer. Unique stencil IDs are assigned to all human characters. In a post-processing script, we then compute a bounding box around each person's instance mask after applying a morphological opening operation to eliminate stray pixels that can result from partial occlusion.

To extract 3D groundtruth, we utilize the simulated human characters' skeleton structure, which is shared by all characters in our simulation. So-called sockets have been attached to the end points of all 23 bones that are of our interest. At every frame through a Blueprint script, their location in sensor coordinate frame is exported into a text file for all characters in front of the camera. The left column of Figure 5.5 shows the resulting joint positions, projected into the RGB image. For each joint, we also determine its visiblity (denoted by hollow circles) by leveraging the instance segmentation mask.

Filtering of groundtruth bounding boxes

In our initial 2D experiments on the RGB modality in Linder et al. (2019), we observed that appropriate filtering of synthetically generated groundtruth bounding boxes is important for successful training of 2D detectors (see also Figure 5.12). The reasoning behind this is that a human annotator would also not annotate all theoretically correct bounding boxes as 'person', *e. g.* those that only consist of a few foreground pixels. Otherwise, we might obtain a different distribution of bounding boxes in synthetic training and real-world test/validation data.

We therefore set 'ignore' flags on all 2D groundtruth person boxes with extremely low contrast, with a groundtruth instance mask that covers less than 300 square pixels, or where more than 80% of the essential body joints are either truncated or occluded after projecting them onto the

Figure 5.8: Randomization of light colors, positions and intensities in the scene

Figure 5.9: View of the same scene without *(top)* and with *(bottom)* 3D augmentation.

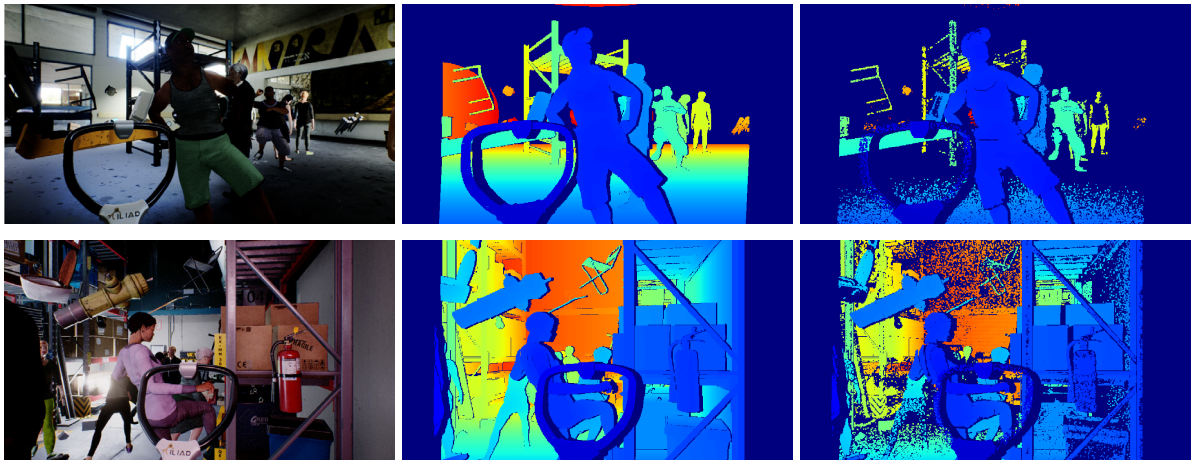


Figure 5.10: Effect of time-of-flight sensor noise modeling *(right)* compared to perfect simulated depth *(middle)*. The left column shows the RGB image.

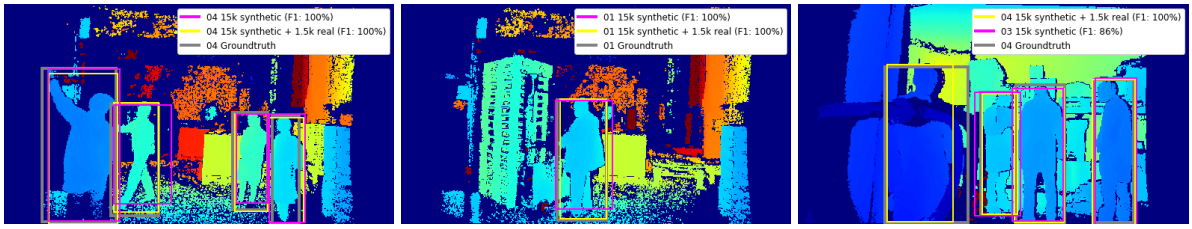


Figure 5.11: Qualitative results for depth-based detection on the real-world test set. Groundtruth in gray, YOLO v3 detector trained on 15k synthetic depth images in magenta, and combined with 1.5k real training images in yellow.

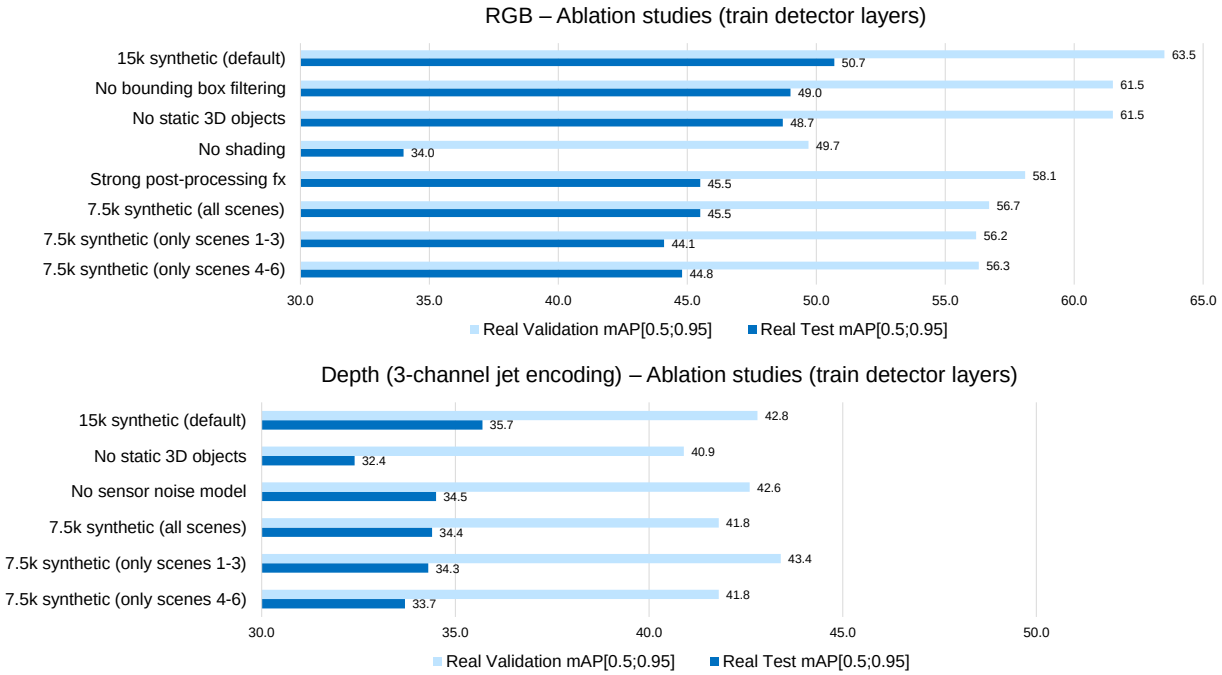


Figure 5.12: Ablation studies for the 2D human detection task on different simulation aspects of an earlier version of our synthetic dataset, when training on synthetic data and testing on our real-world intralogistics dataset. Note the different scaling of the axes. Further details on the experimental setup of these 2D detection experiments can be found in Linder et al. (2019).

corresponding 2D instance mask. These parameters have been empirically tuned to approximate annotation behavior of human annotators. In the last row of Figure 5.5, we can see three such 'ignore' boxes, indicated by grey boxes and red contours around the person.

After computing these ignore flags, we iterate over the 5,000 frames per scene, remove all frames which have less than two remaining (non-ignore) boxes, followed by random subsampling to finally obtain 2,500 sufficiently dense frames per scene. Combining all six scenes, the resulting RGB-D dataset thus consists of 15,000 training samples.

5.6 RGB-D fusion and end-to-end 3D centroid regression

While 2D object detection has made significant progress, robustly localizing objects in 3D space under presence of occlusion is still an unresolved issue. Our focus in this section is on real-time detection of human 3D centroids in RGB-D data. We propose an image-based detection approach which extends the YOLO v3 architecture with a 3D centroid loss and mid-level feature fusion to exploit complementary information from both modalities. We employ a transfer learning scheme which can benefit from existing large-scale 2D object detection datasets, while at the same time learning end-to-end 3D localization from our highly randomized, diverse synthetic RGB-D dataset with precise 3D groundtruth. We further propose a geometrically more accurate depth-aware crop augmentation for training on RGB-D data, which helps to improve 3D localization accuracy. In experiments on our challenging intralogistics dataset, we achieve state-of-the-art performance even when learning 3D localization just from synthetic data.

Overview – Our contributions in this section are:

1. We are, to our best knowledge, the first to propose an RGB-D fusion strategy for the fast YOLO v3 one-stage detector, with an accompanying transfer learning strategy that leverages existing large-scale 2D datasets.
2. Via heavy domain randomization, we are able to learn end-to-end regression of 3D human centroids from the presented synthetic multi-person RGB-D dataset.
3. For optional fine-tuning on real-world images, we propose a semi-supervised strategy to derive groundtruth 3D coordinates from offline articulated 3D human pose estimation.
4. We find that standard 2D crop/expansion augmentations ("zoom-in/out augmentations") are unsuitable for depth data, and propose a geometrically more accurate variant that accounts for the resulting shift of focal length.
5. On a sequence from our challenging real-world RGB-D dataset from the intralogistics domain, our method outperforms existing baselines in 3D person detection without requiring additional hand-annotated 3D groundtruth for training.
6. Our method is integrated with ROS and our human tracking framework (Chapter 8) and achieves real-time speeds of around 25 Hz on a Titan RTX GPU.

We now present our solution for robust detection and 3D localization of persons from RGB-D data, starting with the modified YOLO v3 network architecture that incorporates RGB-D fusion. We then explain how we add 3D centroid regression, and present a transfer learning strategy to benefit from both synthetic 3D and real-world 2D data. We also show how we can derive weak 3D groundtruth for optional fine-tuning on real-world data, using an offline 3D human pose estimation method. Lastly, we present a depth-aware variant of 2D crop/expansion augmentation that leads to better 3D predictions.

5.6.1 Network architecture with RGB-D fusion

Our method is based upon the YOLO v3 network, which we modified to also predict 3D centroids. First, to leverage depth information, we extend the Darknet-53 backbone to accommodate the additional single-channel depth data. Therefore, we duplicate layers up until a fusion point, resulting in an RGB- and depth-specific backbone. The depth backbone is highlighted in blue in Figure 5.13.

We evaluate two different mid-level fusion points, which are placed at the end of residual stages in the Darknet-53 architecture, in our case after Layer 9 or Layer 26. Mid-level fusion has shown to lead to good results (Ophoff et al., 2019; Hazirbas et al., 2016). Furthermore, as shown in Figure 5.13, we fuse the RGB and depth modalities before the pyramid structure of the network begins. The modalities are fused by concatenating the output features of both backbones along the channel dimension and using a 1×1 convolution to halve the number of channels to the original channel dimension.

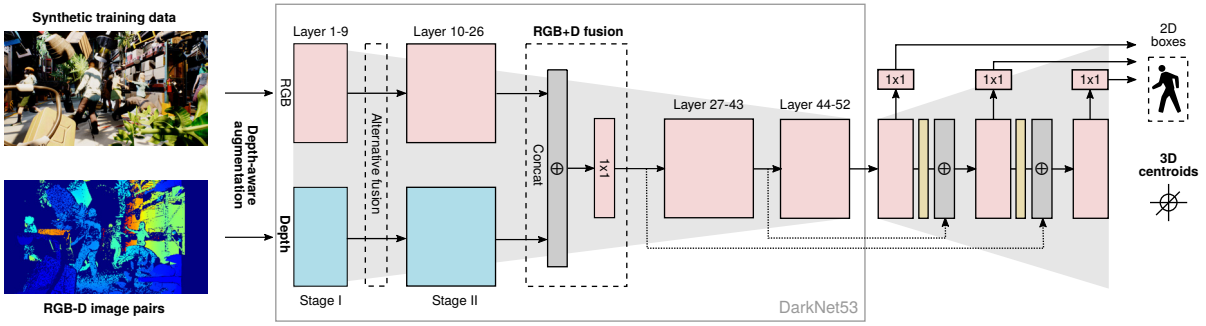


Figure 5.13: Overview of our proposed approach, which extends the YOLO v3 (Redmon et al., 2018) detector with mid-level RGB+D feature fusion, depth-aware augmentation, and 3D centroid regression. We show that the latter can be learned from synthetic RGB-D images.

5.6.2 3D centroid regression

The final 1×1 convolutions of the three output stages are extended to predict a centroid (c_x, c_y, c_z) for each anchor box. c_z is regressed directly in metric scale. The c_x and c_y coordinate are first predicted in image coordinates (c_u, c_v) and then backprojected with help of c_z and the camera matrix. This mirrors the unit system of our input. Furthermore, we formulate the regression targets in a constraint manner, relative to the bounding box coordinates where (b_u, b_v) is the top-left corner of the bounding box with height b_h and width b_w in pixel coordinates:

$$\begin{aligned} c_u &= b_u + b_w \sigma(t_{c_u}) \\ c_v &= b_v + b_h \sigma(t_{c_v}) \\ c_z &= t_{c_z} \end{aligned} \tag{5.1}$$

This limits the centroid to lay within the predicted bounding box. The 2D YOLO loss per anchor box is then extended by an additional term

$$\mathcal{L}_{\text{centroid}} = |t_{c_z} - \hat{t}_{c_z}| + \sum_{i \in \{u,v\}} \text{BCE}(\sigma(t_{c_i}), \sigma(\hat{t}_{c_i})) \tag{5.2}$$

where $\hat{t}_{c_i}$ denotes the groundtruth label. While for c_u, c_v we keep the sigmoid binary cross-entropy loss (BCE) that our YOLO v3 implementation originally uses for 2D bounding box centers, we found that ℓ_1 loss works best for centroid depth c_z.

5.6.3 Transfer-learning strategy

Our transfer learning strategy is inspired by Ophoff et al. (2019), but without an extra step to first train a depth-only detector. To benefit from existing large-scale 2D object detection datasets, we first initialize all layers from existing YOLO v3 RGB detector weights pretrained on ImageNet and MS COCO (Russakovsky et al., 2015; T.-Y. Lin et al., 2014). While other transfer learning strategies, such as proposed by Gupta et al. (2016) and Ophoff et al. (2019), could also be used, we already obtained good results using this approach.

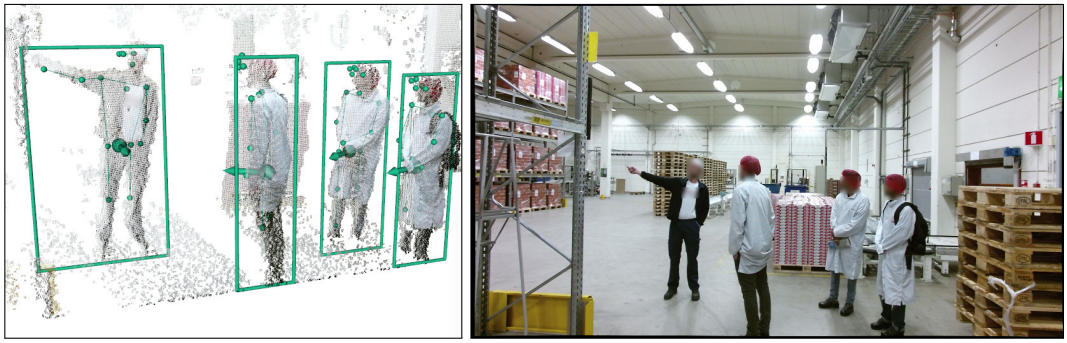


Figure 5.14: 3D body joints derived from offline 3D human pose estimation (Zimmermann et al., 2018) for optional fine-tuning on real-world RGB-D data.

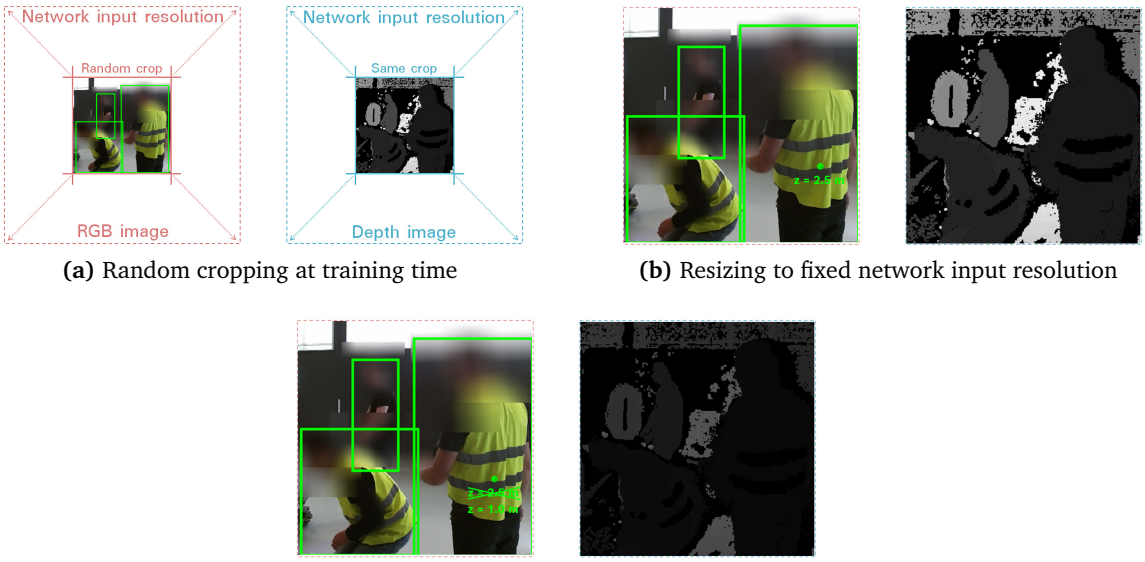
For the depth backbone, we duplicate RGB weights, but initialize the first layer (that takes single-channel depth images) from scratch. As indicated by the coloring in Figure 5.13, the fusion block is initialized using a hardwired fusion scheme such that at the start of training, the existing RGB features are forwarded as-is (*cf.* Hazirbas et al., 2016).

In the three output layers that we extended with 3D centroid regression, we randomly initialize weights for the new outputs, while leaving the original 2D detection weights unchanged. This initialization strategy allows to maintain the pretrained 2D performance despite the changes to the network architecture.

5.6.4 Weak real-world 3D centroid labels from human pose estimation

Kollmitz et al. (2019) propose a clustering-based heuristic to derive groundtruth 3D centroid coordinates for their training set, without requiring manual 3D annotation. This approach can be problematic if persons are truncated, for example when only the head or an arm are visible. Unless such persons are skipped, which can prevent difficult examples from ending up in the training set, the centroid would be offset to the top or side of the body (it would not be an *amodal* centroid, following the definition from Z. Deng et al., 2017).

To obtain more accurate results, we therefore propose 1.) to use a more informed approach, by leveraging offline 3D human pose estimation (Zimmermann et al., 2018) to derive weak groundtruth from predicted 3D body joints, 2.) to select a fixed, central body joint as the amodal 'centroid' regression target, which is more stably attached to the human body under truncation or unsual poses (*e. g.* stretching out a single arm) and thus more suitable for 3D human tracking or 3D articulated pose estimation applications. For the latter, top-down methods such as proposed by us in Sárándi, Linder, et al. (2018a) often require the pelvis joint, localized between the hip joints, as body-centric root joint for input, which we adapt as 3D regression target. However, in principle, our method can be trained on any (derived) body joint, as shown in Figure 5.14.



(a) Random cropping at training time

(b) Resizing to fixed network input resolution

(c) Scaling of groundtruth depth labels (left) and raw depth measurements (right)

Figure 5.15: Depth-aware zoom augmentation

5.6.5 Depth-aware zoom augmentation[3]

The 2D data augmentation pipeline of our underlying YOLO v3 implementation (J. Guo et al., 2020) involves random cropping and expansion of the image with corresponding adjustment of the bounding boxes, originally referred to as "zoom in" and "zoom out" by W. Liu et al. (2016). This is followed by resizing to a fixed-size square input image to our network during training.

Directly applying crop or expansion augmentation plus resizing to an RGB-D image can distort the understanding of objects' metric scale in the perceived environment, which is essential for accurate 3D perception. Therefore, we propose a depth-aware variant of this augmentation that adapts groundtruth depth labels and input depth to the current "zoom level" at training time, and preserves the metric scale and aspect ratio of the cropped image for a physically more well-grounded representation. The concept is visualized in Figure 5.15; our proposed enhancement is shown in the last row, (c).

Our network has a square input resolution of $d_n \times d_n$ pixels during training. Therefore, to preserve aspect ratio, we constrain ourselves to random square crops of varying size $d_c \times d_c$. Under the assumption of a single sensor with known camera matrix $\mathbf{K}$, resizing of a crop to the input dimension $d_n \times d_n$ can be expressed as a zooming operation (Hartley et al., 2003) with zoom factor $s = d_n/d_c$. Zooming is usually attributed to a change in focal length. Instead, to keep the intrinsic parameters intact, we apply the scaling to the depth values:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} s & & \\ & s & \\ & & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \frac{1}{z} = \mathbf{K} \begin{pmatrix} x \\ y \\ \frac{z}{s} \end{pmatrix} \frac{1}{\frac{z}{s}} \tag{5.3}$$

[3]This contribution has been proposed by my co-authors Kilian Y. Pfeiffer and Narunas Vaskevicius and is presented here for better understanding and reproducibility of results.

here (x, y, z) is a 3D point resolved in the RGB-D sensor frame and z/s is the new scaled depth measurement at input pixel (u, v). While this operation is not physically well-grounded for arbitrary crops not centered at the principal point of the RGB-D sensor, this approximation already yields a significant improvement, as shown later in our ablation studies. In addition to the scaling of depth measurements and groundtruth depth labels during training, we scale input depth measurements at inference time based upon the resize transformation applied to the RGB-D image before the forward pass.

5.7 Experiments on the 3D human detection task[4]

5.7.1 Experimental setup

As motivated in Section 5.3, our implementation is based upon MxNet using YOLO v3 from GluonCV (J. Guo et al., 2020). We train for a total of 80 epochs using stochastic gradient descent. During a warmup phase of 20 epochs, we gradually increase the learning rate to 6e-4, after which the learning rate is reduced to 1e-6 over 50 epochs using cosine decay (Loshchilov et al., 2017; Z. Zhang et al., 2019). Finally, the model is trained for another 10 epochs at a constant rate of 1e-6. We train using Volta V100 GPUs and test on a Titan RTX GPU along with an AMD Threadripper 2950X CPU.

Real-world intralogistics RGB-D dataset

Our application use-case is person detection in the intralogistics domain, with the goal of making autonomous guided vehicles (AGVs) human-aware. Human detection in such professional environments brings up certain challenges, such as people wearing special clothing; human forklift operators standing on the footrests of their vehicles; narrow, cluttered spaces with significant occlusion, which the robot observes from an ego-centric perspective; and the lack of publicly available datasets, especially for sensor modalities containing sensitive information, such as RGB-D. To train and evaluate our method, we therefore recorded a diverse dataset using two different AGV platforms equipped with a Kinect v2 sensor at around 1.50m and 1.80m height. Data has been recorded over several weeks at four different locations (two warehouses, a small food factory, and a robotics laboratory with forklifts and warehouse shelves). It includes both scenes with very few people, and very crowded scenes with up to around 20 people that frequently occlude each other and have very similar appearance. More details and example images can be found in Appendix A.

From these recordings, we selected around 3.1k diverse frames and split them into a training set of 1.5k, a validation set of 0.5k and a 2D test set of 1.1k real-world frames, with each split recorded at a different location or day. Each frame consists of a registered pair of RGB-D images, where we manually annotated 2D person bounding boxes. On the training set, we derive weak 3D groundtruth as described in Section 5.6.4.

[4] Our intern K. Y. Pfeiffer helped in conducting several of these experiments, as well as implementing an experimentation framework by adapting the Pytorch-Ignite framework to MxNet. He also implemented the 3D centroid loss and the RGB-D fusion under supervision by N. Vaskevicius and me.

For 3D evaluation, we labeled a continuous 60-second test set sequence from one of the environments with 3D centroids, using our trajectory-based annotation tool from Section 8.8, which we extended for annotation of centroid heights over ground. In this sequence, five persons are highly dynamic, assume different poses, and some push around carts. Additional registered LiDAR data was used to make 3D centroid annotations more accurate. For evaluation, we mark all centroids with ignore flags that are too heavily occluded in the point cloud, or outside of the Kinect v2 depth camera's field of view (with 8m range limit), in order not to penalize baseline methods which rely on the availability of depth data.

3D human detection baselines

We compare our approach with 5 different RGB-D baselines. Besides Munaro and Menegatti (2014), Vasquez et al. (2017), Kollmitz et al. (2019), and Zimmermann et al. (2018) (see Table 5.1), we also include our own RGB-only YOLO v3 baseline that naïvely lifts 2D centroids into 3D by computing median depth (Linder et al., 2018). The 2D detector for this method was trained on MS COCO; with naïve lifting into 3D, we saw no improvement in 3D performance when fine-tuning on our dataset. The HOG-SVM of the method by Munaro and Menegatti had previously been trained by us on a person dataset from SPENCER, recorded in the airport environment (Appendix A). We fine-tuned the method by Kollmitz et al. on our real-world intralogistics training set, as the RGB variant initially did not perform well in our scenario.

For the methods by Vasquez et al. and Kollmitz et al., we show the better variant of RGB or depth. In both cases, we obtained best results when considering only detections for the person class. For a fair comparison, we do not use their Kalman filter-based tracking or HMM smoothing. For the other methods, we use the original, trained model from the publicly available implementations.

The approach by Zimmermann et al. is not fast enough for our use-case (1-2 Hz), but we still decided to include it as we were wondering how a radically different, bottom-up 3D pose estimation approach would perform on the 3D person detection task. At test time, we derive 3D centroids from hip joint positions as described in Section 5.6.4; if no hips are detected, we fall back to the median of essential body joints while excluding the eyes.

3D evaluation metrics

For 3D evaluation on our real-world test sequence, we use a modified variant of COCO metrics (T.-Y. Lin et al., 2014), where instead of bounding box IoU we apply a metric distance threshold and compute 3D average precision (AP) as well as peak-F1 score; see Section 2.5.2 for details.

For methods that do not output robust estimates of centroid height (Vasquez et al., 2017; Kollmitz et al., 2019), we are more lenient and perform detection-to-groundtruth association only on ground plane coordinates while ignoring the height.

For ablation studies on our precise, synthetic 3D groundtruth, we also report root mean square error (RMSE), as well as 2D bounding box AP according to PASCAL VOC criteria (Everingham et al., 2010) at an IoU level of 0.5.

Variant	Modality	2D AP	3D AP ↑		RMSE
		VOC	0.25m	0.5m	↓
COCO model	RGB	71.0	–	–	–
Fine-tuned on ILIAD data	RGB	74.9	–	–	–
+Centroid regression	RGB	72.5	17.5	48.2	56.9
+Depth-aware augmentation	RGB	75.4	46.5	73.4	39.3
Fine-tuned on ILIAD data	RGB+D	75.4	–	–	–
+Centroid regression	RGB+D	73.4	47.6	74.5	34.7
+Depth-aware augmentation	RGB+D	76.6	60.6	81.5	30.8
/ Fusion after stage I	RGB+D	76.5	58.9	80.4	31.8
+More synthetic data	RGB+D	77.0	66.4	83.0	27.8

Table 5.4: Ablation studies on our synthetic validation set with accurate 3D groundtruth. RGB+D fusion after stage II unless noted.

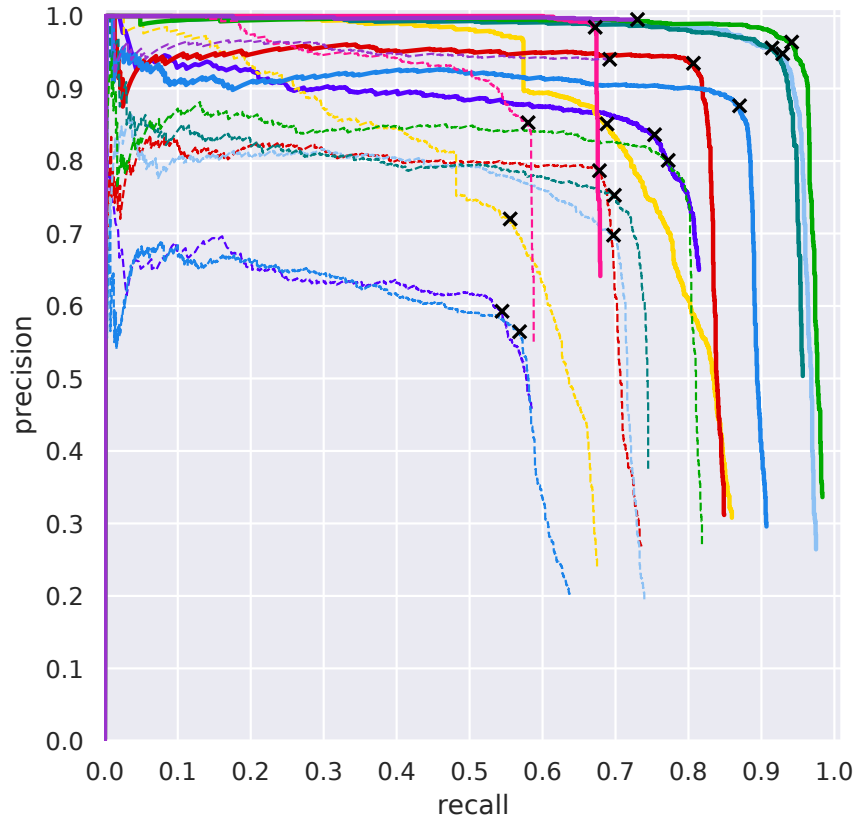
5.7.2 Results

Ablation studies on synthetic RGB-D data

Table 5.4 shows results of ablation studies on our synthetic validation set (2 extra scenes, 5k diverse frames, see Figure 5.7) with precise groundtruth. We initially used half of the synthetic training set (7.5k frames) for training. It can be seen that our depth-aware augmentation scheme boosts accuracy in both 2D and 3D significantly, compared to when using standard rectangular crop augmentation and resizing without scaling the groundtruth labels and input depth measurements appropriately. With the full synthetic training set, especially 3D localization at the smaller distance threshold of 0.25m improves. Incorporating RGB+D fusion leads to significantly higher 3D accuracy, and improves 2D slightly.

3D detection results on a sequence from our real-world intralogistics dataset

In Table 5.5 and the corresponding precision/recall curves, we compare variants of our method to other baselines on a 60-second hand-annotated sequence (1.8k frames) from our real-world intralogistics test set. It can be seen that our RGB-D method fine-tuned only on our real-world data (■) does not achieve a better peak-F1 score than our naïve YOLO v3 baseline trained on MS COCO (■) under the 3D metric with d=0.5m, and performs especially bad at d=0.25m. This could indicate that our real-world training set is too small. Training only on synthetic images (■), where through domain randomization we can generate an unlimited amount of frames and here thus have 10x more data with precise 3D groundtruth, improves performance drastically at both thresholds. Adding real to the synthetic training data (■) further improves performance. The corresponding RGB-only model (■) is similarly strong at d=0.5m. However, at d=0.25m, the combination of both modalities is around +9% better in peak-F1, which shows that our network can exploit depth information for more accurate 3D localization. Yet, as we also observe qualitatively, our approach degrades gracefully when only RGB data is available.



Method	3D AP ↑		Peak-F1 ↑	
	0.25m	0.5m	0.25m	0.5m
Munaro and Menegatti, 2014 (RGB+D)	56.0	77.9	62.7	76.1
Vasquez et al., 2017 (RGB+D, RGB)	66.3	73.1	**79.7**	84.2
Kollmitz et al., 2019 (RGB, VGG-M)	37.9	72.5	56.8	79.3
Zimmermann et al., 2018 (RGB+D)	55.8	67.3	69.1	79.9
Naïve YOLO v3 (RGB+D)	58.1	79.8	72.8	86.6
Ours (RGB, Synthetic+Real)	57.5	95.2	69.8	93.8
Ours (RGB+D, Real)	39.0	82.1	56.7	87.3
Ours (RGB+D, Synthetic)	59.9	93.7	72.5	93.5
Ours (RGB+D, Synthetic+Real)	**68.7**	**96.5**	78.6	**95.3**

Table 5.5: Precision-recall curves for 3D centroids on a 60-sec sequence of our real-world test set. Solid lines correspond to an evaluation radius of 0.5m, dashed 0.25m. Crosses are at peak-F1. For our method, we list several variants that were trained either on synthetic 3D data only, on real-world 3D data, or a combination of both (with pretrained 2D weights from MS COCO/ImageNet).

By combining 3D clustering-based region proposals with a modern deep learning-based 2D detector, Vasquez et al. (■) achieve very good localization accuracy and outperform our method in peak-F1 at d=0.25m. Here, our method might be at slight disadvantage because we train on pelvis joints, whereas our test sequence has been annotated with 3D centroids to be fair to the baseline methods.

Methods which use a geometric 3D point cloud representation (Munaro and Menegatti, 2014; Vasquez et al., 2017; Zimmermann et al., 2018) (■, ■, ■) are more limited in recall compared to our proposed approach, which uses an image-based representation and exploits complementary RGB+D information through feature fusion. If we extend our evaluation to the (wider) RGB sensor FOV, our method makes the most out of the available data, and our peak-F1 margin over the naïve baseline increases to around +13% at d=0.5m.

Qualitative insights on 3D performance

Qualitative 3D detection results are shown in Figures 5.1, 5.16 and 5.17. Like in 2D, many of our baseline methods have problems in localizing persons under partial occlusion, and sometimes even place centroids onto the scene background (such as shelves, pallets or walls). Furthermore, all baselines except for the RGB variant of Kollmitz et al. (2019) are strongly affected by missing depth data at the image boundaries, or at far distances. Our proposed RGB+D method is more robust in both regards.

Runtime performance

Our method runs in real-time on a high-end desktop GPU at around 25 Hz. Compared to the original YOLO v3 detector that ran at approximately 40 Hz in a similar setup, the additional convolutional layers of the Darknet53-based feature extractor for the depth modality, and the preprocessing of depth data (on the CPU) have the biggest impact on performance. Though we did not attempt this, we believe that the feature extractor for depth could be replaced by a more compact network, such as MobileNet or Darknet19. In this case, no easy weights transfer from the RGB backbone would be possible, and use of techniques such as cross-modal distillation (Gupta et al., 2016) might be required.

5.8 Conclusions

In this chapter, we presented a solution to the problem of detecting 3D human centroids from RGB-D data, when only low amounts of real-world data with manually annotated 3D groundtruth are available. This is frequently the case in robotics use-cases, where sensor setups are often specific to a particular application domain and publicly available datasets are limited in scale and diversity. While we have seen that bounding box detectors that were trained on existing large-scale 2D datasets, such as MS COCO, transfer well to the special work-wear worn by subjects in our intralogistics application domain, it is not straightforward to lift these 2D detections into 3D. The reason for this is that bounding boxes are a suboptimal representation under partial occlusion and heavy clutter, when observing the scene from a robot's first-person perspective. We also saw that more fine-grained 2D instance segmentation masks, computed with state-of-the-art

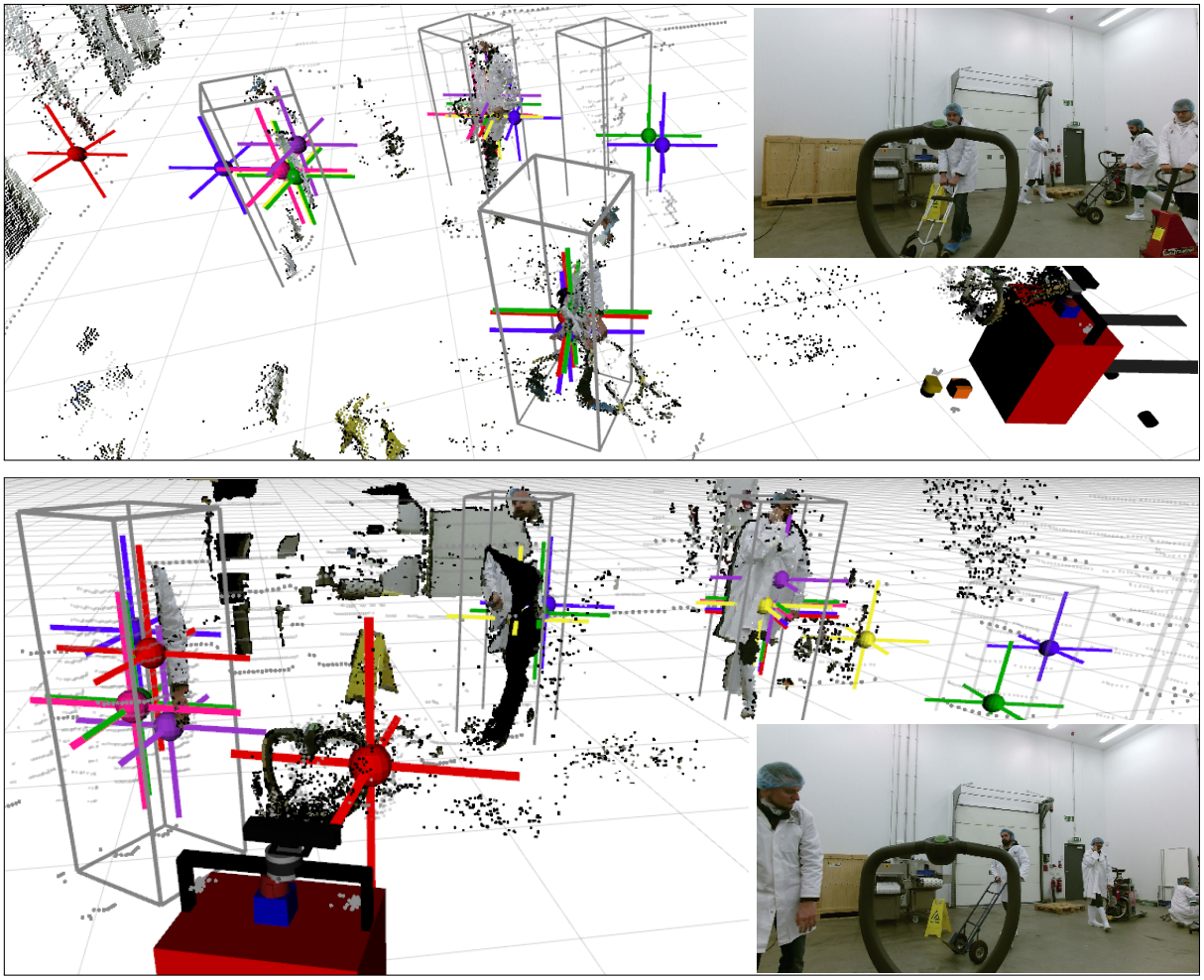


Figure 5.16: Qualitative 3D detection results at peak-F1 from a scene of our RGB-D dataset. Colors from Table 5.5; grey is groundtruth.

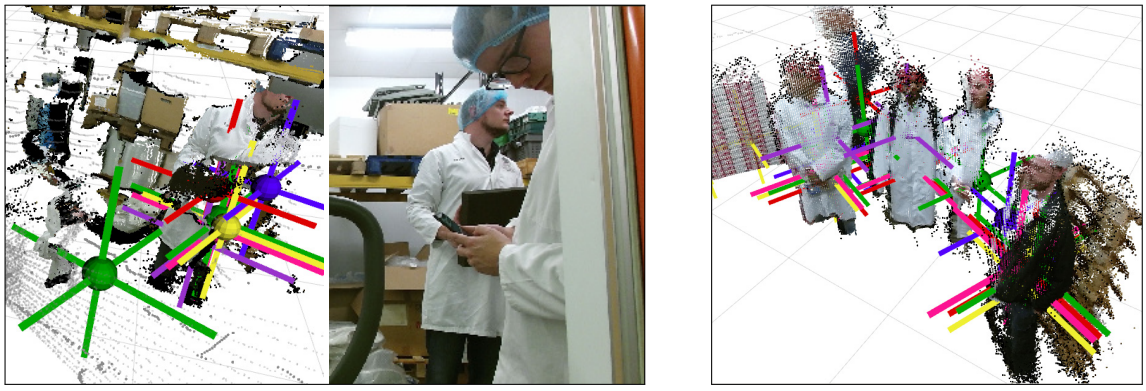


Figure 5.17: Results of methods from Table 5.5 on two further, more cluttered scenes.

methods, are often corrupt in such cases. Therefore, performing end-to-end regression of 3D coordinates appears to be the most promising approach.

The key ingredients of our image-based real-time approach are a heavily randomized synthetic dataset with accurate 3D groundtruth, generated using Unreal Engine 4, and an extension of the YOLO v3 architecture that incorporates RGB+D fusion, 3D centroid regression, and depth-aware augmentation at training time. Our proposed learning strategy benefits from synthetic data with 3D groundtruth, unlabelled real-world RGB-D data, and pre-training on existing labeled datasets for 2D detection. We demonstrated that it is possible to learn precise 3D localization from our diverse, synthetic dataset. In experiments on a sequence of our challenging real-world intralogistics dataset, we have seen that (without fine-tuning on our dataset) our method achieves higher detection accuracy than multiple state-of-the-art baselines trained on their respective datasets. Our method also works when depth information is missing, for instance at the boundaries of the depth image, by automatically and gracefully degrading to a regression solely based upon RGB information.

It is especially noteworthy that although we use the 3D articulated pose estimation method by Zimmermann et al. (2018) as a teacher for optional fine-tuning on real-world data, we surpass its performance. We believe this is partly due to our use of large-scale synthetic RGB-D data with heavy domain randomization and 3D occlusion augmentation, which allows us to better generalize to difficult cases with partial occlusion; and partly due to our proposed architecture: First, through our principled fusion of RGB and depth in the feature extractor, our method is able to exploit complementary RGB information when depth is missing, and vice versa. Second, our method does not suffer from occasionally wrong global localization of persons due to failure in 2D-to-3D lifting of the root joint, as our one-stage detector regresses root joint coordinates end-to-end and can take the entire image context into account.

Can we benefit from synthetic data?

As we have seen in Table 5.5, adding synthetic data clearly leads to a benefit in 3D detection performance. This is because the synthetic groundtruth is accurate and 3D annotations come for free, whereas our real-world training set is limited in both size and diversity, even though we have spent several days on data acquisition in different warehouse environments.

Earlier in this chapter, we posed the question if synthetically generated data could *accelerate* the deployment of such detectors in new scenarios. While we can conclude that the initial setup of such a simulation and the post-processing pipeline, as described in Section 5.5, takes a one-time effort of several months, new scenes and 3D models can now be added within a few hours. More importantly, modifying the extrinsic sensor setup to target a new robot platform, along with regeneration of the entire dataset, takes only 5 minutes of manual effort, without any need for manual 3D annotations.

Limitations and weaknesses

One limitation is that our quantitative evaluation occurred on a rather short (60-second), though challenging, continuous sequence. Clearly, we would expect a strong method to perform

reasonably well across *all* possible scenarios in our use-case. Therefore, in the remaining year of the ILIAD project, we want to extend our evaluation to further scenes to gain deeper insights.

A weakness we observed in qualitative experiments is that our regressed 3D centroids tend to oscillate slightly more (by a few centimeters) than the centroids *e. g.* computed by the method of Vasquez et al. (2017) via the point cloud. These oscillations could affect tracking performance, if they are not properly compensated by a sufficiently large observation noise level (which might affect the tracking system's reactivity in very dynamic situations). The use of a heatmap representation for centroid localization, as is often done in 3D human pose estimation (Sárándi, Linder, et al., 2018b), instead of directly regressing 3D coordinates could be worth studying.

Future work

We believe that our approach also transfers to other RGB(-D) sensors with different intrinsics, such as the upcoming Kinect for Azure sensor, or stereo camera setups. More work is required to implement simulation of sensors with a different working principle, such as 3D lidar; first steps in this direction have been taken in recent versions of the CARLA (Dosovitskiy et al., 2017) and AirSim (Shah et al., 2017) simulators, which are both also based upon Unreal Engine 4. This would allow for truly multi-modal sensor setups, and facilitate training and evaluation of detection algorithms that *e. g.* fuse lidar and RGB data using early or mid-level fusion.

Given the proposed 3D detector with RGB-D fusion, many interesting aspects related to synthetic RGB-D data generation could be studied in the future. For instance, while we have performed some initial ablation studies in a 2D detection context, it still needs to be understood which simulation aspects – *e. g.* sensor noise modeling, 3D augmentation, scene diversity, number and type of human 3D models – have a large impact on 3D human detection performance. To this end, the simulation could easily be extended with 3D scans of real people (derived using photogrammetry methods), potentially in professional clothing or carrying accessories like bags or backpacks. It could also be interesting to study the effect of adding clothing animation, for which there is already support in modern game engines. In terms of simulation fidelity, we observed in our initial 2D experiments (Linder et al., 2019) that there is still a noticeable domain gap especially for the RGB modality. Through use of GAN-based domain adaptation techniques, such as cycle consistent adversarial domain adaptation (Hoffman et al., 2018), the realism of our renderings – and also of the depth data – could potentially be improved without resorting to computationally much more complex ray-tracing techniques.

CHAPTER 6

Classical and deep learning-based methods for human detection in 2D and 3D lidar data

In the previous chapter, we studied how to detect humans using RGB-D data acquired with Kinect-like sensors. One limitation of such devices is that their horizontal field of view and detection range is rather limited. However, in many robotics use-cases, a 360-degree surround view of the environment can be helpful. For this purpose, 2D or 3D lidar sensors with a rotating mirror or sensor array are frequently used to detect humans and other obstacles. In this chapter, we want to compare, improve, and better understand algorithms for human detection using such sensors. For 2D lidar, we here focus on sensors mounted at leg height.

To this end, we first compare a deep-learning based method, previously developed by Beyer et al. (2016) for detection of wheelchairs and walking aids, and in our joint work (Beyer et al., 2018) recently extended to human detection by exploiting temporal information, with multiple model-based approaches for leg detection and tracking in 2D laser. Before evaluation, we re-train these methods and extensively tune their parameters using automatic hyperparameter optimization on the same large-scale dataset from an elderly care facility for the sake of a fair comparison.

In a second evaluation step, we then apply these methods to a challenging application scenario from the intralogistics domain, where we additionally include 3D lidar- and RGB-D-based baselines – motivated by the fact that not many works compare methods for human detection across different sensor modalities on the same multi-modal dataset, particularly not for robotics applications in such challenging scenarios. We provide quantitative and qualitative insights on two novel evaluation sequences. Our cross-modal evaluation provides interesting insights on failure cases and directions for future research.

The experiments on the 2D laser dataset from the elderly care facility and the improved DROW detector with temporal fusion have been previously presented in the joint work "Deep Person Detection in Two-Dimensional Range Data" by L. Beyer, A. Hermans, T. Linder, K. O. Arras and B. Leibe in IEEE Robotics and Automation Letters (RA-L), Vol. 3(3), July 2018.

6.1 Introduction

The sensor setups of many mobile robots in industrial, service and domestic applications include a horizontally mounted 2D laser scanner. Often, these devices are safety-certified for obstacle avoidance, and a mandatory component for autonomous mobile robots *e. g.* in intralogistics or hospitals. This is usually achieved through protective safety fields that monitor a certain zone in front of the sensor for intruding objects, without obtaining an understanding of which type of object is actually involved. Instead, our motivation is to fully exploit 2D laser range data in order to make robots human-aware and enable them to adapt their behavior accordingly.

However, detecting people from 2D range data is challenging, not only because of the dynamics of human leg motion and the high levels of (self-)occlusion particularly in crowded environments. The mounting height of the sensor can have a major impact on how the sensor will perceive humans: While in this chapter, we focus on 2D laser scanners that are mounted close to the floor, like in the example of the intralogistics robots from Figure 1.4 on page 12, on other systems they might be mounted at a higher level – one such example is the SPENCER robot, described in Chapter 9. In such cases, the sensor would perceive the waist, hips, hands, and carried accessories, whereas close to the floor, the legs or feet of the person reside within the 2D scan plane. Due to the sparsity of the data, other vertical structures such as table legs, columns or tree trunks can easily resemble human legs in 2D laser data. In Figure 6.1, we show two examples of how challenging human detection in 2D and 3D lidar can be. One intuitive insight is that motion might provide important cues for detection, though our goal is to be able to detect also still-standing humans.

The majority of existing detection algorithms for 2D laser rely on hand-crafted geometric features that are computed on a segmented laser scan, to be then fed into a classifier and possibly a tracking algorithm. Only recently, researchers have started developing deep learning-based methods for 2D laser-based detection. In this chapter, we want to compare, improve and better understand such methods, to provide recommendations for deploying such systems on future human-aware robots, and to outline possible future research directions. For this, we also establish a quantitative and qualitative comparison against detectors from other modalities, namely 3D lidar and RGB-D. We believe this is a valuable contribution, because we are not aware of many works that compare methods for human detection across multiple modalities on the same multi-modal dataset, particularly not for robotics applications in challenging intralogistics scenarios.

Outline This chapter is divided into two main parts: In the first half, we focus solely on 2D laser-based methods, and extensively compare classical methods against a novel deep learning-based approach on a dataset from an elderly care facility. In the second half, we review existing 3D lidar-based methods, and perform experiments on a novel multi-modal dataset from the intralogistics domain. Here, we present a previously unpublished cross-modal comparison of different detectors, including methods for 2D laser and 3D lidar data and RGB-D baselines from the previous chapter. We conclude with a brief discussion on the fusion of detector outputs, and outline open issues for future research.

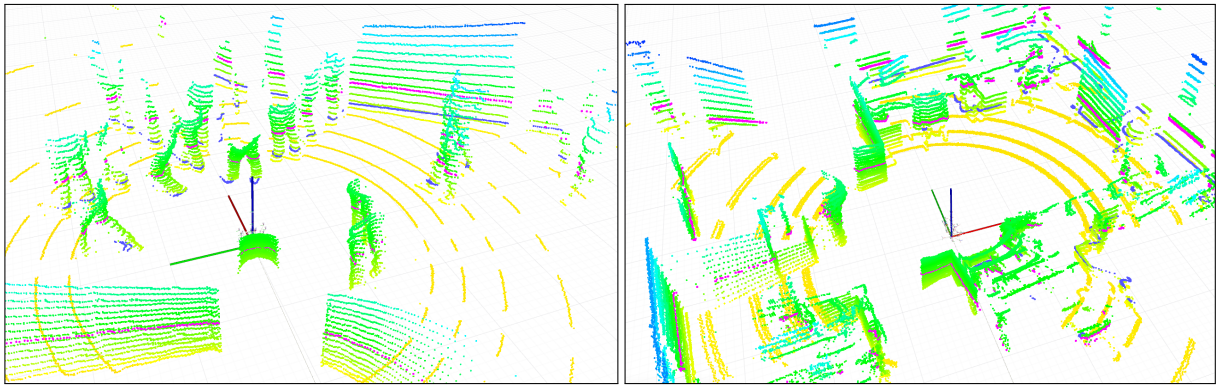


Figure 6.1: Exemplary lidar point clouds showing a human crowd, and a single human within an office environment. Color encoding reflects height above ground. The magenta points highlight the central scan plane of a VLP-16 lidar at 40 cm above ground. The blue points close to the floor have been recorded by a 2D laser scanner at 15 cm and show how a 2D safety scanner would perceive the environment.

6.2 Leg detection and tracking in 2D laser

6.2.1 Related work

We now review prior work on the topic of 2D laser-based leg detection and tracking. For the sake of continuity with the following section, we begin with a discussion of existing datasets for human detection in 2D laser, before we focus on existing algorithms for this purpose. There, we distinguish between classical, model-based approaches with handcrafted feature representations, and more recent deep learning-based methods.

Existing datasets for human detection in 2D laser

Several datasets for human detection in 2D laser have been presented in the past. However, not all of them fit to our application scenario and sensor setup and are at the same time large enough to enable the training of deep learning-based methods.

The Freiburg “Main Station” and “City Center” datasets by Luber, Tipaldi, et al. (2011b) contain around 10k and 6k labeled frames that have been recorded with a stationary SICK scanner with limited background variation at around 80 cm height, where individual legs are not visible. As our focus is on leg height, they are not suitable for the scenario considered in this chapter.

The “Home” and “Reha” datasets of Weinrich et al. (2014) were recorded with a SICK S300 laser scanner at 23 and 40 cm above ground and an angular resolution of 0.5° using mobile robot platforms in assisted living and elderly care scenarios. They are fully annotated with splits for training (22k and 19k frames) and testing (2.5k and 5k).

Leigh et al. (2015) introduced a dataset for person tracking in 2D laser at leg height. It consists of two labeled indoor sequences with 45 and 37 person tracks, comprising 3.2k frames where the robot is stationary and 2.3k where the robot is moving, that are useful also for detector evaluation

but too small for training deep learning-based methods. Instead, the outdoor sequences are targeted at person tracking and following and only contain a single person.

Beyer et al. (2016) introduced a novel, large-scale 2D laser dataset for detection of wheelchairs and walking aids in an elderly care and hospital scenario. The "DROW" dataset has been recorded from a mobile platform equipped with a SICK S300 laser scanner[1] at 12.5 Hz and an angular resolution of 0.5° at 37 cm above ground. Based upon over 10 hours of laser scan recordings (464k frames), the dataset has been split into batches of 100 frames where in every fourth batch, every fifth frame was manually annotated with 2D centroids of wheelchairs and walkers in x, y ground plane coordinates. In total, the resulting 5% of annotated data correspond to 18k training, 4k validation and 2.5k test frames. For the follow-up work (Beyer et al., 2018), 2D centroid annotations for person without walking aids[2] have been added. Compared to the semantically similar datasets by Weinrich et al. (2014), the person density is roughly twice as high. Therefore, the DROW dataset is to our best knowledge the largest publicly available dataset for leg-based human detection in 2D laser, making it best-suited for the evaluation of data-hungry deep learning algorithms.

In concurrent work, Álvarez-Aparicio et al. (2018) presented a novel benchmark dataset for the "evaluation of range-based people tracker classifiers in mobile robots". It has been recorded in a 56 m^2 apartment using a mobile robot with a 2D laser below 20 cm. Its second version includes 14 different, scripted scenarios with a total of around 25k frames of raw laser measurements. Therefore, it is by a factor of around 20 smaller than the DROW dataset, at a comparable number of annotated frames. Groundtruth has been obtained by having people wear ultra-wideband beacons, as opposed to using manual annotations, with a location error of up to 30 cm.

Classical methods for human detection and tracking in 2D laser

2D laser range finders have been used for human and object detection and tracking since long ago, for instance in the works of Prassler et al. (2000), Kluge et al. (2001), Lindstrom et al. (2001), and Schulz et al. (2001). Cui et al. (2007) focus on tracking people in crowds, where single-frame detection approaches based upon clustering techniques often fail. They argue to exploit dynamics of leg motion by incorporating information from successive frames.

However, none of these early works exploit machine learning techniques. We now first focus on "classical" machine learning-based methods for human detection and tracking in 2D laser, that still rely upon hand-crafted geometric features as opposed to deeply learned representations. These works can roughly be grouped into two different categories:

Methods from the first category (Arras et al., 2007; Spinello et al., 2008; Weinrich et al., 2014) propose novel geometric features to distinguish human from non-human laser scan segments. Geometric features proposed by Arras et al. (2007) include *e. g. number of points, linearity, circularity, mean curvature*, and are computed on laser scan segments obtained via jump-distance clustering. Weinrich et al. (2014) propose generic distance-invariant features (GDIF) that may

[1] This is the same sensor that is used on some of the ILIAD robots (see Figure 1.4 on page 12).
[2] Including persons *pushing* a wheelchair (walking on their own), and persons sitting in ordinary chairs.

span across several of such segments. They are computed on fixed-size windows according to the expected target object size and thus avoid problems resulting from over-segmentation. For all these methods, resulting feature representations are then fed into a machine learning approach, *e. g.* Adaboost or a Random Forest, to classify segments as foreground (human) or background.

The works that fall into the second category additionally focus on the tracking aspect and explicitly model human leg dynamics. Most of them are based upon some variant of the hand-crafted geometric features from Arras et al. (2007). In these approaches, association of up to two legs per person (when unoccluded) occurs via hand-designed heuristics (Pantofaru, 2010; Leigh et al., 2015), or complex multi-hypothesis data association (Arras et al., 2008). Due to this model assumption, they cannot easily generalize to classes other than human, and may not work when the laser scanner is mounted too high (such as in Chapter 4) or when legs are not clearly visible (*e. g.* people wearing dresses or professional clothing like lab coats, work uniforms).

Deep learning-based 2D leg pair detection

Only few works so far examined the human detection problem in 2D laser using deep learning-based methods. One reason for this could be that CNN-based approaches, which work well for image-based tasks on a regular grid structure, do not easily lend themselves to the irregular structure of 2D range data that becomes increasingly sparse at larger distances. Therefore, several approaches first transform 2D laser scans into an occupancy grid map format via raytracing to obtain such a regular grid representation.

Ondruska et al. (2016) present a recurrent neural network (RNN) that is trained in an end-to-end fashion to infer object locations and track their identities from raw laser measurements. The method operates on a partially observed occupancy grid, performs context aggregation at multiple scales using stacked dilated 2D convolutions to increase receptive field size, and supports tracking of multiple object classes including pedestrians, vehicles and cyclists. Very recently, Guerrero-Higueras et al. (2019) proposed a CNN-based detection approach on basis of the U-Net architecture (Ronneberger et al., 2015). Here, the input is a binary 2D occupancy grid where unobservable areas are not marked. The output of the network is also a 2D occupancy grid, with a semantic segmentation of leg locations that are in a post-processing step grouped into human centroids. The method is trained on roughly 2.2k input frames, does not exploit temporal information, and outperforms the early leg-tracking method by Pantofaru (2010).

Beyer et al. (2016) propose *DROW*, a CNN-based detector for wheelchairs and walking aids. Unlike the previous two approaches, it does not use a 2D occupancy grid representation as input.

The original DROW approach (Beyer et al., 2016)

We now give a brief overview of the original DROW approach. Figure 6.2a visualizes the overall working principle. We assume that the input is a 2D laser scan consisting of range measurements at a fixed angular interval. In a first step, fixed-size windows (cutouts) are generated based upon expected sizes of target objects, similar to what is done by Weinrich et al. (2014). However, in DROW, such a cutout is created for every single laser point, like in a sliding-window approach. To deal with sparsity of points and varying spatial resolution at different distances, points are

linearly resampled to obtain a distance-invariant fixed resolution. The resulting points in each window are then fed into a convolutional neural network. This can be realized efficiently by stacking all windows of a laser scan along the batch dimension, such that they can all be passed through the network in a single forward pass. For each window, the network outputs softmax probabilities for all target classes and a background class. Each window that is likely to contain an object then *votes* for the centroid location of that object, through a 2D offset vector that is regressed at the same time. Different class-agnostic and class-specific voting schemes have been examined in Beyer et al. (2016) and Beyer et al. (2018). After the votes have been cast into a regular grid spanning the entire sensor's field of view, most likely object locations are determined via non-maximum suppression on a smoothed version of the voting grid.

6.2.2 Proposed approach

We now present an improved version of the deep learning-based DROW method targeted at the detection of legs of persons without walking aids – which is a more difficult problem, due to their smaller spatial extents and subsequently weaker signal in the 2D laser scan. The improved method was first proposed in an article co-authored with Beyer et al. (2018), where we extensively compared it to different baseline methods. In order to highlight our contributions to the approach and the experiments, we first summarize several improvements to the original method that have been *not* been developed and implemented by the author of this thesis.

Improvements over the original DROW method in Beyer et al. (2018)[3]

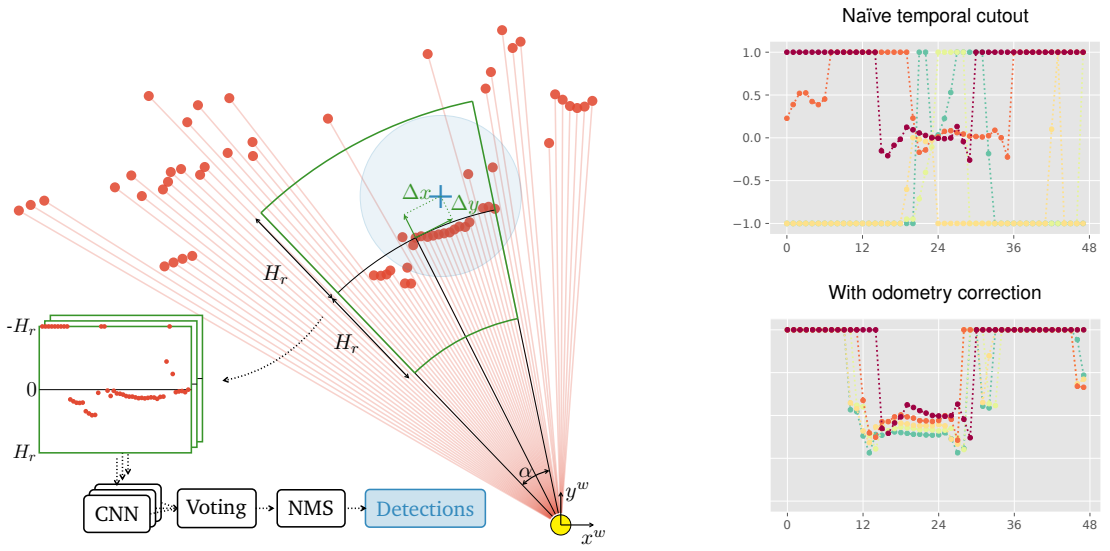
The first change relates to the CNN architecture: The new network architecture, "DROW3x", is deeper and follows latest best practices from research in computer vision. It consists of three stages of three zero-padded 1D convolutional layers of filter size three, where every stage is additionally followed by a max-pooling operation of size two and standard dropout. A fourth stage comprises two more convolutional blocks as previously described that feed into a global average pooling and fully-connected layer to compute final softmax probabilities and vote offsets. A further modification over the original method is an improved voting scheme, which we do not detail here further as it is mainly relevant in the multi-class case when also walking aids shall be detected. More details and ablation studies can be found in the PhD thesis of Beyer (2020).

Temporal fusion[4]

Cui et al. (2007) outlined the importance of exploiting dynamics of human leg motion to robustly detect humans in crowded scenes, due to ambiguities in leg association that arise otherwise. We were therefore wondering if also DROW would benefit from incorporating temporal information, without directly resorting to a tracking approach with explicit data association.

[3] These improvements have been proposed and implemented by L. Beyer and A. Hermans.

[4] The idea to exploit information from multiple frames via early/late fusion to make human leg detection in 2D laser more robust was jointly developed. It was implemented by L. Beyer, who also worked on the odometry correction.



(a) Original DROW approach (Beyer et al., 2016)

(b) New temporal multi-scan fusion

Figure 6.2: *Left:* An overview of the original DROW approach (from Beyer et al., 2016). *Right:* The extended version of DROW described in Beyer et al. (2018) incorporates temporal information from multiple scans. This is the representation fed into the 1D CNN for each generated cutout. (Images ©2016, 2018 IEEE)

Instead, we propose to incorporate temporal information by fusing sensor measurements within the CNN from multiple consecutive laser scans. This can be achieved by feeding not only the current, but also the n (not necessarily directly) preceding scans into the CNN in the form of spatially aligned cutouts. As we only rely on current and past measurements, this will not incur extra latency. Commonly used fusion schemes include early and late fusion, as in the works of Mnih et al. (2013) and Feichtenhofer et al. (2016): To realize early fusion, we can simply concatenate multiple temporal cutouts along the channel axis. For late fusion, we instead replicate the first two stages of DROW3x according to the number of scans to be fused, while keeping their weights tied, and then pass their sum into the third stage, with the rest as-is.

One caveat here is that the cutouts fed into the network need to be *spatio-temporally consistent*: Besides human motion, which we cannot easily anticipate without a prediction step (as *e. g.* in Ondruska et al., 2016), also the robot's own motion might cause misalignment of successive scans. Therefore, some form of odometry correction, as indicated by Figure 6.2b, is necessary[5]. We can distinguish three different variants that are relatively easy to implement while retaining a polar coordinate representation of laser scans with fixed angular intervals: In a naïve version, we create a cut-out at the same point index i for temporally successive scans. In the second variant, we instead fix the location in Cartesian space, such that the cut-out cannot easily jump onto background points. In the final variant, we also compensate for the robot's *rotational* motion.

[5]The previously discussed CNN-based methods which use a 2D occupancy grid representation as input could instead transform grids of successive scans into a common Cartesian coordinate frame during raytracing, to add support for a mobile sensor. While Ondruska et al. (2016) use a stationary sensor, their follow-up work (Dequaire et al., 2018) instead feeds visual odometry into a spatial transformer module (Jaderberg et al., 2015) to transform feature maps from preceding cycles into the current coordinate frame to decouple ego- and object motion.

Automated optimization of detector hyperparameters

Classical detection approaches that first segment a laser scan, then classify it, and potentially also track it, easily possess over a dozen different hyperparameters. These range from jump-distance segmentation thresholds over sampling ratios of positive and negative training samples to process noise levels for leg tracking. Even though there can be complex inter-dependencies between these parameters, they had until now been manually tuned through expert knowledge and coarse grid search, which can be a time-consuming effort. We instead propose to find the best set of hyperparameters using automated hyperparameter optimization – a so far rather underexplored topic in robot perception. To this end, several methods have been developed in the machine learning community, such as SMAC (Hutter et al., 2011), which we used in Chapter 4, or hyperopt (Bergstra et al., 2013). These methods optimize an objective function, which for the detection task could for example yield as a result the area under the precision-recall curve (AUC), or other metrics that we previously introduced in Section 2.5.

6.2.3 Experiments on a 2D laser dataset from an elderly care facility[6]

Considered methods

Taking into account the previously described improvements, we now want to experimentally compare the methods by Arras et al. (2007), Leigh et al. (2015) and the proposed deep learning-based approach on the DROW test set from an elderly care facility, in order to gain further insights. As an additional baseline without re-training or hyperparameter optimization, we include the ROS leg detector by Pantofaru (2010), which is often used in that form in experiments by other researchers. We do not consider the Gandalf detector based upon GDIF features from Weinrich et al. (2014)[7], which was already part of the experimental evaluation of Beyer et al. (2016) and had been outperformed by the original DROW on detection of walkers and wheelchairs.

Experimental setup

Due to the focus of this chapter, we present results only for the person class, even though DROW is capable of detecting multiple classes, *e. g.* different walking aids. For evaluation, we consider a detection as correct if the centroid lies within 0.5 m radius of an annotation, where at most one detection per detector can be assigned to a given groundtruth. All baseline methods have been integrated with the ROS-based human detection and tracking framework (Chapter 8) to facilitate experiments.

For the ROS leg detector by Pantofaru (2010), which involves a tracking step with association of detected single-leg hypotheses, we treat the resulting tracked human centroids on a frame-by-

[6] In this section, the experiments with DROW were conducted by L. Beyer and A. Hermans, wheras the experiments with all other baseline methods have been performed by me.

[7] This method is not straightforward to evaluate in our scenario since there exists no publicly available code to train the classifier, and perform the necessary association of leg hypotheses into human centroids. Furthermore, the available code does not expose confidence values or a detection threshold parameter, which makes it hard to obtain complete precision-recall curves.

frame level as if they were detections. Following the findings of Leigh et al. (2015), we reduce the leg reliability threshold from 0.7 to 0.3 as otherwise too few person tracks are initiated.

For the joint leg tracking method by Leigh et al. (2015), we treat all non-occluded human tracks as detections that have at least one currently associated leg detection. We compute detection confidences from their track confidence scores. For the purpose of a fair evaluation, we disable their proposed online occupancy grid mapping extension that can suppress false positive detections within walls and other static structures, since this is an orthogonal extension that could be combined with any of the other methods.

Re-training and improvement of the methods by Arras et al. and Leigh et al.

We have fully re-implemented the method of Arras et al. (2007) as a ROS node using the machine learning classifiers from the OpenCV library, to facilitate experiments with different classification methods. In earlier experiments, we found that a Random Forest classifier consistently yielded better results than the original Adaboost approach on several of our datasets, which we thus use. In initial experiments on the DROW dataset, we further observed many detection failures (false positives) at the boundaries of the 2D lidar field of view, when an object is entering the scene and is just partially visible. From studying the existing set of geometric features, we realized that the classifier has no way of knowing that an object may be truncated due to being close to the sensor boundaries. We therefore include an additional 4D boundary feature for each segment:

$$f_{\text{boundary}} = \begin{bmatrix} \text{angle to closest boundary} \\ \text{segment average angle} \\ \text{segment minimum angle} \\ \text{segment maximum angle} \end{bmatrix}$$

We have further adapted the training code of the methods of Arras et al. (2007) and Leigh et al. (2015) to the DROW dataset, and carefully re-trained the binary classifiers of both methods on the training set using positive labels of only the person class (without walking aids). DROW is always trained for all three classes (persons, wheelchairs and walkers), which in initial experiments showed no significant difference to training only on the person class.

Hyperparameter tuning setup

As previously described, we propose to find the best set of hyperparameters for the evaluated detection methods[8] through automated hyperparameter optimization. For the experiments in this section, we chose a distributed variant of *hyperopt* (Bergstra et al., 2013) that stores its trial results in a central MongoDB database. Our objective is to maximize detection performance, measured by the area under the precision-recall curve (AUC) on the DROW validation set, after training the methods on the DROW training set with a given set of hyperparameters. Because the detectors are fully embedded into ROS components, in every hyperopt trial we launch an

[8]We did not include the ROS leg detector in the tuning process, since our computational resources were limited and it is superseded by Leigh's approach, which extends the original ROS leg detector.

entire ROS stack, which proved to be non-trivial[9]. For each baseline method, we subsequently ran nearly 2,000 hyperparameter optimization trials using *Tree-structured Parzen Estimator (TPE)* search (Bergstra et al., 2013), which took ten days per method on a quad-core PC.

Similar to the previously described "classical" machine learning-based approaches, also the most important hyperparameters of DROW have been tuned using hyperopt. Due to limited computational budget, we excluded parameters that affect network structure and the training process. Here, the objective function maximizes the sum of the precision-recall AUCs over all classes (walkers, wheelchairs, persons).

In Figure 6.3, we show four exemplary hyperparameters of the method by Leigh et al. (2015). For each parameter, we generate a histogram which shows how frequently the parameter value in the corresponding bin has been sampled by hyperopt, and gives an idea of parameter ranges that are likely to be more promising. This is further supported by the coloring of the bars, which indicates the mean AUC attained within this bin. We can see, for example, that a Euclidean clustering distance of around 13 cm produces a segmentation with the best detection results (in terms of AUC on the validation set), that even very tiny segments with just 3 points should be tracked, that leg tracks should be initiated as quickly as possible, and which confidence percentile should be chosen for the validation gate during data association. For the method by Arras et al. (2007), the optimization *e. g.* yields a random forest with 128 trees of max depth 50 and a segmentation threshold of 23 cm. This is a larger clustering distance than for Leigh et al. (2015), because that method needs to be able to detect both legs individually.

Quantitative results

We now present quantitative results on the DROW test set. First, we examine the impact of temporal fusion. Ablation studies show that the late-fusion strategy significantly boosts performance especially for the person class: In Table 6.1 we see that the person and walker classes benefit most from temporal fusion if odometry is appropriately taken into account, while wheelchairs are mostly unaffected.

Next, we compare the proposed deep learning-based approach against classical methods. Figure 6.4 shows precision-recall curves for all evaluated methods on the task of person detection. We also report area under the curve (AUC), equal-error rate (EER) and peak-F1 measure.

Best detection results are obtained using the extended DROW method that incorporates temporal information via late fusion. Without temporal fusion, it achieves approximately the same performance as the leg-tracking approach by Leigh et al. (2015). This method gains 1.3% in AUC and almost 4% in peak-F1 score through automated hyperparameter optimization, which

[9] Within the objective function of each trial, we launch a new ROS master instance on a randomly assigned TCP port, such that multiple trials can be run in parallel per computer. Hyperparameters are relayed to the particular algorithm via the ROS parameter server. One limitation of our ROS-based approach is that the data playback rate has to be restricted to a factor of around 100 during training such that no frames are dropped. When tuning tracking hyperparameters via SMAC (Chapter 4), we noticed already that a lot of care has to be taken to configure publisher and subscriber queues correctly, to prevent buffers from exceeding their capacities, or non-deterministic synchronization behavior at high playback rates. Otherwise, the metrics computed by the objective function can be inconsistent. Besides increasing queue and buffer sizes, we therefore also added assertions to verify that the expected number of frames have been processed by the detector, and otherwise mark the hyperopt trial as invalid.

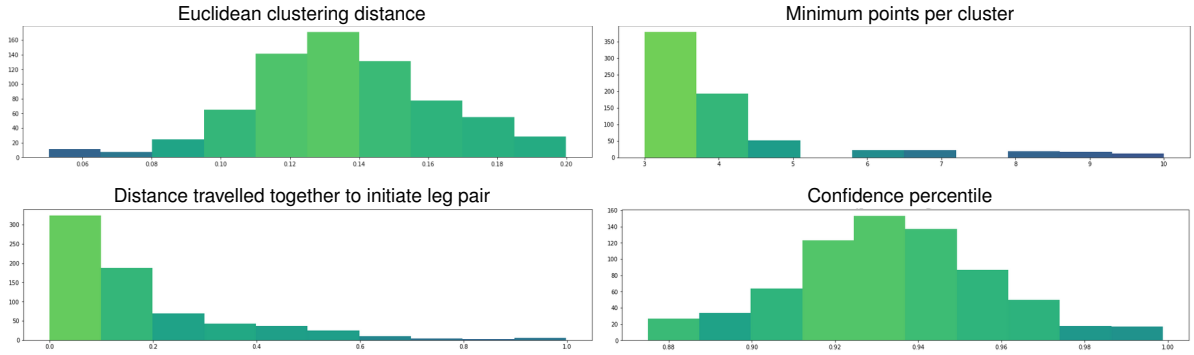


Figure 6.3: Distribution of optimal parameter values for selected parameters after hyperparameter tuning of the leg-tracking approach by Leigh et al. (2015) using *hyperopt* on the DROW validation set. Warmer colors indicate higher mean detection accuracies (AUC) obtained within this bin of the histogram.

	wheelchair			walker			person		
	AUC	p-F1	EER	AUC	p-F1	EER	AUC	p-F1	EER
Single frame	82.1	77.5	76.6	71.0	66.0	65.4	59.4	61.5	61.4
Naïve Late-fusion	79.4	74.9	74.2	73.1	70.7	69.9	64.6	64.5	64.1
+ Fixed location	80.4	76.7	76.1	77.7	73.4	72.1	67.0	65.9	64.9
+ Rotational odom correction	**82.7**	**78.7**	**78.1**	**82.4**	**78.9**	**76.4**	**68.1**	**68.1**	**67.2**

Table 6.1: Positive impact of incorporating temporal information from preceding frames into DROW on the person and walker classes. Results are on the DROW test set.

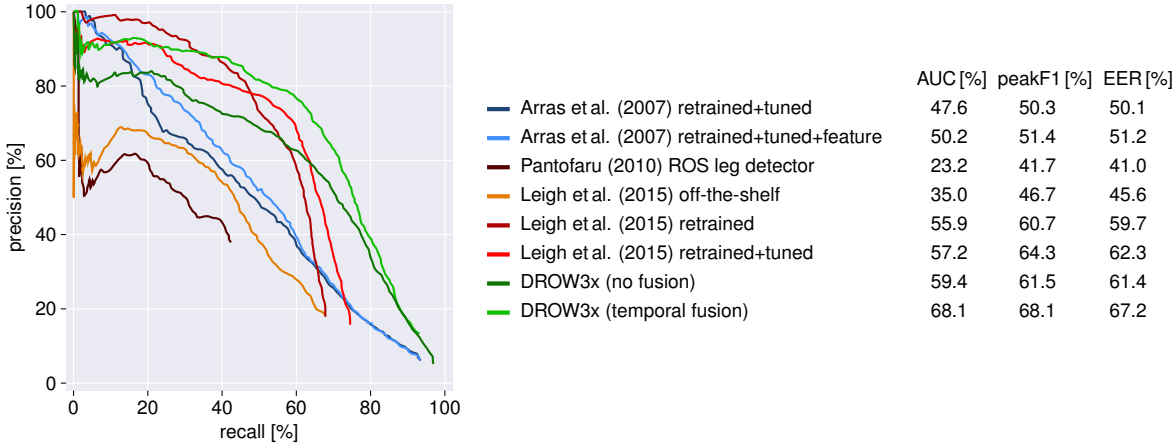


Figure 6.4: Quantitative results for 2D laser-based person detection on the DROW test set.

mainly leads to higher recall at cost of some precision. However, an even larger improvement of 14 percentage points was previously attained through re-training on the DROW training set (with the default parameter set)[10]. It is noteworthy that the approach by Leigh et al. (2015) in every case significantly outperforms the original leg-tracking method by Pantofaru (2010), thereby confirming the results of Leigh et al. (even without suppressing false positives using online occupancy grid mapping). The segment classifier by Arras et al. (2007), which neither performs individual leg detection, nor leg tracking, after retraining achieves better performance than Leigh et al. without retraining, but otherwise cannot beat their more informed method. However, the inclusion of the additional boundary feature that we introduced to reduce false positive detections at sensor boundaries leads to a measurable improvement in performance.

Range limitations of leg-tracking methods on basis of geometric features

We make the interesting observation that Leigh's joint leg tracker, even after extensive automatic hyperparameter optimization, only achieves a maximum recall of around 75%, while DROW and Arras et al. (2007) can achieve up to 95% recall. Further experiments reveal that the method of Leigh outputs basically no person detections beyond a distance of around 7 m, whereas the other methods still generate detections up to the maximum detection distance of 15 m. The reason for this appears to be that at 7 m distance and an angular resolution of 0.5°, a single leg cluster often consists of only two or less points, while some of the used geometric features by design require at least three points[11].

Our re-implementation of Arras et al. (2007), which uses similar geometric features, does not suffer from this issue because the entire person segment (including both legs) is used to compute features. We believe the limited maximum range is a significant limitation of the leg-tracking method of Leigh et al. (2015), as many robots are still equipped with 2D laser range finders with an angular resolution of 0.5° or less (*e. g.* Beyer et al., 2016; Weinrich et al., 2014; Luber, Tipaldi, et al., 2011b; Spinello et al., 2008), especially when considering the increasing adoption of low-cost, low-res 2D laser scanners in domestic applications.

Advantages of the deep learning-based approach

DROW achieves similarly high recall as the segment classifier from Arras et al. (2007), but is overall much more robust because it can learn more expressive features, and take into account temporal information without need for an explicit tracking stage. While Arras et al. (2007) had also tried to include motion-based features, they led to worse performance, which suggests that it is not always obvious how to make good use of temporal information. The proposed fusion methodology for DROW is simple, though more complex methods, e. g. based upon RNNs (Ondruska et al., 2016), could be worthwhile exploring. In addition, DROW is inherently multi-class – a capability that is difficult to integrate into model-based approaches that assume legs to be discernible.

[10] The large difference in performance can partly be explained by the dependence of the learned model's geometric features on the angular resolution of the sensor (originally $1/3$, here $1/2$ degree).

[11] We tried to retrain the classifier after setting the values of affected features to a constant in such cases, but neither obtained better results, nor real-time performance (< 1 Hz) using a lower minimum point count per leg cluster.

Interestingly, DROW did not benefit from automated hyperparameter tuning on the validation set. While the resulting hyperparameter values in some cases deviated drastically from initial hand-tuned estimates, this was not reflected in detection performance, which actually dropped slightly on the test set. Our intuition is that hyperopt slightly overfitted to the validation set, but due to the large number of trainable parameters (around 1.5 million), the CNN can easily compensate for most changes of the hyperparameters (*e. g.* size of the cut-out windows). It shall be noted that the optimization was performed jointly over the summed-up AUC of all classes, and slightly better results could be obtained when optimizing only for the person class.

Runtime performance

The methods by Arras et al. (2007), Pantofaru (2010), and Leigh et al. (2015) use geometric features that are easy to compute, some of them in combination with efficient nearest-neighbor tracking approaches, such that in moderately crowded scenes they easily achieve over 20 Hz on a single CPU core at test time.

While the larger DROW3x network is around 4× slower than the original method, its forward pass still takes less than 10 ms for a full laser scan with a GTX 1080 Ti GPU. However, the largest bottleneck is clearly the generation of cutouts during pre-processing, which involves a linear resampling step on the CPU that is conducted independently for each window. Without any further optimizations that could be implemented here, *e. g.* through parallelization, we obtain a frame rate of around 10 Hz on a single CPU core at 100% and GPU at 30-40%, after wrapping the improved DROW detector into a ROS node[12].

6.3 Comparison of 2D and 3D lidar detectors in an application scenario from the intralogistics domain

We now want to consider also baselines from other sensor modalities, and examine how well the examined methods generalize to different application scenarios from the intralogistics domain in a cross-modal comparison. In particular, we want to understand if recent deep learning-based 3D lidar detectors that have been trained on large-scale autonomous driving datasets have a significant advantage over the discussed 2D laser-based approaches, and how an RGB-D detector from the previous chapter compares to these methods. Such an experimental comparison of different detectors across different modalities on the same dataset is largely missing in the literature, in particular also with regard to challenging intralogistics scenarios.

6.3.1 Related work

Like in the previous section, we begin our analysis with a discussion of available datasets and then discuss methods for human detection using 3D lidar. These sensors are becoming increasingly relevant for mobile robots, as they are already in widespread use in the autonomous driving domain, where subsequent technological advances and expected price reductions would make them more appealing for future use in low-volume robotics products.

[12]The ROS wrapper was implemented by Kilian Y. Pfeiffer during an internship.

Datasets for human detection in 3D lidar

Research on 3D lidar-based detection of humans and other dynamic objects is nowadays mainly driven by autonomous driving scenarios. One of the first and most popular publicly available datasets from the autonomous driving domain is the KITTI benchmark dataset by Geiger et al. (2012), featuring data from a 64-beam lidar and corresponding camera images. The Oxford RobotCar dataset (Maddern et al., 2017) contains data from a 4-beam lidar, which is too sparse for our intended purposes. Just very recently, several commercial entities have started making parts of their autonomous driving datasets publicly available, including nuScenes (Caesar et al., 2020), the Waymo Open Dataset (P. Sun et al., 2019), the Lyft Level 5 Dataset (Kesten et al., 2019), and the Audi A2D2 Dataset (Geyer et al., 2019). All of these are significantly larger than the KITTI dataset, and have been recorded using different sensor platforms mounted on moving cars. They include labeled point cloud data with 3D annotations from one or multiple 3D lidar sensors with 16 to 64 beams, along with camera images annotated with 2D bounding boxes or segmentation masks. Besides pedestrians, further annotated dynamic object classes usually include cyclists and vehicles.

For robotics scenarios in indoor environments, it is still difficult to find a single, multi-modal dataset on which to train and evaluate methods from different sensor modalities. The L-CAS 3D Point Cloud People Dataset (Z. Yan et al., 2017) contains around 5.5k labeled point clouds from a 16-beam sensor on a partially static, partially moving robot platform in a university building. A later version (Z. Yan et al., 2018) adds 2D laser and depth images from an RGB-D camera. One limitation is that groups of people have not been annotated with individual person bounding boxes. For future work, the very recently released JRDB dataset (Martín-Martín et al., 2019) recorded with the "JackRabbot" social robot looks interesting. It contains 360° cylindrical stereo and monocular fisheye images, point clouds from two 16-beam lidars, two 2D laser scanners and RGB-D data from both indoor and outdoor scenes at a university campus, and a novel benchmark for 2D and 3D human detection and tracking.

However, our focus in this section is on the fundamentally different domain of intralogistics, *i. e.* warehouse environments, which is so far relatively underexplored and for which to our best knowledge no publicly available datasets exist.

Methods for human detection in 3D lidar

The field of 3D lidar-based object detection methods is currently rapidly evolving. In the KITTI 3D Object Detection leaderboard, we observe that around three times as many methods have only been evaluated on the comparatively simpler *car* class, and not the more difficult *cyclist* and *pedestrian* classes. The performance of leading 3D methods according to PASCAL VOC criteria (Everingham et al., 2012) for these classes at "moderate" difficulty level is 80%, 70% and 44% respectively, highlighting that pedestrians are indeed to most challenging class to detect. Z. Liu et al. (2020) propose TANet, which outperforms state-of-the-art methods on the pedestrian class. They note that the current detection accuracy for hard objects such as pedestrians is unsatisfactory, and becomes worse when additional noise is introduced. Like other recent methods such as VoxelNet (Zhou et al., 2018), SECOND (Y. Yan et al., 2018) or PointPillars (Lang

et al., 2019), it is based upon a voxel grid representation. However, in contrast to these methods that rely on a voxel feature encoding layer (Zhou et al., 2018) and a single-stage detector head inspired by SSD (W. Liu et al., 2016) to be efficient, and also different from computationally more complex two-stage methods like PointRCNN (Shi et al., 2019) that operate directly on raw point clouds, TANet uses an end-to-end trainable coarse-to-fine regression mechanism inspired by RefineDet (S. Zhang et al., 2018).

A recent, extensive review of further deep-learning based techniques for classification of 3D data is provided by Griffiths et al. (2019). In addition to the previously described approaches, the article describes classification methods that leverage representations on basis of unordered point sets (*e. g.* PointNet by Qi, H. Su, et al., 2017). Some of the derived detection approaches, like Frustum PointNets (Qi et al., 2018), Frustum ConvNet (Z. Wang et al., 2019) or AVOD-FPN (Ku et al., 2018), combine lidar and camera data. We do not consider such methods here.

Instead, in resource-constrained mobile robotics, some methods still rely on Euclidean clustering in the point cloud in combination with hand-crafted geometric features and SVM classifiers (Z. Yan et al., 2017) if no GPU acceleration is available.

6.3.2 Experiments

Dataset

We now present previously unpublished quantitative and qualitative experiments on two continuous sequences from a novel intralogistics dataset, recorded in a food factory with an adjacent storage room using an autonomous pallet truck during the EU project ILIAD. Details on the project and pictures of the robot platforms and environments can be found in Section 1.4.2.

The first of the two dataset sequences from a dynamic open-space scene had already been used in the previous chapter for the 3D evaluation of different RGB-D detectors. The second sequence has been manually labeled with 3D human centroids in the same way using our trajectory-based annotation tool from Section 8.8. It has a length of 120 seconds, and was acquired in a narrow and cluttered storage room with a single subject always in upright standing/walking pose. In comparison to the dataset from the elderly care facility from the previous section, the 2D laser scanner (of same type and resolution) is here mounted slightly lower at a height of around 15 cm, where it can still see legs of persons. The full sensor setup is described in Appendix A.4.

Considered methods[13]

For 2D laser, we include the previously discussed and improved methods by Arras et al. (2007), Leigh et al. (2015), and Beyer et al. (2018) in our experiments. All three have been trained on the DROW training set, which is the largest dataset available at leg height, allowing us to examine the generalization abilities of these methods.

In 3D lidar, we consider the CPU-only approach by Z. Yan et al. (2017), as well as the deep learning-based methods SECOND (Y. Yan et al., 2018) and its computationally more efficient derivative PointPillars (Lang et al., 2019). The first two have been trained on the KITTI dataset

[13] My colleague Robert Schirmer trained the SECOND and PointPillars detectors and integrated them with ROS.

and the latter on nuScenes, using existing training scripts. For PointPillars, we use the improved variant from the official Pytorch implementation of SECOND. Before training, we did not downsample the number of laser beams (64 in KITTI, 32 in nuScenes) to the 16 beams of our VLP-16 lidar, instead using data as-is.

We also include some of the RGB-D baselines from Chapter 5, namely the PCL-based head subclustering method with HOG-SVM by Munaro and Menegatti (2014), the 3D articulated human pose estimation by Zimmermann et al. (2018), and our own naïve YOLO v3 baseline which was trained on MS COCO and estimates depth via the median of the bounding box. For details on these methods, please refer to Section 5.7.1[14].

To gain insights on how multi-modal detection fusion is affected by the performance difference between detectors of different modalities, we further select the best detectors of each modality, and fuse their detections using nearest-neighbor association (*cf.* Section 8.6), before applying the evaluation protocol described in the following. Before the fusion step, we drop all detections with scores weaker than the respective detection threshold at peak-F1.

Experimental setup

As our evaluation is based upon 3D centroids, for all 3D lidar methods we compute centroids from their estimated 3D bounding boxes. We report peak-F1 measure and average precision (AP) following COCO protocol (T.-Y. Lin et al., 2014), where instead of maximizing IoU overlap we minimize the Euclidean distance between centroids with a rather coarse association threshold of 0.5 m. We discount groundtruth annotations that are fully occluded in 3D lidar, and only evaluate detections within the Kinect v2 field of view for a fair comparison of all modalities. We report results for the individual sequences *(a)* and *(b)*, as well as aggregated results over the combined set *(a+b)* of frames from both sequences. Experiments for runtime performance measurements were conducted on an AMD Threadripper 1950X CPU with GTX 1080 Ti GPU.

Figure 6.5 shows our quantitative results, along with the precision-recall curves of all detectors and a representative image for each scene.

6.3.3 Discussion of results

Performance of 2D laser-based detectors

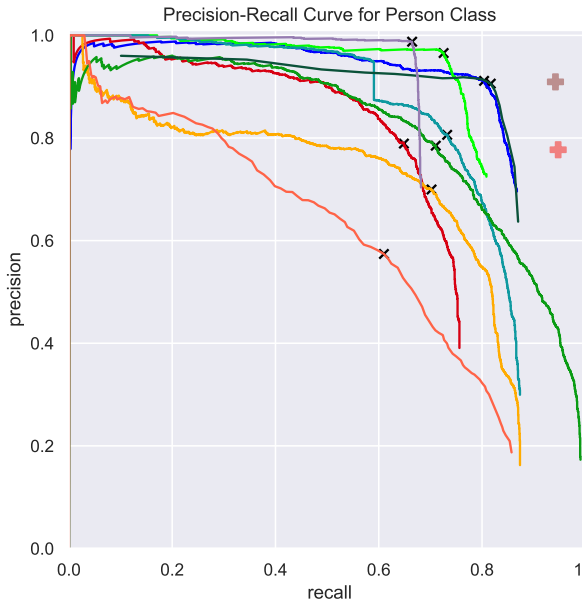
The 2D leg tracker of Leigh et al. and the improved version of DROW that we discussed earlier in this chapter are the only approaches in this comparison that incorporate temporal information; they clearly outperform the single-frame segment classifier of Arras et al. (2007).

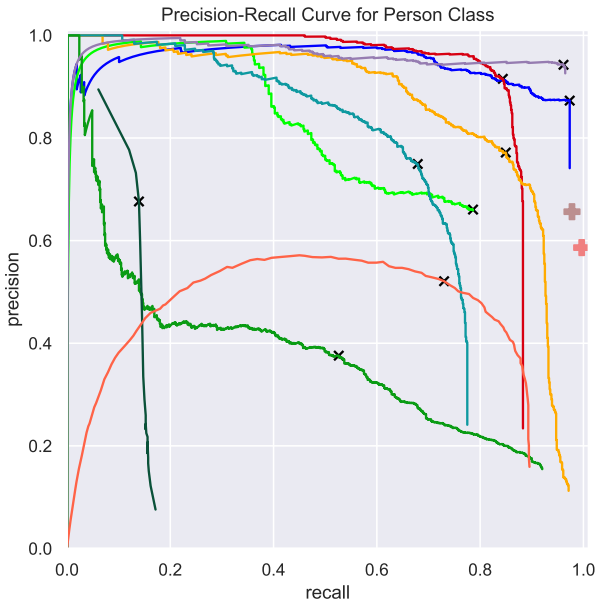
These methods were all trained on a dataset from an elderly care facility, where the same type of 2D laser is mounted at a height of 37 cm, compared to around 15 cm in ILIAD. We believe that the difference in mounting height might explain the somewhat lower precision attained by the deeply-learned DROW method; it appears that the model-based method by Leigh et al. in comparison generalizes better to different datasets and sensor setups. On the two intralogistics

[14]The quantitative results we report here are around 1% higher due to subtle differences in how we compute groundtruth occlusions. However, we made sure that all methods here are evaluated on the same groundtruth.

(a) NCFM Dynamic

(b) NCFM Storage Room





Modality	Method	Hor. FOV°	FPS (Hz)	VRAM (MiB)	AP[0.5m] ↑ in %			Peak-F1 ↑		
					(a)	(b)	(a+b)	(a)	(b)	(a+b)
2D laser	Arras et al. (2007)	180	25	-	56.8	48.4	52.6	0.59	0.61	0.60
	Leigh et al. (2015)		22	-	67.2	86.3	74.0	0.71	0.88	0.76
	DROW3x (temporal fusion)		10	1005	66.2	85.5	72.7	0.70	0.81	0.74
3D lidar	SECOND (Y. Yan et al., 2018)	360	10	3063	78.0	67.2	76.2	0.83	0.72	0.78
	PointPillars (Lang et al., 2019)		18	1200	81.4	36.8	65.6	0.74	0.44	0.61
	Object3D Detector (Z. Yan et al., 2017)		8	-	88.7	34.3	73.4	0.86	0.23	0.65
RGB-D (Kinect v2)	Munaro and Menegatti (2014)	86	21	-	79.1	68.2	76.3	0.85	0.71	0.77
	YOLOv3 (naïve depth) (Chapter 5)		27	1015	82.5	93.1	84.2	0.85	0.92	0.87
	RGBD Pose 3D (Zimmermann et al., 2018)		1	4055	67.8	93.1	76.1	0.80	0.95	0.85
3D+RGB-D	+ Detection fusion (SECOND, YOLOv3)	360	10	4078	-	-	-	0.93	0.79	0.88
2D+3D+RGB-D	+ Detection fusion (Leigh, SECOND, YOLOv3)	360	10	4078	-	-	-	0.85	0.74	0.82

Figure 6.5: Quantitative comparison of different human detectors across sensor modalities on two ILIAD sequences.

sequences, both methods are effectively on par in AP, with DROW again achieving a higher recall than the method by Leigh. Both approaches often misdetect pushcarts as humans, as can be seen in qualitative results in Figure 6.6.

The low mounting height of the safety laser scanner often leads to significant occlusions of the sensor's field of view caused by forks of other pallet trucks, which are being moved around in the first scene, but not in the second. The first, more dynamic scene additionally has people in difficult sitting or kneeing poses, whereas in the storage room scene, legs are mostly clearly visible and the subject is always in upright pose. This explains why the 2D detectors are surprisingly strong in the second scene, where they outperform all 3D lidar methods by a wide margin.

Performance of 3D lidar detectors

Due to the large sensor field of view, the 3D lidar detectors can provide 360-degree coverage around the robot with only a single sensor, which can be of benefit to many higher-level robot functions, while keeping computational requirements within acceptable limits. All three considered methods perform well on the dynamic, open-space scenario of the first scene, with the method by Z. Yan et al. (2017) even coming close to an RGB-D detector that uses high-resolution Kinect v2 images. In this scenario, the clustering-based approach might profit from the fact that all tall objects are indeed humans, and well-separated.

Overall, the SECOND detector (Y. Yan et al., 2018) is clearly the most precise 3D lidar-based method in our comparison. While PointPillars (Lang et al., 2019) is indeed faster, it produces many false positives in regions that are only very sparsely populated with points in the lidar point cloud. Further examination is needed if this is a limitation of the method, its many hyperparameters[15], or related to training on the nuScenes dataset[16].

For PointPillars and, to a lesser extent, the method by Z. Yan et al. (2017), we observe low recall at closer distances to the sensor. This scenario is more common in the second scene, where *all* 3D lidar detectors have very weak performance and are outperformed even by the methods that use the 2D safety laser. We noticed that close to the 3D sensor, only the torso of the person is visible, but neither legs nor head, due to the limited $\pm 15^\circ$ opening angle of our 3D lidar, and its horizontal mounting at around 1.2 m. Likely, such cases are not well-represented in the automotive training data, such that either additional 3D data augmentation, or sufficient training data from the target domain might be needed. However, we observed that the resulting body shape is sometimes also for a human not easy to recognize as a person, due to the lack of discriminative geometric features when the extremities are not visible. The method by Z. Yan et al. (2017) achieves extremely low recall in this scene, and appears to fail already during the clustering step when persons are pulling the ego-vehicle by putting hands on the handle bar.

[15] We tried to carefully hand-tune training and testing hyperparameters for several weeks without significant qualitative improvement. Possibly, a more systematic approach using automated hyperparameter optimization on a separate validation set could yield better results.

[16] Other users of the publicly available implementation also experienced weak performance after training and testing on nuScenes. It appears that this dataset is much harder to train on, because the annotations are multi-modal and sometimes 3D bounding boxes are annotated that barely contain any points in the lidar cloud. This would explain our observed behavior.

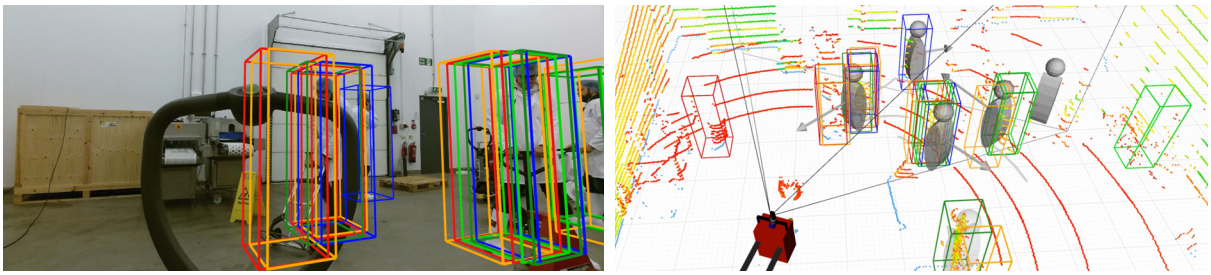


Figure 6.6: Qualitative results for human detection on the NCFM Dynamic scene at peak-F1 detection thresholds. Colors as in Figure 6.5, grey is groundtruth.

All 2D and 3D lidar methods have significant problems detecting a person very close to a wall (Figure 6.7, top left); of these, only the 2D DROW method intermittently detects the person. All lidar methods also often struggle to distinguish stacks of (loaded) pallets and shelves from humans; at least in 2D laser point clouds, the vertical supports of the pallets sometimes indeed have visual similarity to leg shapes.

Performance of RGB-D detectors

While most limited in horizontal field of view, the RGB-D methods clearly achieve the highest performance in both scenes. In this evaluation, we had not yet included our improved RGB-D detector from Chapter 5, which produces even stronger detection results. It can robustly detect humans in any kind of body pose (e. g. sitting, kneeing, lying on the floor), also when leaning against walls or other structures.

Impact on detection fusion

Due to the way how we presently fuse detections, the fusion can mainly increase recall, but not precision. In the first scene, we see that fusing RGB-D and 3D lidar clearly leads to better results. In the second scenario, the weak performance of the 3D lidar method leads to a significantly decreased precision – which means that in this case, it would be better not to fuse at all. This shows that further work is required to improve the way in which multiple modalities are fused, such that we can benefit from the individual strengths of different detectors, but do not succumb to their weaknesses. The additional inclusion of 2D lidar leads to worse results on both scenarios – upon qualitative inspection, it appears that the small latency induced by the Kalman filter of the leg tracker sometimes makes data association at the fusion stage fail. This could likely be alleviated by a small forward-prediction of the leg tracker's track states.

6.4 Conclusions

In this chapter, we first examined and improved methods for human detection in 2D laser range data. In particular, we gained a better understanding of different model-based approaches and a deep learning-based method by evaluating them on the same, large-scale dataset from an elderly care facility. One finding was that incorporating temporal information into the CNN-based approach can significantly boost its performance for the human class, where leg motion provides valuable cues. We further showed how to perform automated hyperparameter optimization with

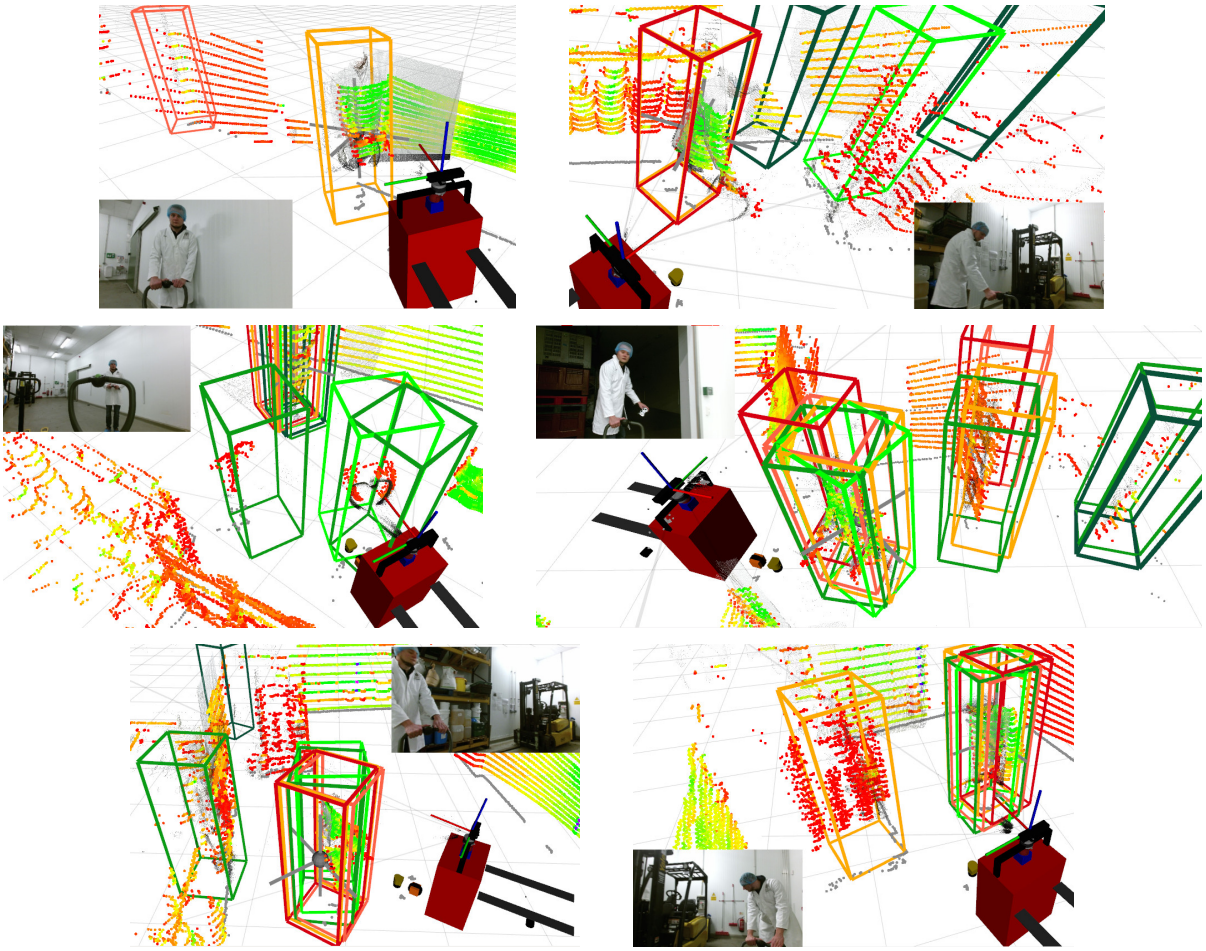


Figure 6.7: Six examples of false negative and false positive detections of the different 2D and 3D lidar detectors in the NCFM Storage Room scene. Colors correspond to those from Figure 6.5, groundtruth is indicated by grey crosses.

detectors that are integrated into a ROS framework. In some cases, this can significantly improve performance with regard to the target metric, though the computational effort is rather high.

We then focused on an application scenario from the intralogistics domain, where we included additional 3D lidar-based and RGB-D baselines to conduct an insightful and valuable cross-modal evaluation – so far an under-explored topic in robot perception. The focus here was not on developing a new method, but rather on trying to get a deeper understanding of the strengths and weaknesses of different detection approaches, and the associated sensor modalities – in particular also with regard to their computational requirements from an application perspective.

We saw that different indoor environments can have a huge impact on how different detectors behave. While RGB-D methods appear rather robust, the accuracy of 3D lidar-based methods can drop drastically in certain narrow and close-distance scenarios, and it needs to be further understood why. We also found that methods which rely on 2D laser at leg height work best if people are in standard (standing, walking) poses and not pushing carts.

From our final experiments, we also learned that significantly more work is required on the fusion part. Ideally, this should happen at a much earlier stage than after detection, if we want to exploit the complementary information from multiple modalities. Mees et al. (2016) proposed an approach for *adaptive* multi-modal fusion by "choosing smartly" via a gating network in a mixture of experts. However, the proposed approach assumes a single two-stage detector, where each expert operates on the same input 2D bounding box from a shared region-proposal stage. This is not so easy to realize in our more heterogeneous multi-modal scenario, where we are dealing with image data on the one hand, and 2D or 3D point clouds on the other. Training such methods further depends on the availability of sufficiently large, multi-modal datasets. One downside of such approaches could then be that the learned models might be rather specific to the particular sensory setup, and not generalize easily to other robots. Possibly, synthetically generated data (Chapter 5) could help in improving this situation.

Ideas for future work

Based upon our experiments and qualitative insights, the inclusion of temporal information appears to be the most promising way of further improving detection performance in lidar data. Here, even subtle cues can be helpful: Even if a person is standing still, small arm movements that are visible in 3D lidar can already tell us that this is a human; this goes into the direction of model-free tracking approaches, which we briefly discussed in Section 2.1, though we believe that more human-specific domain knowledge could be incorporated. More advanced ways of modeling the temporal aspect in a deep neural network, for instance using recurrent neural networks, and possibly up to the point of end-to-end tracking approaches as proposed by Ondruska et al. (2016), should also be considered.

Because the intralogistics dataset that we used in our multi-modal evaluation is very recent, our experiments were so far limited to two short, hand-labeled sequences. It is planned to further extend the amount of labeled data and the evaluation until the end of the ILIAD project.

None of the considered approaches have so far been trained or fine-tuned on data from the intralogistics domain, as we have not yet annotated a sufficiently large training set. Overall, there appears to be a strong need for more domain-specific robotics datasets in 3D lidar. Given the strong performance of modern image-based detectors, it makes sense to automate this as far as possible, *e. g.* using multi-sensor transfer learning as proposed by Z. Yan et al. (2018). The RGB-D detector that we presented in Chapter 5 is a good candidate for this purpose, as it already outputs metric 3D coordinates. Resulting centroids can easily be used to generate training data for the discussed 2D lidar methods. For the 3D detectors, usually oriented 3D bounding boxes are used as training labels, which we want to integrate into our RGB-D detector in the future.

Finally, one – to our best knowledge – still open question is what is the best mounting height for 2D laser- or 3D lidar-based detection of humans. While there are, of course, often technical limitations on where on a robot a sensor can even be installed, it would be interesting to see a systematic evaluation of different mounting levels, and their impact on detection performance on the same use-case. Possibly, this could be achieved through a sufficiently sophisticated simulation, which might also be useful for training a height-agnostic detector given enough data.

Part IV

Taking a closer look at humans

CHAPTER 7

Human attribute recognition in RGB-D

After having examined how we can robustly detect humans, we now want to take a closer look at them. In this chapter, we address the problem of recognizing binary human attributes, for example related to gender and clothing, from RGB-D point clouds. Unlike previous work which typically considered faces or frontal body views in image data, we address the problem from side and back views as well. We propose to solve the task with a tessellation-based, boosted classifier which selects the most informative subset of features and the best scales and locations at which to compute these features for a given attribute. The method, based upon earlier work by Spinello, Luber, et al. (2011) for the human detection task, uses a set of predefined geometric point cloud features which we extend with further geometric and color-based cues.

We present a large, novel, gender-balanced RGB-D human attributes dataset with full-body views of over a hundred different persons captured with the Kinect v2 sensor at varying distances and body orientations in both standing and walking poses in a controlled environment. We evaluate the proposed method using our dataset on six human attributes, namely *gender*, *has long hair*, *has long trousers*, *has long sleeves*, *wears skirt/dress* and *wears jacket*, with regard to an image-based HOG-SVM baseline. For the gender attribute, we additionally consider two previously published deep learning-based approaches for RGB-D object classification. Our experiments show that the method achieves competitive results and can robustly recognize multiple attributes across different view directions and distances to the sensor with accuracies up to 90%. We further present previously unpublished qualitative results for gender recognition on challenging in-the-wild data from an airport environment, where we gain interesting insights on remaining failure cases. Our method runs in real-time at around 300 Hz for a single attribute without requiring GPU acceleration. This makes it well-suited for deployment on a resource-constrained mobile robot.

Parts of this chapter have been presented in a paper "Real-Time Full-Body Human Gender Recognition in (RGB)-D Data" by T. Linder, S. Wehner and K. O. Arras at the IEEE International Conference on Robotics and Automation (ICRA) 2015 in Seattle, WA, USA.

Subsequent parts have been presented in a paper "Real-Time Full-Body Human Attribute Classification in RGB-D Using a Tessellation Boosting Approach " by T. Linder and K. O. Arras at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2015 in Hamburg, Germany.

7.1 Introduction

The ability to describe and re-identify individual persons that interact with a robot is an important perceptual skill for robots in human environments, for example, when providing personalized, user-friendly services. The *fine-grained recognition* task of extracting human attributes such as gender, age group, or clothing-related attributes from a person's appearance can help in solving these higher-level tasks. For example, C. Su et al. (2016) show that mid-level human attributes, such as the ones considered in this chapter, can be successfully used for person re-identification.

The problem of gender recognition[1], as a particular instance of human attribute recognition, has traditionally been studied in the computer vision and surveillance communities on the basis of facial appearances. Initially, few authors examined the use of upper-body (B. Li et al., 2012) or full-body views (Collins et al., 2009; Bourdev et al., 2011; Ng et al., 2013). More recently, challenging, diverse, large-scale "in-the-wild" datasets to benchmark deep learning-based approaches for generic human attribute recognition have surfaced that cover a larger variety of human poses and camera viewpoints (Y. Deng et al., 2014; Y. Li et al., 2016; X. Liu et al., 2017; D. Li et al., 2019).

However, even as of 2019, human attribute recognition is relatively unexplored for RGB-D data, although particularly relevant for robotics. Unlike methods that rely on image data only, which may suffer from varying illumination conditions when deployed on a mobile robot, RGB-D approaches can be more robust to lighting conditions, as 3D point clouds allow for the extraction of geometric cues in addition to visual appearance. There is a limited number of RGB-D methods in this area, focusing *e. g.* on recognizing color-based clothing attributes (W. Liu et al., 2012), person re-identification (Barbosa et al., 2012; Munaro, Fossati, et al., 2014), or full-body human attribute recognition while, however, requiring accurate estimates of skeletal joint angles (H.-J. Wang et al., 2013). Therefore, our proposed method makes an interesting contribution to a so far under-explored field with high relevance to robotic applications.

Outline This chapter is structured as follows: After discussion of related work, we present an initial version of the tessellation-boosting approach for human attribute recognition, which uses only geometric 3D features. We then present our novel RGB-D human attributes dataset, before conducting experiments on the gender recognition task. Next, we extend the method with further geometric and color-based features, and evaluate it on a larger set of human attributes. Finally, we integrate a learned gender classifier with our human detection and tracking system, and perform qualitative experiments on real-world data recorded inside an airport terminal. We conclude with a number of ideas on how the method could be further improved.

[1] For comparability of our results with existing methods, we treat gender recognition here as a binary classification problem, as is common in literature and datasets from the computer vision community. A recent article on *"Gender Recognition or Gender Reductionism? The Social Implications of Embedded Gender Recognition Systems"* by Hamidi et al. (2018) discusses concerns and potential negative consequences of using such systems *e. g.* in social robotics, and provides recommendations on how to accommodate gender diversity when designing new systems. The method we propose in this chapter is technically limited to solve binary classification problems, where gender is one out of several binary attributes that we consider. Wengefeld, Lewandowski, et al. (2019) have recently extended the method to multiple classes for rough body orientation estimation. In a similar fashion, additional classes could be considered for particular human attributes, given the availability of sufficient training data.

7.2 Related work

Computer vision methods for human attribute recognition in RGB images

A recent survey on pedestrian attribute recognition is presented by X. Wang et al. (2019). It covers 14 different datasets and over 30 recent methods, mostly dealing with binary attribute classification on full-body RGB images, starting with a popular parts-based approach by Bourdev et al. (2011). They learn 1200 *poselets* that represent small parts of the human body under a specific pose, with the goal of decomposing view point and human pose from appearance. On these poselets, feature vectors based upon histogram of oriented gradients (HOG), HSB color histograms and skin mask features are computed which are fed into three layers of SVM classifiers. Example attributes include *is male*, *has hat*, *has t-shirt*, *has shorts*, *has glasses*, or *has long pants*. The authors report an average precision (AP) of around 82.4% for gender, 73% for long hair, 74% for long sleeves and 90% for long pants on a database of 8000 color images containing persons in a large number of different poses and viewpoints under mostly favorable illumination conditions. According to Bourdev et al., the method *"automatically determines the optimal location, scale and viewpoint to look for evidence for a given attribute"* – which is similar to what our proposed method aims at, but on RGB-D point clouds instead.

The PANDA approach by N. Zhang et al. (2014) replaces the SVMs and manually crafted features by "pose-aligned" deep convolutional neural networks for the individual poselets, as well as the overall image. The approach increases the mean average precision by 13% over the results of Bourdev et al. (2011). However, they require large datasets: Images of around 25,000 people are extraced from Facebook, in addition to 8000 color images from an existing database to avoid overfitting. Gkioxari et al. (2015) match the performance of PANDA without using parts. They show that the gain from explicitly modeling parts is only marginal, and conjecture that it might vanish with more powerful, future CNN architectures. Indeed, Sarafianos et al. (2018) recently proposed an end-to-end trainable network that incorporates a *visual attention mechanism* with only attribute-level supervision. They argue that *"Giving emphasis to the upper part of an image, where the face is located, for attributes such as 'glasses' and to the bottom part for attributes such as 'long pants' can increase the recognition performance as well as the interpretability of our models"*, which coincides with our motivation. Our point cloud-based tessellation-learning approach does not rely on any explicitly modeled body part decomposition or articulated human pose estimates, and can be trained on a dataset with significantly fewer person instances because it uses lower-dimensional hand-crafted feature representations.

Human attribute recognition in 3D and RGB-D

Tang et al. (2011) recognize gender from a large set (2484 persons) of 360° full-body high-resolution laser scans created with an expensive stationary scanner. The approach requires several costly steps including hole-filling, mesh smoothing and normal computation and has been evaluated in a setting where people wear no clothing.

Using RGB-D data, the only approach to our knowledge is due to H.-J. Wang et al. (2013) who recognize multiple attributes using a similar approach to Bourdev et al. (2011) and N. Zhang et al. (2014) in the sense that individual SVM classifiers are trained on body parts instead of

the whole body. Instead of poselets, they use articulated human pose estimates provided by the Kinect sensor to generate sampling regions around body limbs for the computation of HOG, LBP, Gabor filter and color/depth histograms. Trained on a dataset with full-body views of over 4,600 persons recorded by a sensor mounted above the entrance of an elevator, they outperform a reimplementation of Bourdev et al. (2011) and achieve equal precision and recall (EPR) rates of 87% on gender, 87.4% on short sleeves and 91.6% on long pants.

Unlike these methods, our approach does not require skeleton estimates and instead fully leverages 3D shape data that is relatively robust with regard to illumination conditions. As a result, our method achieves good recognition performance already using only 3D point cloud data. Finally, due to its simplicity, our approach is extremely efficient at 125 Hz on a single CPU core (300 Hz on four cores) without GPU acceleration, which is highly relevant for real-time implementations on resource-constrained mobile robots.

Datasets for human attribute recognition in RGB-D

While there exists a large number of RGB image datasets for gender recognition and human attribute recognition, such as the recent "PETA"[2], "WIDER", "PA-100K" and "RAP" datasets (Y. Deng et al., 2014; Y. Li et al., 2016; X. Liu et al., 2017; D. Li et al., 2019), there are only few suitable full-body RGB-D datasets.

The "RGB-D people dataset" by Spinello and Arras (2011) for person tracking or the "BIWI RGBD-ID dataset" by Munaro, Fossati, et al. (2014) for person reidentification contain too few individuals or are insufficiently gender-balanced to *e. g.* train a reliable gender classifier. The "IIT RGB-D Person Re-identification Dataset" by Barbosa et al. (2012) includes 79 people walking down a hallway mostly in frontal or rear view. In the "TUM Gait from Audio, Image and Depth Database" (Hofmann et al., 2013), people also walk mostly into the same direction, thus these datasets lack variety in body orientations. None of these datasets include data captured with the Kinect v2 sensor, which provides a significantly improved depth resolution over the first-generation sensors.

The dataset that we will present in Section 7.3.2 is well-suited for learning human attribute recognition from RGB-D point clouds. Human subjects follow multiple complex walking patterns, such that the dataset contains views of people at various relative orientations and distances to the sensor. It also contains a close-up sequence that is valuable for human-robot interaction. Unlike other RGB-D datasets, our dataset includes more subjects (118 persons) and is largely gender-balanced. This made it the largest and most complete dataset for the purpose of human attribute recognition in RGB-D at the time it was published (Linder, Wehner, et al., 2015a).

7.3 Full-body human gender recognition in depth data

We now introduce our proposed approach to full-body human attribute recognition from RGB-D point clouds. As mentioned in the beginning, our approach is based upon a tessellation-learning

[2]PETA is based upon 10 different datasets originally created for person re-identification.

method originally developed by Spinello, Luber, et al. (2011) for human detection, where it has been evaluated on 3D lidar data. For our work in this chapter, we re-implemented the method, show that it can be successfully applied to the human attribute recognition task in RGB-D data, and extend it with further geometric and color features. We perform qualitative and quantitative experiments and ablation studies on our novel RGB-D human attributes dataset.

In this section, we present the initial version of our proposed method, which relied solely on geometric 3D features and did not yet take any color information into account. This initial variant has been evaluated on *gender recognition*[1], with some qualitative results visualized in Figure 7.1 to highlight the idea.

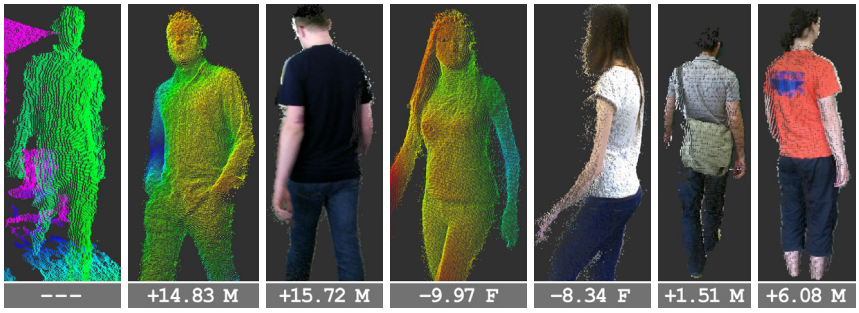


Figure 7.1: 3D and RGB-D data from our human attribute dataset with corresponding *gender* classification results using only *geometric features* (no RGB). The leftmost image has been taken with a Kinect v1 sensor, the remaining ones with a Kinect v2. Numbers below the images are confidences of the boosted classifier, the sign indicates the predicted label (M for male, F for female).

7.3.1 Proposed approach for human attribute recognition

We assume that we are given 3D person detections in an RGB-D point cloud, for example from a human detector as presented in Chapter 5. We then center a fixed-size bounding volume $\mathcal{B}$ around the median in x and y of all points that we consider as belonging to the person. The size of $\mathcal{B}$ can either be fixed and taken from the maximum expected object size, or be learned from a training set. It should represent some form of upper bound, and not vary across persons.

We now describe the process of generating axis-parallel tessellations of the bounding volume $\mathcal{B}$. Then, we present our set of geometric features and how to train an AdaBoost classifier on these features in the resulting set of tessellations, as proposed by Spinello, Luber, et al. (2011).

Generation of tessellations

Our first goal is to subdivide the 3D bounding volume $\mathcal{B}$ into voxels, in which we can then compute local point cloud features. This leads us to the question of how a volume can be tessellated into a collection of smaller volumes, a problem well known as *tiling* in computational geometry. For the sake of simplicity, we consider only *axis-parallel voxels* which reduces the

Algorithm 2: Compute all axis-parallel tessellations $\mathcal{T}$ of a volume $\mathcal{B}$.

Input: Bounding volume $\mathcal{B}$ of size $w_\mathcal{B} \times d_\mathcal{B} \times h_\mathcal{B}$, set of voxel proportion constraints $\mathcal{C}$, list of voxel scaling factors $\mathbf{s}$, protrusion tolerance θ. **Output:** Set of all possible tessellations $\mathcal{T}$

$\mathcal{T} \leftarrow \{\}$
foreach $s_j \in \mathbf{s}$ **do**
  foreach $\mathbf{c}_k = (w_k, d_k, h_k) \in \mathcal{C}$ **do**
    $w = s_j \cdot w_k;\ d = s_j \cdot d_k;\ h = s_j \cdot h_k$
    if $\mathtt{rem}(w_\mathcal{B}, w) < \theta \wedge \mathtt{rem}(d_\mathcal{B}, d) < \theta \wedge \mathtt{rem}(h_\mathcal{B}, h) < \theta$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \mathtt{Tess}(\mathcal{B}, w, d, h, 0, 0, 0)$
      $\mathcal{T} \leftarrow \mathcal{T} \cup \mathtt{Tess}(\mathcal{B}, w, d, h, \frac{w}{2}, \frac{d}{2}, \frac{h}{2})$
return $\mathcal{T}$

complexity of the problem but still leaves an infinite number of tessellations of $\mathcal{B}$. Therefore, we introduce a set of *proportion constraints* $\mathcal{C}$ to exclude extreme aspect ratios of voxels and a list of *increments* $\mathbf{s}$ by which voxels will be enlarged. Each element $\mathbf{c} = (w, d, h) \in \mathcal{C}$ is a width-depth-height triplet with multipliers for the respective voxel dimension.

The resulting procedure in Algorithm 2 from Spinello, Luber, et al. (2011) generates all possible voxel sizes subject to $\mathcal{C}$ and $\mathbf{s}$. Defining the remainder after ceiling-division $\mathtt{r}em(a, b)$ as $|a - \lceil \frac{a}{b} \rceil b|$, the algorithm tests whether voxels can fill a volume $\mathcal{B}$ without gaps and subdivides $\mathcal{B}$ into a regular grid. The function $\mathtt{T}ess(\mathcal{B}, w, d, h, \Delta_w, \Delta_d, \Delta_h)$ produces a regular face-to-face tessellation of $\mathcal{B}$ with voxels of size (w, d, h) and offset $(\Delta_w, \Delta_d, \Delta_h)$ to also allow voxels that overlap each other. The algorithm generates gapless subdivisions of $\mathcal{B}$ that are complete in that no tessellation is missing under the given constraints. We also allow slightly protruding voxels with a tolerance θ.

Tessellation constraints

For our experiments on human attribute recognition, we deviate from the original constraints by Spinello, Luber, et al. and use the scaling factors $\mathbf{s} = (0.1, 0.2, ..., 0.8)\ [m]$ and proportions $\mathcal{C}$ being the set of all permutations of $\{\{1, 1, 1\}, \{1, 1, 1.25\}, \{1, 1, 2\}, \{1, 1, 2.5\}, \{1, 1, 3\}, \{1, 1, 4\}, \{1, 1, 5\}, \{1, 1, 6\}, \{1, 1, 8\}, \{1, 1, 10\}, \{2, 2, 3\}, \{4, 4, 2\}, \{4, 4, 3\}\}$. In particular, we allow for smaller minimum voxel sizes and more elongated voxel proportions[3] than in the original method, to be able to recognize more fine-grained details. These constraints lead to 134 valid tessellations, of which some examples are shown in Figure 7.2.

Computation of voxel features

Let $\mathcal{T}_j$ be the jth valid tessellation and $\mathcal{V}_j^i$ its ith voxel. Then, for each $\mathcal{V}_j^i$ of all generated $\mathcal{T}_j$'s, we determine the set $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of points from the point cloud that reside within this voxel. With the goal to describe shape properties locally, we then compute a set of RGB-D point cloud features f_i that characterize geometrical and statistical properties of $\mathcal{P}$. Initially, we use the same set of geometric features from Spinello, Luber, et al. (2011), listed in Table 7.1. Most of them can be computed very efficiently from the points' scatter matrix via eigenvalue decomposition and none of them require costly computation of surface normals.

[3] These elongated voxels are *e. g.* used by the learned classifier in Figure 7.11 (right) on page 157.

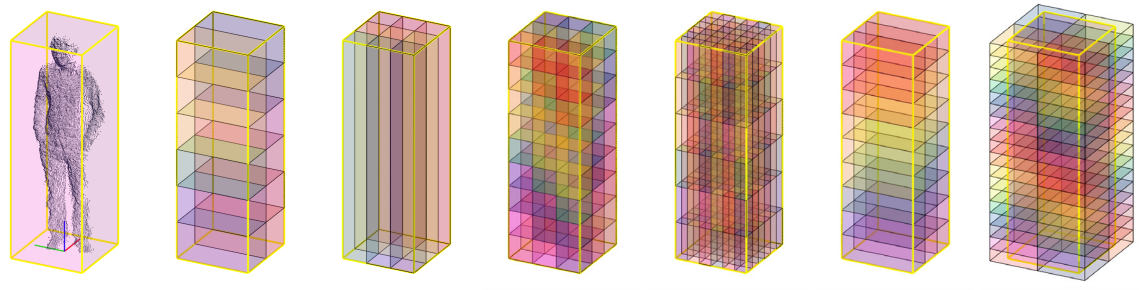


Figure 7.2: *Left:* Bounding volume $\mathcal{B}$ centered around the median of the person candidate point cloud in x and y. *Remaining pictures:* Example tessellations of the bounding volume generated using our tessellation algorithm. We also allow protruding voxels, shown in the rightmost picture.

Training of the AdaBoost classifier

Training samples are formed by stacking the features of *all* voxels of *all* tessellations into one large feature vector and associating the corresponding ground truth class label. We train an AdaBoost classifier with n_{weak} decision stumps as weak learners. After training, the final model is given by the collection of all voxels in which *at least one feature* has been selected. The resulting strong classifier achieves a double objective, it selects the best features ("best" quantified by the AdaBoost voting weights) and selects the optimal subdivision $\mathcal{T}_{opt}$ of $\mathcal{B}$ for the classification task at hand. The method can select an arbitrary number of features in each voxel – a large number, for instance, means that the voxel contains a particularly salient local shape – and may also select a mixture of voxels from *different* tessellations. This implicit feature selection is performed separately for each human attribute, yielding an attribute-specific classifier.

7.3.2 Human attributes dataset

Before we describe the further setup of our experiments, we now introduce our novel human attributes dataset, motivated by the previously described limitations of existing datasets for the full-body, multi-perspective human attribute recognition task on RGB-D data. We acquired and annotated[4] an RGB-D dataset including 118 persons (54 male, 64 female) under controlled conditions with the Kinect v2 sensor. The subjects' mean age is 27 years ($\sigma = 8.7$), the age of the youngest participant is 4 and the oldest 66 years. All subjects, or their respective legal guardian, signed a mandatory consent form, in which they were informed about their personal rights, data security measures taken and the intended usage of the data. Through an opt-in checkbox, the subjects could voluntarily agree for the data to be made publicly available for academic research purposes, which 105 out of 118 participants agreed to. Participants received a small monetary compensation for their participation, regardless of whether they opted in or not. In the resulting

[4]My master student S. Wehner, with the help of M. Timon, performed most of the data acquisition under my supervision. Further attributes besides gender and age were annotated by me and two colleagues.

#	**Description**	**Expression**
1	Number of points	The point count of $\mathcal{P}$ denoted as n. $f_1 = n$
2	Density	Captures the normalized point density w.r.t. the entire point cloud: $f_2 = \frac{n}{N_\mathcal{B}}$
3	Sphericity	Captures the level of sphericity from the ratio of the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ extracted from the scatter matrix of $\mathcal{P}$. $f_3 = 3\frac{\lambda_3}{\sum_i \lambda_i}$ where $\lambda_1 > \lambda_2 > \lambda_3$
4	Flatness	Measures the degree of planarity from the eigenvalues. $f_4 = 2\frac{\lambda_2 - \lambda_3}{\sum_i \lambda_i}$
5	Linearity	Captures the level of linearity from the eigenvalues. $f_5 = \frac{\lambda_1 - \lambda_2}{\sum_i \lambda_i}$
6	Standard deviation w.r.t. centroid	Measures the compactness of points in $\mathcal{P}$, $f_6 = \sqrt{\frac{1}{n-1}\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})^2}$ where $\bar{\mathbf{x}}$ is the centroid.
7	Kurtosis w.r.t. centroid	Captures the peakedness of points in $\mathcal{P}$, fourth centralized moment of the data distribution in $\mathcal{P}$. $f_7 = \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})^4 / f_6$.
8	Average deviation from median	Alternative measure of compactness. $f_8 = \frac{1}{n}\sum_i \|\mathbf{x}_i - \tilde{\mathbf{x}}\|$ where $\tilde{\mathbf{x}}$ is the vector of independent medians $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z})$.
9	Normalized residual planarity	Alternative measure of flatness. Squared error sum of a plane fitted into $\mathcal{P}$ normalized by n. $f_9 = \sum_i^n (a\,x_i + b\,y_i + c\,z_i + d)^2$ where a, b, c, d are the parameters of the plane derived from the eigenvalues of the scatter matrix.

Table 7.1: Geometric features originally introduced by Spinello, Luber, et al. (2011)

dataset, no names or contact information are associated with the recordings that belong to a specific person; instead, recordings have been numbered in sequential order.

All data was recorded at 15 Hz in three different indoor locations at the University of Freiburg under controlled lighting conditions. The locations include a robotics laboratory (60% of all recordings), a lecture hall (6%), and a seminar room (34%), which have different backgrounds and are shown in Figure 7.3. Example color frames with foreground are shown in Figure 7.4.

Subjects were asked to perform several standing and walking patterns that were demonstrated beforehand and have been designed to cover all relative body orientations and the full Kinect v2 RGB-D sensor range between 0.5 m to 4.5 m[5].

As visualized in Figure 7.5, we recorded four distinct sequences for each participant. In sequence 1, the subject is standing at a fixed distance of around 2.5 m from the sensor, and rotating clockwise in 45° steps (one image per step). The continuous sequence 2 (typical length ∼370 frames) consists of a video of the person performing a complex walking pattern so as to capture various distances from the sensor and relative orientations. In sequence 3 (∼300 frames), the person walks on a circle that covers almost the entire view frustum, in both clockwise and

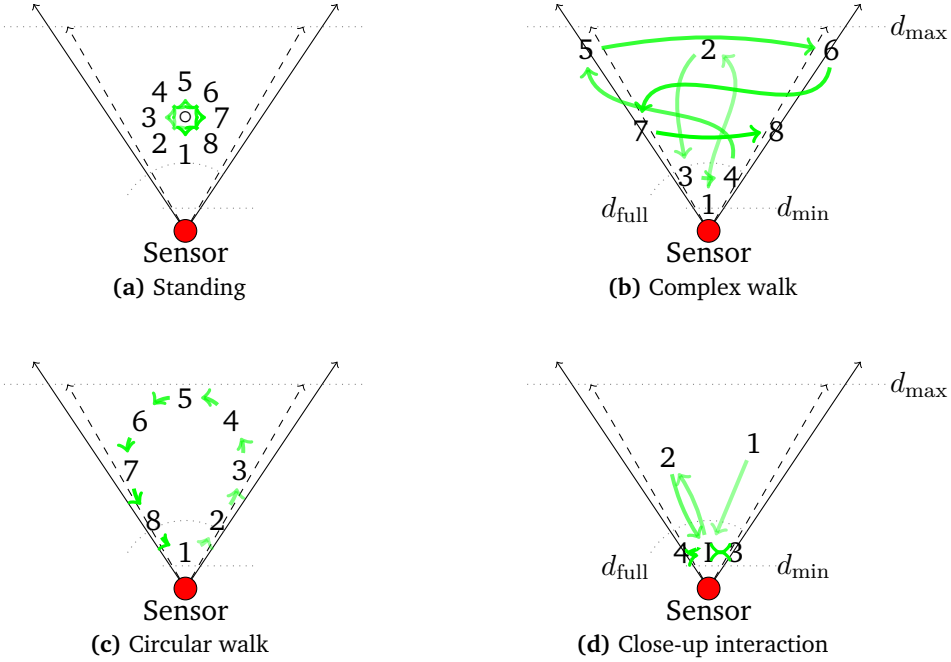
[5] About 75% of the data were recorded using a developer preview version of the Kinect v2 sensor. We did not observe any notable difference in data quality compared to the final version, apart from the 4.5 m range limitation which was lifted to around 9 m for the final version.

(a) Robotics laboratory **(b)** Lecture hall **(c)** Seminar room

Figure 7.3: Recording locations of our RGB-D human attributes dataset

(a) Sequence 1 (static) **(b)** Sequences 2+3 (walking) **(c)** Sequence 4 (close-up)

Figure 7.4: Example images from the sequences in our RGB-D human attributes dataset. Faces have been blurred for publication, but not in the original dataset.



(a) Standing **(b)** Complex walk **(c)** Circular walk **(d)** Close-up interaction

Figure 7.5: Standing and walking patterns performed by human subjects in our dataset. (a) Standing pose in 8 different orientations, at around 2.5 m distance to the sensor. (b) A complex walking pattern, designed to capture a large variety of poses and distances to the sensor. (c) Circular walking pattern in both directions. (d) A close-up human-robot interaction pattern. $d_{\min}$ and $d_{\max}$ denote the Kinect v2's minimum and maximum depth sensor distance and d_{full} the distance above which a person is fully visible in our setup.

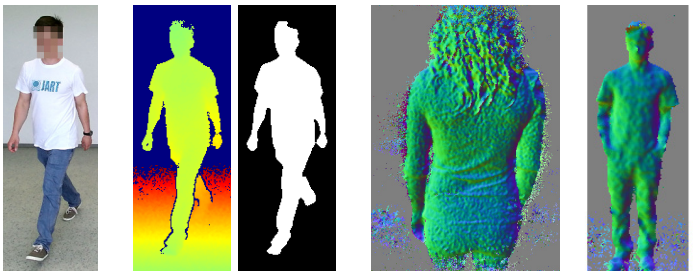


Figure 7.6: Post-processing of the data for baseline methods. *Left:* Initial color image and colorized depth image of a person from sequence 2. Middle: foreground segmentation mask. *Right:* Point cloud normals for a person close to the sensor (sequence 4, distance <1.0 m) and a subject in a static pose (seq. 1, distance approx. 2.5 m).

anti-clockwise direction. Finally, sequence 4 ($\sim$280 frames) simulates a close-up interaction with a robot, where the subject steps back, forth and sideways in front of the sensor as if he/she is physically interacting with the robot's touch screen or manipulator. The sequence is thought to be a relevant benchmark for human-robot interaction that contains many vertical and horizontal occlusions of people as well as cases of missing out-of-range depth data.

In addition to the Kinect v2 data, we recorded in parallel data with a first-generation ASUS Xtion Pro Live RGB-D sensor, to allow for a comparison in data qualities (though for our experiments in this chapter, we solely use the Kinect v2). The sensors were stacked on top of each other and recorded all sequences at the same time. We did not notice any cross-talk effects between the sensors which is likely due to the different measurement principles – structured light for Kinect v1, time-of-flight for Kinect v2. They were mounted at around 1.5 m height and tilted slightly downwards, approximately replicating the setup of a mobile robot like SPENCER (Chapter 9). All recording was done under Windows using the official SDK, as no Linux-based driver was available yet. In post-processing, we converted all data into a ROS-compatible format.

Post-processing and labeling

During post-processing, we first segment the person from scene background by projecting the point cloud into a 2D height map and finding its peak within the designated recording area. All points within 0.6 m radius of the peak are considered as belonging to the person, while filtering out points in the ground plane. From this, we also derive 2D foreground segmentation masks and point cloud surface normals that are required for the deep learning-based baselines in our experiments. Normals are computed using a k-NN method with $k = 25$. As a result, our dataset's format is largely compatible with the University of Washington's RGB-D object dataset by Lai et al. (2011).

Each person instance has been labeled with age and gender, based upon data provided by the subjects in a questionnaire. For later experiments that incorporate color features, we further annotated the persons with additional binary attributes *has long trousers*, *has long sleeves*, *has long hair*, *wears jacket*, *wears skirt/dress*. These attributes are instance-specific and do not vary across frames. Each subject has been independently annotated by three persons and in case of

conflicting annotations, we chose the majority vote. This happened *e. g.* when a person was wearing $^3/_4$ trousers (which are not clearly *long trousers*, but also not shorts) or with medium-length hair. The absolute frequencies of these attributes across the dataset can be seen in the first three columns of Table 7.6.

In total, the dataset contains around 131,800 RGB-D frames which results in approximately 300 GB of data. Subsampled point cloud and image-based versions of the dataset, containing the subjects that agreed for their data to be shared, are available online for academic research purposes, upon request (Linder, Wehner, et al., 2015b).

7.3.3 Experimental setup for gender recognition

We now investigate the ability of the tessellation learning approach to recognize the gender attribute using RGB-D data from the Kinect v2 sensor. We compare our approach, which in this section relies solely on geometric 3D features, with two state-of-the-art deep-learning methods for RGB-D object recognition, and an RGB-D histogram of oriented gradients (HOG) approach[6]. Concretely, the considered baselines are:

- *Histogram of oriented gradients (HOG)* and *histogram of oriented depths (HOD)*, computed on the greyscale (from RGB) and depth images, respectively. HOG is a widely used descriptor by Dalal and Triggs (2005) for person detection in image data. HOD was proposed for the same purpose on depth data by Spinello and Arras (2011). The two feature vectors are *concatenated* and then fed into a linear SVM. We evaluate two window sizes, 32×64 and 64×128 pixels, and use the HOG and SVM C++ implementations provided by OpenCV.
- *Convolutional-recursive neural networks (CRNN, or CNN-RNN)*: At the time of our experiments, they were a recent deep-learning method by Socher et al. (2012) for RGB-D object recognition. In contrast to modern multi-layer CNN classifiers, like the DarkNet53 backbone used in Chapter 5, their feature extraction stage is pretrained without backpropagation in an unsupervised fashion by sampling random image patches and then computing convolutional filters via k-means clustering. Only a *single* convolutional layer is used, in combination with average-pooling. *Recursive* – not to be confused with "recurrent" – hints at the fact that the same neural network is then applied recursively in a tree structure to obtain a hierarchical feature representation. Its weights are initialized randomly, but shared by all layers of the RNN. Multiple random RNNs are combined to compute the final feature vector for a softmax classifier.

  We use the Matlab code provided at the authors' website, which we had to modify to use distinct training/test splits on a per-person basis, and to reduce memory consumption by computing less features in parallel, in order to allow training on our HPC cluster (as this early deep learning-based method did not utilize GPU acceleration). As an alternative to softmax, we tried a linear SVM classification stage. Our results in this section have been

[6] CRNN and HMP experiments were conducted by my master student S. Wehner under my supervision.

obtained using the SVM; experiments showed no significant difference between the two classifiers for the task at hand.

- *Hierarchical matching pursuit (HMP)* by Bo et al. (2014) is another deep learning-based method, which represents a multi-layer sparse coding network. Like CRNN, this method has been designed specifically for RGB-D object recognition, where it outperforms *e. g.* fast point feature histograms (FPFH, Rusu et al., 2009) significantly in their experiments. Its feature extraction stage is trained in a fully unsupervised fashion, given that no large-scale labeled RGB-D datasets for supervised pretraining such as SceneNet RGB-D (McCormac et al., 2017) existed back then. During training, the method alternates between updating a codebook using a variant of singular value decomposition (SVD), and encoding image patches as a linear combination of K codebook words via a greedy algorithm called orthogonal matching pursuit. A feature hierarchy is then obtained through spatial pyramid pooling over sparse codes computed within spatial cells. A second layer is structured similarly and combines high-level features with low-level features from the first layer, which are then fed into a linear SVM to solve classification tasks.

  In addition to foreground masks needed by CRNN, HMP also requires point cloud normals as an input, which we computed during post-processing of our data (see Section 7.3.2). For HMP, we also use the Matlab implementation provided by the authors, which we modified slightly to process our training and test splits.

For training of these classifiers, we form the training set by concatenating all four sequences of all 118 persons and split it into equally-sized training and test splits on a per-person basis. By never including the same person instance in both training and test set, we try to ensure that the classifiers do not learn individual person appearances. For each method, we perform at least 10 runs of repeated *random sub-sampling validation* with a training/test set split ratio of 1:1 to ensure that there is always sufficient person variety in the training set. We did not choose k-fold cross-validation due to attribute class imbalances and the limited number of person instances.

As sequences 2–4 are recorded with 15 frames per second, which yields a large number of locally similar frames, we subsample the number of frames by a factor of 5 for HOG and our tessellation-boosting method, and a factor of 20 for CRNN and HMP, to keep training times and memory usage within manageable bounds[7].

We evaluate our tessellation learning approach for both $n_{\text{weak}} = 100$ and $n_{\text{weak}} = 500$ weak classifiers. We have re-implemented the original method by Spinello, Luber, et al. (2011) in C++ in two variants, a regular non-optimized version that runs on a single core and an optimized version that uses OpenMP to parallelize feature calculations on the CPU. The two variants will be considered when evaluating inference performances of the different approaches. Different from

[7] While this might not look fair with regard to the deep learning-based methods, we note that already at this subsampling factor we were severely limited by available system memory with their MATLAB-based implementations. Given that the omitted frames are still quite similar to adjacent frames, probably a higher number of unique person instances would be much more worth striving for.

Sequence	(1) Standing	(1)–(3) +Walking	(1)–(4) +Interact.	(1)–(4) $d>0.8$ m
HOG-SVM (32×64 px)	84.78%	70.55%	70.70%	70.55%
HOG-SVM (64×128 px)	86.27%	74.10%	74.36%	74.33%
CRNN (Socher et al., 2012)	86.64%	83.46%	83.73%	84.10%
HMP (Bo et al., 2014)	88.07%	85.28%	**86.10%**	85.42%
Ours	**91.07%**	**86.41%**	85.29%	**87.47%**

Table 7.2: Average gender classification accuracy obtained with the baselines from Section 7.3.3. For the top-down classifier, which operates solely on the depth modality, we use $n_{\text{weak}} = 500$. The other methods exploit both visual and depth information. The last column shows results if we ignore all test samples closer than 0.8 m sensor distance.

Spinello, Luber, et al., we use the more modern AdaBoost C++ implementation from OpenCV. Our code has been made publicly available as a ROS package (Linder, 2015).

7.3.4 Results for gender recognition

Table 7.2 shows average classification accuracies (over multiple validation runs) for the different sequences in the dataset: for the eight standing poses of sequence 1, the standing poses and the two walking patterns (seq. 1–3), the entire dataset (seq. 1–4) and the entire dataset except test samples with persons closer than 0.8 m to the sensor, below which we observed noise and missing data artifacts produced by the Kinect v2.

It can be seen that all methods achieve more than 80% accuracy under the rather controlled conditions of sequence 1. As soon as people start walking, and the shapes of people become more diverse and articulated, motion blur starts to play a role and the HOG-based methods drop to around 70%. CRNN and HMP both perform fairly solid across all sequences, with HMP in all cases being about two percentage points better than CRNN.

Despite not using any color information from the RGB image, our proposed tessellation learning method generally outperforms all other methods on the task of full-body gender recognition. In Figure 7.1, we saw qualitative classification results of our method (using only geometric features) along with their input point clouds. Only for close-range subjects in the interaction sequence 4, HMP is slightly more accurate. Examples for which our methods fails under close-range conditions are shown in Figure 7.7.

Next, we analyze these results with respect to relative distance and body orientation.

Impact of distance to sensor

Figure 7.8 shows the classification accuracy as a function of person distance to the sensor. HOG, CRNN and HMP deliver relatively constant results in terms of accuracy across the available RGB-D sensor range. At very close-range (<0.6 m), our tessellation approach breaks down as the point clouds provided by the sensor extend beyond the near clipping plane, such that the shape

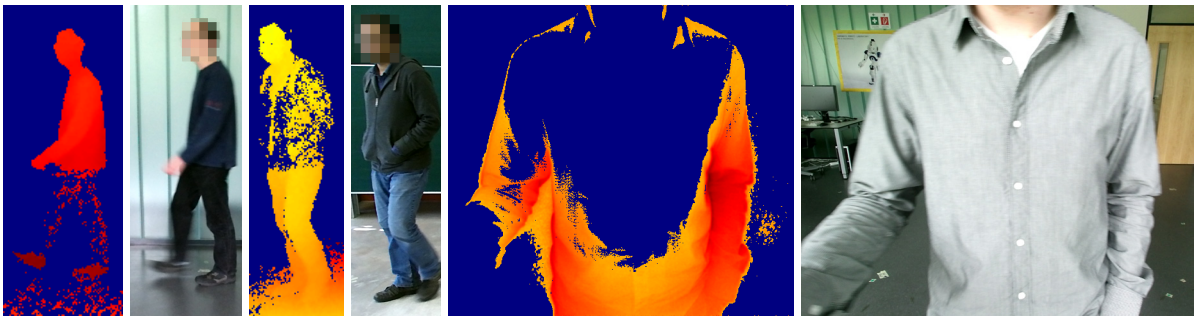
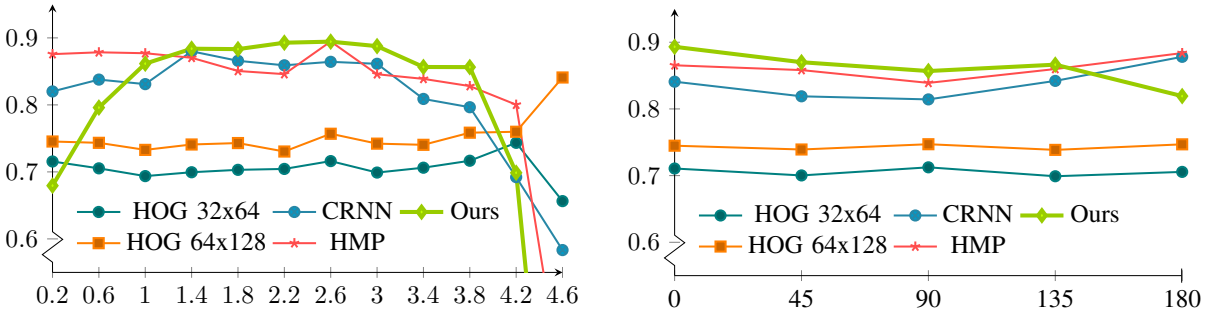


Figure 7.7: Example RGB-D images where our tessellation learning approach fails to classify the person's gender correctly. Problems mainly occur at the sensor's near and far clipping planes, and with certain types of black clothing. The RGB image is shown just for illustration purposes, and not used here.



(a) Distance to sensor

(b) Person orientation

Figure 7.8: Average accuracy of *gender* classifiers as a function of (a) person-to-sensor distance in meters and (b) person orientation in degrees, obtained on the full dataset containing sequences (1) to (4). The sharp decline at around 4.5 m distance is due to the maximum range limit of the developer preview of the Kinect v2 sensor, leading to truncated point clouds. 0° orientation means that the person is looking towards the RGB-D sensor.

of the person becomes very hard to distinguish (Figure 7.7). Here, the other methods could be in advantage because they can fall back onto the RGB image, which – at this close distance – is potentially of very high resolution. At >4.2 m distance, parts of the point cloud start to vanish at the far clipping plane[8], which also means there are only few person samples at this far range. The drop in performance of the tessellation learning approach is thus due to limitations of the depth sensor rather than the method itself.

[8]This is a limitation of the development version of the Kinect v2 sensor that we used in our experiments. The final version has a larger range of around 9 m.

	n_{weak}	
Tessellation type	100	500
Regular tessellation 0.1 m	75.93%	76.49%
Regular tessellation 0.2 m	83.19%	84.16%
Regular tessellation 0.3 m	81.54%	82.96%
Learned tessellation	**89.75%**	**91.07%**

Table 7.3: Comparison of *gender* classification accuracy of the tessellation learning approach using the learned tessellations with mixed-size voxels (selected by AdaBoost from all voxels across all generated tessellations), against the same method with only a set of fixed-size voxels with side length 0.1 m, 0.2 m and 0.3 m.

Impact of person orientation

To analyze the impact of relative person orientation, we derive the body yaw angle in the following way: For sequence 1 of the dataset, we can directly determine person's orientations in 45° steps from the frame number. For the other sequences, we track the center of the person blob using a Kalman filter-based nearest-neighbor tracker with constant velocity motion model, and then determine the orientation from the track velocity estimates. We filter out frames where persons were barely moving, and note that subjects were instructed not to walk backwards. As can be seen in Figure 7.8, gender recognition is rather stable across relative view angles for all methods. The deep-learning methods excel on rear views (180°), whereas for our tessellation learning method, rear views appear to be slightly more difficult than frontal or side views.

Resulting learned tessellation for the gender attribute

With the encouraging results of the previous subsection, we seek to better understand the learning of tessellations. First, we examine if it actually makes sense to *learn* a volume subdivision over the object as a combination of voxels from different tessellations – such as those shown in Figure 7.2 –, versus a predefined regular tessellation. We compare the learning approach with a grid of cube-shaped voxels of proportions $\{1, 1, 1\}$ and fixed side length. Such a regular tessellation is shown in the center of Figure 7.2. We use the same geometric point cloud features, and train an AdaBoost classifier with the same hyperparameters on sequence 1 of our dataset. Table 7.3 shows the classification accuracy over several grid sizes. It can be seen that the *learned* tessellation performs clearly better than the best regular tessellation using cube-shaped voxels of size 0.2 m.

Figure 7.9 (left and middle) shows the resulting tessellation learned by our method using 500 weak classifiers on the full dataset. The most commonly used features in descending order of frequency (in brackets) are $f_3\,(94)$, $f_5\,(84)$, $f_2\,(64)$, $f_6\,(64)$, $f_9\,(61)$, $f_4\,(50)$, $f_8\,(50)$, $f_7\,(20)$, $f_1\,(13)$. From the wireframe representation, it can be seen that the highest concentration of voxels is located at above waist height and around the upper body, indicating that these regions contain more relevant information than for instance the legs. A similar observation can be made when allowing only fixed-size cube-shaped voxels of size 0.2 m with $n_{\text{weak}} = 100$, leading to the tessellation shown in Figure 7.9 (right).

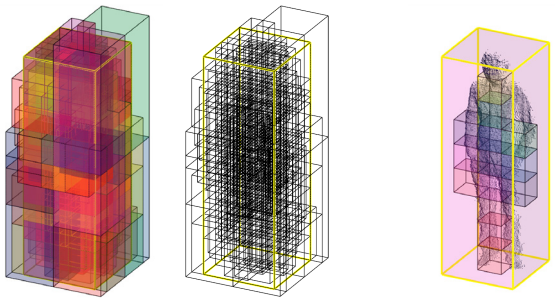


Figure 7.9: *Left and middle:* Learned best tessellations for the *gender* attribute using 500 weak classifiers, trained on the full dataset. *Right:* Learned regular tessellation using only cube-shaped voxels of side length 0.2 m, trained on sequence (1).

	Training full seq. (1)	**Testing** single frame
HOG (32×64 px)	$<$ 1 min	90 ms
HOG (64×128 px)	4 min	120 ms
CRNN (Matlab) (Socher et al., 2012)	420 min	7500 ms
HMP (Matlab) (Bo et al., 2014)	35 min	3000 ms
Ours, 1 thread	–	24.0 ms
Ours, 4 threads	17 min	6.7 ms

Table 7.4: Average training and testing durations for the different methods. The tessellation learning approach is clearly the fastest classification method even if we assume a conservative speed up factor of 100 for a C++ implementation of CRNN and HMP.

Evaluation of training and inference speeds

Runtime performance is key in real-world robotics applications, particularly because the robot may be surrounded by several persons that all need to be classified within a short time frame. We therefore analyze the time it takes for all methods to predict the gender class for a given person. We assume that the RGB-D data have already been pre-processed, *i. e.* the person has been detected and cropped out from the RGB-D image or point cloud; that point cloud normals have been calculated for the HMP method; and that the sub-cloud containing the person has been transformed into the origin of a local coordinate frame. Besides inference times, which we average over a large number of frames, we also measure the time it takes to train the classifiers on seq. 1 of the dataset that contains the eight standing poses.

We anticipate that the comparison is not fully fair at this point because CRNN and HMP are implemented in Matlab without GPU acceleration, whereas all other methods are implemented in C++. We believe, however, that the comparison is still able to reveal a trend that we may see in the light of Matlab code running 10$\times$ to 100$\times$ slower than C++ code. The computer used is a regular desktop PC with Intel Core i7-2600 CPU.

As expected, the processing times for training and inference in Table 7.4 show that the deep-learning methods are computationally most expensive. The HMP approach, although it uses much higher-dimensional feature vectors at the classification stage than CRNN (188,300 *vs.*

32,768) still performs about 2.5× faster during testing. Our tessellation learning approach is very fast. The non-optimized implementation achieves more than 40 Hz for classification, the parallelized variant that we described in this section even 150 Hz on four CPU cores. This is still clearly the fastest method even if we assume a conservative speed up factor of 100 for a C++ implementation of CRNN and HMP. Our method does not require any GPU acceleration, which can be an advantage when deploying the method on resource-constrained mobile robots.

7.4 Incorporating color cues and further attributes

In this section, we extend the tessellation boosting approach that we described in the previous section, and which so far only relied on geometric 3D features, with new geometric extent and color features and two additional pre-processing steps. We evaluate the extended method on a larger set of six human attributes, including the previously considered gender attribute.

7.4.1 Proposed extensions to the method

Geometric extent features

In the previously described set of geometric point cloud features from Table 7.1, *standard deviation w.r.t. centroid* and *average deviation from median* measure the compactness of points $\mathcal{P}$ within a voxel, expressed as two single, scalar values. However, these provide statistics on point-to-point distances and do not capture well the overall, geometric extents of $\mathcal{P}$ per axis, useful *e. g.* to distinguish vertically and horizontally elongated shapes that are approximately axis-aligned. Therefore, we propose to extend the feature set by the *depth*, *width* and *height* of $\mathcal{P}$ as defined in Table 7.5.

Color features

All features considered so far are geometric in nature and do not encode color information – clearly a very informative object property. We now propose a very simple way of incorporating RGB information based upon low-dimensional statistical measures computed over the points $\mathcal{P}$ in a voxel. This is motivated by a current limitation of our proposed method, which relies on decision stumps as weak learners that work best on low-dimensional feature representations. In the conclusion, we discuss further ideas for integrating more complex representations.

To accommodate for this, we propose to include the average color of the points $\mathcal{P}$ within a voxel $\mathcal{V}_j^i$ by computing the component-wise mean of the red, green and blue channels. We also include the component-wise standard deviations. Additionally, since RGB color values are not invariant with regard to illumination, we do the same in HSV (hue, saturation, value) color space. The expectation here is that the V feature should be chosen less often by the AdaBoost classifier due to its dependence on lighting. Finally, we also conduct experiments in $\mathrm{Y'C_BC_R}$ color space, where $\mathrm{C_B}$ and $\mathrm{C_R}$ represent the blue-difference and red-difference chroma components, which are illumination-independent, and $\mathrm{Y'}$ the luma. The advantage of this color space, used *e. g.* in JPEG compression and digital video, is that skin colors are very localized in the chroma

#	Description	Expression
10	Depth	Geometric extent of $\mathcal{P}$ in x direction. $f_{10} = \max_{\mathcal{P}}(x_i) - \min_{\mathcal{P}}(x_i)$
11	Width	Geometric extent of $\mathcal{P}$ in y direction. $f_{11} = \max_{\mathcal{P}}(y_i) - \min_{\mathcal{P}}(y_i)$
12	Height	Geometric extent of $\mathcal{P}$ in z direction. $f_{12} = \max_{\mathcal{P}}(z_i) - \min_{\mathcal{P}}(z_i)$
13 – 15	RGB component-wise mean	Mean of the red, green, blue components of each point in $\mathcal{P}$. $(f_{13}, f_{14}, f_{15}) = (\bar{r}, \bar{g}, \bar{b})$
16 – 18	RGB component-wise standard deviation	Standard deviation of the red, green, blue components of each point in $\mathcal{P}$. $(f_{16}, f_{17}, f_{18}) = (\sigma(r), \sigma(g), \sigma(b))$
19 – 21	HSV component-wise mean	Mean of the hue, saturation, value components of each point in $\mathcal{P}$. $(f_{19}, f_{20}, f_{21}) = (\bar{h}, \bar{s}, \bar{v})$
22 – 24	HSV component-wise standard deviation	Standard deviation of the hue, saturation, value components of each point in $\mathcal{P}$. $(f_{22}, f_{23}, f_{24}) = (\sigma(h), \sigma(s), \sigma(v))$
25 – 27	$\mathrm{Y'C_BC_R}$ component-wise mean	Mean of the luma, blue-difference, red-difference components of each point in $\mathcal{P}$. $(f_{25}, f_{26}, f_{27}) = (\bar{Y'}, \bar{C}_B, \bar{C}_R)$
28 – 30	$\mathrm{Y'C_BC_R}$ component-wise standard deviation	Standard deviation of the luma, blue-difference chroma, red-difference chroma components of each point in $\mathcal{P}$. $(f_{28}, f_{29}, f_{30}) = (\sigma(Y'), \sigma(C_B), \sigma(C_R))$

Table 7.5: New geometric extent and color features

channels and that their values do not vary much between subjects of different ethnicity, which can be useful for robustly localizing skin colors (*i. e.* uncovered body parts) in the point cloud. Skin-specific features have also been used successfully by Bourdev et al. (2011) for detecting long sleeves or short trousers. For conversion into HSV and $\mathrm{Y'C_BC_R}$ space, we use formulas described by Szeliski (2010). The resulting list of new features is shown in Table 7.5. They are used in combination with the existing features from Table 7.1.

Scaling of input clouds

So far, we assumed a fixed-size bounding box $\mathcal{B}$ in which voxels are generated. The size of this bounding volume can be learned from training data or set to some upper bound. This, however, is problematic when there is a significant variation in person size: if the training set includes very large and very small subjects (*e. g.* children), the classifier may fail to locate specific scales on the human body that are informative for a particular attribute (*e. g.* hips and waist for gender, or the head for *long hair*), or it might at least spend a significant number of weak classifiers to accommodate for the differences in size. Learning multiple attribute classifiers specific to different person sizes would require even larger amounts of training data to cover different person sizes across all the attributes. As a more effective alternative to deal with this issue, we scale the input person cloud in vertical z direction to a fixed size. In our case, we stretch the point cloud to a height of $h = 1.8$m while leaving the x and y coordinates unaltered.

Pruning of non-informative voxels

We further propose to discard non-informative voxels in a filtering step, in order to reduce memory consumption at training time and accelerate the training process: In the OpenCV implementation of AdaBoost that we are using, the feature matrix F must include the stacked features of all person clouds over all voxels and tessellations[9]. The resulting size $|\mathcal{V}_j^i| \cdot n_{\text{features}} \cdot n_{\text{samples}}$ of F scales linearly with the number of voxels $|\mathcal{V}_j^i|$. Thus, with just the nine geometric features from Table 7.1, already 12 GB RAM are consumed to train on the full dataset. The addition of the proposed geometric extent and color features would have prevented us from training the method (without expensive swapping to hard-drive) within reasonable time frame.

To deal with this, before training, we filter out voxels that are mostly empty across the entire training set. For each voxel $\mathcal{V}_j^i$ we count, over all training samples, how many times it contains less than 4 points[10]. If this is the case for at least 30% of all training samples, we remove the voxel $\mathcal{V}_j^i$ from the corresponding tessellation $\mathcal{T}_j$. Voxels that are discarded in this way encode a non-informative location or scale for the characterization of the object. Examples include small voxels around the head, which is smaller in diameter than other body regions, and usually centered inside the bounding volume $\mathcal{B}$ if we restrict ourselves to standing and walking poses.

7.4.2 Experimental setup for the extended method

In the following experiments, we compare the extended method with geometric extent features, color features and point cloud scaling from this section against the baseline approach from the previous section, which relies only on the original features from Table 7.1. This time, we only use 100 instead of 500 weak classifiers to limit training time as we now evaluate upon six different human attributes. We further compare classification performance quantitatively against a linear SVM classifier baseline trained on HOG features in RGB-D (as in the previous experiments), using an input image size of 64×128 px.

We focus on human attributes that are reasonably well presented in our dataset, namely *gender*, *has long trousers*, *has long sleeves*, *has long hair*, *wears jacket* and *wears skirt/dress*. For other attributes like *wears hat* and *has backpack*, there are only very few instances that are insufficient for reliable training and testing. However, we believe that in principle our algorithm is applicable to a broader range of appearance-based human attributes if sufficient data is available.

We again perform 10 rounds of repeated random sub-sampling validation. To keep training times within reasonable limits, we subsample the frames of the larger sequences 2–4 of the dataset by a factor of 5. For *class balancing*, we conduct undersampling on a frame-by-frame basis by discarding excess samples of the majority class, such that each train and test set eventually contains 50% positive and 50% negative sample frames.

[9] More recent boosting libraries like XGBoost (Chen et al., 2016) that became popular after our initial publication also support use of external memory, distributed training, and GPU acceleration.

[10] Four is the minimum amount of points required to be able to robustly compute all geometric features.

	n_{pos}	n_{neg}	Baseline	+Extents	+Scaling	+RGB /	HSV /	$\text{Y}'\text{C}_\text{B}\text{C}_\text{R}$	HOG
gender (m/w)	68	69	89.4%	89.0%	**91.7%**	90.2%	90.1%	91.3%	85.2%
long trousers	111	26	73.9%	72.6%	73.9%	80.8%	85.0%	**85.7%**	70.6%
long sleeves	60	77	63.9%	65.2%	63.6%	65.5%	71.6%	**73.2%**	69.8%
long hair	70	67	85.1%	84.0%	**87.7%**	86.8%	87.1%	86.2%	83.4%
jacket	20	117	62.8%	**63.8%**	61.4%	61.8%	57.4%	59.9%	56.5%
skirt/dress	16	121	74.1%	76.1%	76.5%	74.1%	74.0%	70.0%	**82.5%**

Table 7.6: Ablation studies on the additional geometric extent features, point cloud scaling, and color features in three different color spaces when training and testing on seq. 1 *(static poses)*. The second and third column show the number of positive and negative class instances per attribute. The fourth column shows the accuracy of our baseline method (from Section 7.3) using only geometric features from Table 7.1. Our approach, with $n_{\text{weak}} = 100$, outperforms the HOG-SVM on all but one attribute (which has the lowest number of positive samples), while being around $15\times$ faster on a single core.

7.4.3 Results for the extended method on all human attributes

Ablation studies on sequence 1 (static poses)

In Table 7.6, we quantitatively compare the different extensions discussed in this section against the baseline methods on sequence 1 of the dataset – containing only standing poses – over different human attributes.

We can see that all extensions are able to improve performance on certain attributes, but not all extensions work equally well in every case. For every human attribute, there is always at least one variant of our method that clearly outperforms the HOG-SVM, except for the *skirt/dress* attribute. This is the attribute with the smallest number of positive samples (11.7%), which (due to class balancing of training samples) will lead to an overall low amount of training samples. Also, the attribute was maybe poorly chosen, as it mixes two different concepts ("wears skirt" affecting only the lower body, while "wears dress" affects the full body). It is interesting that for this attribute, the inclusion of $\text{Y}'\text{C}_\text{B}\text{C}_\text{R}$, but not RGB, color features leads to significantly weaker performance, while for long trousers and long sleeves, these features clearly improve results (as we had hoped for). Again, this could be due to the low number of positive examples, leading to overfitting on certain color components in $\text{Y}'\text{C}_\text{B}\text{C}_\text{R}$ space. As we often achieve an accuracy of 90 to 100% on the corresponding training set, we are confident that this is a sign of overfitting to the training data. A significantly larger training set (which, in RGB-D, is very expensive to acquire) or various kinds of training-time augmentations could help to alleviate these effects.

As the attributes *long hair* and *gender* are highly correlated in the groundtruth ($\rho = -0.90$), we conduct an additional experiment to analyze if the *long hair* classifier still performs well when only the upper body including the head is visible, while excluding the region below the waist – whose shape can be an important indicator for gender. To do so, we cut all point clouds below a fixed height of 1.2 m to focus on the upper body. In this case, accuracy only drops from 87.7% to 86.1%. If we instead only consider the uppermost 0.35 m of each point cloud, which contain the

Figure 7.10: Extracted RGB-D point clouds (faces anonymized for publication) with corresponding single-frame classification results for three different human attributes. The text at the top of each image is the attribute in question and the number below each image the confidence of the boosted classifier, where the sign indicates an association with the negative or positive class.

head, we obtain an average accuracy of 83.5%, which is still a good result, but at the same time visualizes how individual attribute classifiers can benefit from additional context information.

Quantitative and qualitative results on the full dataset

In Table 7.7, we show results when training and testing a classifier on the full dataset, including walking and close-up interaction sequences. To obtain these results, we employed a fixed combination of original geometric features, geometric extent features, $\mathrm{Y'C_BC_R}$ color features and point cloud scaling. In this setting, our approach outperforms the baselines in all cases except for the *has jacket* attribute, which does not seem to benefit from the color features. We also want to note that this is a difficult binary attribute to detect, as sometimes the subject is wearing a cardigan, which in its appearance often resembles a jacket but has not been annotated as such (see Figure 7.12). For *long trousers* and *long sleeves*, we improve the accuracy by around 12% compared to the original method from Section 7.3. As expected, the overall performance goes down by 3–5% under less controlled conditions when persons are walking instead of just standing (seq. 2–3), and in very close proximity to the sensor (seq. 4). Figure 7.10 shows some correctly classified qualitative results.

Insights on feature selection

At training time, our method computes for *all* voxels of *all* tessellations *all* features that the respective column in Table 7.6 refers to. For instance, for the second-last column, the features (1)–(9), (10)–(12), (25)–(30). This leads to a very high-dimensional feature vector. During testing, however, only the *most informative* features in the most informative voxels of all tessellations are calculated; that selection of features is implicitly given by the best n_{weak} weak classifiers chosen by AdaBoost.

To find out how often the proposed new features are selected by our boosting approach, we count the absolute usage frequency of each feature type across all 100 weak classifiers. Figure 7.11 shows the 10 most frequently used features for the *long trousers* attribute. It can be seen that the

Gender	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG-SVM	78.0%	76.9%	77.0%	77.4%
Our baseline (Section 7.3)	89.8%	83.7%	82.6%	85.1%
With proposed extensions	**90.4**%	**87.0**%	**86.3**%	**87.7**%
Long trousers	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG-SVM	65.0%	60.0%	59.4%	60.7%
Our baseline (Section 7.3)	69.4%	66.0%	64.1%	67.0%
With proposed extensions	**83.6**%	**78.0**%	**76.2**%	**79.9**%
Long sleeves	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG-SVM	63.2%	60.8%	60.7%	61.9%
Our baseline (Section 7.3)	62.3%	61.8%	61.0%	61.8%
With proposed extensions	**76.9**%	**73.8**%	**72.8**%	**74.3**%
Long hair	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG-SVM	74.3%	72.6%	72.7%	73.3%
Our baseline (Section 7.3)	83.7%	77.9%	77.2%	79.3%
With proposed extensions	**87.2**%	**83.3**%	**82.9**%	**83.9**%
Jacket	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG-SVM	56.7%	56.5%	56.8%	57.0%
Our baseline (Section 7.3)	**62.8**%	**62.3**%	**61.5**%	**62.1**%
With proposed extensions	60.8%	59.3%	59.0%	59.1%

Table 7.7: Classification accuracies for static poses (seq. 1) and in combination with walking sequences (seq. 2+3) and close-up interaction (seq. 4). For these results, we trained on the full dataset and combined the original feature set with geometric extent and YCbCr color features plus point cloud scaling. The last column excludes all frames where the person is closer than 0.8 m to the sensor, where only a very limited part of the body is visible and clipping artefacts sometimes appear.

means of the illumination-independent chroma color channels are being used fairly often, but geometric features such as density, standard deviation w.r.t. centroid and planarity still play a large role. Also taking other human attributes into account, we note that standard deviations in the color channels are being used less often than the corresponding means, and that the geometric extent features (width, depth, height) are used less than 5 times on average across all human attributes. Along with the results from Table 7.6, this indicates that the geometric extent features are not as useful as we had hoped for. One reason could be that they assume the point cloud to be aligned with the x, y, z axes – which might often not be the case due to variation in human pose.

On the right side of Figure 7.11, we see the learned, best tessellations for the *long trousers* attribute with 2, 5 and 10 weak classifiers. Here, we see that most selected voxels end up in the lower part of the bounding volume, which is exactly what we would expect to happen. With

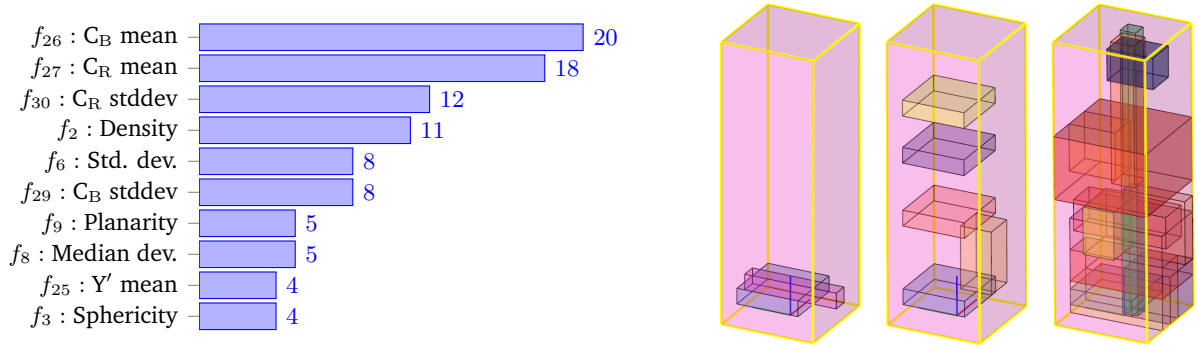


Figure 7.11: *Left:* Best 10 geometric and $Y'C_BC_R$ color features selected for the *long trousers* attribute on the full dataset. Absolute frequencies are with regard to the total number of 100 weak classifiers used during training. *Right:* Learned best tessellations for the same attribute using 2, 5 and 10 weak classifiers. As expected, most voxels are placed in the lower half of $\mathcal{B}$.

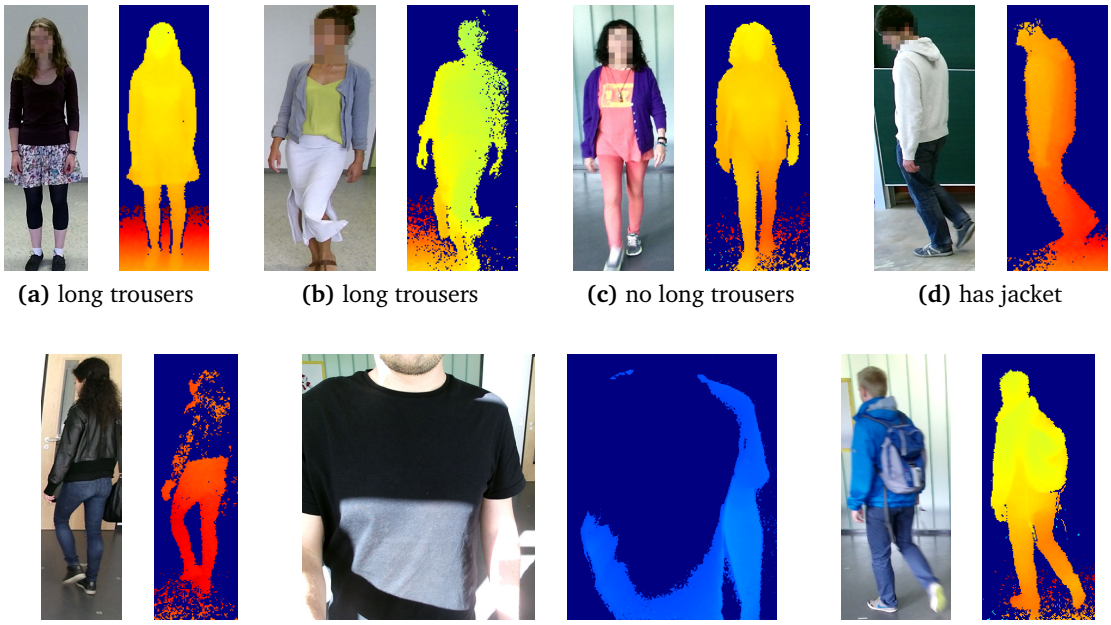
increasing number of weak learners, voxels are also placed in the upper-body region. We believe that these voxels provide contextual cues, for instance knowledge of the person's overall height could be helpful to determine leg lengths.

Failure cases

In Figure 7.12, we show example RGB and depth images depicting cases where our approach typically fails. The first row shows the *long trousers* attribute; here, we notice that our classifier sometimes fails to detect very thin trousers/tights, that do not significantly affect the 3D shape (first image), or when the clothing is highly deformable and its shape varies a lot during motion (middle image). Also, nearly skin-colored trousers (last image) are not detected as such, which we believe is due to this trousers color not otherwise appearing in the training set. For the *long hair* attribute (second row), the classifier may fail when depth data is totally missing due to clothing surface properties that irritate the RGB-D sensor (first image), when the person is too close to the near clipping plane such that a significant part of the point cloud is missing (middle image), or when accessories like hoods or backpacks generate unusual shapes in the point cloud which are significantly under-represented in our training set (last image).

Computational efficiency

Impact of voxel pruning on memory usage during training: Pruning mostly empty voxels from the set of tessellations before training as described in Section 7.4.1 reduces memory consumption significantly from 12 GB to around 2.4 GB (-80%) on the full dataset using just the geometric features. Due to the otherwise extremely high-dimensional feature vectors, this pre-processing step becomes even more important when additionally including color features. While testing on seq. 1, no negative impact on classification accuracy was observed when including this additional filtering step. Although the pre-filtering incurs a processing time overhead, this is remedied by the fact that the feature vectors fed into the learning algorithm become much shorter.



(a) long trousers **(b)** long trousers **(c)** no long trousers **(d)** has jacket

(e) no long hair **(f)** long hair **(g)** long hair

Figure 7.12: Examples of typical failure cases for the *long trousers*, *jacket* and *long hair* attributes. The caption below RGB and depth images shows the *predicted*, incorrect class label.

Training time: Using 4 parallel threads for feature computations, training on a single randomized 50% subset of the full dataset (while using every 5^{th} frame of seq. 2–4) takes between 0.5 and 2 hours for a single human attribute, depending on the number of features being used.

Inference time: Efficient runtime performance is important for resource-constrained mobile service robots. Since a single scene might contain multiple persons, each with multiple attributes to detect, the processing time per input person cloud should be low. With $n_{\text{weak}} = 100$ and using 4 threads, our classifier runs in real-time at around 300 Hz (for a single attribute) on pre-extracted RGB-D person point clouds without requiring GPU acceleration. Using just a single thread, we still achieve about 125 Hz. If we instead use $n_{\text{weak}} = 500$ as in Section 7.3, accuracies improve by around 1-3% while still allowing processing at 120 Hz (4 cores) or 40 Hz (1 core). The new features added in this section do not have a significant impact on performance, as we selectively only compute the best features (found by AdaBoost) in each selected voxel, and all features are very simple to compute. Also, adding additional features at training time does not increase the feature vector size during testing as long as the number of weak classifiers remains constant.

7.4.4 Further experiments in the wild

In this section, we present qualitative results that were obtained in-the-wild when trying out the *gender* classifier in the airport environment with the SPENCER robot.

Integration with human detection and tracking

To evaluate human attribute recognition online on a robot system, we first need to extract person-specific sub-clouds from the RGB-D point cloud of the entire scene. For this, we rely on our human detection and tracking framework (Chapter 8).

For human detection, we use a variant of the PCL detector by Munaro and Menegatti (2014), which we adapted to work with the Kinect v2 sensor and then integrated it with our tracking framework. It extracts 3D candidate bounding boxes from the point cloud using a head-based subclustering, and then confirms them using a HOG-SVM. To extract person clouds after detection, we initially experimented with two variants: In one variant, we directly collect all points within the detected 3D bounding box. In the second variant, we instead use the human centroids from our tracking system, and then extract all points within a fixed-radius cylinder around the centroid. In both cases, we remove the ground plane and center the points along their x, y median as described at the beginning of Section 7.3.1.

We obtained visually more convincing results, which were closer to our dataset that has been recorded under controlled conditions, using the first variant, because of a more accurate 3D localization when no tracking, smoothing and prediction is involved. A good compromise (that we did not attempt) could be to proceed with the first variant, but still try to associate detections with tracks in order to filter out spurious false alarms.

Temporal filtering of attribute classification results

We also experimented with temporal filtering of classification results. For this, we first map the score output of the AdaBoost classifier to a positive-class probability using Platt scaling (Platt, 1999). We then associate the human attributes with tracks (through associated detection IDs), and retain a history of the N last classification results per attribute for each track. Finally, we compute the mean over the last N probabilities to obtain a simple moving average.

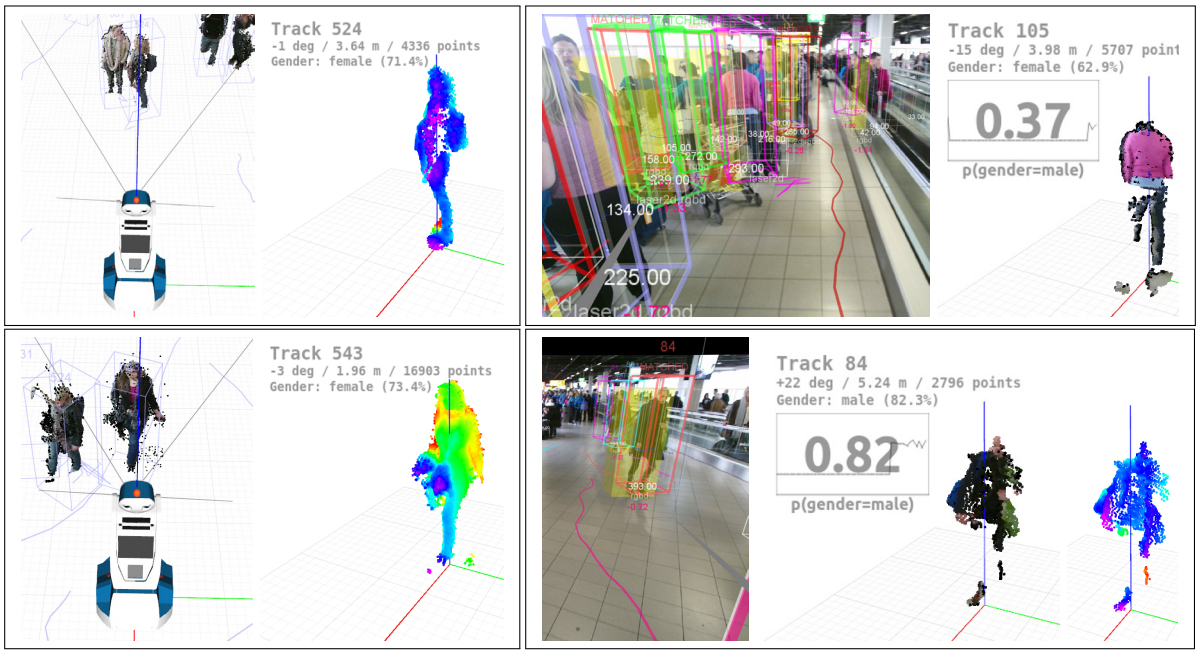
Qualitative results on the airport scenario

We now show some qualitative results obtained with the extended method, using the same combination of features as in Table 7.7. First, Figure 7.13 shows qualitative results on four different scenes for single persons. In the two examples on the left side, classification results are per-frame and without smoothing. On the right-hand side, temporal filtering over at most 300 frames has been applied. It is noteworthy how the number of points per person cloud varies between less than 3,000 and almost 17,000, depending on distance and material properties.

In Figure 7.14, we show recognized gender attributes for entire scenes, after associating per-frame classification results with human tracks and filtering them over at most 10 seconds[11].

In our experiments, we made the following qualitative observations:

[11] When the robot is in motion and people are walking into opposite direction at fast pace, or are initially occluded, they might actually be visible for only 2–3 seconds in the RGB-D view frustum.

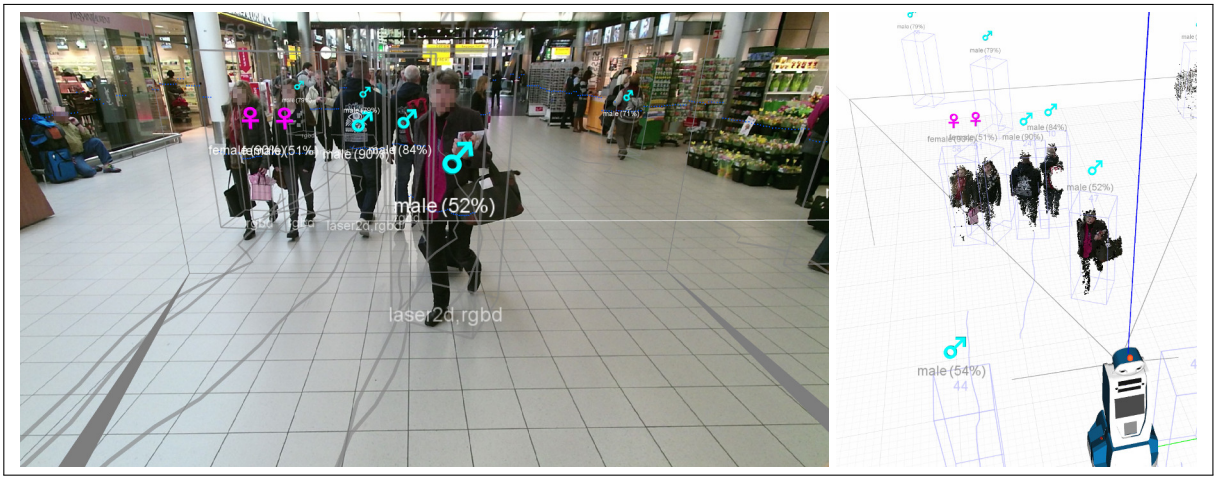


(a) Per-frame classification results **(b)** Different scenes with temporal filtering

Figure 7.13: *Gender* classification results for four different scenes. *Left:* Correct single-frame results for the person in the center, after performing Platt scaling to obtain probability estimates from the output of the boosted classifier. *Right:* Probability for male gender after temporal filtering of different scenes over 10 seconds. Probably due to large holes in the point cloud, the woman in the bottom right is consistently misclassified despite smoothing.

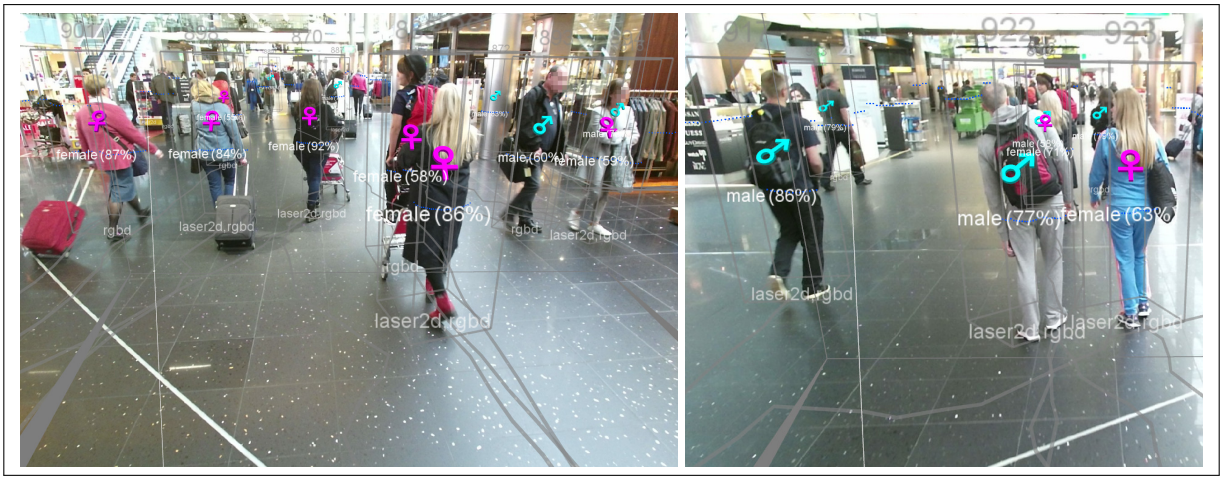
- Overall "in-the-wild" performance is weaker than on the test set which we recorded in controlled environments. The positive-class probabilities obtained after Platt scaling are often relatively close to the decision boundary of 50%.
- When misclassifications occur, they are mostly systematic, rather than spurious (*i. e.* only in isolated frames). For instance, an approaching person is correctly classified as "male" between 8–4 m distance, but then incorrectly classified as "female" between 4–0 m distance, or vice versa. Therefore, temporal filtering does not have the desired positive impact, and we believe that also more advanced techniques would not lead to further improvement[12].
- A major contributing factor is the fact that our training set only contains persons up to a distance of 4.5 m (due to limitations of the prototype sensor used for recording). However, even at such smaller distances, we observe more misclassifications than on our controlled-condition test set from Section 7.3.2.

[12] Breuers et al. (2018) make similar findings for temporal filtering of head-pose and skeleton estimates with *"rather minor quality gain"*. They remark that *"for perfect analysis modules, a filter cannot improve the result, which also holds for bad performing observations with systematic noise"*.



(a) Woman in front misclassified (no handbags and trollies included in training set!)

(b) Consistent misclassification of girl in pink trousers (children underrepresented in training set!)



(c) Good gender classification results, but false positive human detections in left image

Figure 7.14: Qualitative results for the *gender* attribute on in-the-wild data from the SPENCER robot inside the airport, temporally smoothed over 300 frames.

- Persons carrying accessories, such as shoulder bags, hand bags, backpacks and trollies, appear to have a higher likelihood of being misclassified. Except for few backpacks, such accessories are underrepresented in our training data, though they can have a large impact on the 3D shape of the person (*e. g.* Figure 7.13, bottom left).
- In Figure 7.14 (a), person clouds in the background contain almost no points, but the classifier still outputs high probabilities of 79% for the male class. Similarly in Figure 7.13 (b), the classifier should be able to notice that half of the point cloud is missing due to the dark trousers absorbing the sensor's infrared light emissions.

We will further discuss these results in the following section.

7.5 Conclusions

In this chapter, we applied and extended a tessellation learning method, originally developed by Spinello, Luber, et al. (2011) for human detection in 3D lidar, to the human attribute recognition task in RGB-D point clouds. The approach characterizes the point cloud by a set of features computed on measurements within axis-aligned voxels of the 3D object and uses AdaBoost to create a strong classifier with the best features and voxels. What distinguishes this method is that the boosted classifier not only selects the best features and thresholds, but also the best combination of voxels on which these features have found to be informative. Thus, the classifier jointly learns the best scales and locations of features on the 3D object for the classification task at hand. This allows to robustly and stably describe complex articulated shapes, as shown for the example of gender recognition, where the learned tessellation outperformed a fixed tessellation by a large margin of 6-10 percentage points.

For the experiments, we presented a large, annotated RGB-D dataset for learning and testing of human attributes with full-body views of 118 persons, captured with the Kinect v2 sensor. Using only geometric 3D features, we achieved an accuracy of up to 91% for standing people and 87% including walking people for recognition of the *gender* attribute. We outperform an RGB-D HOG baseline by a wide margin and two deep-learning methods for RGB-D object recognition by a small margin.

We further extended the method with additional processing steps and color features, which turn out to significantly improve classification accuracy on certain attributes such as *has long trousers* or *has long sleeves*. The extensions also lead to shorter training times and significantly lower memory consumption during training. Finally, we conducted experiments in-the-wild on challenging data from a robot operating inside an airport terminal. For this, we integrated the method with our human detection and tracking framework from Chapter 8, and presented a simple temporal filtering approach.

In conclusion, our extensive experiments show that the tessellation-boosting method can be successfully applied to the problem of human attribute recognition in RGB-D for people in standing or walking poses. Performance is highest on the *gender* attribute, which is rather global than fine-scale. Carried accessories such as handbags or backpacks can negatively impact performance, but are not sufficiently represented in our training data, so further study is needed

here to investigate if this is a limitation of the method. Additional temporal filtering has no major positive impact, because recognition failures are rather systematic in nature, than random. The method is very efficient, achieving classification rates per attribute of up to 300 Hz without GPU acceleration, which is important for resource-constrained mobile robots.

Limitations and weaknesses of method and dataset

Based upon our experiments and more recent insights from the literature, we believe that the proposed method has the following weaknesses that could be improved upon:

- From our ablation studies, it appears that not all proposed extensions work well for all human attributes. However, we did not try to combine all possible color features in the same experiment, to let the AdaBoost classifier select the most informative type of color features for a given attribute.
- It would make sense to study color representations that are more complex than our simple statistical features. The key question is how to incorporate higher-dimensional representations, given that our weak learners are currently one-dimensional decision stumps[13]. One simple experiment would be to feed the corresponding person image through a CNN classifier (pre-trained on existing large-scale datasets), and include the resulting scalar classification score as (global) input feature during training of the classifier.
- Our method did not perform very well on attributes that are underrepresented in our dataset. Recent research on visual attribute recognition (Sarafianos et al., 2018) stresses that *"accounting for class imbalance is essential during learning from large datasets"*. To cope with this, they propose a loss function for their CNN to handle class imbalance at class and at instance level. We believe that similar thoughts could help our method, which currently undersamples the majority class to obtain class balance (at the cost of a significantly smaller training/test set). With regard to AdaBoost, Galar et al. (2012) note that it is *"an accuracy-oriented algorithm, when the class distribution is uneven, this strategy biases the learning (the weights) toward the majority class, since it contributes more to the overall accuracy."*. Subsequently, they describe different strategies for *cost-sensitive boosting*, which modify the weight update rule in various ways.
- Currently, we only train to recognize a single attribute at a time. There is no sharing of selected voxels or weak classifiers when aiming to recognize multiple attributes, whereas modern convolutional neural networks often learn a common, shared feature representation for multiple classes or tasks on the same input data. They achieve better performance through multi-task learning, which has been shown to allow neural networks to generalize better (*e. g.* Caruana, 1997). Instead, our approach relies on pre-existing, rather general, hand-crafted features, and selects the best features specific to the attribute at hand. However, there exists literature on *boosted* multi-task learning (Chapelle et al., 2011).

[13] We also tried decision trees of depth 3 and 5 for *gender*, and got slightly worse results.

- We do not perform any training-time augmentation, as is commonly done in modern deep learning frameworks. This is because in the presented implementation, features for all training samples need to be pre-computed before training, and we are limited by available system memory. However, with the proposed voxel pruning strategy, or a newer boosting implementation like XGBoost (Chen et al., 2016) that supports external memory, such online augmentations could be feasible. For instance, one could horizontally flip the point clouds, perform random point drop-out, or add mild perturbations to the points' coordinates and color values.

The dataset presented in Section 7.3.2 has the following limitations:

- Due to the prototype Kinect v2 sensor that we used, maximum range in our dataset is limited to 4.5 m. It would be interesting to train and evaluate a classifier at larger distances of 8–9 m, where point clouds become increasingly sparse.
- Our dataset contains only upright standing and walking poses, but no sitting or more unusual poses. Also, persons are never occluded. Covering such cases could improve results during training, and provide valuable insights during testing. It would be interesting to see if synthetically generated data and augmentation with 3D occluder objects, as presented in Chapter 5, could help here, and how much variety in synthetic humans would be needed.
- The dataset has been recorded in summer in a single city in Germany. Thus, it lacks the seasonal and cultural differences in clothing likely to be observed *e. g.* at an airport. Since such data is difficult to collect in a laboratory, one idea would be to automatically mine such data from in-the-wild datasets by exploiting human detection and tracking. We did some initial steps in this direction, but found that detection results were not yet sufficiently accurate for this purpose, therefore we started investigating a more robust RGB-D detection method in Chapter 5.

In terms of experiments, it would be interesting to see a comparison in accuracy to a recent image-based CNN trained for human attribute recognition, which could benefit from pre-training on ImageNet (J. Deng et al., 2009), SceneNet RGB-D (McCormac et al., 2017) or other large-scale image-based datasets. Likewise, we would be interested to see a comparison against a neural network learning point cloud features, such as PointNet++ (Qi, Yi, et al., 2017).

Ideas for future work

We also have a number of ideas for extending the method, that go beyond the scope of the presented work:

- Our in-the-wild experiments in Section 7.4.4 led us to the observation that temporal filtering has no significant, positive impact, because misclassifications are rather systematic than spurious in nature. This observation reminds us of one of our key findings from Chapter 4, that detector performance is the single largest factor influencing tracking results, and that the choice of the actual tracking algorithm does actually not matter so much. We believe that future work in this direction should concentrate on improving classifier

accuracy, for instance by following some ideas from the previous subsection, rather than filtering. However, early- or mid-level temporal fusion of several frames as in Chapter 6 could help to resolve ambiguity arising from variations in human pose.

- The final three qualitative observations listed in Section 7.4.4 indicate that some form of *confidence output* and *introspection capabilities* (*cf. e. g.* Grimmett et al., 2016) would be helpful. If the classifier were *aware* that it had never been trained on distances larger than 4.5 m, or that it never saw anybody carrying a handbag, it could refrain from outputting classification results, or assign a low confidence.

- We had the idea that the tessellation-learning approach should be well-suited for *estimation of rough body orientations* (yaw angles). However, this requires either multi-class classification at discretized intervals, or angular regression, both of which are not directly supported by the OpenCV AdaBoost implementation we use.

  Very recently, Wengefeld, Lewandowski, et al. (2019) independently came up with the same idea and modified our method for this use-case. They compared a variant using multi-class classification, implemented as a one-vs-all approach that combines multiple AdaBoost classifiers, and a regression variant using gradient boosted trees. They experienced similar memory issues with the OpenCV implementation, leading them to also subsample the dataset to fit into memory, and later integrate the more recent XGBoost library (Chen et al., 2016) that became popular after our initial publication. In their best configuration, they achieve a mean angular error of 12.2° on a novel Kinect v2 motion capture dataset, outperforming the deep learning-based articulated 3D human pose estimation method by Zimmermann et al. (2018) that we used as a strong baseline in Chapter 5.

- We only considered binary human attributes, and not continuous attributes such as *age*, or multi-class attributes like *age group*, which could also be highly relevant for socially aware robots. Age estimation has a large research community of its own, often focusing on facial images. One general challenge, due to the large variance in human appearance and the large number of age groups, is the need for a very large-scale dataset. Given such data, the extended version of our method proposed by Wengefeld, Lewandowski, et al. (2019) could be trained for age estimation, which is a use-case that they explicitly mention. They propose *weight* as a further example for real-valued human attributes.

Part V

Practical applications

CHAPTER 8

A modular framework for multi-modal people tracking

We now examine how we can bring together the individual detection, tracking and analysis modules that have been presented in the previous chapters. To this purpose, we present a unified multi-modal person and group detection and tracking framework which was initially developed during the EU project SPENCER. Based upon the Robot Operating System (ROS) middleware, it was designed with a modular architecture to make it robot- and application-agnostic and permit easy reuse under different sensor setups. The same framework has later been deployed in the EU project ILIAD on a heterogeneous fleet of autonomous guided vehicles (AGVs) without any modifications to its architecture or the underlying message definitions. Instead, research could focus completely on improving individual detection and tracking components. To our knowledge, the presented framework is the functionally most complete one that is currently available for ROS as open-source software, and has facilitated research in several projects also by other researchers.

Parts of this chapter have been presented in a book chapter entitled "People Detection, Tracking and Visualization using ROS on a Mobile Service Robot" by T. Linder and K. O. Arras in Robot Operating System (ROS) – The Complete Reference (Vol. 1), Springer Studies in Systems, Decision and Control, Springer Verlag, Anis Koubaa (Editor), 2016.

8.1 Introduction

In this chapter, we present a concept for, and experiences with, a robot- and application-agnostic multi-modal people detection and tracking framework, which has been developed during the EU project SPENCER. It is based upon the Robot Operating System (ROS) middleware and supports robots which are equipped with a heterogeneous array of sensors. It combines detectors from different modalities in a unified framework.

One key contribution of this chapter is a set of reusable message definitions for a modular people and group detection and tracking pipeline. In our research, we demonstrated that these can successfully be applied across different sensor modalities and real-time detection and tracking algorithms, for which we provide exemplary implementations. Based upon these message definitions, we provide further tooling including a reusable and highly configurable set of visualization plugins for the RViz visualization tool.

The code for most detection and tracking modules, as well as our message definitions and visualization plugins, can be found online in a Git repository (Linder et al., 2016), many times under a BSD license. Binary DEB packages are provided and built by the L-CAS build farm[1]. Our components have been tested on ROS Hydro and Indigo on 64-bit Ubuntu 14.04 and 16.04.

Human detection and tracking pipeline

Figure 8.1 gives an overview of the real-time people and group detection and tracking pipeline developed in the context of the SPENCER project. For the later ILIAD project, we were able to reuse most components of the pipeline without major modifications.

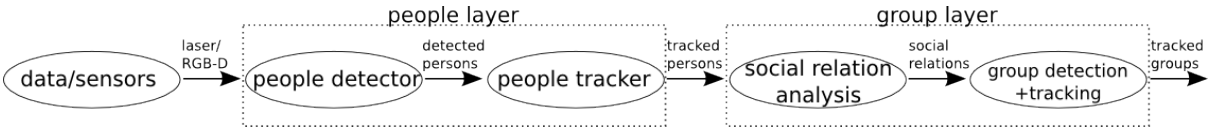


Figure 8.1: People and group tracking pipeline developed during the SPENCER project.

In Figure 8.2, we can see individual modules, sorted by category, that have up to now been integrated into the proposed framework.

This chapter is structured as follows: Starting with sensory data such as 2D laser scans, RGB-D point clouds or stereo or monocular camera images, we first detect people using detectors devised specifically for the particular sensor modality (Section 8.3). The resulting person detections are then fed into a person tracker (Section 8.4), which integrates the information over time and attempts to maintain a consistent ID for a given person for as long as possible. If desired, these person tracks can also be clustered into groups and be tracked over time (Section 8.5).

All of this becomes more complex if information from multiple detectors operating on different sensor modalities, such as 2D laser and RGB-D, shall be combined to make tracking more robust and cover a larger field of view, as we will discuss in Section 8.6. Using the powerful RViz visualization tool and custom plugins developed by us as well as a custom SVG exporter script, the outputs of the tracking pipeline can be visualized (Section 8.7). In Section 8.9, we briefly outline how to generate test data for experiments using a pedestrian simulator. Finally, we present different robot platforms on which the proposed tracking framework has successfully been deployed, which highlights how the framework can easily generalize to other robot setups.

The entire communication between different stages of our pipeline occurs via ROS messages defined in the corresponding sections, which we explain in detail to encourage reuse of our components in custom setups. The architecture allows for easy interchangeability of individual components in all stages of the pipeline. Because most components are implemented as separate ROS nodes and executed as standalone processes, they automatically benefit from parallelization on multi-core CPUs.

[1]Thanks to help from Marc Hanheide at L-CAS, University of Lincoln.

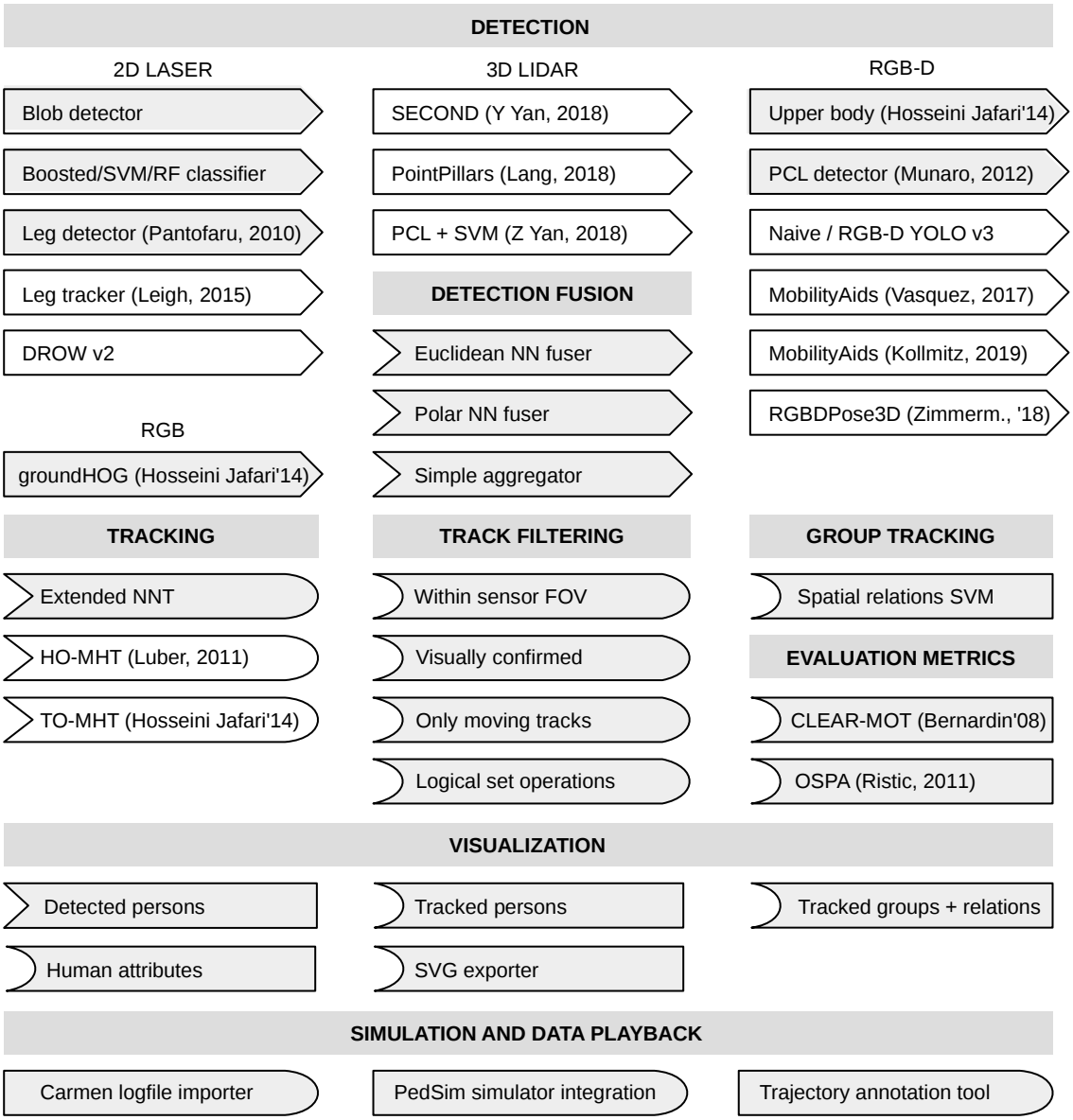


Figure 8.2: Own and third-party components that have been integrated into the modular ROS-based people tracking framework. The shaded modules are already available online in the framework's GitHub repository.

8.2 Related work

While there has been plenty of research in human detection and tracking for robotics, for example documented in a recent overview article by Bellotto et al. (2018), we are aware of only few publicly available, reusable, modular, multi-modal detection and tracking frameworks that are ready for live deployment on a robot. The ROS people tracking stack (Pantofaru, 2010) includes

facial detectors for stereo cameras and a leg detector for 2D laser, but no additional functionalities such as evaluation metrics, group tracking or visualization plugins. The Bayes Tracking Library by Bellotto et al. (2015) focuses solely on the tracking algorithms and includes implementations of NNSF- and JPDAF-based tracking methods using Extended/Unscented Kalman Filters or Particle Filters. It supports fusing detections from multiple sensor sources. The OpenPTrack framework by Munaro, F. Basso, et al. (2016) is targeted at networks of mostly statically mounted RGB-D sensors. The Intel Object Analytics module (2017) is also targeted at RGB-D sensors, and has recently been extended with ROS 2 support. Very recently, Wengefeld, Mueller, et al. (2019) presented a novel multi-modal tracking framework similar to ours, but to our best knowledge it is not publicly available.

8.3 Human detection

In this section, we present generic ROS message definitions for human detection across different modalities. We then briefly discuss own and third-party methods that have been integrated into our ROS-based framework so far. The shaded components in Figure 8.2 are already available online; we are planning to make further of these modules available in the future where the licensing situation permits us to do so.

ROS message definitions for human detection

We designed a reusable set of ROS message definitions that act as a language-independent interface between detection and tracking modules. Our message definitions for human detection are intentionally kept as simple and generic as possible, to allow reuse over a wide range of possible sensor modalities and detection methods. Therefore, we *e. g.* do not include image bounding boxes or visual appearance information which would be specific to vision-based approaches and do not exist *e. g.* in 2D range data.

Our definition of a detected person is similar to a geometry_msgs/PoseArray, but additionally, for each detection we also specify a unique detection ID, a confidence score, a covariance matrix and the sensor modality. The detection ID allows the detection to be matched against *e. g.* the corresponding image bounding box, published under the same detection ID on a separate ROS topic. The covariance matrix expresses the detector's uncertainty in position (and orientation, if known). It could, for instance, be a function of the detected person's distance to the sensor, and is therefore not necessarily constant. The confidence score can be used to control track initialization and to compute track scores. Finally, information about the modality can be used by a tracking algorithm to *e. g.* assign a higher importance to visually confirmed targets.

A spencer_tracking_msgs/**DetectedPerson** thus has the following attributes:

- *detection_id* [uint64]: Unique identifier of the detected person, monotonically increasing over time. To ensure uniqueness, different detector instances should use distinct ID ranges, by *e. g.* having the first detector issue only IDs that end in 1, the second detector IDs that end in 2, and so on.

- *confidence* [float64]: A value between 0.0 and 1.0 describing the confidence that the detection is a true positive.
- *pose* [geometry_msgs/PoseWithCovariance]: Position and orientation of the detection in metric 3D space, along with its uncertainty (expressed as a 6×6 covariance matrix). For unknown components, *e. g.* position on the z axis or orientation, the corresponding elements should be set to a large value[2]. The pose is relative to the coordinate frame specified in the DetectedPersons message (see below).
- *modality* [string]: A textual identifier for the modality or detection method used by the detector (*e. g.* RGB-D, 2D laser). Common string constants are pre-defined in the message.

A **DetectedPersons** message aggregates all detections found by a detector in the same detection cycle, and contains the following attributes:

- *header* [std_msgs/Header]: Timestamp and coordinate frame ID for these detections. The timestamp should be copied from the header of the sensor_msgs/LaserScan, Image or PointCloud2 message that the detector is operating on. Similarly, the coordinate frame can be a local sensor frame as long as a corresponding transformation into the tracking frame (usually "odom") exists in the TF hierarchy (see *e. g.* Section 9.5.1).
- *detections* [array of DetectedPerson]: The array of persons that were detected by the detector in the current time step.

Human detection in 2D laser

These 2D laser-based human detectors have been integrated with the framework:

- **Boosted/SVM/Random Forest segment classifier:** Our re-implementation of the boosted classifier by Arras et al. (2007) using machine learning methods from the OpenCV library. We have trained Adaboost, SVM and Random Forest classifiers. Prior to classification, the laser scan is segmented by a separate ROS node using either jump distance or agglomerative hierarchical clustering with a configurable distance threshold. Exemplary detection results in the airport environment are shown in Figure 8.3.
- **ROS leg detector:** A wrapper to make the existing ROS `leg_detector` package by Pantofaru (2010), used in Chapter 6, compatible with our message definitions.
- **DROW v2:** We have wrapped the CNN-based method with temporal integration from Chapter 6 into a ROS node.
- **Leg tracker** by Leigh et al. (2015): Also see Chapter 6. While this method performs tracking internally, we present the tracked person centroids as detection input to our multi-modal tracking system.

[2] We usually use a value of 10^5 to indicate this. If set to infinity, the covariance matrix becomes non-invertible, causing issues later on during tracking.

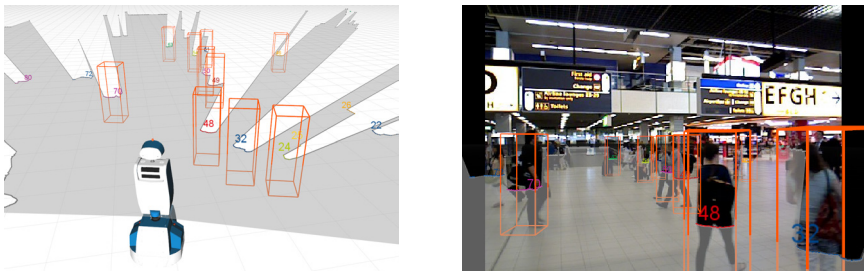


Figure 8.3: *Left:* Visualization of laser scan segments (identified by numbers), human detections (orange boxes), and the area that is visible to the front laser scanner. *Right:* Projection of the laser-based person detections into a color image of the scene.

Human detection in RGB/monocular vision

When no depth or range sensor is available, the following method can be used:

- **groundHOG:** The GPU-accelerated medium- to far-range groundHOG detector from Hosseini Jafari et al. (2014)[3] requires a groundplane estimate (*e. g.* from TF).

Human detection in RGB-D

The following RGB-D detectors have been integrated:

- **Upper-body detector** by Hosseini Jafari et al. (2014)[3]: As shown in Figure 8.4 (left), this methods slides a learned upper-body template over the depth image. No RGB information is used, except for visualization. The method requires a ground plane estimate, and was used on the SPENCER robot – see Chapters 4 and 9.
- **PCL people detector** by Munaro et al. (2012): Uses the Point Cloud Library (PCL) to which extract interest region proposals from a depth-based height map and then applies a linear HOG-SVM classifier. We extended the code to output markers for visualization of ROIs (Figure 8.4, middle) and output DetectedPersons messages.
- **ComboHOD** by Spinello and Arras (2011): This detector has been used in Chapter 3, and is visualized in Figure 8.4 (right).
- **YOLO v3 with RGB-D fusion**: Our proposed method from Chapter 5, and its naïve baseline without 3D centroid regression. Used in the ILIAD project.
- **MobilityAids** (Vasquez et al., 2017): Used as a baseline in Chapter 5.
- **MobilityAids** (Kollmitz et al., 2019): Also baseline in Chapter 5.

[3]This method has been integrated by Stefan Breuers from RWTH Aachen.

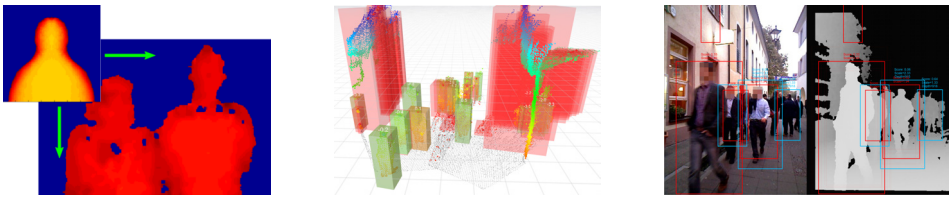


Figure 8.4: Different RGB-D detectors which we integrated into our framework. *Left:* Upper-body detector from Hosseini Jafari et al. (2014) which slides a normalized depth template over the depth image. *Middle:* RGB-D detector from PCL which first extracts regions of interest, visualized here by boxes, from the point cloud (Munaro et al., 2012). *Right:* Combo-HOD detector (Spinello and Arras, 2011) from Chapter 3.

- **RGBD Pose 3D** by Zimmermann et al. (2018): An articulated 3D human pose estimation method based upon OpenPose, from which we extrapolate human centroids. Used in Chapter 5.

8.4 Human tracking

The output of a person detector represents only a single snapshot in time and may be subject to false alarms and missed detections. Therefore we want to track detections over time. If only a single sensor is used for human detection, the resulting detections can directly be fed into one of our tracking algorithms to bridge short moments of occlusion, filter out intermittent false alarms and maintain a persistent track identifier. If multiple sensors are being used, the resulting detections first need to be fused as described in Section 8.6.

ROS message definitions for human tracking

Like in the detection case, we tried to keep the message definitions for human tracking as generic as possible such that they can be re-used across a large variety of different tracking methods.

A **TrackedPerson**, according to our definition, possesses the following attributes:

- *track_id* [uint64]: An identifier of the tracked person, unique over time.
- *is_matched* [bool]: False if no matching detection was found close to the track's predicted position. If false for too long, the track usually gets deleted.
- *is_occluded* [bool]: True if the person is physically occluded by another person or obstacle. False if the tracking algorithm cannot determine this.
- *detection_id* [uint64]: If is_matched is true, this is the unique ID of the detection associated with the track in the current tracking cycle. Otherwise undefined.
- *pose* [geometry_msgs/PoseWithCovariance]: The position and orientation of the tracked person in metric 3D space, along with its uncertainty (expressed as a 6×6 covariance matrix). The pose is relative to the coordinate frame specified in the TrackedPersons message (see below).

- *twist* [geometry_msgs/TwistWithCovariance]: The linear and possibly angular velocity of the track, along with their uncertainties.
- *age* [duration]: The time span since when this track has existed in the system.

The **TrackedPersons** message aggregates all TrackedPerson instances that are currently being tracked:

- *header* [std_msgs/Header]: The coordinate frame is usually a locally fixed frame that does *not* move with the robot, such as "odom". The timestamp is copied from the incoming DetectedPersons message; this is important to synchronize DetectedPersons and TrackedPersons messages to be able to look up details about a DetectedPerson (identified by its detection_id).
- *tracks* [array of TrackedPerson]: All persons that are being tracked in the current tracking cycle, including occluded and unmatched tracks.

Tracking algorithms integrated into the framework

We have integrated the following tracking methods, which are discussed more extensively in Chapter 4:

- **Extended nearest-neighbor tracker** from Linder, Girrbach, et al. (2015). Optionally subscribes to a second DetectedPersons ROS topic with high-recall detections from a low-confidence detector (*e. g.* laser blob detector). After regular data association, we perform another round of data association where we associate tracks that have not yet been matched with these high-recall detections, for a certain maximum number of frames.
- **Hypothesis-oriented MHT**, originally developed by Luber, Tipaldi, et al. (2011a), also including the variant with group models from Linder and Arras (2014) presented in Chapter 3;
- **Track-oriented MHT**: The MDL tracker from Hosseini Jafari et al. (2014), initially proposed by Ess et al. (2009), was integrated[3] for the experiments in Chapter 4.

For backwards compatibility with older data recordings, *e. g.* the *Freiburg City Center* and *Main Station* datasets from Luber, Tipaldi, et al. (2011b) and Luber, Tipaldi, et al. (2011a), we provide an import module that republishes CARMEN logfiles as ROS messages.

8.4.1 Tracking metrics

Having a robust implementation of metrics for tracking is important for benchmarking, regression testing and automated hyperparameter optimization (*cf.* Chapter 4). Therefore, we have wrapped two well-tested open-source Python implementations of the **CLEAR-MOT** (Bernardin et al., 2008) and **OSPA** (Ristic et al., 2011) multi-target tracking metrics into ROS nodes that adhere to our

message conventions. They assume that groundtruth trajectories are given in the form of time-synchronized `TrackedPersons` messages. Ignore flags can optionally be set using the `is_occluded` field. Groundtruth can be obtained from a motion capture system, such as in the Toulouse Motion Capture dataset (see Appendix A), or manually using a trajectory annotation tool like the one presented later in this chapter.

8.4.2 Post-processing filters for tracking and detection

Higher-level reasoning components, such as motion planning or HRI, are often not interested in all tracks that are output by the people tracking system. We therefore offer a series of ROS nodes which can be attached to the output of a tracking algorithm. They consume and output `TrackedPersons` messages and can thus be chained in series if desired. The provided filters include:

- Selecting only visually confirmed tracks,
- Selecting non-static tracks,
- Selecting tracks within a certain sensor's field of view,
- Logical set operations (union, intersection, complement),
- Computing occlusion status – either via raytracing, or by searching for points in 2D laser –, which is useful for setting ignore flags on groundtruth trajectories.

For `DetectedPersons` messages, we offer the following filtering modules:

- Filtering out (false) detections using a static obstacle map,
- Filtering detections within another sensor's field of view.

8.5 Group tracking

In Chapter 3, we have presented strategies for detecting and tracking groups of people from a mobile robot, based upon coherent motion indicator features. Here, we summarize how group tracking is modeled within our ROS-based framework.

ROS message definitions for group tracking

In Figure 8.5, we show how a `TrackedGroup` is constituted of multiple tracked persons (or rather, their unique IDs). In addition, we keep track of when the group was formed, and compute its center of gravity from the positions of the individual person tracks. The `TrackedGroups` message contains all groups tracked within the current step.

As we have seen in Chapter 3, one way of detecting groups is by computing pairwise social relations to build a social relationship graph (Figure 3.1b on p. 30). To make this more modular, the graph can be computed and published by a separate ROS node in the form of `SocialRelations`

messages (Figure 8.6), which are weighted edges in the graph with a specific predicate or *type* (for instance “social” or “spatial”). In our framework, we provide such a ROS node for group detection based upon coherent motion indicators. We also provide an implementation of the computationally efficient baseline group tracking method which we presented at the end of that chapter.

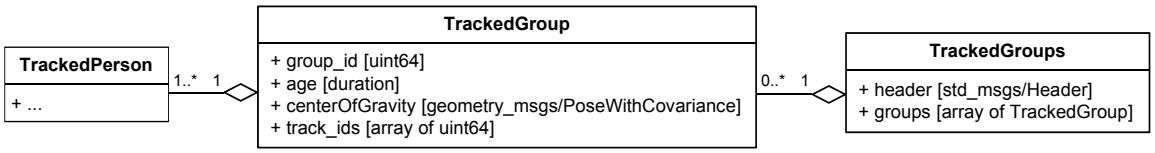


Figure 8.5: Message definition for a single tracked group, a collection of all tracked groups in one cycle, and their relation towards tracked persons.

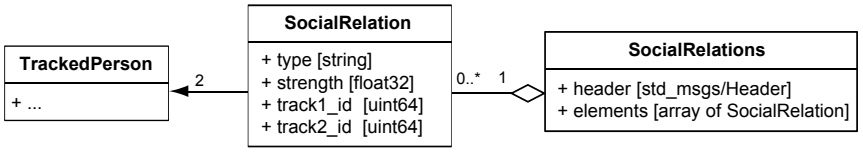


Figure 8.6: Message definitions for the social relationship graph. The strength of a social relation is specified as a real number between 0.0 and 1.0. The type string can be used to distinguish between different types of relations.

8.6 Tracking in multiple modalities

In this section, we describe simple primitives for fusing the output of multiple detectors that potentially operate on different sensor modalities. For a broader overview of fusion strategies for multi-modal detections, see Bar-Shalom et al. (2011). Here, we mainly focus on fusion at the detection level under the assumption that we are operating a single, centralized tracker[4].

ROS message definitions for detection fusion

We introduce a `CompositeDetectedPerson` message (Figure 8.7) to keep track of which detections have been fused. Retaining this information is essential for components at later stages in the perception pipeline, like a human attribute classifier that takes as an input the regions of interest output by a vision-based person detector. If these ROIs and human attributes likewise bear detection IDs, they can later on be associated with the corresponding tracked persons such that the extracted information may for instance be smoothed over time.

A composite detection is composed of the original set of detections, which we copy here for simplicity; a fused pose; scalar values that denote the minimum, maximum and average detection confidence of the underlying detections (before fusion); as well as a unique detection ID. How the fused pose and its covariance are computed is determined by the respective fusion primitive.

[4]Track-to-track fusion, which is computationally more complex, can become more interesting for information exchange in multi-robot systems, such as the fleet of AGVs in the ILIAD project.

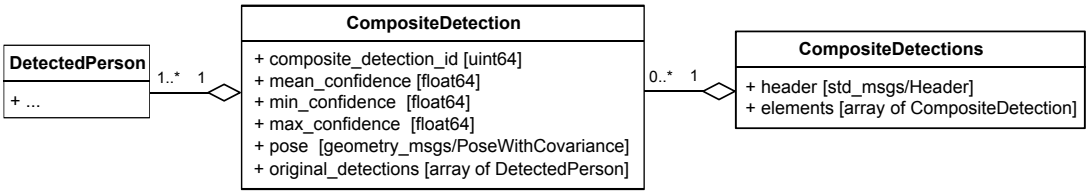


Figure 8.7: Message definitions for composite detections. Retaining the information which original detections have been fused into a composite detection is important for perception components at later stages in the pipeline.

In order to provide all fusion primitives described in the following with a homogenous interface, we initially convert *all* `DetectedPersons` messages via a converter node into `CompositeDetectedPersons`, even if only a single detection is involved and no actual fusion has occurred. This allows for easy chaining of the fusion primitives.

Primitives for fusion at the detection level

Our framework allows for two basic fusion strategies at the detection ("measurement") level, which we describe briefly in the following.

Detection-to-detection fusion

Fusing detections to other detections has got the advantage that the resulting composite detections can be fed into any existing (unimodal) tracking algorithm, without requiring special provisions to cope with information from multiple detectors. This makes it easier to compare different tracking methods, which was one of our motivations. Furthermore, this approach can also be helpful if the tracking algorithm itself is computationally very complex and scales badly with the number of detections.

While much more elaborate multi-sensor fusion strategies have been studied in the literature, see for example the survey article by Khaleghi et al. (2013), this detection-to-detection fusion approach yielded good results in our experiments. We have implemented a series of reusable **fusion primitives** as ROS nodelets which can be used to flexibly compose a fusion pipeline by means of roslaunch XML files:

- *Converter nodelets* convert `DetectedPersons` into `CompositeDetectedPersons` messages, and vice versa, at the beginning and end of the fusion pipeline.
- *Aggregator nodelets* compute the union of two sets of `CompositeDetectedPersons`, without any further fusion logic. Their main use-case is to combine detections from sensors with non-overlapping fields of view, in which case no data association is required.
- *Fusion nodelets* perform data association between two spatially overlapping sets of `CompositeDetectedPersons`, and geometrically fuse associated detections. We provide

exemplary implementations of fusion nodelets with greedy nearest-neighbor data association in two variants: One with Euclidean association metric, and one that operates on polar coordinates and offers a separate weighting factor for the range value, which is useful *e. g.* for monocular vision-based detectors that do not output accurate depth estimates. In these implementations, fusion is done in a naïve fashion, by computing a weighted mean of the detection centroids using detector-specific weights. More elaborate fusion schemes like covariance intersection as used by Volkhardt et al. (2013) could easily be integrated.

Both aggregator and fusion nodelets operate on *pairs* of `CompositeDetectedPersons` topics. If more than two detection sources shall be fused, the nodelets can be concatenated in series. For every pair of topics, incoming messages are synchronized using an approximate time synchronizer. The disadvantage of this approach, over detection-to-track fusion, is the added latency of each fusion step, and the reduction of the frequency of fused detections to the frequency of the slowest detector (or sensor).

The nodelets support dynamic and automatic reconfiguration of the fusion pipeline: By monitoring their input topics, they switch automatically from simple forwarding of composite detections into fusion mode, and vice versa, when a detector comes online or goes offline. This makes it possible to selectively disable certain detectors for experiments and troubleshooting, without compromising the entire tracking system.

Detection-to-track fusion

Detection-to-track fusion has not yet been implemented in any of our tracking algorithms, but is possible using the proposed message definitions. In this case, the fusion stage needs to be implemented within the tracker itself. It then becomes the tracker's responsibility to publish a corresponding `CompositeDetectedPersons` message to let other perception components know which original detections have been fused. This type of fusion scheme would incur no extra latency during the fusion process, because detections could be fused asynchronously. As the tracking algorithm's knowledge of a track's previous trajectory can help with the association of incoming detections, this approach is generally more informed and may provide better results in certain cases.

8.7 Visualizing outputs of the perception pipeline

Powerful visualization tools can be helpful for debugging and analyzing the performance of a complex perception pipeline. The standard visualization tool of ROS is *RViz*, which uses the OGRE graphics engine as its visualization backend. The core idea behind RViz is that it provides different visualization plugins *(displays)* for different types of ROS messages, *e. g.* sensor_msgs/LaserScan or nav_msgs/Odometry. In general, RViz provides two ways of implementing custom visualizations:

- *Markers and marker arrays*, using existing Rviz displays and message types from the `visualization_msgs` package. They allow to easily display primitives such as lines, tri-

angles, texts, cubes or static 3D meshes. Shape color, pose and dimensions are specified within the message itself, published by any kind of ROS node.

- *RViz plugins*, written in C++, can benefit from the entire set of capabilities offered by the OGRE graphics engine. Each display comes with a property panel based on the UI framework Qt, allowing for easy customization of display options from within RViz. Without extra effort, settings can be persisted into configuration files, allowing for entire visualization setups to be loaded with a mouse click.

While markers allow to quickly implement component-specific visualizations without much developer effort, RViz plugins are especially suited for more complex visualizations that are intended to be reused often (like the components of our ROS framework). They offer a better end-user experience (due to the ability to change settings on-the-fly inside the RViz GUI) at the cost of larger implementation effort.

Custom RViz visualization plugins

We thus developed a series of custom RViz plugins for displaying detected persons, tracked persons, social relations, tracked groups and human attributes. They are highly customizable and provide many useful display options:

- Different visual styles: 3D bounding box, cylinder, crosshairs, animated human
- Coloring: 6 different color palettes
- Display of velocity arrows
- Visualization of the 99% covariance ellipse for position uncertainty
- Display of track IDs, status (matched, occluded), associated detection IDs
- Configurable reduction of opacity when a track is unmatched or occluded
- Track history (trajectory) display as dots or lines
- Configurable font sizes and line widths

Throughout this thesis, we have seen different visualizations generated using our RViz plugins. We sometimes also use them in combination with the existing RViz *Camera* display to project our 3D visualization into a given camera image.

ROS-based SVG exporter for detections, tracks and groups

For the visualization of detected and tracked persons and groups, we additionally provide a Python-based ROS node that can generate scalable vector graphics (SVGs) for display in a web browser. An example is shown in Figure 8.8. This is useful to analyze person trajectories from a 2D top-down view of the scene, which can be animated to display the evolution of tracks over time. In contrast to videos recorded from Rviz, they support free zooming to permit the analysis of smaller details (*e. g.* the cause of track identifier switches, by looking at the positions and timestamps of individual detections).

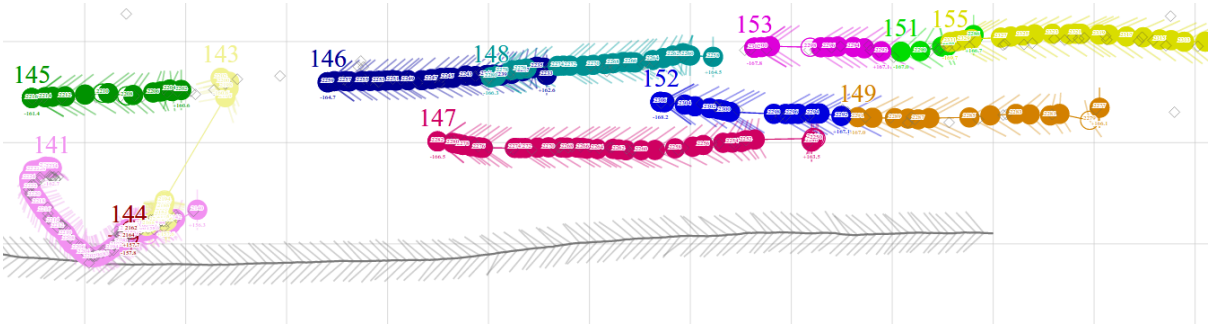


Figure 8.8: SVG rendition of multiple trajectories in a 2D top-down view. Grey diamonds are detections, arrowheads denote velocity, the grey track indicates robot odometry.

8.8 Trajectory-based annotation tool

Figure 8.9: Our trajectory-based annotation tool, integrated with Rviz

At the time it was first proposed by us (Linder, Breuers, et al., 2016), no multi-modal track annotation tool for 2D/3D laser, RGB-D and stereo data had been publicly available to our best knowledge. Our ROS-based multi-modal annotation tool, shown in Figure 8.9, leverages the powerful visualization capabilities of the ROS visualization tool *RViz* and *rqt* to enable annotating people directly in 3D world space in RGB-D, 2D laser and 3D lidar point clouds. For reference purposes, annotations and 2D laser scans are also projected into the camera images. This assumes a good extrinsic sensor calibration, for instance obtained using the tool we will present in Section 9.5.2. By placing trajectory waypoints at regular intervals – usually every 0.5-2.0 sec, depending on the dynamics of the scene – and linearly interpolating in between, the annotation process is sped significantly up.

Related to our work, Manen et al. (2017) present a fast trajectory-based annotation tool with path supervision for video data. Similar to what we propose, their method lets *"the annotator loosely track the object with a cursor while watching the video"*. However, in contrast, our method supports multi-modal data annotation, as long as point clouds from at least one range sensor (2D laser, 3D lidar, RGB-D) are available. We do not link the annotated paths with object detections, as we are only interested in obtaining centroid trajectories and do not require object extents. Their experiments show that *"annotating paths is only 33% slower than watching the video in real-time"*, which is slightly faster than what we are able to obtain in practice: Due to computational restrictions, we usually play sequences at 0.2x – 0.5x rate, depending on the required level of granularity.

8.9 Integration with simulation tools

To be able to test tracking algorithms in simulation, we integrated our framework with a physics-based robot simulator, and a pedestrian simulator.

Integration with the Gazebo robot simulator

We use the robot simulator Gazebo to generate simulated 2D laser and RGB-D sensor data of moving persons from a mobile robot platform (Figure 8.10, left). The noise characteristics of the simulated 2D laser have been matched to datasheet specifications of sensors on our robot.

When simulating a large number of human agents, due to scene complexity we disable the Gazebo physics engine and instead manually update the agents' and robot's position at 25 Hz using the Gazebo ROS API, while keeping collision checking enabled. Person shapes are approximated by static 3D meshes that are processed by a GPU-accelerated raytracing algorithm in Gazebo to simulate laser scans (right picture). The resulting `sensor_msgs/LaserScan` messages are fed to our laser-based person detector.

Integration with the pedestrian simulator PedSim

Simulating realistic pedestrian motion behaviors is a research topic of its own. Here, we use the publicly available PedSim library[5], which simulates pedestrian crowd behaviors using a social force model from behavioral sciences (Helbing et al., 1995). The library has been wrapped into a ROS node[6] and extended with an RViz-based visualization of persons and obstacles using visualization markers (visible in Figure 8.10, right). The positions of the simulated pedestrians are published as `TrackedPersons` messages. A separate converter node then sends these positions to Gazebo, where they are used to position 3D meshes and simulate sensor data. This setup has been used for group tracking experiments at the end of Chapter 3. Since accurate groundtruth positions are known, it can also be used to evaluate person-level tracking.

[5] http://pedsim.silmaril.org/ – Find our ROS wrapper at: https://github.com/srl-freiburg/pedsim_ros
[6] This is joint work with Billy Okal, Dizan Vasquez, Sven Wehner, Omar Islas Ramírez and Luigi Palmieri.

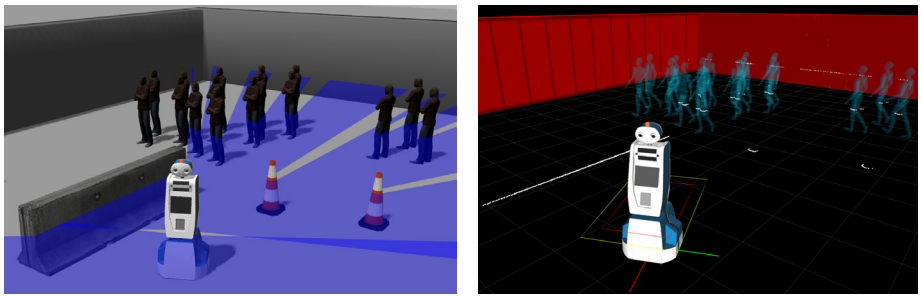


Figure 8.10: *Left:* Positions of simulated pedestrians are constantly sent to the Gazebo simulator via ROS, which then generates simulated laser scans and again publishes them via ROS. These simulated laser scans can then be fed into the people tracking pipeline for synthetic experiments. *Right:* Moving pedestrians in a ROS adaptation of PedSim, the pedestrian simulator, visualized in RViz. The pedestrian behavior is modelled using a social force model from behavioral sciences.

8.10 Practical experiences during deployments

To demonstrate that the proposed framework can generalize across a range of different mobile platforms, we have experimentally deployed it on a number of different systems, visualized in Figure 8.11. We now briefly introduce a few of them.

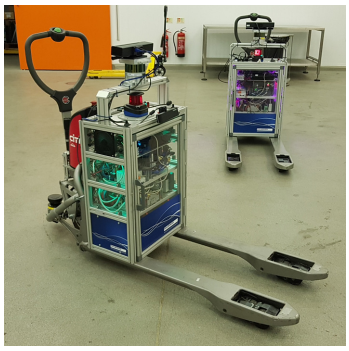
ILIAD: a heterogeneous fleet of AGVs

The heterogeneous fleet of autonomous forklifts, consisting of four smaller pallet trucks (Figure 8.11c) and a larger forklift (Figure 8.11d), represents a challenge for human tracking in the EU project ILIAD. Only two pairs of the smaller trucks share the same sensor setup; this means that a flexible, easily reconfigurable tracking pipeline is required in which individual detector modules can easily be exchanged. The proposed framework offers these capabilities: Because the detection-fusion pipeline automatically reconfigures itself, on each robot, we just start up the detectors for available sensors, and the appropriate fusion pipeline is set up automatically following our generic launch file template. To keep up with recent developments in deep learning, new 2D laser and 3D lidar detectors from Chapter 6 and the RGB-D detector from Chapter 5 have been integrated. Heavy use has been made of the calibration tool that we will present in Section 9.5.2, to efficiently calibrate all vehicles in the fleet.

Because the proposed framework was originally not designed for multi-robot setups (which would lead to topic naming conflicts), we had to ensure that the perception modules of different robots stay isolated. This was achieved initially through a ROS multi-master setup, later replaced by a modified variant of rosduct[7]. In both cases, only the outputs of the human tracking pipeline are forwarded to a central coordinator PC; all other ROS topics stay internal to the particular AGV. For future versions of the framework, it would seem worthwhile to parameterize robot names in all ROS nodes and launch files.

[7] http://wiki.ros.org/multimaster_fkie and https://github.com/LCAS/rosduct

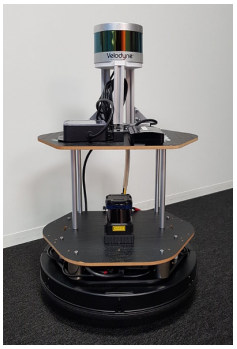
(a) SPENCER



(b) ILIAD Cititrucks (4x)

(c) ILIAD BT Truck

(d) Autonomous Shuttle



(e) TurtleBot

(f) DARYL

Figure 8.11: Different robot platforms on which the proposed ROS-based framework, or parts of it, have been deployed by the author. (Autonomous Shuttle picture by Holger Banzhaf, used with permission.)

SPENCER robot

While we provide further details on SPENCER in the next chapter, Figure 8.12 shows a multi-modal tracking setup built using the framework that we tested in the airport environment.

Autonomous shuttle bus

For demonstration purposes, we showcased the human detection and tracking framework on recorded bagfiles from an autonomous campus shuttle, shown in Figure 8.11 (d). As this platform is equipped with an array of 3D lidar sensors and our framework did not include a lidar-based detector at that point, we integrated a previously existing, proprietary method with our tracking algorithms. Data from different sensors was fused using detection-to-detection fusion to enable tracking over longer distances, shown in Figure 8.13 (right picture). For performance evaluation, we also made use of the trajectory-based annotation tool (left and middle picture).

TurtleBot

We deployed the 2D laser-based leg detector on a small Intel NUC i7 computer inside a TurtleBot platform to evaluate laser-based human detection and tracking in office spaces.

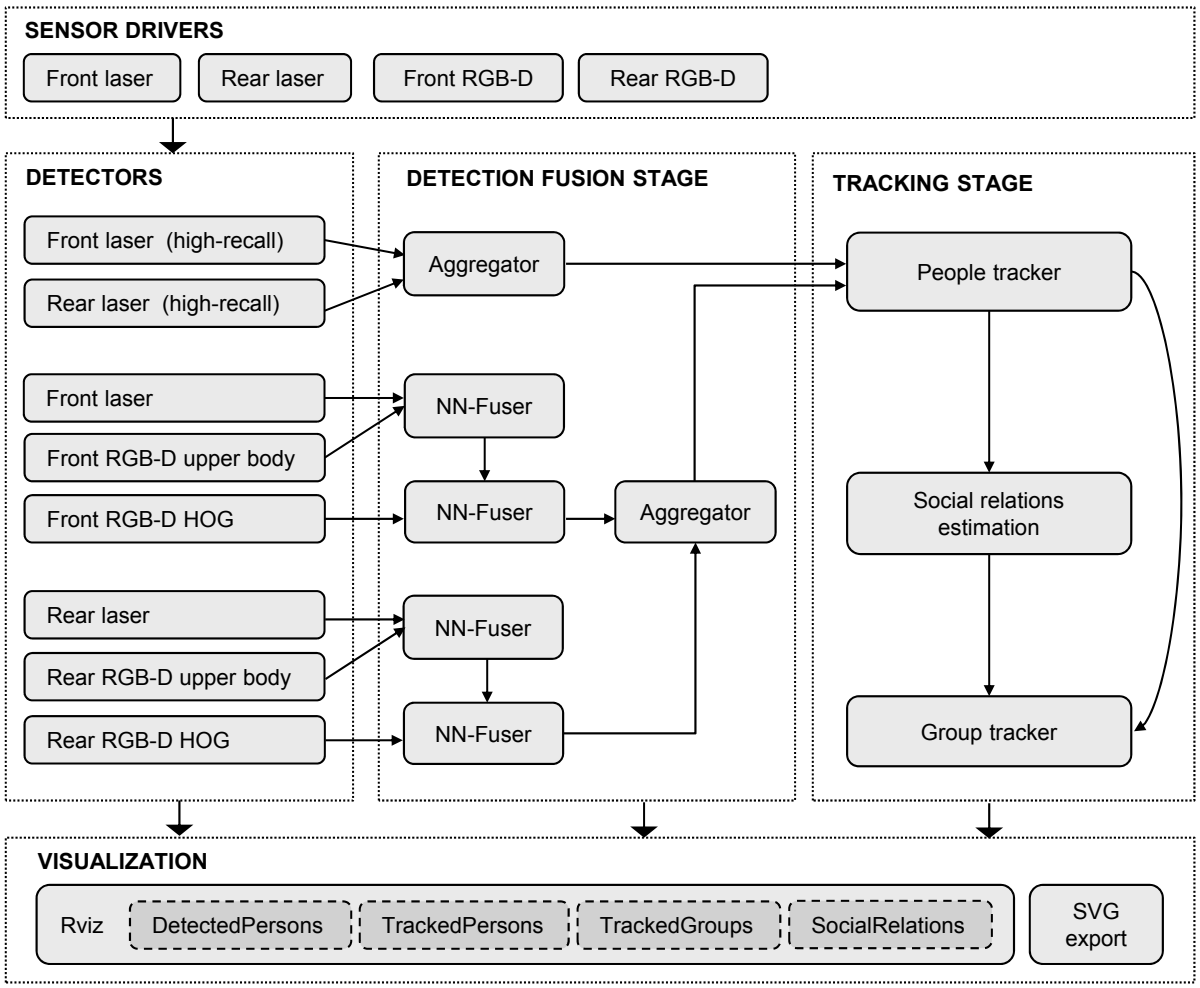


Figure 8.12: Multi-modal human detection and tracking setup on the SPENCER robot, realized with reusable components from the proposed ROS framework. Arrows represent message flows; these connections are flexibly configured in roslaunch XML files. Because individual components are separate ROS nodes, they are easily distributed across several computers (*cf.* Section 9.4).

Further applications of the framework by others

The proposed framework, or parts of it, have been used in research by others, for example a recent paper on sidewalk navigation by Du et al. (2019), learning to navigate robotic wheelchairs from demonstration (Kutbi et al., 2019) or model-predictive control for collision avoidance (Brito et al., 2019). A visualization generated by our Rviz plugins is featured in a video[8] of the JackRabbot project (2015–2020) on a social robot.

[8] https://youtu.be/U7LcN-mdoVs (JackRabbot at Samsung CEO Summit 2018), accessed Nov 7, 2019

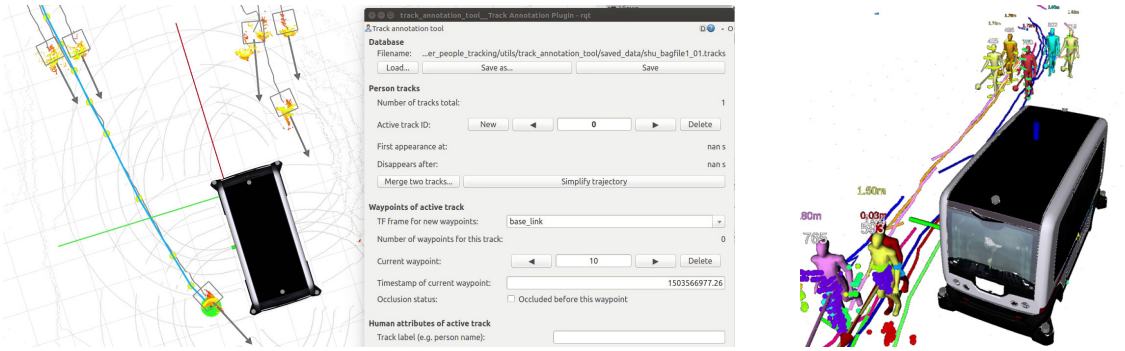


Figure 8.13: Data annotation and tracking visualization on a dataset recorded from an autonomous shuttle bus. For this demonstration, we were able to quickly integrate an existing 3D lidar-based detector into our ROS framework.

8.11 Conclusions

In this chapter, we presented a unified, multi-modal, ROS-based people detection and tracking framework. It has initially been deployed on a person guidance service robot and was later tested on further kinds of robots, including a heterogeneous fleet of AGVs, to demonstrate that its modular structure with clearly defined ROS interfaces allows for reuse across different sensor setups and robotic applications.

We have seen from our experiments in Chapter 4, 5 and 6 that the proposed message interfaces allow to easily replace individual components of the pipeline, such as the core tracking algorithm, by different implementations for comparative studies. While doing so, the user can select from a broad range of existing visualization, evaluation and simulation tools to quickly set up a usable tracking system. To our knowledge, the presented framework is the functionally most complete one that is currently available as open-source ROS software. Because modules are implemented as individual ROS nodes which spawn separate processes, systems built using this framework can easily be distributed across multiple computers, and leverage parallelism of multi-core processors.

The main weakness of the proposed framework, which on the other hand also offers a lot of flexibility, lies in how we fuse information from multiple sensors. The currently implemented sequential detection-to-detection fusion pipeline is highly modular and easy to introspect (*e. g.* using Rviz), but restricted to operate in a synchronous fashion. This means that composite detections are output at the lowest sensor rate of all fused sensors, and every single fusion step incurs extra latency. Albeit less modular, this could be addressed by implementing asynchronous detection-to-track fusion inside the tracking algorithm. In Section 8.6, we outlined how this would be possible using the existing ROS message definitions.

Inclusion of appearance information (for data association and person re-identification) could bring a large benefit; however, it needs to be understood how to design an architecture and data association scheme that is tolerant to missing appearance information in a multi-modal setup, for example when only detections from 2D laser are available for a given person.

CHAPTER 9

Deploying a 250 kg person guidance robot in a crowded airport terminal

Figure 9.1: SPENCER robot at Amsterdam-Schiphol airport

9.1 Introduction

In Chapter 1, we introduced the SPENCER project (2013–2016), a three-year research project funded by the EU's Seventh Framework Programme comprising eight international partners from academia and industry. To demonstrate the technologies developed in SPENCER in a practical application scenario, the defined goal was to deploy a service robot inside a crowded terminal at Amsterdam-Schiphol airport in order to efficiently guide passengers to their departure gate while adhering to normative social behaviors.

In this chapter, we describe the technical challenges of such a deployment that go beyond the theoretical contributions presented in earlier chapters. Deploying such a complex, first-of-its-kind robotic system in an uncontrolled, real-world scenario is a joint team effort which requires the integration of dozens of individual software components.

For a three-year project, SPENCER had very ambitious research goals. According to the official project website, the overall objectives included:

- Robust detection, tracking and multi-person analysis of individuals and groups,
- Recognition of human social relations, social hierarchies and social activities,
- Normative human behavior learning and modeling,
- Socially-aware task, motion and interaction planning,
- Learning socially annotated maps in highly dynamic environments,
- Empirical evaluations to assess socio-psychological effects of normative behaviors.

Because of this ambitious set of goals and the desire to demonstrate many of these technologies in real-time on a robot, a significant amount of computing power was required, together with demands on long battery runtime resulting in a very heavy and tall platform. At the same time, the demonstration use-case was targeted at passengers with short connection times, requiring the robot to operate at fast walking speeds.

As such, to the author's best knowledge, SPENCER was the first fully autonomous service robot with such a high mass and operating at such high speeds in an indoor environment as crowded and complex as an airport terminal. However, the combination of a very *heavy, tall and fast robot operating among humans* introduces additional safety-related challenges, which cannot be neglected in an uncontrolled public environment – even if it is just a research prototype.

Outline This chapter is structured as follows: After discussing related work, we introduce the SPENCER hardware platform, including its sensor suite which is essential for the perception components presented within this thesis. We then describe the overall ROS-based software architecture, individual software components developed by the author of this thesis, and the integration of the human perception pipeline from Chapter 8. Next, we present hardware and software measures that had to be implemented on a lower level of the system to enhance the practical safety of the robot. We conclude with practical experiences made with the system.

9.2 Related work

To get a better understanding of the unique characteristics of the SPENCER robot, we now discuss other mobile robots with ability to detect and track humans that have been developed in the last two decades. We first describe past and present developments in the field of human-aware mobile robots, and then we focus on socially-aware robots. For each category, we explain how SPENCER distinguishes itself from these robots. Successors of SPENCER that were developed after the end of the project are briefly discussed at the end of the chapter.

Human-aware mobile robots

First experiences with mobile robots that are able to sense humans in their surroundings date back to RHINO (Burgard et al., 1998), the "interactive museum tour-guide robot", MINERVA (Thrun et al., 1999), the EXPO.02 robotics exhibition with eleven freely navigating, interactive robots (Siegwart et al., 2003), and TOURBOT (Trahanias et al., 2005). Further museum robots are presented in Arras et al. (2002). RHINO, for instance, used a probabilistic novelty filter approach for detecting people and other objects through a combination of tactile, infrared, sonar and laser range sensors.

Later robots also operating in indoor environments include multiple generations of mobile shopping assistant robots, with more advanced multi-modal person detection, tracking and guidance capabilities (Gross et al., 2008; Gross et al., 2009). In extensive field trials of the "Shopping Guide" project, a series of MetraLabs SCITOS A5 platforms have been driving autonomously for over 2187 kilometers. One limitation of their human detection and tracking system that fused information from laser, sonar and panoramic cameras was the limited range of 3 m. The STRANDS project (2013–2017), started at the same time as SPENCER, also used several SCITOS platforms, deployed in hospitals and elderly care facilities. Similar to SPENCER, human-awareness has been achieved through 2D laser and RGB-D cameras (Hawes et al., 2017).

With focus on outdoor use-cases, the EUROPA project (2009–2013) and the EUROPA2 project (2014–2016) investigated how mobile robots can autonomously navigate in urban scenarios, like a pedestrian zone. Research topics also included detection and tracking of dynamic objects, including pedestrians. The robot Obelix was equipped with multiple 2D laser, 3D lidar, and stereo cameras. At about the same time, the FROG project (2011–2014), short for "Fun Robotic Outdoor Guide", aimed to engage tourists in a fun exploration of outdoor activities. Similar to Obelix, FROG had to drive on uneven terrain and non-smooth surfaces. A combination of stereo vision and 2D laser was used for person detection and guidance.

Out of the previously described robots, none had been autonomously operating inside a crowded airport terminal, which – as described in Chapter 1 – can be a very complex and challenging indoor environment where people often walk at a fast pace. Most similar in its use-case to SPENCER is a mobile information kiosk, Robi, that project partner BlueBotics had previously deployed in 2013 at Geneva airport. While able to interact with humans and greet them at the arrivals area, it was operating at much slower speed than SPENCER, whose goal is to guide passengers with short connection times.

Social and socially-aware robots

Recently, there has been a stronger research focus on so-called social, or socially aware, robots. In addition to being able to sense humans in their surroundings, they are able to understand and/or express human emotions, understand social relationships, activities and norms and behave and react accordingly. A general overview and examples of many different social robots is given by Tzafestas (2015) and Belpaeme (2019).

The previously described MINERVA robot can be considered as an early type of social robot, as it was able to express facial emotions which led people to "clear the path much faster than reported by the Rhino team" (Thrun et al., 1999). Expression of emotions has also been studied with DARYL (Embgen et al., 2012; Becker-Asano et al., 2014), a companion robot called PARO the seal (Šabanović et al., 2013), and the semi-humanoid Pepper robot (Pandey et al., 2018).

Pepper is being used extensively in the MuMMER project (2016–2020), which aims to develop a humanoid robot "with the social intelligence to interact autonomously and naturally in the dynamic environments of a public shopping mall" (Foster et al., 2016). Its research objectives include audiovisual scene processing, social signal processing, high-level action selection and human-aware robot navigation. The Pepper robot weighs 28 kilograms, is 1.2 meters tall and has a maximum travel speed of around 0.6 m/s. It is equipped with a single quad-core CPU. Thus, it is smaller and much more lightweight, slower, and computationally less powerful than the SPENCER platform, which does all of its sensing onboard without relying on additional infrastructure. Two generations of social robots from the JackRabbot project (2015–2020) have been in use since 2015 to study different aspects of social navigation in crowded public environments such as malls, terminals or campuses. They have onboard sensing, but are significantly smaller and lighter than SPENCER. Instead, several social robots that have served as shopping assistants, in hospitals, schools, museums, elderly care and at trade fairs (Kanda et al., 2008; Kanda et al., 2012) relied on the availability of ubiquitous sensor networks.

The aim of the SPENCER project was to develop a socially-aware robot that learns normative social behaviors, recognizes social activities and relations, and utilizes this information to behave in a socially compliant way. Besides group detection (Chapter 3), these aspects of higher-level social understanding, reasoning and behavior modeling are not covered in this thesis, but are described partially in the works by Okal et al. (2014) and Okal and Arras (2016), Fiore et al. (2015), Ramírez, Khambhaita, et al. (2016) and Ramírez, Varni, et al. (2016), Joosse (2017) and Palmieri (2018). Because all navigation, perception, and higher-level reasoning tasks were to be executed onboard to make it *fully autonomous* – thus not requiring *e. g.* cloud infrastructure – the technical system requirements for the platform, described in the following, were high.

9.3 SPENCER hardware platform

The SPENCER robot is a 250 kg heavy, 1.93 meter tall, 80 $\times$ 81 cm wide differential-drive platform manufactured by BlueBotics, an industrial partner within the project. Figure 9.2 (left) shows a frontal view of the robot with its most important hardware components. For human-robot interaction, the head is controllable by a pan/tilt system and the eyes by a pan joint. In the front, a touchscreen and boarding pass reader have been integrated. The rear of the robot offers a mounting option for a small basket, to symbolize the robot's ability to carry hand luggage.

Around 120 kilograms of lead-acid batteries inside the lower base of the robot ensure by design sufficient physical stability through a low center of gravity, and provide up to 12 hours of runtime for the entire system with its 6 onboard computers:

- An embedded PowerPC controller for low-level motion execution, connected to the drive motors and safety loop, powered by the proprietary BlueBotics ANT system;

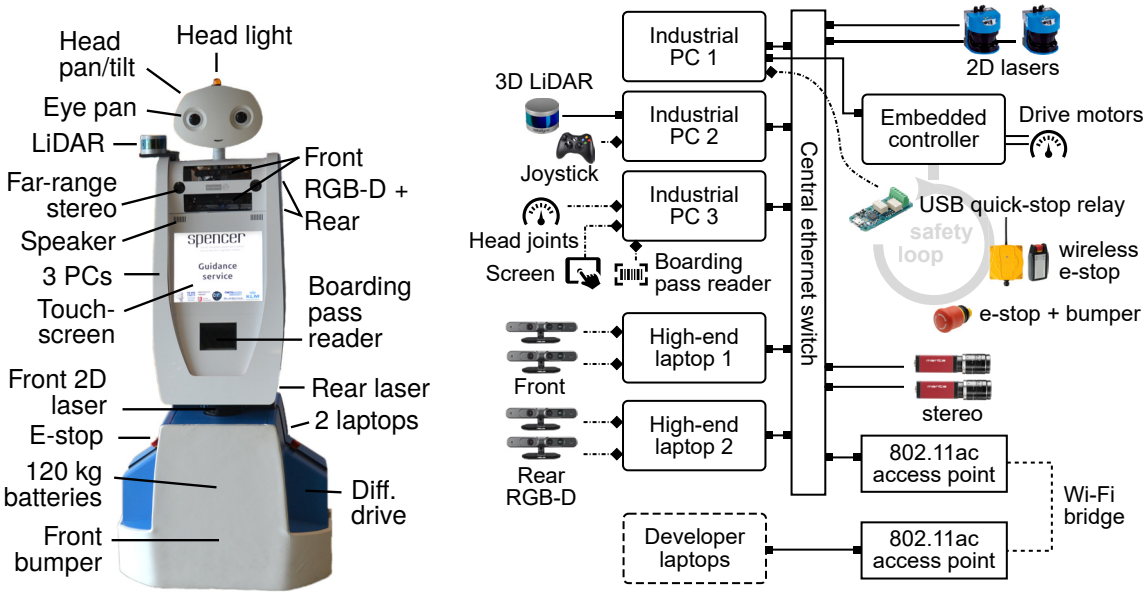


Figure 9.2: Hardware setup and architecture of the SPENCER robot. In the right figure, black squares denote Ethernet links and black diamonds USB connections.

- Two industrial PCs with Intel Core i7-3520 CPU;
- One industrial PC with Intel Core i3-2120 CPU;
- Two high-end laptops with Intel Core i7-4700MQ and Nvidia GTX 765M GPU.

Total power consumption of the entire system was up to 1000W. As shown in Figure 9.2 (right), the three PCs and the two laptops are interconnected through a central Gigabit Ethernet switch, while the embedded controller is directly connected to one of the PCs through a second Ethernet port. All computers are running Ubuntu 14.04 with ROS Indigo.

9.3.1 Sensor setup

The robot is equipped with the following exteroceptive sensors:

- Two 2D SICK LMS 500 laser range finders, mounted back-to-back at around 0.7 m height for 360 degree coverage at 35 Hz;
- Four Asus Xtion Pro Live RGB-D sensors, two in the front and two in the rear at around 1.6 m height, providing a horizontal field of view of around 57 degrees in front and rear directions, at a frequency of 30 Hz;
- A stereo camera rig of 0.4 m baseline for far-range perception, consisting of two AVT Manta cameras with 4.5 mm lenses (in the end not used in the project);
- A Velodyne VLP-16 LiDAR, mounted on the shoulder at 1.8 m height, solely for localization purposes. This sensor was initially not planned, but had to be added for reliable localization inside the highly dynamic airport environment.

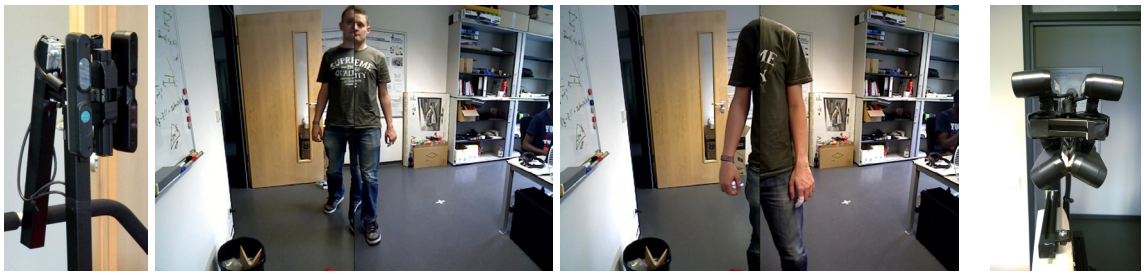


Figure 9.3: Initially planned vertical RGB-D sensor setup, as used for the *Rathausgasse* dataset (Appendix A), and resulting RGB images of a person at mid- and close-range. Right image shows the final sensor setup used for data recordings at the airport.

Choice of RGB-D sensors

One important decision in the design phase of the robot concerned the choice of RGB-D sensors. In early experiments with the mobile data recording platform (Appendix A), it was found that having more than one Asus sensor connected per USB 2.0 bus would often cause one of the sensors not to be detected. At the same time, Microsoft had just announced the new USB 3.0-based Kinect v2, with a larger horizontal field of view (86 vs. 57°) and higher depth and image resolution.

Because each Kinect v2 requires an extra 25W power supply, a GPU and around one CPU core for RGB-D processing and registration, it would not have been possible to replace all Asus sensors by Kinect v2 devices. Based upon qualitative insights from data recordings at the airport, it seemed that the time-of-flight-based Kinect v2 would be better suited for perception of people, whereas the structured-light Asus sensors are more useful for collision detection – especially if tilted downwards – due to a significantly lower noise amplitude at the ground plane, making it easier to segment obstacles (as described in Section 9.7.2); here, the depth measurements of the Kinect v2 would often fluctuate by $\pm$20 cm. For these reasons, we in the end decided to stick with the Asus sensors.

A Kinect v2 sensor was temporarily mounted in the robot's "neck" for data recordings at the airport, and experiments on human attributes recognition (see Chapter 7). In Chapter 5, an improved RGB-D human detector for the Kinect v2 has been presented and evaluated in an intralogistics scenario of the subsequent ILIAD project.

Sensor placement

For the RGB-D sensors, initially a vertical arrangement was favored, similar to the one shown in Figure 9.3 (left) that was used for data recordings in the *Rathausgasse* dataset. Here, the vertical arrangement of two Asus sensors had led to a considerable increase in horizontal field of view.

However, even when two sensors are mounted as closely as possible to each other, a vertical setup would result in a large blind spot at close range ($<$1m)[1], as shown in the middle of Figure 9.3.

[1] If one were instead to reduce the angular rotation between the two sensors, the combined field of view would not be much larger than that of a single sensor, which undermines the original motivation.

Because in SPENCER, humans closely approach the robot and interact with its touchscreen, this sensor setup was not deemed to be optimal. Instead, the setup on the right-hand side was used for initial data recordings at the airport, and subsequently also realized on the SPENCER robot platform. As visible in Figure 9.8 (c) on page 202, human upper bodies stay intact with this horizontal arrangement.

The 2D laser range finders (SICK LMS 500) in SPENCER are mounted back-to-back at around 70 cm height above ground. This is an unusual configuration, as especially safety-certified sensors such as the S300 (used on the forklifts in the ILIAD projects) are normally mounted close to the floor at a height of less than 20 cm. This would not have been possible on the SPENCER robot, with its 120 kg of batteries in the base, without integrating the sensors into the bumpers, relaxing the constraint of having both sensors back-to-back, and subsequently creating a large blind spot on the robot's sides. Another concern was that the bumpers would induce vibration into the sensors while driving.

Because at 70 cm height, the 2D lasers cannot perceive smaller obstacles such as shoes, backpacks, or small children, it became necessary to integrate an RGB-D-based collision checking module, described in Section 9.7.2. The mounting height, mandated by the size and location of the batteries, also turned out to be a difficult height for 2D laser-based person detection, because at this height, depending on how tall the person is, the sensors can either see the upper legs, hips, waist, or hands of a person (possibly carrying a shopping bag).

9.4 SPENCER software architecture

Figure 9.4 provides an overview of the ROS-based software architecture of SPENCER. While previous chapters have already discussed the components, shaded in different colors, for group tracking (Chapter 3), multi-modal people tracking (Chapter 4) using the 2D laser detectors (Arras et al., 2007) and the upper-body detector (Mitzel et al., 2012), and human attribute recognition (Chapter 7), the remaining software components that are highlighted in bold will be described in the following sections. The module for head pose estimation by Beyer et al. (2015) and the rough body pose estimation by Rafi et al. (2016) are not part of this thesis. Like the human attribute classifier from Chapter 7, they have been tested on the robot, but were inactive during guidance missions due to limited computational resources.

It can be seen from Figure 9.4 that the software components have been distributed across computers primarily by functionality (navigation, interaction, perception). However, another important factor that had to be taken into account, was to avoid sending large amounts of sensor data over the network at high frequency, in particular RGB-D point clouds. Therefore, modules that consume RGB-D data had to be placed on the same laptop to which the corresponding sensor is connected.

9.5 Additional software components for SPENCER

In addition to the human tracking framework described in the previous chapter, the following SPENCER software components have been developed (to a large part) by the author.

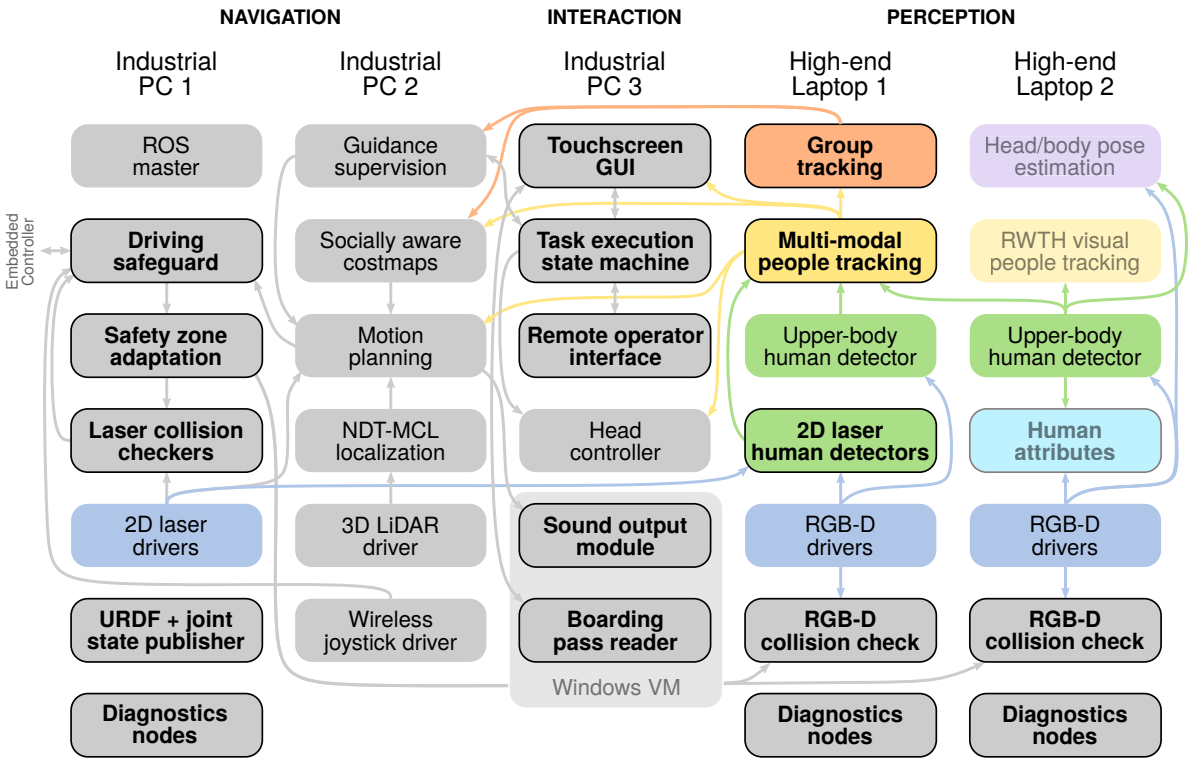


Figure 9.4: Software architecture of SPENCER. All modules communicate with each other over Ethernet via ROS; arrows indicate the most important connections. Colors encode components essential for perception of humans. The faded ones were not enabled during live guidance missions. Modules highlighted **in bold** were developed during this thesis; several of the grey ones are described in this chapter.

Most of the software components described in this section have not been part of the original research plan for SPENCER. However, they were either required to conduct experiments safely and efficiently, needed for a successful final demonstration at the airport or for studies on aspects of human-robot interaction (Joosse, 2017), or were prerequisites for the deployment of the human detection and tracking framework described in Chapter 8.

9.5.1 URDF model and transform tree

URDF (Unified Robot Description Format) is a universal representation of robot models that can be used for visualization purposes (*e. g.* in Rviz), and if enhanced with simulated sensors and actuators, for simulation (*e. g.* in Gazebo). It is further used in ROS by `robot_state_publisher` and `joint_state_publisher` to publish the transform (TF) tree of the robot, which can be used by any ROS node to transform for instance between local sensor, robot and world coordinate frames as shown in Figure 9.5 (right).

A visual robot model in Rviz provides a sense of scale and orientation, and is helpful when trying to interpret raw sensor data *e. g.* from RGB-D sensors. The visual model of SPENCER in

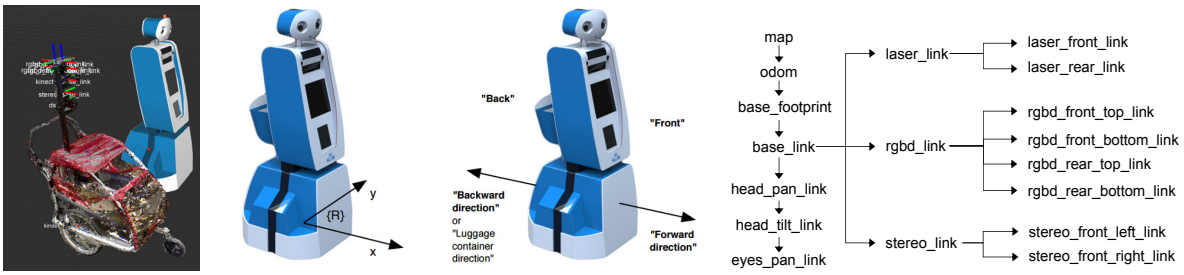


Figure 9.5: *Left:* URDF models of the SPENCER robot platform and the mobile data recording platform (described in Appendix A). *Middle:* Robot coordinate frame. *Right:* Simplified version of the TF hierarchy used on SPENCER.

Figure 9.5 (left) was created from a CAD model provided by the robot manufacturer, by removing its inner parts using FreeCAD and conversion into a simplified Collada polygon mesh using MeshLab. For the mobile data recording platform (see Appendix A), a point cloud registration software was used together with an RGB-D sensor that was rotated around the platform at different azimuths and elevations to obtain a dense representation from multiple viewpoints. A transform hierarchy of similar structure and naming conventions to that of the real robot has been set up, to ensure that the same perception algorithms can be run on recorded and live data without modifications.

In the TF tree, resulting *static* transforms are published at high rate (1000 Hz) or as latched ROS topics, whereas *non-static* transforms (*e. g.* for SPENCER's head joints) are published at 30 Hz. We found it important not to have too many of these, as they slow down `rospy` nodes significantly (with a single node easily using up one CPU core).

9.5.2 Extrinsic sensor calibration toolbox

A reoccurring problem in SPENCER and later on ILIAD has been the extrinsic calibration of the multi-sensor setups of these platforms. While in SPENCER, especially the RGB-D sensors would often shift during transports of the robot, in ILIAD the sensor arrangement had to be modified multiple times when *e. g.* switching the five forklifts from manual mode (during data collection) to autonomous operation. However, a precise calibration is important for detection-to-detection fusion (Chapter 4) to function correctly, where the goal was to obtain person centroid offsets of less than 10 cm across all sensors.

In SPENCER, we therefore came up with a relatively simple, but efficient strategy for calibrating all onboard range sensors (2D laser, 3D lidar, RGB-D) relative to one sensor of known pose with reference to the robot's base[2]. By modifying the URDF model from the previous section to include a "sensor calibration assembly" macro for every sensor, the sensor's fixed 6D joint essentially gets replaced by a non-static transform, that we split up for practical reasons into 9 individual components for pre-translation, rotation, and post-translation $(x, y, z, r, p, y, x', y', z')$. In the

[2]In practice, the position of the 2D lasers is often known from data sheets or can be measured using a rule, as they are usually rigidly mounted to the robot base at an upright angle.

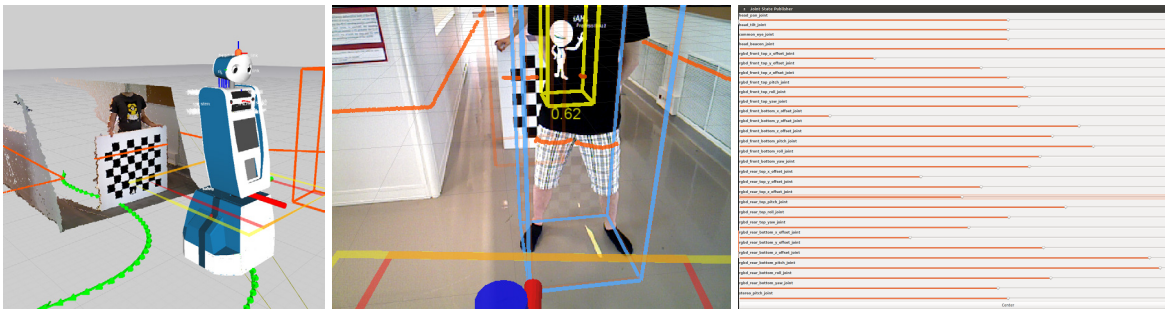


Figure 9.6: Extrinsic calibration by aligning the point clouds of multiple sensors (here: 2D laser and RGB-D). The middle shows a projection of 2D laser into the RGB image (and an interesting false positive human detection from a HOG-based detector, in yellow). The right screenshot shows the slider-based GUI used to adjust the sensor poses based upon `joint_state_publisher`.

URDF, the translation components are implemented as 1D prismatic joints, and the rotation components as 1D revolute joints. Applying these individual transforms sequentially from left to right, it becomes possibly to adjust every 1D transform value using a slider-based GUI as shown in Figure 9.6 (right), where for instance one slider represents the yaw value of the frontal upper RGB-D sensor. The resulting changes are done *online* and immediately reflected in Rviz (Figure 9.6, left and center image) and all other systems of the robot.

After calibration is completed, using TF and a Python script, the 9D transform vector is collapsed again into a fixed joint of 6D representation (x'', y'', z'', r, p, y) and saved into a calibration file, such that the resulting static transform can again be published by a latched publisher.

Recently, progress has also been made on the development of automatic multi-sensor calibration techniques. For instance, Della Corte et al. (2019) propose a unified motion-based calibration of mobile multi-sensor platforms with time delay estimation. In ILIAD, both approaches have been successfully used in tandem (*e. g.* using the manual method to fix a remaining z offset after automatic calibration, or to provide an initial rough estimate). On SPENCER, a sufficiently accurate calibration for multi-modal human detection took around 10 minutes with the proposed manual approach, while on ILIAD around 5 minutes were required per robot.

9.5.3 Human-robot interaction

Graphical user interface

Already the very first guide robots, such as RHINO (Burgard et al., 1998), possessed a dedicated and multi-modal user interface, “integrating text, graphics, pre-recorded speech and sound”. Back then, four colored buttons were used to enable user inputs. SPENCER is equipped with a touchscreen for this purpose. Its graphical user interface has been implemented using pyQt. The Qt framework offers template and style sheet mechanisms to re-use layouts and designs across several UI pages, while allowing for easy ROS integration through `rospy`.

The GUI has been designed with the following user experience-related criteria in mind:

- Choice of high-contrast colors (black font on white background) to ensure readability under varying lighting conditions including direct sunlight (Figure 9.1);
- Use of large fonts and buttons to ensure readability and accessibility while in motion;
- Consistent design over all pages through use of templates and style sheets;
- Clear communication of the current system status;
- Anticipation, and proper handling, of unexpected user behavior;
- Ability to interrupt the current mission at any time, leaving the user in control;
- Not relying on speech recognition or synthesis because of high ambient noise and many spoken languages at the airport;
- Clearly indicating that this is an ongoing research project (by prominently showing the university partners' logos and the full project title).

Figure 9.7 shows selected parts of the resulting SPENCER GUI. For demonstration purposes, the GUI has only been implemented in English language, though in final user studies participants expressed the need for additional languages (Joosse, 2017). The GUI has been integrated with the robot's SMACH-based task execution system (Bohren et al., 2010), by exposing its state and reacting to external state transitions via ROS topics.

During guidance missions, the robot drives backwards (with the head facing forward) such that guided passengers can track the progress on screen, and pause or cancel guidance through a large on-screen button. Progress in percent, remaining distance and time are computed using a symbolic graph of the terminal within the guidance supervision module. A progress indicator is an important feature: During a pre-study with 72 participants by Joosse (2017, p. 135), in which this progress display had not yet been activated, participants were asked about possible improvements of the robot. 43% of the comments suggested to provide more feedback to the user, *"for example route information through a map and the progress of the tour"*.

Via a "hidden" button in one of the screen corners, a diagnostics screen (Section 9.5.4) as well as a visual representation of the robot's safety zones (Section 9.7.2) along with a button to reset the robot's software emergency stop could be displayed on-screen. These turned out to be a very useful features for debugging.

Boarding pass reader integration

SPENCER's built-in boarding pass reader is controlled through a Windows API running inside a virtual machine on the interaction PC (see Figure 9.4). A background service that initiates the scan process has been implemented in C#, and connected to the ROS system using a .NET ROS client via `rosbridge`. There, a ROS node listens to incoming string representations of scanned bar codes in PDF417 or Aztec Code format, parses them using the BCBP protocol (IATA resolution 792), before providing results to the GUI.

In the final user studies by Joosse (2017, p. 149), one participant praised the overall quite intuitive process of starting a guidance mission as *"you just have to keep following, you hold up your boarding pass, it reads it, and then it says just follow me. Three steps. Simple."*. However,

(a) Welcome screen

(b) Switch to start screen when user approaches

(c) Scanning boarding pass

(d) Starting guidance to gate

(e) Display of remaining distance and time

(f) Progress display seen from larger distance

Figure 9.7: Graphical user interface on the robot's touchscreen developed by the author for human-robot interaction. Pictures by KLM

other participants experienced issues while scanning the boarding passes, most likely related to sunlight interference because no sun shield was installed.

Sound and speech output module

Inside the Windows VM, a sound and speech synthesis module has been implemented. One of its purposes is to emit warning sounds when the robot is manually controlled via wireless joystick and its safety features from Section 9.7.2 are overridden by hand (*e. g.* to fit the robot into a narrow elevator). Also, as described by Palmieri (2018), the human-aware motion planner

would trigger an “Excuse me!” phrase when the robot is blocked by humans that were detected by the human detection and tracking framework. In the user study of Joosse (2017, p. 149) with 12 participants, *“the majority of participants (10) indicated auditory feedback- and communication could, or should, be improved”*, which, however, was not a planned research focus in SPENCER.

Remote operator interface

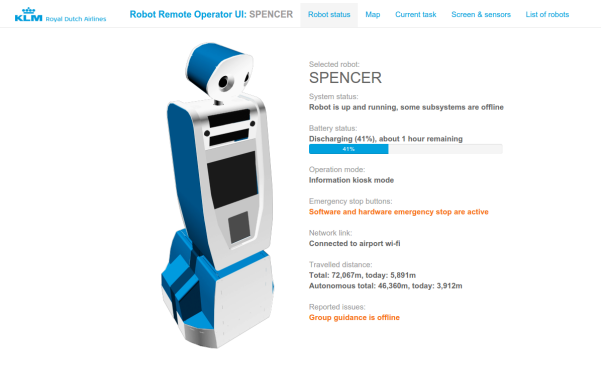
To demonstrate the system to interested stakeholders, who might want to operate a fleet of such robots, a web-based remote operator interface has been developed. Such web interfaces were already used in earlier systems like RHINO (Burgard et al., 1998). As depicted in Figure 9.8, the SPENCER remote operator interface, based upon `roslibjs` and `rosbridge`, provides an overall system status summary with battery level and driven kilometers, the current operating mode, whether important subsystems are offline, and the status of the emergency stops. In (b), a schematic live map, the robot’s current position within the terminal, its destination, remaining time and closest point of interest are shown. These are retrieved from the symbolic map of the guidance supervisor. Such a map display could *e. g.* be provided to a gate agent, to decide whether to wait for remaining passengers being guided by the robot, and was noted as a possible key benefit of such a robot in an end user workshop.

A camera stream from all four RGB-D sensors at low frame rate, and an image of the current touchscreen content in Figure 9.8 (c) enable a remote operator to see what is happening on screen and in the robot’s surroundings, without having to physically reach out to the robot. In (d), the current high-level task and the present sub task are shown (*e. g.* waiting for a boarding pass to be scanned). Furthermore, there exist buttons for switching between operating modes, setting or releasing (only) the software emergency stop, resetting the current scenario, or diverting the robot to a different gate.

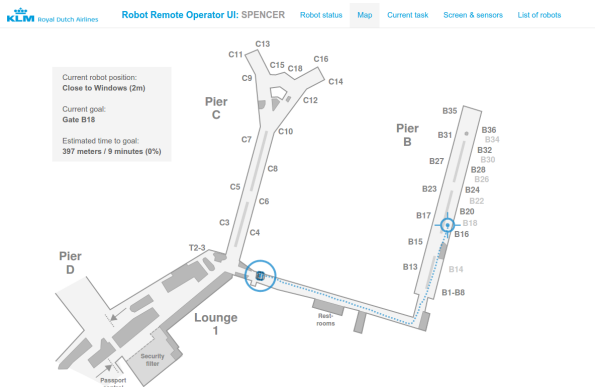
9.5.4 Diagnostics instrumentation

In a complex system like the SPENCER robot with its 5 onboard PCs and over 30 different software modules, many components can fail which would prevent a successful guidance mission. Even for a trained person, it can be hard to troubleshoot such issues with the large number of log messages output at every second. The difficult working conditions during an integration week, or while testing in a public environment like the airport, as indicated by Figure 9.9 (left and center) are a contributing factor.

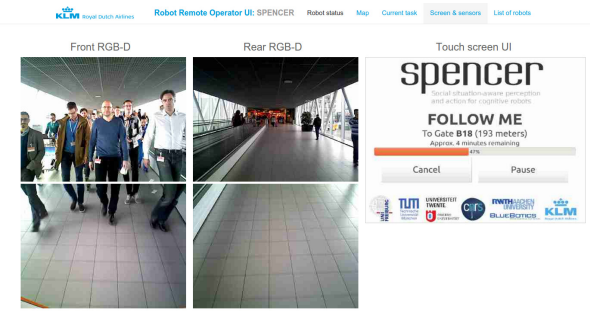
We therefore found it to be very helpful to have a central dashboard that aggregates and displays all information at a single glance. Our solution based upon `rqt` is shown in Figure 9.9 (right): In a nominal operating state of the robot, the GUI would not show any red errors or warning signs. In the depicted case, we can see that one laptop battery is critically low (because it was not correctly plugged into the robot’s DC supply), that the other laptop’s hard drive is almost full (which would prevent further data recording with the front RGB-D sensors), that motion planning is not working correctly (velocity command time out), that localization is not running (transform timeout), that RGB-D collision checking is nonoperational, but all computers’ clocks



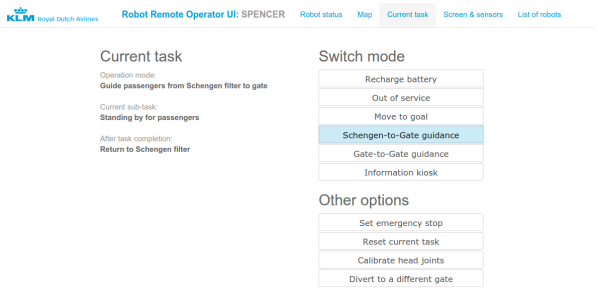
(a) Status overview



(b) Live map with robot position and progress



(c) Camera stream and touchscreen visualization



(d) Switching operation modes

Figure 9.8: Remote operator interface developed for end user demonstration.

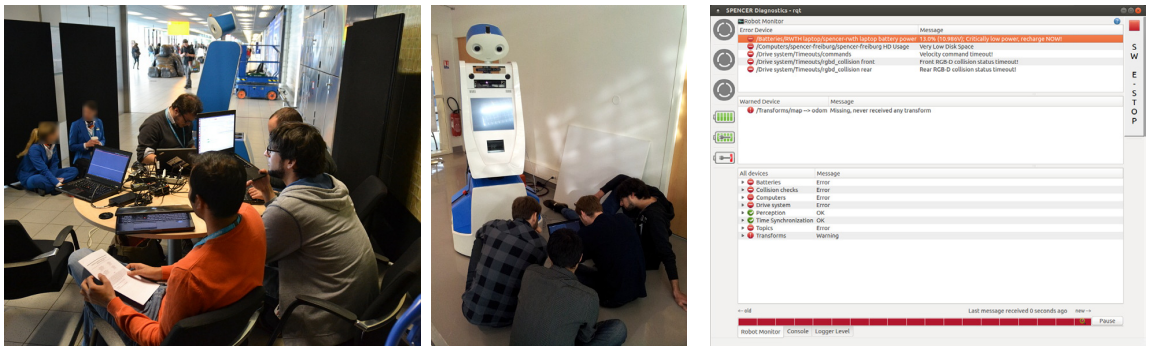


Figure 9.9: A graphical user interface for robot diagnostics based upon `rqt`, as shown on the right, proved to be highly useful for quick troubleshooting and system monitoring in the difficult working conditions during integration.

are synchronized correctly. We can also see that none of the emergency stops (either in hardware, or in software) is engaged; via a button, we can trigger or release the software e-stop.

In the background, this is achieved through a number of diagnostics nodes which monitor important ROS topics (such as human tracks), as well as their latencies and publish rates that are compared against preset thresholds. Important components, such as the collision checkers, also send a regular heartbeat message of a predefined format. Additional ROS nodes in the background monitor the battery status of the robot, the laptops' batteries, as well as CPU and GPU loads, thermal limits and disk capacity.

9.6 Integration of human detection and tracking

In this section, we outline how the human detection and tracking framework from Chapter 8 has been integrated with the other SPENCER subsystems.

Motion planning integration

The integration with motion planning happens in three ways. First, tracked humans and groups are fed into a special cost map layer of the ROS navigation stack[3]. This social cost layer models three different types of socially normative navigation behaviors *(polite, sociable* and *rude)*. As described by Okal and Arras (2016), they have been learned in a PedSim[4] environment based upon the social force model (Helbing et al., 1995) using Bayesian inverse reinforcement learning. An exemplary situation for the *sociable* behavior in simulation, and the resulting cost map, is shown in Figure 9.10 (left).

The social cost maps are subsequently used to adapt the elastic band of a socially-aware elastic band local planner as described in the PhD thesis by Palmieri (2018, pp. 142ff.). The local planner is coupled with a multi-hypothesis global path planner that dynamically selects between a socially-aware path, and a static-world path. For the latter, information from human tracking is used to trigger an HRI event ("Excuse me!" phrase) if persons are obstructing the path. Lastly, tracked humans are also erased from the (static) obstacle layer by filtering their associated 2D laser and RGB-D points.

For the integration with motion planning, only the current human positions have been utilized, but no predictions. The reason for this is that when directly using predictions from the Kalman filter – which has been tuned for robust data association, not smooth predictions –, predictions would oscillate too wildly especially in crowded areas of the airport terminal. In experiments that used these predictions to scale the translational velocity component of the socially-aware elastic band planner, Palmieri (2018, p. 156) observed reduced legibility. This indicates a need for further research on human motion prediction – see *e. g.* the survey by Rudenko et al. (2020).

[3] http://wiki.ros.org/navigation
[4] https://github.com/srl-freiburg/pedsim_ros

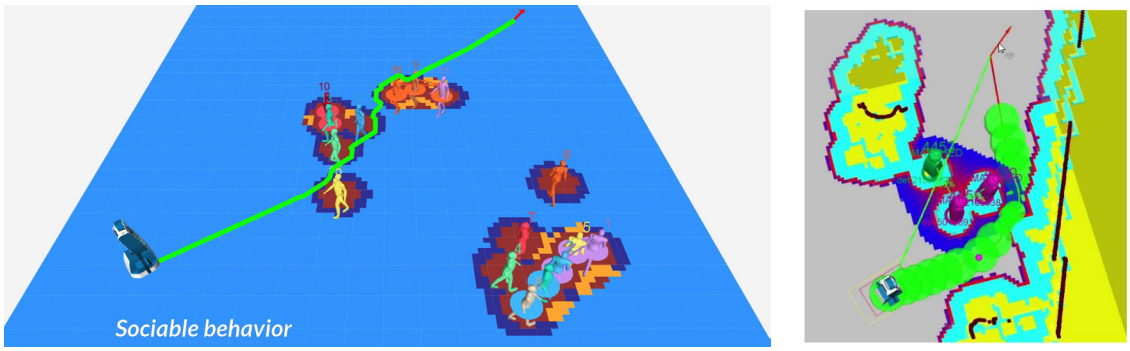


Figure 9.10: Integration of human detection, tracking and group tracking with the motion planning components in SPENCER. *Left:* Social costmap learned using Bayesian IRL, here shown in simulation (Source: Okal and Arras, 2016). *Right:* A socially-aware motion planner that incorporates knowledge about tracked human positions (Source: Palmieri, 2018).

Integration with the guidance supervision module

The main demonstration use-case of SPENCER, the person guidance scenario, strongly depends on information from human and group tracking. A ROS-based guidance supervision module based upon hierarchical MOMDPs (Mixed Observability Markov Decision Processes) has been implemented by Fiore et al. (2015). It expects centroids from group tracking (Chapter 3) as a spatial representation of the input data. When guidance begins, the group that is closest to the robot's touchscreen is used to initialize the main guidance MOMDP, which is composed of a "speed adaptation" and a "suspend" model (*i. e.* the robot waiting for the group).

Speed adaptation based upon information from human tracking is an important aspect, as can for instance be seen from the right picture of Figure 1.3 (a) on page 10, where a guided passenger is tying her shoes: With regard to a pre-study conducted at the University of Freiburg, where speed adaptation had not yet been implemented, Joosse (2017, p. 136) reports that *"20.8% indicated the robot should be equipped with some form of speed adaptation, a feature not only related to motion planning, but requiring the integration of various components (such as people tracking, and potentially re-identification)"*.

HRI integration

Human tracking information is also used inside the graphical user interface. A trapezoid-shaped area in front of the robot's touchscreen is constantly monitored for human presence by an HRI component to show a "Welcome!" message. This should encourage people to interact with the robot while it is idle.

Human tracking is further used to make the robot turn its head towards passengers that are approaching the touchscreen, to make the robot appear more "alive"; secondly, during guidance missions the robot occasionally looks at persons walking towards it, to signal that it is aware of their presence. Our subjective impression was that it makes people come closer to the robot, in comparison to a static head. Further human-aware head behaviors, including looking back at guided passengers from time to time, have been studied in the PhD thesis by Joosse (2017).

9.7 Safety considerations

One important aspect that has so far not been considered is safety. While the project's original research plan did not include any safety-related tasks, the project's reviewers had requested a supplementary safety audit report during the first review meeting. This was prompted by the surprisingly large mass of the robot, after its specification and assembly had been completed around one year into the project. SPENCER weighs 250 kg and can move at up to 1.8 m/s.

Improving practical robot safety in crowded environments

Safety guidelines for personal care robots, including mobile servant robots such as SPENCER, are documented in ISO 13482:2014. This norm came into effect one year into the project, after the robot's hardware platform had been finished. As a research prototype, the robot did not receive a dedicated safety certification. The safety concept therefore involved a dedicated, well-trained human safety operator with a safety-certified wireless emergency stop device to always be in close vicinity of the robot.

Still, given that SPENCER would operate in a public, uncontrolled, crowded environment, we wanted to learn more about possible failure modes and try to improve the robot's autonomous safety systems as much as possible within reasonable effort. To address these concerns, the following measures have been implemented:

1. An FMEA (Failure Mode and Effects Analysis) was conducted together with the robot manufacturer, highlighting the most hazardous, highest-risk failure modes.
2. An analysis of bumper effectiveness has been performed by the robot manufacturer.
3. Subsequently, a laser-based and an RGB-D-based *collision checker* have been implemented as ROS components (as part of this thesis), acting as a "virtual bumper". They operate on *speed-adaptive safety zones*.
4. The collision checkers have been integrated with a ROS-based *driving safeguard*, which slows down and then stops the robot when a collision is imminent, while also incorporating watchdog timers for other essential ROS components such as motion planning.
5. Braking tests have been performed to measure resulting braking distances at various robot speeds. Subsequently, the deceleration ramp of the robot's motor controllers was adjusted to enable faster deceleration.
6. A *quick-stop relay* has been integrated into the safety loop to enable the driving safeguard software to stop the robot as quickly as possible via a USB interface.

Results of the FMEA can be found in the public extra SPENCER deliverable, D6.6. In the next sections, we focus on a description of the ROS components developed by the author to make the robot safer *in practice* by reducing the risk of human injury and property damage. Figure 9.11 gives an overview of the robot's safety-relevant hardware and software components. In the right diagram, black arrows indicate data flow and red color indicates the hardware-based safety loop, which, if opened, immediately brings the robot to a full stop by triggering a quick stop on the drive motor amplifiers.

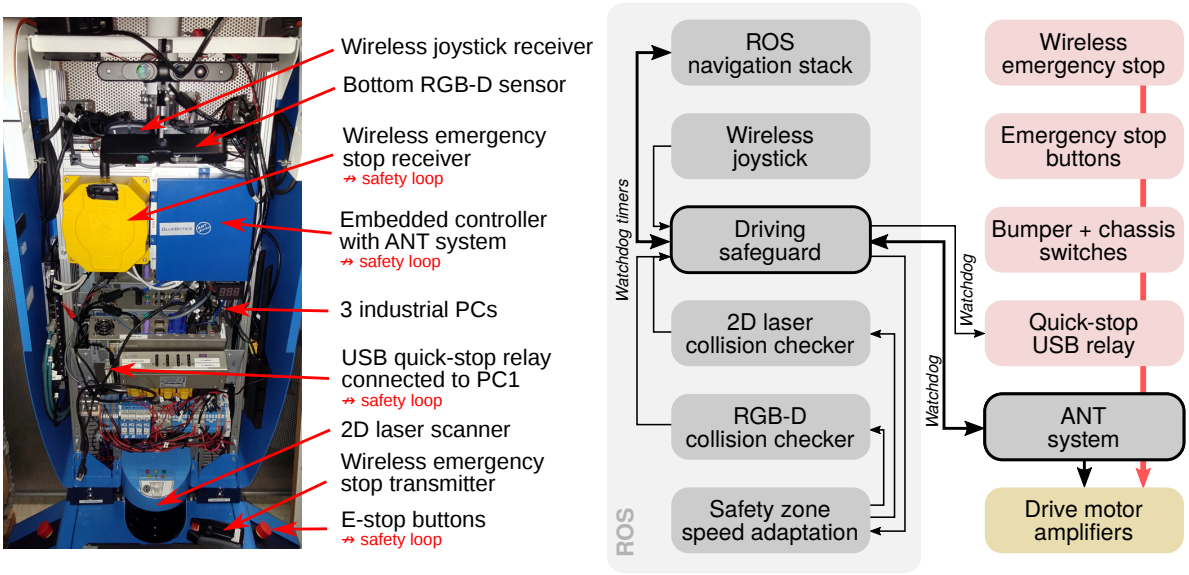


Figure 9.11: Overview of safety-relevant hardware and software components of SPENCER. In the right picture, red color symbolizes the safety loop. The quick-stop USB relay was added at a later point to significantly reduce braking distances when needed, and adds a layer of redundancy.

9.7.1 Driving safeguard

The driving safeguards acts as an interface between the ROS system and the embedded controller that runs the ANT system (Tomatis et al., 2003; Tomatis, 2011) on a PowerPC platform. ANT is based upon XO/2, a deadline-driven hard real-time operating system. It provides advanced autonomous navigation functionality for automating different types of robots. In SPENCER, it was decided not to use the built-in planning, localization, and collision avoidance capabilities of ANT, to leave space for additional research and experiments on these topics using dedicated ROS components.

Communication between the ROS-based driving safeguard and ANT occurs via the *LOS RPC (remote procedure call) protocol* over Ethernet using a C++ API. In its main loop, the driving safeguard publishes odometry and status information from the low-level system such that other ROS components like the ROS navigation stack can access it. This happens at a frequency of around 20 Hz and also includes a TF transform from odometry frame to the robot's base frame. At the same frequency, velocity commands (v, ω) from the navigation stack are forwarded to the embedded system.

Since ROS and the underlying Linux operating system provide no real-time guarantees – which is a problem when operating a robot in a safety-critical public environment – the driving safeguard at this stage introduces the following extra safety measures:

- Velocity messages from the navigation stack and the wireless joysticks are monitored by a watchdog timer. If no new message is received within 100 milliseconds, *e. g.* due to a crash of a ROS node, a quick-stop is triggered.
- Angular and linear velocity limits are enforced, which depend on if the robot is in free space, close to an obstacle, or too close (see next subsection).
- If a collision is imminent, or some other critical error has occurred, a quick-stop is triggered both by sending a zero velocity command to ANT, and opening the hardware safety loop over a USB relay. This mechanism is also used to implement a software-based emergency stop that can be initiated by any other ROS component (or from the joystick) if an abnormal situation is detected.
- A heuristic checks if the wheel encoder values reported by the embedded controller are changing as expected for the given (v, ω), within some tolerance. This is a failsafe to detect loose wheel encoder cabling, which we encountered once.

The watchdog timers shown in Figure 9.11 (right), via transitivity, also cover cases where for instance a sensor driver is malfunctioning, causing the collision checkers to stop sending messages to the driving safeguard. In the unlikely case that the driving safeguard itself would crash or fail to respond, both the ANT system and the microcontroller of the quick-stop USB relay have also been configured with a 100 ms watchdog timer, both of which would again bring the robot to a full stop.

For testing the SPENCER software stack in simulation, a simulated version of the ANT system and the safety loop have been implemented, such that the same driving safeguard and collision checking code can be used in Gazebo.

9.7.2 Speed-adaptive safety zones for collision avoidance

With its approximately 3 cm thick foam layer, SPENCER's front and rear bumpers have theoretically and practically been shown by the robot manufacturer to be safe up to a velocity of 0.28 m/s. This was insufficient to meet the research goals in SPENCER, which included efficient navigation at fast walking human speeds up to 1.5 m/s; at that speed, the foam on the bumper would have needed to be around 1 meter thick.

We therefore implemented a *virtual bumper* using the robot's onboard 2D laser and RGB-D sensors[5]. Our virtual bumper, shown in Figure 9.12, is speed-adaptive and therefore does not restrict the robot too much when navigating in very tight spaces. Though developed independently, our speed-adaptive safety zones partially implement the guidelines from ISO 13482:2014 that came into effect during the project: In our implementation, the "protective stop space" from the norm is called *error zone*, while the "safeguarded space" corresponds to our *warning zone*. In the warning zone, v and ω are limited to 0.5 m/s and 0.8 rad/s (where the smaller of the two components is scaled appropriately to maintain curvature). In the error zone, no motion is allowed.

[5] 3D lidar was not used because the sensor was added later in the project solely for localization purposes.

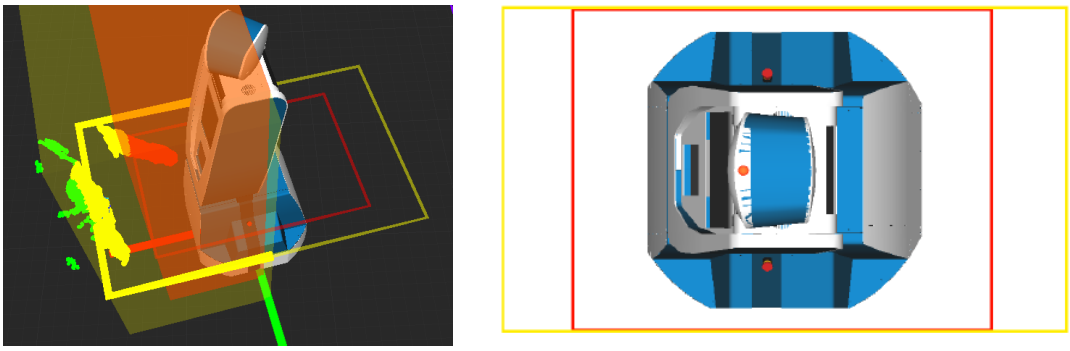


Figure 9.12: *Left:* A person, visualized by the point cloud, violates the front safety zones of the robot: For RGB-D, both the yellow warning volume and the red error volume are violated. For 2D laser as visualized by the safety rectangles, only the warning zone is triggered because the person's protruding arm is not visible to the 2D laser. Note that the RGB-D sensors cannot actually observe the full safety volume, due to their limited horizontal field of view of 57°. *Right:* Top view of the safety zones at 0.0 m/s. The extra 0.25 m safety margin covers people's shoes, which cannot be observed by sensors.

To define the velocity-dependent extents of the safety zones, we performed a series of braking tests with the robot, described in the next subsection. Depending on if the robot is going forwards or backwards, only the corresponding safety zone in driving direction is activated. During tight or on-the-spot turns, both zones are taken into account. Unless overridden by a special button, which triggers an audible alert, this also happens when a human operator is steering the robot via joystick, which greatly simplifies the handling of the robot and has prevented several accidents.

This low-level collision avoidance module does not process information from human detection and tracking, which, as a research-level system, might fail in some cases. All safety margins have been experimentally defined under the assumption that the human is cooperative, *i. e.* does not intentionally run into the robot.

2D laser collision detection

Collision detection using the 2D laser scanners (at 0.7 m height) runs at the sensor rate of 35 Hz. For each incoming laser scan, laser points are first transformed from sensor frame into a local coordinate frame at the center of the speed-adaptive safety rectangles. Since this transform is static, it is looked up once at startup using TF (Section 9.5.1). As 2D laser echoes are very robust against noise, an intrusion into the safety warning or error zone can be detected with as few as 3 points. We did not use the protective field feature provided by many 2D safety laser scanners in hardware, to be more flexible with our software-based implementation in crowded scenarios.

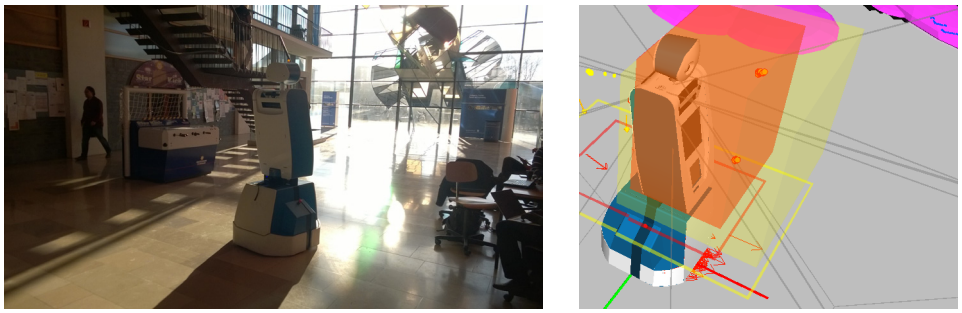


Figure 9.13: The left picture shows a challenging lighting situation at University Building 101 in Freiburg, where direct sunlight from the front causes infrared interference with the front RGB-D sensors. This leads to false positives in the RGB-D collision detector, visualized in the center of the right picture by yellow and red dots, triggering the safety warning and error zones, and erroneously bringing the robot to a full stop.

RGB-D collision detection[6]

Due to the mounting height of the 2D laser scanners, they are unable to detect obstacles close to the ground (such as infants, backpacks) or higher up (like a stretched out arm in front of the robot). It was therefore decided to re-purpose the lower RGB-D sensors for collision checking.

RGB-D collision checking also operates at sensor rate (30 Hz). The sensors have been tilted downwards to be able to observe the entire area in front of the bumper. For real-time performance, it was necessary to subsample the RGB-D point clouds randomly by a factor of 0.1, which delivered better results than a voxel-grid subsampling with fixed voxel size. This way, we achieve a mean latency of around 50 ms. We filter out all points within ± 10 cm of the ground plane, assuming a good extrinsic calibration using the method from Section 9.5.2.

To deal with randomly occurring artifacts (flying pixels), we average the number of points in the warning and error zones over five consecutive frames, and report an intrusion at a threshold of over 25 points. Because we use the arithmetic mean, tiny artifacts get filtered out, while large objects are still detected immediately, without inducing extra latency. In experiments, this threshold was shown to be sufficiently low for detection of a small backpack, but not for a coffee cup-sized object.

While we initially also included the upper RGB-D sensors to detect hanging structures, they were later excluded from collision checking due to sunlight interference issues (Figure 9.13 and Figure 9.14, left). As shown in Figure 9.14 (right), certain metal surfaces and advertising boards are not detected in RGB-D , but in most cases covered by 2D laser. Large glass surfaces were problematic for both RGB-D and 2D laser and had to be manually annotated in the map.

[6]The RGB-D collision checking module is joint work with Tomasz Kucner from Örebro University.

Figure 9.14: Sensing challenges for SPENCER due to difficult surfaces. *Left:* Shiny floor with sparkling particles causes locally clustered false positive readings of the RGB-D sensors (appearing randomly at around 1.5m height). *Right:* Glass side walls of moving sidewalks are not detected by any sensor. Metal surfaces are sometimes not seen, depending on incident angle. Back-lit advertising boards cause problems for structured-light RGB-D.

9.7.3 Analysis of braking performance

During first tests of the SPENCER platform at higher velocities up to 1.5 m/s, braking distances appeared rather long. Initial experiments with a stopwatch and rule seemed to confirm this observation. To obtain more accurate results, timing instrumentation has been added to the *driving safeguard* to measure latencies between sending a deceleration command, start of deceleration, and coming to a standstill.

As suspected, and visible in the plots in Figure 9.15, there was an around 0.6 second delay between triggering the safety warning zone (which should immediately lead to a deceleration), and an actual speed reduction; in fact, the robot kept accelerating. Only in the leftmost case of the left plot, where the robot had already reached the maximum velocity limit set at that time, the robot decelerated instantaneously.

Further investigation revealed that this was caused by a too slowly responding PID controller in the drive motor amplifiers, when only a speed reduction, but no “stop motion” command had been issued. Still, after parameter tuning through the manufacturer, further experiments revealed that the best possible deceleration could only be obtained by opening the safety loop using the wireless emergency stop[7]; this activates a dedicated “quick stop” velocity profile in the amplifiers, which is unsuitable for normal operation and could not be triggered from the ROS-based driving safeguard by any means.

Quick-stop relay

As a workaround, we therefore added an additional relay[8] into the safety loop, which can directly be controlled by the driving safeguard software via USB. The relay module, which is controlled

[7] Yielding, for instance, 0.68 m instead of 1.2 m braking distance at 1.5 m/s.
[8] http://www.yoctopuce.com/EN/products/usb-actuators/yocto-relay

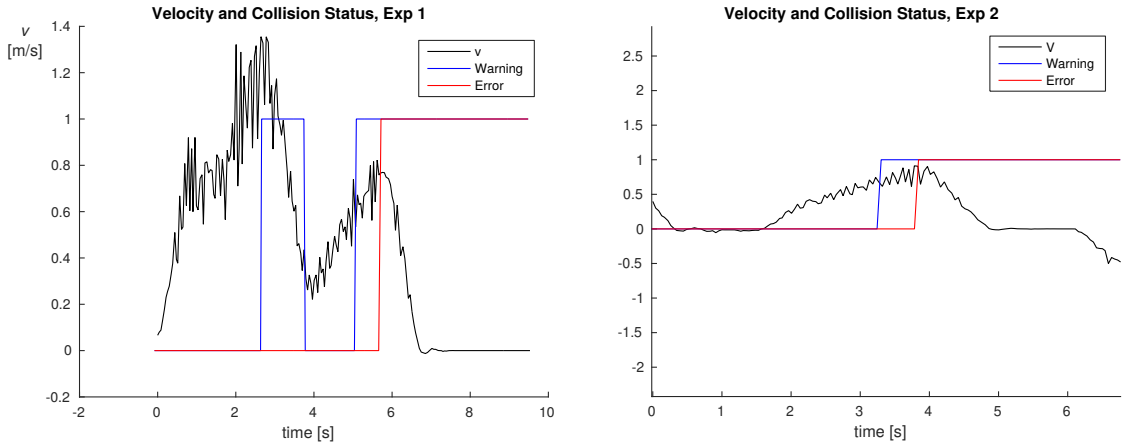


Figure 9.15: Measurements of how quickly the robot starts to decelerate and brake upon collision warning and error, respectively. In the second acceleration phase of the first experiment, and in the second experiment, the robot was still accelerating and had not yet reached the speed limit; in this case, we observed a high latency in deceleration apparently due to smoothing in the low-level PID controller of the embedded control unit.

by a microcontroller and in fact includes two relays on one circuit board, has been integrated into the safety loop with both relays connected in series for redundancy, such that if a single relay gets mechanically stuck in 'on' state, the safety mechanism is still functional. A watchdog timer ensures the driving safeguard is still alive. Otherwise, or when the coils are unpowered *e. g.* at startup or if the USB connection fails, the safety loop is opened automatically. Measured response times, after triggering through the safeguard, were consistently below 5 ms.

Derivation of speed-adaptive safety zones

To derive parameters for the safety zones in a conservative setting, we conducted further braking tests on a slippery floor with low friction coefficient[9]. Figure 9.16 shows the (undesired) outcome of one such experiment. After integration of the quick-stop relay, we experimentally obtained the braking distances in Table 9.1.

v	Braking distance	+Target distance	+Safety margin	=Sum
0.0 m/s	–	0.25 m	–	0.25 m
0.1 m/s	0.02 m	0.25 m	–	0.27 m
0.5 m/s	0.10 m	0.25 m	0.1 m	0.45 m
0.8 m/s	0.20 m	0.25 m	0.1 m	0.55 m
1.0 m/s	0.30 m	0.25 m	0.2 m	0.75 m
1.6 m/s	0.70 m	0.25 m	0.3 m	1.25 m

Table 9.1: Measured braking distances and resulting size of the error zone

[9] At the airport, actual braking performance was better.

Figure 9.16: Braking tests conducted at LAAS-CNRS with the laser-based collision avoidance module. Here, the braking distance was longer than expected.

The desired, remaining target distance (after braking) of 0.25 m accounts for the feet of people. An extra safety margin has been added to account for measuring uncertainties and possible latencies. The resulting sum, *i. e.* the required size of the error zone in driving direction[10] can then be approximated by fitting an exponential function

$$d(v) = 0.260846 \cdot \mathrm{e}^{0.986263 \cdot v}.$$

The *warning zone* has empirically been defined to start 0.3 m before the speed-adaptive error zone. On both sides of the robot, we add 0.2 m and 0.03 m to the robot's width to obtain the lateral extents of the warning and error zones, respectively.

9.7.4 Practical experiences

In Figure 9.17, we see in temporal order the six different environments in which SPENCER has been deployed and tested to date. At the airport, the robot spent around four weeks in total, split into a final integration week in December 2015, and three weeks in March 2016 for final demonstration and user studies. During the final deployment, the robot drove 46.4 km autonomously, and 73.1 km in total (including data recording, and mapping sessions).

Before the measures described in this section had been implemented, a handful of collisions occurred due to human operator error or failure in motion planning and localization at the LAAS-CNRS testing site. Since introduction of the measures, no further crashes have been reported. In fact, several collisions due to failed localization have been avoided by the system during tests at the University of Freiburg.

Figure 9.18 shows a situation encountered during deployment at the airport, where the low-level collision avoidance module prevented an imminent collision with a human. While the robot was driving at relatively high speed and the socially-aware motion planner was replanning to avoid one person in (a), its new path in (b) actually led it into collision course with another person, for which no motion prediction had been considered. At this time, the motion planner would,

[10] Measured from the surface of the respective bumper.

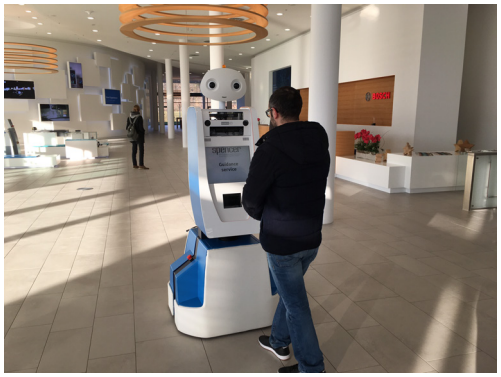
(a) LAAS-CNRS, Toulouse

(b) University Building 101, Freiburg

(c) Amsterdam-Schiphol Airport

(d) TV studio

(e) Robotics Laboratory

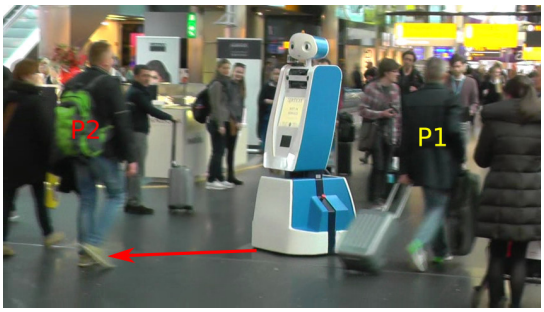


(f) Main Lobby

Figure 9.17: Different environments in which the SPENCER robot was autonomously guiding people. (d), (e) and (f) happened after official project end.

most likely, not have been able to avoid a collision anymore due to time required for replanning. Also for the human safety operator with the wireless emergency stop, this situation was hard to keep track of. However, the driving safeguard resolved the situation safely by initiating a quick stop (while still causing some mental discomfort for the involved person). In Figure 9.19, we see a further example where a person was trying to cause a collision *on purpose* with his trolley. Again, a quick stop was initiated by the driving safeguard.

(a) Replanning path to avoid P1



(b) New path is on collision course with P2

(c) Quick stop is triggered

(d) Robot stops just in time

Figure 9.18: *Example of software quick-stop:* Motion planner adjusts its initial path (dotted line) due to person P1 approaching. Adjusted path, however, now leads it onto direct collision course with person P2 (red arrow). Collision checker registers violation of safety zones, and driving controller immediately slows down robot to a complete stop. Even though person P2 does a step to the side in last picture because he feels uncomfortable, the robot stopped in time to prevent a collision.

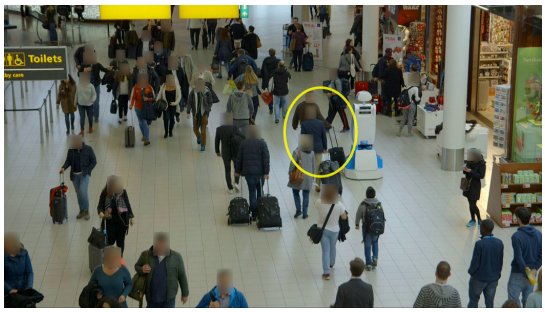


Figure 9.19: *Example of software quick-stop:* A person with luggage is walking within safe distance from the robot and observing its behavior. A split second later, the person intentionally places the luggage in front of the robot to test its reaction. The imminent collision is detected, and the driving controller quickly brings the robot to a stop by opening the safety loop.

In the final user studies conducted by Joosse (2017) at the airport, some participants complained that the robot was sometimes braking abruptly in crowded situations, and then taking a moment to re-plan. We observed this behavior mainly after the robot had been joining a flow of people, and closely driving behind a person at approximately the same speed. In such cases, the speed-adaptive safety zones do not appropriately take the dynamic nature of the moving obstacles into account. This could be improved with a more informed collision detection mechanism that integrates predictions from human detection and tracking (Chapter 4) or model-free tracking. However, as discussed in Section 9.6, obtaining robust predictions is a topic of ongoing research and might require integration of further cues such as head pose.

The presented system describes a compromise between best possible performance, in terms of reactivity and smoothness (which could be improved by implementing *all* components inside the real-time operating system of the embedded controller, including the socially-aware motion planning described by Palmieri, 2018), and being able to profit as much as possible from the rich ROS ecosystem along with its navigation stack. Despite initial skepticism, we were able to achieve sufficiently small reaction times to ensure collision-free operation at up to 1.5 m/s in the crowded airport terminal environment.

While there exists room for improvement, we believe that the overall goal of enhancing the robot's practical safety has been achieved: Joosse (2017, p. 151) concludes that *"During this part of the deployment, SPENCER drove in total 3865 meters autonomously without any critical failures that resulted in aborting trials or having to trigger the emergency stop button. Therefore, we feel that we can conclude that the robot was technically safe."*.

9.8 Conclusions

In this chapter, we presented a series of contributions and findings related to the deployment of the SPENCER robot at Amsterdam-Schiphol airport that go beyond the more theoretical, algorithmic aspects discussed in earlier chapters. These contributions have led to a very successful final robot demonstration, and resulted in positive reception in the media (see Section 1.5.5). The SPENCER robot distinguishes itself from many earlier systems in that it is largely ROS-based, heavier, taller, and operating at higher speeds, while operating in a very crowded indoor environment in which people are often in a hurry. One challenge was that the platform had to be custom-built by an industrial project partner to serve the ambitious research goals of SPENCER. The robot became ready for integration rather late (1.5 years into the 3-year project), and was shipped without ROS integration. The author was heavily involved in implementing this core functionality. Further contributions concern extrinsic calibration, human-robot interaction, safety, and the software architecture and sensor setup.

The algorithmic contributions from earlier chapters have been integrated with other systems such as motion planning, person guidance and HRI, and were successfully tested on the robot, while taking computational resource constraints into account and aiming for real-time performance at low latencies. All presented components performed well across different deployment scenarios; no major changes in functionality or parameters were required. Integration of the human detection and tracking pipeline enabled further studies concerning human-robot interaction and socially-aware motion planning.

One important aspect that we did not take lightly is the safety of this 250 kg robot platform. Despite the permanent presence of a human safety operator with a wireless emergency stop, additional safety measures have been taken as a precautionary measure. With the solutions presented in this chapter, we were able to find a good compromise between making the robot prototype more safe from a practical application standpoint, while still being able to benefit from the advantages of modern frameworks like ROS with its modular navigation stack.

Outlook: What happened after SPENCER?

After the successful completion of the SPENCER project and its positive public perception in the media, several airlines, airport operators and technology companies carried out further experiments with autonomous mobile robots in airport environments.

Project partner BlueBotics deployed the "Leo" robot at Geneva airport, which was able to scan boarding passes, let passengers drop off their luggage inside its secured luggage compartment, to then transport it to a designated baggage drop-off area. Similar to SPENCER's small luggage bin, this platform demonstrated the major advantage that a robot platform can offer – the ability to perform *physical tasks*. Several airports, including Munich, Taipei and Haneda, also started experiments with smaller humanoid robots connected to cloud services, such as Softbank's Pepper or Hitachi's EMIEW3[11], mainly targeted at improving the customer experience through various social aspects while also triggering an emotional response.

In 2017, LG deployed a first generation of airport cleaning and multi-lingual guide robots at Seoul's Incheon airport[12]. Figure 9.20 (left)[13] shows the second-generation AIRSTAR robot, deployed in 2018 as a commercial airport guide robot. While smaller than SPENCER, it incorporates a large touchscreen with a moving map display, speech recognition, and similar person guidance, speed adaptation and HRI functionalities as demonstrated in SPENCER, operating at 1 m/s[14]. In 2018, based upon lessons learned from SPENCER, Care-E[15], an HRI experiment in form of a self-driving trolley, was deployed for two days at international airports. As can be seen in Figure 9.20 (right), it again showcases the physical benefits of a human-aware robot. Like SPENCER, it autonomously drives to a gate while tracking a person using an RGB-D sensor.

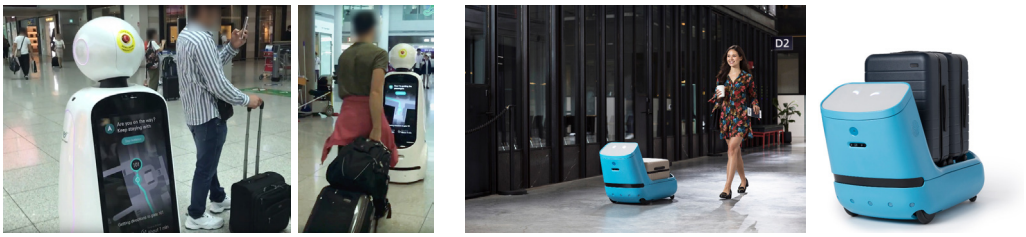


Figure 9.20: *Left:* A guidance robot deployed at Seoul airport since 2018. *Right:* A follow-up study based upon lessons from SPENCER. (Source: Press kit by Air France KLM)

[11] https://social-innovation.hitachi/de-de/case_studies/emiew3/
[12] https://lg.com/sg/press-release/lg-airport-robots-take-over-koreas-largest-airport
[13] Picture source: https://youtube.com/KvonComedy, CC-BY license
[14] http://koreabizwire.com/incheon-airport-introduces-airstar-passenger-aiding-robot/121298
[15] http://www.klmcare-e.info

Part VI

Conclusion

CHAPTER 10

Final conclusions and outlook

Figure 10.1: SPENCER looking out of the window

This thesis investigated methods for multi-modal perception of humans in challenging cluttered and crowded environments, that are suitable for real-time usage on mobile platforms. The presented chapters cover various aspects of a multi-modal tracking-by-detection perception pipeline, from sensing and low-level collision detection over human detection, tracking, group tracking to fine-grained recognition of human attributes. The considered methods operate on 2D laser, 3D lidar or RGB-D sensor data.

The author studied both classical, model-based and deep learning-based, data-driven approaches in the context of two publicly funded EU projects, SPENCER and ILIAD, involving a socially-aware service robot for person guidance in a crowded airport terminal, and a heterogeneous fleet of autonomous forklifts in intralogistics scenarios.

The methods presented herein have been validated “in the field” in complex and challenging scenarios on several different real robots. They were designed with system complexity and limited computational resources in mind, run in real-time, and have been integrated with higher-level components such as motion planning, person guidance and HRI. Through a strong team effort, these integration efforts resulted in a very successful final demonstration of the SPENCER project at Amsterdam-Schiphol airport. Three years after its project end, both the robot and the developed human detection and tracking framework are still operational. The ILIAD project had a successful intermediate demonstration and will go on until the end of 2020.

10.1 Lessons learned

In the problem statement of this thesis in Section 1.1, we outlined a number of key questions. We now want to briefly summarize the answers that we found and lessons that have been learned.

1. How can we robustly and efficiently track individual persons and groups of people in crowded environments from a robot-centric perspective?

In Chapter 3 we followed up on a previous line of works by Lau et al. (2009) and Luber et al. (2013) on multi-model multi-hypothesis group tracking to gain a broader understanding of the problem. In the joint individual-group tracking approach of Luber et al., the data association of regular hypothesis-oriented MHT is extended with an intermediate model hypotheses layer that tracks group formation processes in the form of merge/split/continuation events. One key benefit of jointly reasoning about both is that information from group tracking can easily be fed back into person-level tracking to improve tracking of occluded group members. However, this had already been shown by Luber et al. and was therefore not our key focus in this chapter.

With focus on robust group tracking in RGB-D, we partially re-implemented and improved the existing method (originally targeted at 2D laser range data) by formulating a more stable data-driven likelihood term for continuation events, and proposing a mechanism to retain stable group IDs across merge and split events. In experiments on a novel multi-sensor RGB-D dataset recorded with a mobile platform in a moderately crowded pedestrian zone, we found that the proposed approach delivers good real-time tracking performance for groups that are moving and whose members are spatially close to each other. The underlying group detection mechanism based upon coherent motion indicators shows weaknesses with static people because we currently do not estimate their body orientations, such that their feature representations tend to be less informative. In such cases, additional social cues (*e. g.* human attributes) should be considered.

In further experiments on a highly crowded, simulated scenario, we found that HO-MHT data association, and especially track initiation and deletion at the person level, suffers under the high amount of tracks and measurements. This problem is most severe under real-time constraints, where only a limited amount of global hypotheses can be generated in a given time span. In such cases, HO-MHT with its delayed decision-making can be inferior to simpler, deterministic NN approaches if not the right hypotheses are generated, or get pruned too quickly. At the end of the chapter, we discussed several possible ways of improving the current implementation, for example by reducing the detection radius, spatial subdivision, or ensuring that more diverse data association hypotheses are generated.

For deployment on the SPENCER robot, we instead devised a computationally less complex group tracking approach that retains the probabilistic SVM classifier based upon coherent motion indicator features, but combines it with a more efficient approach for tracking individual humans (see next question).

2. How large are the relative impacts of the chosen detection and tracking algorithms and sensor modalities on tracking performance?

Based upon the availability of first multi-modal datasets from the airport environment, we revisited in Chapter 4 the problem of tracking individual persons in crowded environments from a mobile platform. Previous works in the field of social robotics by Luber (2014) had examined how to incorporate social constraints into a probabilistic HO-MHT approach. While these have led to considerable improvements in tracking performance, the underlying data association method is computationally complex, as previously seen, due to the combinatorial explosion of likely global hypotheses particularly in crowded environments. Therefore, we examined the question if not a simpler data association method, combined with the right "bells and whistles", could yield similar tracking performance, and possibly be more suitable to run on a socially-aware mobile guidance robot that also has to perform a large number of other tasks at every instant.

Our positive findings in this direction, which have very recently been confirmed by independent research on the KITTI dataset (Weng and Kitani, 2020), indicate that state-of-the-art results can be obtained with simple nearest-neighbor data association, when using adequate track management (such as logic-based track initiation) and properly modeling human motion, for instance through IMM or higher levels of process noise to account for dynamics in crowded scenarios. In our experiments, we found it helpful to tune such parameters using automatic hyperparameter optimization. Our proposed simpler tracking approach, which does not consider appearance information, significantly outperforms a hypothesis-oriented MHT and track-oriented vision-based MDL tracker on the MOTA metric that we used as objective function to maximize.

We further evaluated different combinations of existing detectors for 2D laser, RGB and depth data regarding their impact on tracking performance. We found that under the utilized detection-to-detection fusion scheme, a combination of the 2D laser-based detector and a depth-based upper-body detector performs best. While 2D laser provides 360-degree coverage to maximize track recall, the vision-based upper-body detector offers high precision. Instead, a monocular HOG-based method that first has to extrapolate person distances from height over the ground plane leads to worse results if included. This was a first indication that further work on robust 3D localization of detected persons is required.

One of our key findings was that detector performance is the single, most influential factor with impact on tracking performance, that goes far beyond the impact of the chosen data association method. Because most tracking mistakes result from detection errors, which are often of systematic nature, it appears that focusing on improving detector performance would be the most promising path to better tracking results.

In Chapter 6, we subsequently compared different detection methods in 2D laser, 3D lidar and RGB-D in a cross-modal evaluation in an intralogistics scenario. While the RGB-D approaches based upon Kinect v2 data are overall the most robust methods, the 2D laser and 3D lidar methods have their own strengths and weaknesses depending on the application context. 2D laser-based detectors that observe leg motion over time, either through classical Kalman filter-based tracking or by temporally fusing a sequence of frames in the proposed CNN-based detector without explicit tracking stage, perform well when persons are in upright standing or walking poses and

do not push carts in front of them. The 3D lidar-based methods in general appear slightly more robust, but their performance breaks down in very cluttered and narrow environments, which most likely are underrepresented in the autonomous driving datasets they have been trained on. This indicates a need for more large-scale domain-specific training data sets, or methods that generalize better with less data. In any case, we believe that further investigation is necessary to systematically understand why these methods fail in such scenarios.

In conclusion, we can say that the relative impact of the choice of data association method on tracking performance is lower than expected in our scenarios, while the impact of tracking hyperparameters and especially detectors is much higher than initially suspected.

3. How can we robustly detect and localize persons not only in 2D image space, but also in metric 3D coordinates – which is essential for robotics applications?

In Chapter 5, we evaluated several recent deep learning-based approaches for 2D object detection, including Mask R-CNN (K. He et al., 2017) and YOLO v3 (Redmon et al., 2018), on the human detection task in intralogistics scenarios. Compared to earlier methods used in the airport environment (Chapter 4), due to training on diverse and large-scale datasets such as MS COCO (T.-Y. Lin et al., 2014), they achieve higher precision and recall especially when people are in challenging poses, partially truncated or occluded, or wear unusual professional clothing.

However, since such datasets often do not contain 3D groundtruth, the prevalent methods output only 2D bounding boxes. We found that 2D bounding boxes are a suboptimal intermediate representation when the ultimate goal is to estimate 3D coordinates – as often required in robotics applications. For instance, when a person is stretching out both arms, the largest part of the 2D bounding box will contain background. Therefore, naïve RGB-D methods that estimate a person's depth *e. g.* using the median of the bounding box will fail.

We showed that 2D instance segmentation masks can help in such cases; however, because they do not exploit depth information, they are prone to error especially in cases where persons interact with other unknown foreground objects. We therefore proposed to directly regress 3D coordinates, without such an intermediate representation, using a real time-capable detector like YOLO v3, that we extend via mid-level feature fusion to benefit from additional depth data (if available). To learn end-to-end 3D regression, a sufficiently large and diverse dataset with 3D groundtruth is required.

4. How can we train recent deep learning-based detection methods using only limited available amounts of labeled real-world 3D data? How can synthetic data help?

A recent trend in the computer vision community to obtain better perception models is the inclusion of synthetically rendered images in the training phase, especially in the field of articulated human pose estimation and 3D object pose estimation. In Chapter 5, we presented a novel, highly randomized RGB-D dataset for 2D and 3D person detection in cluttered environments, which has been synthetically rendered using Unreal Engine 4. Through extensive domain randomization, we achieve a level of diversity in human poses, backgrounds and lighting setups that is hard to obtain in real-world datasets from a single application domain. However, especially in the RGB modality, we still observe a significant domain gap when training 2D detectors on the dataset, and

further investigation concerning the use of domain adaptation techniques is required. Instead, on the depth modality, it appears that modeling sensor noise does not have a large impact on detection performance. In further experiments on the 3D detection task, we found that we can learn 3D localization solely from synthetic RGB-D data with good results. This is enabled by a clever transfer learning strategy in the previously mentioned RGB-D YOLO v3 variant, which leverages existing large-scale real-world 2D datasets to learn the core detection task, and then uses our accurate synthetic 3D groundtruth to learn 3D localization. When we combine this with a depth-aware augmentation scheme during training, we outperform several state-of-the-art baselines, including a computationally more complex articulated 3D human pose estimation method by Zimmermann et al. (2018) that was trained on real-world datasets. In contrast, when we use only real-world data during training, 3D localization accuracy is significantly lower.

By this, we have shown that we can clearly benefit from synthetic data, and at the same time reduce the need for manually labeled large-scale datasets with 3D groundtruth. While the initial effort to set up such a simulation should not be underestimated, adapting the simulation to new sensor arrangements is then very easy, which is especially relevant for robotics applications where sensor setups often vary.

5. Can we learn more about humans by looking at their point clouds, *e. g.* attributes?

Our experiments showed that we can recognize many attributes just by looking at point cloud geometry, while color cues can be of additional help. In order to detect binary human attributes (*e. g. has long trousers* or *is female*) from RGB-D point clouds, in Chapter 7 we proposed an efficient tessellation-boosting approach based upon a method originally developed for human detection by Spinello, Luber, et al. (2011). The method has been trained and evaluated on a novel RGB-D human attributes dataset, comprising over 100 persons, and compared against a HOG-SVM classifier on RGB images, as well as two early deep learning methods in RGB-D. On the gender attribute, we clearly outperform the early deep learning-based RGB-D methods and the HOG-SVM, while at the same time being more efficient without requiring GPU acceleration. Attributes on which our method does not perform as well are the ones that are also underrepresented in our dataset, indicating that even larger datasets might be needed. As we discussed at the end of the chapter, it would be interesting to see a comparison against more recent image-based deep learning approaches, as well as methods such as PointNet that also leverage a geometric representation.

Other researchers have recently extended our open-sourced implementation to the task of body orientation angle regression (Wengefeld, Lewandowski, et al., 2019). They conjecture that their extension of our method should also generalize to continuous human attributes such as *age*, for which there is currently a lack of sufficiently large RGB-D datasets.

6. How can we make a multi-modal human detection and tracking pipeline more modular and reusable across different robot platforms and application scenarios? Which additional engineering aspects are relevant when deploying such a system on a real robot that is operating at fast human walking speeds in a crowded airport environment?

In Chapter 8, we proposed an application- and robot-agnostic ROS-based multi-modal human detection and tracking framework. The modular framework and its reusable components have been successfully applied in the EU projects SPENCER and ILIAD, and are publicly available as open-source software. This has allowed several other international research groups to utilize it on their robot platforms for research *e. g.* on human detection and tracking, group detection, long-term learning of motion patterns, human-aware navigation and human-robot interaction. The proposed framework generalizes to other scenarios and through its reusability has already helped to advance the state of the art. One key concept we developed is a set of ROS message definitions that is largely independent from the particular sensor modality or detector at hand.

In Chapter 9, we presented the SPENCER robot, a first-of-its-kind socially-aware service robot for person guidance in a crowded airport terminal. To be able to deploy the robot in the field, we devised a modular, distributed, ROS-based robot software architecture including several essential components for sensor calibration, task planning and human-robot interaction. We integrated the human detection and tracking framework with motion planning, person guidance and HRI components that are vital for the envisaged person guidance scenario. An aspect not to be underestimated with a 250 kg robot operating at up to 1.5 m/s in an uncontrolled, public environment is the one of safety. Here, we devised several mechanisms based upon low-level collision avoidance in RGB-D and 2D laser, watchdog timers, and redundant hardware solutions that act upon the robot's safety loop in order to enhance safety in practice. Even though there was always a human safety operator with a remote control in the vicinity, our practical experiences show that in such crowded environments, the system must react autonomously to dangerous situations because human reaction times can be too high. Altogether, these contributions have enabled a very successful final project demonstration at Amsterdam-Schiphol.

10.2 Outlook and recommendations for future work

We now discuss, at a higher level than at the end of each individual chapter, recent and possible future developments in the different areas that have been examined in this thesis.

Human detection and pose estimation

As we have seen in Chapter 5, modern vision-based detection approaches such as Faster R-CNN (Ren et al., 2017) or YOLO v3 (Redmon et al., 2018) obtain very good 2D detection results for the human class even in challenging industrial environments and under unusual appearance (*e. g.* workers wearing protective clothing). This can also be attributed to the high diversity of modern large-scale 2D datasets such as MS COCO (T.-Y. Lin et al., 2014). In 2019, the 2D bounding box detection task has been removed from the COCO challenge, indicating that the vision community considers this task as largely solved. Small improvements might still be attained especially under severe occlusion in crowded environments, *e. g.* by incorporating temporal information from preceding frames.

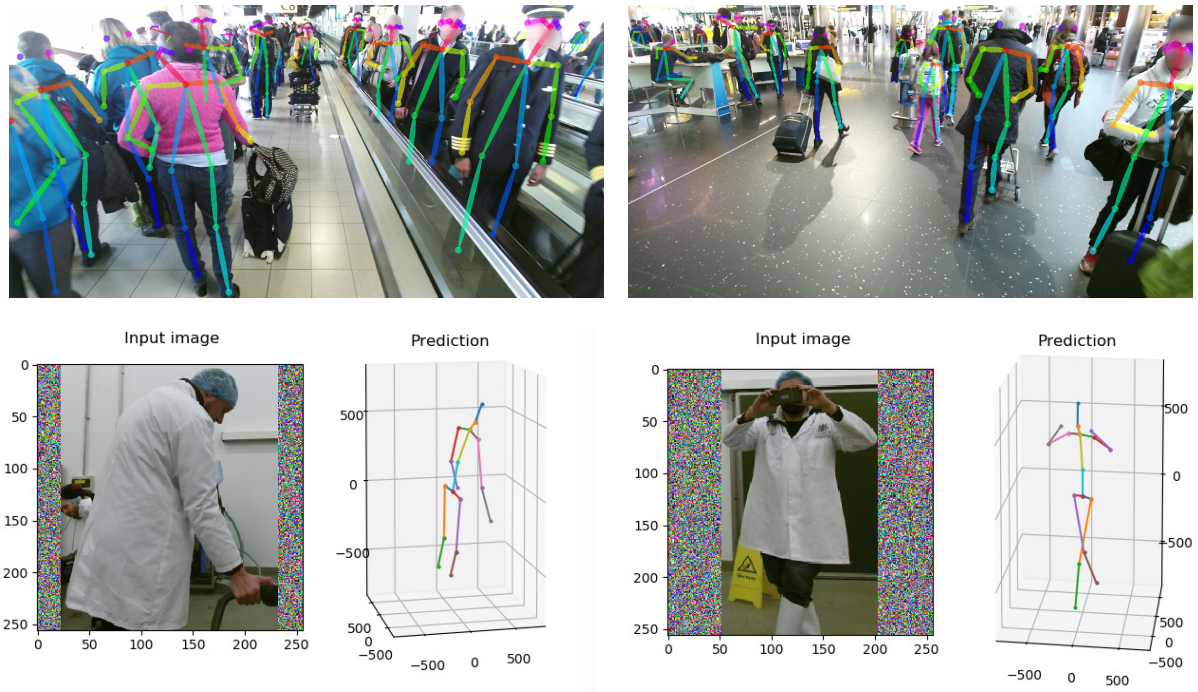


Figure 10.2: Real time-capable articulated human pose estimation methods are starting to compete with pure detection approaches. *Top:* 2D human pose estimation results from OpenPose (Cao et al., 2017) on SPENCER data from the airport. *Bottom:* Root joint-centric 3D pose estimation results on ILIAD data from the truncation-robust method by Sárándi, Linder, et al. (2018a).

Recent research has been focused on 2D instance segmentation as a more fine-grained representation, which – as we have seen in Chapter 5 – can be useful also for robotics tasks. Also detection in 3D has recently been receiving more attention in the vision community – our RGB-D method presented in the same chapter deals with this task, but under the assumption that depth data is available. As we have seen, our method is less accurate in 3D localization when using only monocular RGB data, and we believe there is still room for improvement here. The author's contributions to research on monocular single-person *articulated 3D human pose estimation* (Sárándi, Linder, et al., 2018a) go into this direction, but as shown in Figure 10.2 (bottom) they assume that root joint depth is known. Thus, they still need to be combined with a method (such as the proposed RGB-D human detector) that provides 3D detections in global coordinates.

Very recently, Mehta et al. (2020) proposed a system for monocular multi-person 3D human pose estimation that runs at 30 Hz on a GPU, incorporates tracking, and does not rely on 2D bounding boxes as input. In contrast to the related method by Zimmermann et al. (2018), which we considered as a baseline in Chapter 5 and which already yielded promising results, their method is faster and does not require a depth channel. While the accompanying videos still show failure cases in 3D joint localization, sometimes entire persons are not being detected, and unlike in depth-based approaches a ground plane assumption is required to estimate person heights, such specialized 3D human pose estimation methods might replace more generic object detection

and tracking approaches in the future – i. e. *human detection and pose estimation may converge*, if humans are the only kind of objects to be detected and tracked by the robot. Otherwise, they could be combined with category-agnostic, motion-based, *model-free tracking approaches* that exist for vision (Ošep et al., 2018), 2D laser (D. Z. Wang et al., 2015) and 3D lidar (Dewan et al., 2016). Until then, a smaller step would be to enhance the existing 3D human detectors with *rough body pose estimation* capabilities, such as proposed in Lewandowski et al. (2019) or Wengefeld, Lewandowski, et al. (2019). For this, the existing 3D body joint groundtruth from our synthetic RGB-D dataset in Chapter 5 could be exploited.

For both detection and pose estimation, we believe that further research is required concerning the handling of localization *uncertainties* (Y. He et al., 2019; Kraus et al., 2019; Feng et al., 2019 provide initial insights), the integration of *introspective capabilities* (Grimmett et al., 2016), and study of different *calibration methods* (C. Guo et al., 2017). This might be critical especially for safety-relevant use-cases (Czarnecki et al., 2018) and against *adversarial attacks* (Y. Zhao et al., 2019), where further study is required to assert how such systems can be proven to be safe.

What is overall still missing is an understanding of *underlying physical concepts* – such as distinguishing between a person being reflected in a glass window, and somebody actually standing behind a glass pane. Here, vision-based stuff segmentation (Caesar et al., 2018) and panoptic segmentation (Kirillov et al., 2019; Weber et al., 2019) could provide important contextual cues. In the airport scenario, we also observed that recognizing certain contextual *affordances* and activities can be important – for instance, a mother carrying her baby, a person riding an electric vehicle, or walking on a horizontal escalator, in which cases the motion models should be adapted accordingly during tracking.

Concerning *lidar-based human detection* in indoor environments, there still appears to be more room for improvement. In Chapter 6, we observed low recall especially in *narrow environments*, when persons are close to walls or other objects. Further study is needed to understand if these issues are mainly due to lack of suitable large-scale indoor training datasets from the application domain (*e. g.* no low ceilings in automotive datasets), or if novel methods are required.

Regarding datasets, given the availability of strong detectors in one domain such as RGB-D, *multi-sensor transfer-learning strategies* as *e. g.* proposed by Z. Yan et al. (2018) could be exploited to obtain large-scale training datasets for 3D lidar, for instance for the intralogistics domain. The synthetic dataset from Chapter 5 could also be extended with simulated lidar data (similar to Yue et al., 2018), and *domain adaptation techniques* like cycle-consistent adversarial domain adaptation (Hoffman et al., 2018) could be applied to generate more realistic RGB-D images, or to synthesize realistic lidar intensity readings.

Multi-modal fusion

In this thesis, we chose a late fusion approach after the detection stage because it allows for more modularity and reusability of the detection components, which is beneficial when deploying a tracking system on a heterogeneous fleet of robots. Instead of fusing detections with other detections, an alternative *detection-to-track fusion* approach (as used *e. g.* in Spinello et al., 2010) would offer the potential of making more informed decisions and allow for asynchronous fusion at detector-specific frame rates.

An interesting field of study are multi-modal *early- or mid-level fusion* approaches, which fuse modalities already at the level of raw sensor data or in the region proposal stage (*e. g.* Spinello et al., 2010; Ku et al., 2018; Qi et al., 2018). Except for the RGB-D fusion in Chapter 5, such methods have not been considered in this thesis. The challenge is that the resulting learned models are often specific to a particular sensor setup and dataset at hand – most recent methods are trained and tested on the KITTI benchmark suite (Geiger et al., 2012) –, and do not generalize easily to other platforms without additional large-scale efforts to record and annotate new data. Here again, *synthetically generated data* could be of great help, offering the option of regenerating a new dataset for a different sensor setup with just a few minutes of manual effort. An alternative, more decoupled approach would be to project *e. g.* lidar point clouds into camera images, and then use computed per-pixel instance segmentation masks as a bridge between modalities. In general, it needs to be understood in which cases such techniques have an advantage over late fusion at the detection stage; maybe, early fusion could help far-range perception, or solve detection challenges involving *reflective and translucent materials* such as glass.

Tracking

In Chapter 4, we have taken an important step towards evaluating a human tracking system for mobile robots in the wild in very crowded indoor environments. In contrast to existing benchmarks such as the MOT challenge series (Leal-Taixé et al., 2015; Leal-Taixé et al., 2017), our focus has been on multi-modal sensor setups from an egocentric perspective. We believe that further work is needed to *standardize benchmarks* in this area; as mentioned in the introduction, there are some very recent, promising developments in the computer vision community, with a workshop and the release of a new benchmark dataset (Martín-Martín et al., 2019).

For the presented human tracking system, probably the largest improvement can still be achieved through incorporation of *appearance information*, for instance as done earlier by Luber, Spinello, et al. (2011), or with a deep association metric like in DeepSORT (Wojke, Bewley, et al., 2017). Such feature embeddings can also be used for person re-identification (*e. g.* Hermans et al., 2017), which is an important missing feature in our framework that could be used to bridge longer occlusions (*e. g.* when the robot is waiting in front of the restrooms for guided passengers). From a research standpoint, this is a challenging *open-world re-identification problem*, which is subject of ongoing research and discussed in a recent survey article (Leng et al., 2019). Most work in the vision community so far focused on closed-world problems, where recent methods now surpass human performance on benchmark datasets. Earlier methods in robotics also evaluated multi-modal reidentification approaches (Barbosa et al., 2012; Munaro, A. Basso, et al., 2014), which could be re-examined using recent deep learning techniques.

When we restrict ourselves to the core tracking algorithms, the problem of tracking multiple person centroids has, in the author's opinion, been mostly solved. In very dense crowds as in Figure 10.3 (left), due to the limited observability from the first-person perspective, it is however questionable whether tracking of individual persons at far range still makes sense. One issue we did not consider in SPENCER is the *tracking of extended objects*, such as chains of luggage carts or large vehicles as in Figure 10.3 (middle and right picture). Group tracking in Chapter 3 was purely based upon spatial relationships; with advances in recognition of human attributes, social relationships and activities, such *further social cues* should also be taken into account.

Figure 10.3: In very dense crowds, tracking individual people might not make sense. Also, certain extended dynamic objects are hard to track by their centroids.

New challenges such as tracking instances on a *per-pixel level* (Voigtlaender et al., 2019), or previously considered problems such as *distributed people tracking* for fleets of mobile robots (Chou et al., 2011; Tsokas et al., 2012) over lossy wireless connections, could become more relevant in the future. For the latter, existing methods developed for radar-based tracking (Bar-Shalom et al., 1995) should also be considered.

Fine-grained recognition of human attributes

One important problem (for a social robot) that this thesis did not tackle is *age estimation*. Most computer vision methods for age estimation operate on facial images (Moschoglou et al., 2017; Rothe et al., 2018), and only few consider full-body information (Yuan et al., 2018), which is important for one-shot robotics applications where *e. g.* the robot might only see a person from behind. Some progress has been made in more general human attribute recognition (Y. Li et al., 2016; Sarafianos et al., 2018), summarized in a recent survey (X. Wang et al., 2019), where new large-scale datasets have been released that are composed of *full-body images* (often from a surveillance camera perspective). Interestingly, no significant progress has been made in *multi-modal* human attribute recognition, presumably due to the lack of sufficiently large datasets on which deep learning methods could be applied. With 3D lidar sensors becoming more capable and affordable, they could become more relevant for this problem in the future.

Understanding of social cues, intentions, activities and higher-level reasoning

If we consider again the example of a socially-aware guide robot in an airport, it is apparent that despite the scientific progress made in the SPENCER project (2013–2016) and elsewhere like in the JackRabbot project (2015–2020), most open challenges still relate to a lack of higher-level understanding and reasoning: We would like the robot to respect various *social norms* and consider relevant social cues, while being able to *recognize social activities* and *human intentions*. The main reason for this research gap, besides a lack of domain-specific large-scale datasets, is presumably that in the past, lower-level perception components were not yet sufficiently robust enough to enable such higher-level perception tasks.

Now that essential perception components such as human detection, pose estimation, and attribute recognition are becoming more mature, the resulting semantic information can be used to better approach higher-level problems such as the recognition of social relations and activities. For instance, with the availability of gender classification and age estimation, a robot

Figure 10.4: Different queues encountered by SPENCER in the airport environment.

might be able to infer if a group of people is a family (or a couple), allowing for adaptations in human-robot interaction. A survey article by Ziaeefard et al. (2015) outlines how such semantic cues could be exploited for *human activity recognition*. Broader overviews of relevant datasets and recent methods, many of which instead directly operate on raw image or video data, are provided by Vrigkas et al. (2015), J. Zhang et al. (2016), Rodríguez-Moreno et al. (2019), and H.-B. Zhang et al. (2019). Detection of *queuing behaviors* as in Figure 10.4 and other *social activities* from semantic data have been studied by Okal et al. (2014) and Okal, Triebel, et al. (2016). In the computer vision literature, the problem is also known as *group activity recognition* (Ibrahim et al., 2016; Vahora et al., 2019). Here, the robot could also integrate information over longer periods of time: Whilst driving past a queue for instance, it could discover the length of the queue, observe how quickly people move forward, and combine this with prior semantic knowledge on flight departure times for further reasoning.

Recognizing *human-object interactions* (Lu et al., 2018; Y.-L. Li et al., 2019) is an essential ingredient for understanding human activities, and benefits from an understanding of visual affordances (Hassanin et al., 2018). For interactions between humans, the ability to recognize *social affordances* (Shu et al., 2016), such as “shaking hands” or “handing over an item”, can provide further insights. Eventually, such understanding could help a robot to resolve situations that we motivated in the beginning in Figure 1.3 (p. 10), such as a person taking a photograph of a group, which the robot could recognize and then react upon by following social conventions and not crossing in between.

Part VII

Appendix

APPENDIX A

New datasets for multi-modal people tracking

In this appendix, we give an overview of the novel datasets that have been acquired during this thesis. With robot perception being a data-driven research field, they were of vital importance for this research as no publicly available alternative datasets of this kind did exist. They allowed us to gain a deeper understanding of the problems that arise in different robotic application scenarios. We also describe the sensor platforms with which we recorded the real-world datasets.

A.1 Mobile data recording platform

At the start of this thesis, no suitable mobile platform with a multi-modal sensor setup for data acquisition was available to us. As the SPENCER robot was only to be finished around 1.5 years into the project, we devised an interim solution for first data recordings in the airport based upon a bicycle trailer that we equipped with an array of sensors including wheel encoders for odometry, battery packs and DC-DC converters to supply power to sensors. The final sensor setup used at the airport is shown in Figure A.1 (middle). We constructed the platform such that it can easily be folded together for transport, while replicating the planned setup of the SPENCER robot from Section 9.3 as closely as possible. An additional DSLR camera with fisheye lens was added to aid in data annotation. To ensure that sufficient recording bandwidth is available, we used up to three high-end laptops with fast SSDs and synchronized clocks in parallel.

Figure A.1: The mobile data recording platform with its initial (left) and final sensor setup (middle left and right). To obtain wheel odometry, we added quadrature encoders in dynamo housings on both main wheels (top right). They are connected to an Arduino microcontroller (bottom) that communicates with a ROS node.

A.2 Rathausgasse dataset[1]

This dataset has been recorded with the early prototype of the mobile data recording platform while it was still equipped with fewer sensors and had the RGB-D sensors arranged vertically, as shown in Figure A.1 (left). Table A.1 lists the most important facts concerning this dataset.

Scenario	Outdoor, urban pedestrian zone, semi-crowded / Freiburg im Breisgau, Germany
Purpose	Tracking groups of people (Chapter 3), training of 2D laser-based detector (Chapter 4)
Recorded with	Mobile data recording platform
Sensor setup	Two Asus Xtion Pro Live RGB-D (vertical) at 1.4 m, 640×480 px, 76° hFOV, 30 Hz SICK LMS 500 at 0.7 m, 0.25° resolution, 190° hFOV, 35 Hz Wheel odometry, 10 Hz
Time of recording	October 1st, 2013 shortly before sunset (17:00–19:00)
Total raw duration	25:14 min (42 GB)
Distinct sequences	Rathausgasse (18:18 min), Uni-Café (4:12 min), Saturn (2:44 min)
Groundtruth labels	Aggregated statistics on individuals and groups in Rathausgasse sequence

Table A.1: Rathausgasse dataset

A.3 SPENCER datasets[2]

Airport recordings from June 2014

Initial data recordings in the airport environment have been conducted in June 2014 with the mobile data capture platform. This data has been used mainly to gain an initial understanding of the challenges for human detection and tracking. One sequence from this dataset was used for the quantitative, multi-modal experiments in Section 4.5. Details are shown in Table A.2, Figure A.2 depicts typical recording situations. Figure A.3 shows objects and structures that we encountered at the airport which might challenge a robot's perception system.

Scenario	Indoor, airport environment, empty to highly crowded Amsterdam-Schiphol airport, international terminals E, F, G, H
Purpose	Human tracking (Chapter 4), mapping
Recorded with	Mobile data recording platform
Sensor setup	Four Asus Xtion Pro Live RGB-D (horizontal) at 1.65 m, 640×480 px, 57° hFOV, 30 Hz Two SICK LMS 500 at 0.7 m back-to-back, 0.25° resolution, 360° hFOV, 35 Hz Bumblebee 2 stereo camera rig at 1.4 m, 1024×768 px, 12 cm baseline, 20 Hz DSLR camera with 8 mm fisheye lens at 1.25 m, 1920×1080 px, 180° hFOV, 25 Hz Kinect v2 at 1.5 m, 1920×1080 px (RGB) and 512×424 px (D), 84° hFOV, 15 Hz Wheel odometry, 10 Hz
Time of recording	June 11–12, 2014 (afternoon and morning)
Total raw duration	4h 34 min (982 GB)
Distinct sequences	10 different sequences at different gates and in different terminals
Groundtruth labels	One 4-minute sequence annotated with 172 groundtruth trajectories (human centroids)

Table A.2: SPENCER airport recordings from June 2014

[1] L. Palmieri and B. Okal helped with logistics for this recording.
[2] These were recorded with help of L. Palmieri, S. Breuers, T. Kucner, M. Magnusson, L. Beyer and R. Triebel.

Figure A.2: Data recording at the airport

Figure A.3: Challenging objects and structures encountered at the airport

Airport recordings from December 2015

First data with the actual robot platform (Section 9.3) could only be recorded in the airport environment around 4 months before end of the SPENCER project (Table A.3). Based upon findings from the previous recordings with the mobile data capture platform, it was decided by the responsible project partners to include an additional 3D lidar sensor for mapping and localization as 2D lasers were deemed insufficient in highly crowded situations. A Kinect v2 was added temporarily to test human attribute recognition. Not all sensors were used at all times.

Scenario	Indoor, airport environment, empty to highly crowded Amsterdam-Schiphol airport, B and C piers and central plaza
Purpose	Human tracking (Chapter 4), attributes (Chapter 7), social navigation (Section 9.6); research of project partners on head pose estimation, mapping and localization
Recorded with	SPENCER robot
Sensor setup	Four Asus Xtion Pro Live RGB-D (horizontal) at 1.6 m, 640×480 px, 57° hFOV, 30 Hz Two SICK LMS 500 at 0.7 m back-to-back, 0.25° resolution, 360° hFOV, 35 Hz VLP-16 LiDAR at 1.8 m, 360° hFOV, 10 Hz Kinect v2 at 1.7 m, 1920×1080 px (RGB) and 512×424 px (D), 15 Hz [only 1st day] Wheel odometry, 20 Hz
Time of recording	Nov 30–Dec 3, 2015 (morning till evening; mapping at night)
Total raw duration	14h 20 min (292 GB)
Distinct sequences	21 different sequences in the piers and central shopping area 2h 31 min with multi-modal setup for perception (13 sequences) 0h 49 min to test socially-aware navigation 10h 45 min for mapping and localization
Groundtruth labels	These recordings were only used for qualitative experiments in this thesis.

Table A.3: SPENCER airport recordings from December 2015

Airport recordings from March 2016

A final data recording took place at Amsterdam-Schiphol airport shortly before and after the final project demonstration, details can be found in Table A.4:

Scenario	Same as in Table A.3
Purpose	Human tracking (Chapter 4), social navigation (Section 9.6); research of project partners on human-robot interaction, end-user studies, mapping and localization
Recorded with	SPENCER robot
Sensor setup	Same as in Table A.3
Time of recording	March 17–18 and March 23–24, 2016 (always in the morning)
Total raw duration	4h 41 min (163 GB)
Distinct sequences	20 different sequences in the piers and central shopping area
	1h 49 min with multi-modal setup for perception (9 sequences)
	0h 52 min for social navigation and end-user studies
	2h 00 min for mapping and localization
Groundtruth labels	These recordings were only used for qualitative experiments in this thesis.

Table A.4: SPENCER airport recordings from March 2016

Motion capture dataset

To analyze and improve human tracking in highly dynamic situations, we recorded two short sequences with accurate groundtruth trajectories from a motion capture system at the partner site of LAAS-CNRS in Toulouse, France. People move randomly in highly erratic patterns with frequent stopping and acceleration. Table A.5 shows details on the recording setup:

Scenario	Indoor, narrow and controlled laboratory environment (around 30 m^2)
Purpose	Human tracking in dynamic scenes (Chapter 4)
Recorded with	SPENCER robot (not moving)
Sensor setup	Two Asus Xtion Pro Live RGB-D (horizontal) at 1.6 m, 640×480 px, 57° hFOV, 30 Hz
	Two SICK LMS 500 at 0.7 m back-to-back, 0.25° resolution, 360° hFOV, 35 Hz
Time of recording	August 14, 2015 (afternoon)
Total raw duration	6:10 min (3 GB)
Distinct sequences	Two sequences of 4:10 min (4 persons) and 2:00 min (single person)
Groundtruth labels	Accurate head positions from Optitrack motion capture system

Table A.5: Motion capture dataset

A.4 ILIAD datasets

NCFM food facility sequences from September 2018[3]

Multiple sequences of several minutes were recorded with the Cititruck platforms and a multi-modal sensor setup at a small-scale food factory and adjacent storage rooms in the UK during an integration week of the ILIAD project (see Figure 1.5 (a) on page 12). In these scenes, all people wear protective garments. Table A.6 provides details on the dataset, whereas Figure 6.5 on page 127 shows representative images of the two labeled sequences.

[3] Kevin Li Sun and Manuel Fernández Carmona assisted me during these recordings. Robert Schirmer helped to annotate the shorter of the two sequences.

Scenario	Indoor, small food factory and storage rooms, 1–20 individuals in protective clothing NCFM Facility, UK
Purpose	Human detection (Chapters 5 and 6), human tracking
Recorded with	Two autonomous pallet trucks (Cititruck), manually moving in some sequences
Sensor setup	Kinect v2 at 1.5 m (rear), 1920×1080 px (RGB) and 512×424 px (D), 84° hFOV, 30 Hz SICK S300 at 0.15 m (rear), 0.5° resolution, 270° hFOV, 12.5 Hz VLP-16 LiDAR at 1.2 m, 360° hFOV, 10 Hz Odometry, 16.5 Hz
Time of recording	September 24–27, 2018 (morning till afternoon)
Total raw duration	51:54 min (43.9 GB)
Distinct sequences	9 different sequences with varying levels of dynamics and crowdedness
Groundtruth labels	Two test sequences manually labeled with centroid trajectories: Dynamic scene (60 s, 5 persons pushing carts/trucks, open space, static robot) Storage room (120 s, 1 person, narrow environment, mobile robot)

Table A.6: NCFM recordings from September 2018

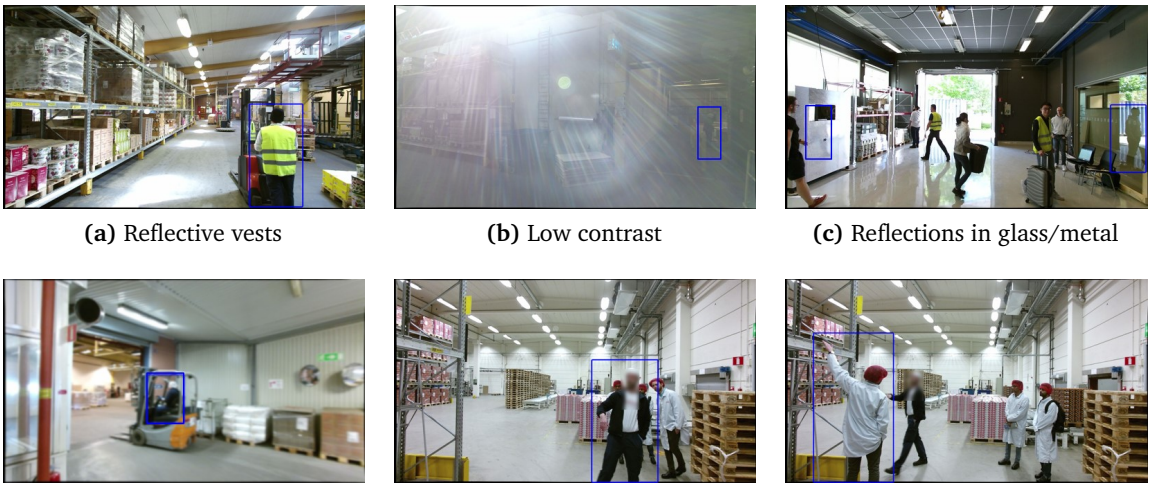
Image-based intralogistics RGB-D dataset (2018)[4]

To train and evaluate 2D image-based RGB-D detectors, we included further data recorded with a larger forklift in long-term recordings at the Swedish food factory warehouse (see Fig. 1.5 (b) on p. 12), and from a robot lab. From several terabytes of data, we hand-selected 3,100 challenging frames that contain at least one person each. Table A.7 provides further information and Figure A.4 highlights typical challenges encountered in this dataset.

Scenario	Two food factories and adjacent storage rooms in UK and Sweden, where people wear protective clothing and safety vests; Örebro AAAS robot lab with shelves and pallet trucks, some people in safety vests. Overall around 1–5 persons per scene, up to 20 in some sequences.
Purpose	Human detection in RGB-D (Chapter 5) Learning of long-term dynamics and site-specific activity patterns (project partner)
Recorded with	Two autonomous pallet trucks (Cititruck), manually moving in some sequences One Toyota BT forklift, manually moving in some sequences
Sensor setup	Cititruck platforms: see Table A.6. Toyota BT truck: Kinect v2 at 1.8 m (front), 1920×1080 px (RGB) and 512×424 px (D), 84° hFOV, 30 Hz SICK S3000 at 0.15 m (rear), 0.25° resolution, 190° hFOV, 16.5 Hz HDL-32E LiDAR at 1.6 m, 360° hFOV, 10 Hz Odometry, 16.5 Hz
Time of recording	May 7–9, 2018 (Swedish food factory), May 18, 2018 (AAAS), Sep. 24–27, 2018 (NCFM) All during regular working hours
Total raw duration	21:05 min (69.3 GB) at Swedish food factory (hand-selected from in total 24 h / 4.8 TB) 15:28 min (55.1 GB) at Örebro AAAS robot lab 51:54 min (43.9 GB) at NCFM facility
Selected frames	3,100 RGB-D frames, split up into training, validation and test set (1.5k / 0.5k / 1.1k) Train/validation/test splits from temporally non-overlapping sequences
Groundtruth labels	All persons manually annotated in RGB with 2D bounding boxes and occlusion flags

Table A.7: Image-based intralogistics RGB-D dataset

[4] M. Magnusson and D. Adolfsson performed the recordings at the Swedish food factory with help of our industrial partner. N. Vaskevicius and D. Grießer annotated around one third of the frames, the rest was done by myself.



(a) Reflective vests **(b)** Low contrast **(c)** Reflections in glass/metal

(d) Motion blur **(e)** Strong occlusion **(f)** Unusual pose + clothing

Figure A.4: Challenges for a computer vision system encountered in data from ILIAD

A.5 Synthetic datasets

Synthetic PedSim dataset

As first presented in Section 3.5.2, we generated a synthetic dataset for human tracking in very crowded scenarios using the PedSim simulator, which uses the social force model (Helbing et al., 1995) to simulate pedestrian interactions. Table A.8 gives a short summary of the dataset.

Scenario	Crowded open space (30×30 m) with simulated pedestrian flows, queues and individuals
Purpose	Tracking of groups (Chapter 3) and humans (Chapter 4) in very crowded scenes
Generated with	PedSim ROS and Gazebo integration (Section 8.9)
Sensor setup	Simulated mobile SPENCER robot with 2D laser scanner at 70 cm height, 0.25° resolution, 360° hFOV, 20 m range to determine pedestrian visibility/occlusion
Total duration	3:00 min (<100 MB) in our experiments
Groundtruth labels	Accurate groundtruth centroids and trajectories of all humans, visibility flags

Table A.8: Synthetic PedSim dataset

Synthetic RGB-D dataset generated with Unreal Engine 4[5]

Since there were no sufficiently diverse and large enough multi-person RGB-D datasets publicly available to learn robust 3D human detection, we synthesized a highly randomized RGB-D dataset with accurate 3D groundtruth for all human body joints using Unreal Engine 4. Section 5.5 introduced the dataset and explained the different aspects of the simulation. Table A.9 summarizes the key facts on the resulting dataset that we used in our experiments.

[5] My master student M. Hernandez set up the initial simulation environment in Unreal Engine, and prepared the first version of the post-processing scripts.

Scenario	Highly randomized dataset from a robot-centric perspective with varying person density (scattered people – denser crowds), over 700 different flying 3D objects for augmentation, randomized lighting, character randomization among 81 different person meshes and over 100 animations, character texture augmentation, randomized navigation of robot and characters, robot teleports to new random position every 15 seconds
Purpose	Training of RGB-D human detectors (Chapter 5), in particular 3D localization
Generated with	Unreal Engine 4, custom Blueprints + post-processing scripts (Section 5.5)
Sensor setup	Simulated Kinect v2 time-of-flight sensor at 1.5 m on a simulated Cititruck, 1920×1080 px (RGB) and 512×424 px (D), 84° hFOV, 10 Hz. Sensor is horizontally mounted on a randomly rotating pan unit. Approximate simulation of time-of-flight sensor noise (depth and amplitude distortion, axial noise, illumination interference, material IR response, reflections, registration shadowing) based upon empirical measurements.
Total size	558 GB raw data, 88 GB after post-processing for a single version of the dataset
Distinct sequences	Training set with 6 scenes (warehouses, train station, factory district), 15,000 frames Validation set with 2 scenes (industrial assembly line, office building), 5,000 frames
Groundtruth labels	Accurate 3D body joint positions in camera coordinates, centroids (derived from pelvis joints), upper-body orientation angle, head orientation angle, 2D instance segmentation masks, 2D bounding boxes with occlusion ratio and ignore flags

Table A.9: Synthetic RGB-D dataset generated with Unreal Engine 4

A.6 Human attributes dataset[6]

In Chapter 7, we introduced a novel RGB-D dataset for human attribute recognition. Table A.10 gives an overview of the so-called SRL Human Attributes Dataset:

Scenario	Single persons in different standing and walking poses in front of a static RGB-D sensor
Locations	Social Robotics Lab / Building 101 Lecture Hall / Room at Department of Psychology All at the University of Freiburg, Germany
Purpose	Human attribute recognition in RGB-D (Chapter 7)
Recorded with	Static sensor on a tripod at 1.5 m height, 14° downward tilt
Sensor setup	Kinect v2 (partially a prototype version), 1920×1080 px (RGB) and 512×424 px (D), 84° hFOV, 15 Hz, 0–4.5m recording distance
Post-processing	Extraction of cropped person images and person clouds
Time of recording	Over 3 months in summer 2014 during daylight hours
Total duration	122 GB (image-based variant), 28 GB (point cloud variant)
Distinct sequences	138 recording sessions with 118 individual persons 4 sequences per session (static poses / two walking patterns / close-up interaction)
Demographics	Mean age 27.0 years (std dev 8.7) / 45.76% of male gender (= 50% of all sessions)
Groundtruth labels	Gender, age, has long trousers, has long sleeves, has long hair, wears skirt/dress, wears jacket, has glasses; rough body pose derived from tracking

Table A.10: Human attributes dataset

[6] This dataset has been recorded and post-processed by my master student S. Wehner under my supervision.

List of Figures

List of Tables

Bibliography

W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, and A. E. Sallab (2018). "YOLO3D: End-to-End Real-Time 3D Oriented Object Bounding Box Detection from LiDAR Point Cloud." In: *ECCV Workshops*. Vol. 11131. Lecture Notes in Computer Science. Springer.

C. Álvarez-Aparicio, Á. M. Guerrero-Higueras, M. C. C. Olivera, F. J. Rodríguez-Lera, F. Martín, and V. Matellán (2018). "Benchmark Dataset for Evaluation of Range-Based People Tracker Classifiers in Mobile Robots." In: *Frontiers in Neurorobotics* 11, p. 72. DOI: 10.3389/fnbot.2017.00072.

K. O. Arras, S. Grzonka, M. Luber, and W. Burgard (2008). "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1710–1715. DOI: 10.1109/ROBOT.2008.4543447.

K. O. Arras and W. Burgard, eds. (2002). *Robots in Exhibitions, IROS'02 Workshop Proceedings*. Lausanne, Switzerland.

K. O. Arras, O. M. Mozos, and W. Burgard (2007). "Using Boosted Features for the Detection of People in 2D Range Data." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3402–3407. DOI: 10.1109/ROBOT.2007.363998.

T. Bagautdinov, F. Fleuret, and P. Fua (2015). "Probability occupancy maps for occluded depth images." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Y. Bar-Shalom (1987). *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional.

Y. Bar-Shalom, P. Willett, and X. Tian (2011). *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing.

Y. Bar-Shalom and X.-R. Li (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing.

Y. Bar-Shalom, X. Li, and T. Kirubarajan (2001). *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Wiley. DOI: 10.1002/0471221279.

I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino (2012). "Re-identification with RGB-D Sensors." In: *European Conference on Computer Vision Workshops (ECCVW)*, pp. 433–442.

L. Bazzani, M. Zanotto, M. Cristani, and V. Murino (2015). "Joint Individual-Group Modeling for Tracking." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 37.4, pp. 746–759. DOI: 10.1109/TPAMI.2014.2353641.

C. Becker-Asano, K. O. Arras, and B. Nebel (2014). "Robotic tele-presence with DARYL in the wild." In: *Int. Conf. on Human-Agent Interaction (HAI'14)*. Tsukuba, Japan.

N. Bellotto and H. Hu (2010). "Computationally Efficient Solutions for Tracking People with a Mobile Robot: an Experimental Evaluation of Bayesian Filters." In: *Autonomous Robots (AURO)* 28.4.

N. Bellotto and H. Hu (2009). "Multisensor-based Human Detection and Tracking for Mobile Service Robots." In: *Trans. Sys. Man Cyber. Part B* 39.1, pp. 167–181. DOI: 10.1109/TSMCB.2008.2004050.

N. Bellotto, C. Dondrup, and M. Hanheide (2015). *bayestracking: The Bayes Tracking Library v1.0.5*. DOI: 10.5281/zenodo.15825.

N. Bellotto, S. Cosar, and Z. Yan (2018). "Human Detection and Tracking." In: *Encyclopedia of Robotics*. Ed. by M. H. Ang, O. Khatib, and B. Siciliano. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–10. DOI: 10.1007/978-3-642-41610-1_34-1.

T. Belpaeme (2019). "Social Human-Robot Interaction." In: *Encyclopedia of Robotics*. Ed. by M. H. Ang, O. Khatib, and B. Siciliano. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–5. DOI: 10.1007/978-3-642-41610-1_31-1.

J. Bergstra, D. Yamins, and D. D. Cox (2013). "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures." In: *International Conference on Machine Learning (ICML)*.

K. Bernardin and R. Stiefelhagen (2008). “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics.” In: *Journal on Image and Video Processing* 2008.1, p. 246309. DOI: 10.1155/2008/246309.

R. Best and J. Norton (1997). “A New Model And Efficient Tracker For A Target With Curvilinear Motion.” In: *IEEE Transactions on Aerospace and Electronic Systems* 33.3.

L. Beyer (2020). “Deep Visual Human Sensing with Application in Robotics.” PhD thesis. RWTH Aachen University, Aachen, Germany.

L. Beyer, A. Hermans, and B. Leibe (2015). “Biternion Nets: Continuous Head Pose Regression from Discrete Training Labels.” In: *Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*. Ed. by J. Gall, P. Gehler, and B. Leibe. Cham: Springer International Publishing, pp. 157–168. DOI: 10.1007/978-3-319-24947-6_13.

— (2016). “DROW: Real-Time Deep Learning based Wheelchair Detection in 2D Range Data.” In: *IEEE Robotics and Automation Letters (RA-L)*.

L. Beyer, A. Hermans, T. Linder, K. O. Arras, and B. Leibe (2018). “Deep Person Detection in Two-Dimensional Range Data.” In: *IEEE Robotics and Automation Letters (RA-L)* 3.3, pp. 2726–2733. DOI: 10.1109/LRA.2018.2835510.

S. Blackman and R. Popoli (1999). *Design and Analysis of Modern Tracking Systems*. Artech House.

L. Bo, X. Ren, and D. Fox (2014). “Learning Hierarchical Sparse Features for RGB-(D) Object Recognition.” In: *International Journal of Robotics Research* 33.4.

I. Bogoslavskyi and C. Stachniss (2017). “Efficient Online Segmentation for Sparse 3D Laser Scans.” In: *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, pp. 1–12.

J. Bohren and S. Cousins (2010). “The SMACH High-Level Executive [ROS News].” In: *IEEE Robotics & Automation Magazine* 17.4, pp. 18–20. DOI: 10.1109/MRA.2010.938836.

D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee (2019). “YOLACT: Real-time Instance Segmentation.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

L. D. Bourdev, S. Maji, T. Brox, and J. Malik (2010). “Detecting People Using Mutually Consistent Poselet Activations.” In: *Proc. European Conference on Computer Vision (ECCV)*. Vol. 6316. LNCS.

L. D. Bourdev, S. Maji, and J. Malik (2011). “Describing People: A Poselet-Based Approach to Attribute Classification.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

S. Breuers, L. Beyer, U. Rafi, and B. Leibe (2018). “Detection-Tracking for Efficient Person Analysis: The DetTA Pipeline.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 48–53. DOI: 10.1109/IROS.2018.8594335.

B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora (2019). “Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments.” In: *IEEE Robotics and Automation Letters (RA-L)* 4.4, pp. 4459–4466. DOI: 10.1109/LRA.2019.2929976.

D. Bugental (2000). “Acquisition of the algorithms of social life: a domain-based approach.” In: *Psychol Bull* 126(2), pp. 187–219.

W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun (1998). “The Interactive Museum Tour-Guide Robot.” In: *Proc. AAAI Conference on Artificial Intelligence (AAAI)*.

H. Caesar, J. R. R. Uijlings, and V. Ferrari (2018). “COCO-Stuff: Thing and Stuff Classes in Context.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1209–1218. DOI: 10.1109/CVPR.2018.00132.

H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom (2020). “nuScenes: A multimodal dataset for autonomous driving.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh (2017). “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

R. Caruana (1997). “Multitask Learning.” In: *Machine Learning* 28.1, pp. 41–75. DOI: 10.1023/A:1007379606734.

O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng (2011). “Boosted multi-task learning.” In: *Machine Learning* 85.1, pp. 149–173. DOI: 10.1007/s10994-010-5231-6.

K. Charalampous, I. Kostavelis, and A. Gasteratos (2017). “Recent trends in social aware robot navigation: A survey.” In: *Journal of Robotics & Autonomous Systems* 93, pp. 85–104. DOI: https://doi.org/10.1016/j.robot.2017.03.002.

S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun (2011). “Probabilistic Reasoning for Assembly-Based 3D Modeling.” In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30.4.

T. Chen and C. Guestrin (2016). “XGBoost: A Scalable Tree Boosting System.” In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, pp. 785–794. DOI: 10.1145/2939672.2939785.

W. Choi and S. Savarese (2012). “A Unified Framework for Multi-Target Tracking and Collective Activity Recognition.” In: *Proc. European Conference on Computer Vision (ECCV)*.

C. T. Chou, J.-Y. Li, M.-F. Chang, and L. C. Fu (2011). “Multi-robot cooperation based human tracking system using Laser Range Finder.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 532–537. DOI: 10.1109/ICRA.2011.5980484.

G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera (2019). “Deep learning in video multi-object tracking: A survey.” In: *Neurocomputing*. DOI: https://doi.org/10.1016/j.neucom.2019.11.023.

M. Collins, J. Zhang, P. Miller, and H. Wang (2009). “Full Body Image Feature Representations for Gender Profiling.” In: *9th IEEE Int. Workshop on Visual Surveillance, ICCV 2009 Workshops*.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). “The Cityscapes dataset for Semantic Urban Scene Understanding.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

J. Correa, J. Liu, and G.-Z. Yang (2013). “Real Time People Tracking in Crowded Environments with Range Measurements.” In: *Social Robotics*. Ed. by G. Herrmann, M. Pearson, A. Lenz, P. Bremner, A. Spiers, and U. Leonards. Vol. 8239. Lecture Notes in Computer Science. Springer International Publishing, pp. 471–480.

I. J. Cox and S. L. Hingorani (1996). “An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 18.2, pp. 138–150. DOI: 10.1109/34.481539.

I. J. Cox and M. Miller (1995). “On finding ranked assignments with application to multi-target tracking and motion correspondence.” In: *IEEE Trans. on Aerospace and Elect. Sys.* 31.1.

M. Cristani, G. Paggetti, A. Vinciarelli, L. Bazzani, G. Menegaz, and V. Murino (2011). “Towards Computational Proxemics: Inferring Social Relations from Interpersonal Distances.” In: *IEEE Int. Conf. on Social Computing*. Boston, USA.

J. Cui, H. Zha, H. Zhao, and R. Shibasaki (2007). “Laser-based detection and tracking of multiple people in crowds.” In: *Computer Vision and Image Understanding* 106.2-3.

K. Czarnecki and R. Salay (2018). “Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving.” In: *Computer Safety, Reliability, and Security*. Ed. by B. Gallina, A. Skavhaug, E. Schoitsch, and F. Bitsch. Cham: Springer International Publishing, pp. 439–445.

N. Dalal and B. Triggs (2005). “Histograms of oriented gradients for human detection.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, pp. 886–893. DOI: 10.1109/CVPR.2005.177.

M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg (2019). “Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

B. Della Corte, H. Andreasson, T. Stoyanov, and G. Grisetti (2019). “Unified Motion-Based Calibration of Mobile Multi-Sensor Platforms With Time Delay Estimation.” In: *IEEE Robotics and Automation Letters (RA-L)* 4.2, pp. 902–909. DOI: 10.1109/LRA.2019.2892992.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). “Imagenet: A large-scale hierarchical image database.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255.

Y. Deng, P. Luo, C. C. Loy, and X. Tang (2014). “Pedestrian Attribute Recognition At Far Distance.” In: *Proceedings of the 22nd ACM International Conference on Multimedia*. MM ’14. Orlando, Florida, USA: ACM, pp. 789–792. DOI: 10.1145/2647868.2654966.

Z. Deng and L. J. Latecki (2017). “Amodal Detection of 3D Objects: Inferring 3D Bounding Boxes from 2D Ones in RGB-D Images.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 398–406. DOI: 10.1109/CVPR.2017.50.

J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner (2018). “Deep tracking in the wild: End-to-end tracking using recurrent neural networks.” In: *International Journal of Robotics Research (IJRR)* 37.4-5, pp. 492–512. DOI: 10.1177/0278364917710543.

A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard (2016). “Motion-based detection and tracking in 3D LiDAR scans.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4508–4513. DOI: 10.1109/ICRA.2016.7487649.

L. Ding and A. Yilmaz (2011). “Inferring Social Relations from Visual Concepts.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

P. Dollár, C. Wojek, B. Schiele, and P. Perona (2012). “Pedestrian Detection: An Evaluation of the State of the Art.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 34.

C. Dondrup, N. Bellotto, F. Jovan, and M. Hanheide (2015). “Real-time multisensor people tracking for human-robot spatial interaction.” In: *Workshop on Machine Learning for Social Robotics at IEEE International Conference on Robotics and Automation (ICRA)*.

A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun (2017). “CARLA: An Open Urban Driving Simulator.” In: *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16.

Y. Du, N. J. Hetherington, C. L. Oon, W. P. Chan, C. P. Quintero, E. Croft, and H. F. Machiel Van der Loos (2019). “Group Surfing: A Pedestrian-Based Approach to Sidewalk Robot Navigation.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6518–6524. DOI: 10.1109/ICRA.2019.8793608.

S. Embgen, M. Luber, C. Becker-Asano, M. Ragni, V. Evers, and K. O. Arras (2012). “Robot-specific social cues in emotional body language.” In: *IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*.

A. Ess, B. Leibe, K. Schindler, and L. Van Gool (2008). “A mobile vision system for robust multi-person tracking.” In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8.

A. Ess, B. Leibe, K. Schindler, and L. van Gool (2009). “Robust Multiperson Tracking from a Mobile Platform.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 31.10, pp. 1831–1846. DOI: 10.1109/TPAMI.2009.109.

EUROPA project (2009–2013). *Project website of the EU FP7 publicly funded project EUROPA*. https://europa.informatik.uni-freiburg.de. Last visited on November 6th, 2019.

EUROPA2 project (2014–2016). *Project website of the EU FP7 publicly funded project EUROPA2*. https://europa2.informatik.uni-freiburg.de. Last visited on November 6th, 2019.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2012). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://host.robots.ox.ac.uk/pascal/VOC/voc2012/.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman (2010). “The PASCAL Visual Object Classes (VOC) Challenge.” In: *Int. Journal of Computer Vision* 88.2.

M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara (2018). “Learning to Detect and Track Visible and Occluded Body Joints in a Virtual World.” In: *Proc. European Conference on Computer Vision (ECCV)*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, pp. 450–466.

P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart (2015). “Kinect v2 for mobile robot navigation: Evaluation and modeling.” In: *Proc. International Conference on Advanced Robotics (ICAR)*, pp. 388–394. DOI: 10.1109/ICAR.2015.7251485.

C. Feichtenhofer, A. Pinz, and A. Zisserman (2016). “Convolutional Two-Stream Network Fusion for Video Action Recognition.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer (2019). “Leveraging Heteroscedastic Aleatoric Uncertainties for Robust Real-Time LiDAR 3D Object Detection.” In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1280–1287. DOI: 10.1109/IVS.2019.8814046.

M. Fernandez-Carmona, T. Parekh, and M. Hanheide (2019). *Making the Case for Human-aware Navigation in Warehouses*. Towards Autonomous Robotic Systems—20th Annual Conference, TAROS (extended abstract). Queen Mary University, London, UK.

M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung (2019). “Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends.” In: *ACM Comput. Surv.* 52.2, 43:1–43:44. DOI: 10.1145/3309665.

M. Fiore, H. Khambhaita, G. Milliez, and R. Alami (2015). “An Adaptive and Proactive Human-Aware Robot Guide.” In: *Social Robotics*. Cham: Springer International Publishing, pp. 194–203.

D. R. Forsyth (2019). *Group dynamics*. Seventh edition. Cengage.

M. E. Foster, R. Alami, O. Gestranius, O. Lemon, M. Niemelä, J.-M. Odobez, and A. K. Pandey (2016). “The MuMMER Project: Engaging Human-Robot Interaction in Real-World Public Spaces.” In: *Social Robotics*. Ed. by A. Agah, J.-J. Cabibihan, A. M. Howard, M. A. Salichs, and H. He. Cham: Springer International Publishing, pp. 753–763.

FROG project (2011–2014). *Project website of the EU FP7 publicly funded project FROG*. https://www.frogrobot.eu. Last visited on November 6th, 2019.

A. Gaidon, Q. Wang, Y. Cabon, and E. Vig (2016). “Virtual Worlds as Proxy for Multi-Object Tracking Analysis.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera (2012). “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches.” In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4, pp. 463–484. DOI: 10.1109/TSMCC.2011.2161285.

M. Gao, J. Jiang, G. Zou, V. John, and Z. Liu (2019). “RGB-D-Based Object Recognition Using Multimodal Convolutional Neural Networks: A Survey.” In: *IEEE Access* 7, pp. 43110–43136. DOI: 10.1109/ACCESS.2019.2907071.

A. Geiger, P. Lenz, and R. Urtasun (2012). “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka (2017). “Synthesizing Training Data for Object Detection in Indoor Scenes.” In: *Robotics Science and Systems (RSS)*. DOI: 10.15607/RSS.2017.XIII.043.

J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, and P. Schuberth (2019). *A2D2: AEV Autonomous Driving Dataset*. http://www.a2d2.audi.

F. Girrbach (2015). “People Tracking with a Mobile Robot: Revisited.” MA thesis. Germany: Albert-Ludwigs-Universität Freiburg.

R. Girshick (2015). “Fast R-CNN.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

G. Gkioxari, R. Girshick, and J. Malik (2015). “Actions and Attributes from Wholes and Parts.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 2470–2478. DOI: 10.1109/ICCV.2015.284.

A. Goel, K. T. Ma, and C. Tan (2019). “An End-To-End Network for Generating Social Relationship Graphs.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

K. Granström, M. Baum, and S. Reuter (2017). “Extended Object Tracking: Introduction, Overview, and Applications.” In: *Journal of Advances in Information Fusion* 12.2.

D. Griffiths and J. Boehm (2019). “A Review on Deep Learning Techniques for 3D Sensed Data Classification.” In: *Remote Sensing* 11.12. DOI: 10.3390/rs11121499.

H. Grimmett, R. Triebel, R. Paul, and I. Posner (2016). “Introspective Classification for Robot Perception.” In: *International Journal of Robotics Research (IJRR)* 35.7, pp. 743–762. DOI: 10.1177/0278364915587924.

A. P. Gritti, O. Tarabini, J. Guzzi, G. A. Di Caro, V. Caglioti, L. M. Gambardella, and A. Giusti (2014). “Kinect-based people detection and tracking from small-footprint ground robots.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4096–4103. DOI: 10.1109/IROS.2014.6943139.

G. Groh, A. Lehmann, J. Reimers, M. R. Friess, and L. Schwarz (2010). “Detecting Social Situations from Interaction Geometry.” In: *Proc. of the IEEE Int. Conf. on Social Computing*.

H.-M. Gross, H. Boehme, C. Schroeter, S. Mueller, A. Koenig, C. Martin, M. Merten, and A. Bley (2008). “ShopBot: Progress in developing an interactive mobile shopping assistant for everyday use.” In: *2008 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3471–3478. DOI: 10.1109/ICSMC.2008.4811835.

H.-M. Gross, H. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley (2009). “TOOMAS: Interactive Shopping Guide robots in everyday use - final implementation and experiences from long-term field trials.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2005–2012. DOI: 10.1109/IROS.2009.5354497.

Á. M. Guerrero-Higueras, C. Álvarez-Aparicio, M. C. Calvo Olivera, F. J. Rodríguez-Lera, C. Fernández-Llamas, F. M. Rico, and V. Matellán (2019). “Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments.” In: *Frontiers in Neurorobotics* 12, p. 85. DOI: 10.3389/fnbot.2018.00085.

J. Guerry, B. L. Saux, and D. Filliat (2017). “"Look at this one": Detection sharing between modality-independent classifiers for robotic discovery of people.” In: *Proc. European Conference on Mobile Robotics (ECMR)*. DOI: 10.1109/ECMR.2017.8098679.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger (2017). “On Calibration of Modern Neural Networks.” In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, pp. 1321–1330.

J. Guo, H. He, T. He, L. Lausen, M. Li, H. Lin, X. Shi, C. Wang, J. Xie, S. Zha, A. Zhang, H. Zhang, Z. Zhang, Z. Zhang, S. Zheng, and Y. Zhu (2020). “GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing.” In: *Journal of Machine Learning Research* 21.23, pp. 1–7.

S. Gupta, J. Hoffman, and J. Malik (2016). “Cross Modal Distillation for Supervision Transfer.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

F. Hamidi, M. K. Scheuerman, and S. M. Branham (2018). “Gender Recognition or Gender Reductionism?: The Social Implications of Embedded Gender Recognition Systems.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. Montreal QC, Canada: ACM, 8:1–8:13. DOI: 10.1145/3173574.3173582.

R. Hanten, P. Kuhlmann, S. Otte, and A. Zell (2018). “Robust Real-Time 3D Person Detection for Indoor and Outdoor Applications.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. DOI: 10.1109/ICRA.2018.8461257.

R. Hartley and A. Zisserman (2003). *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press.

M. Hassanin, S. Khan, and M. Tahtali (2018). *Visual Affordance and Function Understanding: A Survey*. arXiv: 1807.06775 [cs.CV].

N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner, R. Ambrus, N. Bore, J. Folkesson, P. Jensfelt, L. Beyer, A. Hermans, B. Leibe, A. Aldoma, T. Faulhammer, M. Zillich, M. Vincze, E. Chinellato, M. Al-Omari, P. Duckworth, Y. Gatsoulis, D. C. Hogg, A. G. Cohn, C. Dondrup, J. Pulido Fentanes, T. Krajnik, J. M. Santos, T. Duckett, and M. Hanheide (2017). "The STRANDS Project: Long-Term Autonomy in Everyday Environments." In: *IEEE Robotics & Automation Magazine* 24.3, pp. 146–156. DOI: 10.1109/MRA.2016.2636359.

C. Hazirbas, L. Ma, C. Domokos, and D. Cremers (2016). "FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-based CNN Architecture." In: *Asian Conference on Computer Vision (ACCV)*.

K. He, G. Gkioxari, P. Dollár, and R. Girshick (2017). "Mask R-CNN." In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

X. He, R. Tharmarasa, T. Kirubarajan, and M. Pelletier (2013). "Modified Murty's Algorithm for Diverse Multitarget Top Hypothesis Extraction." In: *IEEE Transactions on Aerospace and Electronic Systems* 49.1, pp. 602–610. DOI: 10.1109/TAES.2013.6404123.

Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang (2019). "Bounding Box Regression With Uncertainty for Accurate Object Detection." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

D. Helbing and P. Molnár (1995). "Social force model for pedestrian dynamics." In: *Phys. Rev. E* 51 (5), pp. 4282–4286. DOI: 10.1103/PhysRevE.51.4282.

A. Hermans, L. Beyer, and B. Leibe (2017). *In Defense of the Triplet Loss for Person Re-Identification*. arXiv: 1703.07737v2 [cs.CV].

M. J. Hernandez Leon (2019). "Synthetic data generation using game engines for deep learning in robotics." MA thesis. Germany: Institute for Control Engineering of Machine Tools and Manufacturing Units, University of Stuttgart.

J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell (2018). "CyCADA: Cycle Consistent Adversarial Domain Adaptation." In: *International Conference on Machine Learning (ICML)*.

M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll (2013). "The TUM Gait from Audio, Image and Depth (GAID) Database: Multimodal Recognition of Subjects and Traits." In: *Journal of Visual Communication and Image Representation* 25 (1).

O. Hosseini Jafari, D. Mitzel, and B. Leibe (2014). "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

Z. Hu, H. Leung, and M. Blanchette (1997). "Statistical performance analysis of track initiation techniques." In: *Signal Processing, IEEE Transactions on* 45.2, pp. 445–456.

F. Hutter, H. H. Hoos, and K. Leyton-Brown (2011). "Sequential model-based optimization for general algorithm configuration." In: *Learning and Intelligent Optimization*. Springer, pp. 507–523.

M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori (2016). "A Hierarchical Deep Temporal Model for Group Activity Recognition." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Intel Object Analytics module (2017). https://github.com/intel/ros_object_analytics.

JackRabbot project (2015–2020). http://svl.stanford.edu/projects/jackrabbot/. Last visited on November 19th, 2019.

M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu (2015). "Spatial Transformer Networks." In: *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*. NIPS'15. Montreal, Canada: MIT Press, pp. 2017–2025.

R. Jonker and A. Volgenant (1987). "A shortest augmenting path algorithm for dense and sparse linear assignment problems." In: *Computing* 38.4, pp. 325–340. DOI: 10.1007/BF02278710.

M. Joosse (2017). "Investigating positioning and gaze behaviors of social robots: people's preferences, perceptions, and behaviors." PhD thesis. Netherlands: University of Twente. DOI: 10.3990/1.9789036543767.

T. Kanda and H. Ishiguro (2012). *Human-Robot Interaction in Social Robotics*. 1st. Boca Raton, FL, USA: CRC Press.

T. Kanda, D. F. Glas, M. Shiomi, H. Ishiguro, and N. Hagita (2008). “Who Will Be the Customer?: A Social Robot That Anticipates People’s Behavior from Their Trajectories.” In: *Proceedings of the 10th International Conference on Ubiquitous Computing*. UbiComp ’08. Seoul, Korea: ACM, pp. 380–389. DOI: 10.1145/1409635.1409686.

R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet (2019). *Lyft Level 5 AV Dataset 2019*. https://level5.lyft.com/dataset/.

B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi (2013). “Multisensor data fusion: A review of the state-of-the-art.” In: *Information Fusion* 14.1, pp. 28–44. DOI: https://doi.org/10.1016/j.inffus.2011.08.001.

H. Khambhaita, J. Rios-Martinez, and R. Alami (2016). *Head-Body Motion Coordination for Human Aware Robot Navigation*. 9th International workshop on Human-Friendly Robotics (HFR).

C. Kim, F. Li, A. Ciptadi, and J. M. Rehg (2015). “Multiple Hypothesis Tracking Revisited.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 4696–4704. DOI: 10.1109/ICCV.2015.533.

A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar (2019). “Panoptic Segmentation.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

B. Kluge, C. Köhler, and E. Prassler (2001). “Fast and robust tracking of multiple moving objects with a laser range finder.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Seoul, Korea.

M. Kollmitz, A. Eitel, A. Vasquez, and W. Burgard (2019). “Deep 3D perception of people and their mobility aids.” In: *Robotics and Autonomous Systems* 114, pp. 29–40. DOI: 10.1016/j.robot.2019.01.011.

F. Kraus and K. Dietmayer (2019). *Uncertainty Estimation in One-Stage Object Detection*. arXiv: 1905.10296 [cs.CV].

T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch (2013). “Human-aware Robot Navigation: A Survey.” In: *Robotics and Autonomous Systems* 61.12, pp. 1726–1743. DOI: 10.1016/j.robot.2013.05.007.

J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander (2018). “Joint 3D Proposal Generation and Object Detection from View Aggregation.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

T. Kurien (1990). “Issues in the design of practical multitarget tracking algorithms.” In: *Multitarget-Multisensor Tracking: Advanced Applications*. Ed. by Y. Bar-Shalom. Artech House.

M. Kutbi, Y. Chang, B. Sun, and P. Mordohai (2019). “Learning to Navigate Robotic Wheelchairs from Demonstration: Is Training in Simulation Viable?” In: *IEEE International Conference on Computer Vision (ICCV) Workshops*.

E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer (2015). “Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling.” In: *Remote Sensing* 7.

K. Lai, L. Bo, X. Ren, and D. Fox (2011). “A Large-Scale Hierarchical Multi-View RGB-D Object Dataset.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom (2019). “PointPillars: Fast Encoders for Object Detection from Point Clouds.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

B. Lau, K. O. Arras, and W. Burgard (2009). “Tracking groups of people with a multi-model hypothesis tracker.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3180–3185. DOI: 10.1109/ROBOT.2009.5152731.

L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn (2011). “Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker.” In: *ICCV Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*.

L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler (2015). *MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking*. arXiv: 1504.01942v1 [cs.CV].

L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth (2017). *Tracking the Trackers: An Analysis of the State of the Art in Multiple Object Tracking*. arXiv: 1704.02781v1 [cs.CV].

B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool (2008). “Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 30.10.

A. Leigh, J. Pineau, N. Olmedo, and H. Zhang (2015). “Person tracking and following with 2D laser scanners.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 726–733. DOI: 10.1109/ICRA.2015.7139259.

Q. Leng, M. Ye, and Q. Tian (2019). “A Survey of Open-World Person Re-identification.” In: *IEEE Transactions on Circuits and Systems for Video Technology*. DOI: 10.1109/TCSVT.2019.2898940.

H. Leung, Z. Hu, and M. Blanchette (1996). “Evaluation of multiple target track initiation techniques in real radar tracking environments.” In: *IEE Proceedings - Radar, Sonar and Navigation* 143.4, pp. 246–254. DOI: 10.1049/ip-rsn:19960404.

B. Lewandowski, J. Liebner, T. Wengefeld, S. Müller, and H.-M. Gross (2019). “Fast and Robust 3D Person Detector and Posture Estimator for Mobile Robotic Applications.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4869–4875. DOI: 10.1109/ICRA.2019.8793712.

B. Li, X.-C. Lian, and B.-L. Lu (2012). “Gender Classification by Combining Clothing, Hair and Facial Component Classifiers.” In: *Neurocomputing* 76.1.

D. Li, Z. Zhang, X. Chen, and K. Huang (2019). “A Richly Annotated Pedestrian Dataset for Person Retrieval in Real Surveillance Scenarios.” In: *IEEE Transactions on Image Processing* 28.4, pp. 1575–1590. DOI: 10.1109/TIP.2018.2878349.

J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli (2017). “Dual-Glance Model for Deciphering Social Relationships.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 2669–2678. DOI: 10.1109/ICCV.2017.289.

Y. Li, C. Huang, C. C. Loy, and X. Tang (2016). “Human Attribute Recognition by Deep Hierarchical Contexts.” In: *Proc. European Conference on Computer Vision (ECCV)*.

Y.-L. Li, S. Zhou, X. Huang, L. Xu, Z. Ma, H.-S. Fang, Y. Wang, and C. Lu (2019). “Transferable Interactiveness Knowledge for Human-Object Interaction Detection.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Y. Li, C. Huang, and R. Nevatia (2009). “Learning to associate: Hybridboosted multi-target tracker for crowded scene.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). “Microsoft COCO: Common Objects in Context.” In: *Proc. European Conference on Computer Vision (ECCV)*, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

W. Lin, Y. Li, H. Xiao, J. See, J. Zou, H. Xiong, J. Wang, and T. Mei (2019). “Group Reidentification with Multigrained Matching and Integration.” In: *IEEE Transactions on Cybernetics*, pp. 1–15. DOI: 10.1109/TCYB.2019.2917713.

T. Linder, S. Wehner, et al. (2015a). “Real-time full-body human gender recognition in (RGB)-D data.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3039–3045. DOI: 10.1109/ICRA.2015.7139616.

T. Linder, D. Griesser, N. Vaskevicius, and K. Arras (2018). “Towards Accurate 3D Person Detection and Localization from RGB-D in Cluttered Environments.” In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’18) – Workshop on Robotics for Logistics in Warehouses and Environments Shared with Humans*.

T. Linder (2015). *SPENCER Human Attribute Recognition*. https://github.com/spencer-project/spencer_human_attribute_recognition.

T. Linder and K. O. Arras (2014). “Multi-model hypothesis tracking of groups of people in RGB-D data.” In: *Proc. 17th Int. Conf. Information Fusion (FUSION)*.

T. Linder and S. Breuers (2016). *SPENCER People Tracking Framework*. https://github.com/spencer-project/spencer_people_tracking.

T. Linder, S. Breuers, B. Leibe, and K. O. Arras (2016). "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5512–5519. DOI: 10.1109/ICRA.2016.7487766.

T. Linder, F. Girrbach, and K. O. Arras (2015). "Towards a Robust People Tracking Framework for Service Robots in Crowded, Dynamic Environments." In: *Assistance and Service Robotics Workshop (ASROB-15) at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

T. Linder, S. Wehner, and K. O. Arras (2015b). *SRL Human Attributes Dataset*. http://srl.informatik.uni-freiburg.de/human_attributes_dataset.

T. Linder, M. J. Hernandez Leon, N. Vaskevicius, and K. O. Arras (2019). "Towards Training Person Detectors for Mobile Robots using Synthetically Generated RGB-D Data." In: *Computer Vision and Pattern Recognition (CVPR) 2019 Workshop on 3D Scene Generation*.

M. Lindstrom and J. Eklundh (2001). "Detecting and tracking moving objects from a mobile platform using a laser range scanner." In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3, pp. 1364–1369. DOI: 10.1109/IROS.2001.977171.

L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen (2019). "Deep Learning for Generic Object Detection: A Survey." In: *International Journal of Computer Vision*. DOI: 10.1007/s11263-019-01247-4.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg (2016). "SSD: Single Shot MultiBox Detector." In: *Proc. European Conference on Computer Vision (ECCV)*.

W. Liu, T. Xia, J. Wan, Y. Zhang, and J. Li (2012). "RGB-D Based Multi-attribute People Search in Intelligent Visual Surveillance." In: *Advances in Multimedia Modeling*. Ed. by K. Schoeffmann, B. Merialdo, A. Hauptmann, C.-W. Ngo, Y. Andreopoulos, and C. Breiteneder. Vol. 7131. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 750–760.

X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang (2017). "HydraPlus-Net: Attentive Deep Features for Pedestrian Analysis." In: *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 350–359. DOI: 10.1109/ICCV.2017.46.

Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai (2020). "TANet: Robust 3D Object Detection from Point Clouds with Triple Attention." In: *Proc. AAAI Conference on Artificial Intelligence (AAAI)*.

I. Loshchilov and F. Hutter (2017). "SGDR: Stochastic Gradient Descent with Warm Restarts." In: *International Conference on Learning Representations (ICLR)*.

C. Lu, H. Su, Y. Li, Y. Lu, L. Yi, C.-K. Tang, and L. J. Guibas (2018). "Beyond Holistic Object Recognition: Enriching Image Understanding With Part States." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Luber (2014). "People Tracking under Social Constraints." PhD thesis. University of Freiburg.

M. Luber and K. O. Arras (2013). "Multi-Hypothesis Social Grouping and Tracking for Mobile Robots." In: *Robotics Science and Systems (RSS)*.

M. Luber, L. Spinello, and K. O. Arras (2011). "People Tracking in RGB-D Data With Online-Boosted Target Models." In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

M. Luber, G. D. Tipaldi, and K. O. Arras (2011a). "Better Models For People Tracking." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China.

— (2011b). "Place-Dependent People Tracking." In: *International Journal of Robotics Research (IJRR)* 30.3.

M. Luber, K. O. Arras, C. Plagemann, and W. Burgard (2009). "Classifying Dynamic Objects: An Unsupervised Learning Approach." In: *Autonomous Robots (AURO)* 26.2-3, pp. 141–151.

M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras (2010). "People Tracking with Human Motion Predictions from Social Forces." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim (2014). *Multiple Object Tracking: A Literature Review*. arXiv: 1409.7618 [cs.CV].

W. Maddern, G. Pascoe, C. Linegar, and P. Newman (2017). "1 Year, 1000km: The Oxford RobotCar Dataset." In: *International Journal of Robotics Research (IJRR)* 36.1, pp. 3–15. DOI: 10.1177/0278364916679498.

R. P. S. Mahler (2003). "Multitarget Bayes filtering via first-order multitarget moments." In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4, pp. 1152–1178. DOI: 10.1109/TAES.2003.1261119.

R. P. S. Mahler (2007). *Statistical Multisource-Multitarget Information Fusion*. Artech House.

— (2014). *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House.

S. Manen, M. Gygli, D. Dai, and L. Van Gool (2017). "PathTrack: Fast Trajectory Annotation With Path Supervision." In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

N. Mansfeld, M. Hamad, M. Becker, A. G. Marin, and S. Haddadin (2018). *Safety Map: A Robotics Safety Evaluation and Safe Robot Design*. Workshop on Autonomous Robot Design at the IEEE International Conference on Robotics and Automation (ICRA).

C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross (2006). "Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking." In: *Journal of Robotics & Autonomous Systems* 54.9. Selected papers from the 2nd European Conference on Mobile Robots (ECMR 2005), pp. 721–728.

R. Martín-Martín, H. Rezatofighi, A. Shenoi, M. Patel, J. Gwak, N. Dass, A. Federman, P. Goebel, and S. Savarese (2019). *JRDB: A Dataset and Benchmark for Visual Perception for Navigation in Human Environments*. arXiv: 1910.11792 [cs.CV].

E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan (1998). "Interacting multiple model methods in target tracking: a survey." In: *IEEE Transactions on Aerospace and Electronic Systems* 34.1, pp. 103–123. DOI: 10.1109/7.640267.

J. McCormac, A. Handa, S. Leutenegger, and A. J.Davison (2017). "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?" In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

O. Mees, A. Eitel, and W. Burgard (2016). "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments." In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 151–156. DOI: 10.1109/IROS.2016.7759048.

D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt (2020). "XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera." In: vol. 39. 4. DOI: 10.1145/3386569.3392410.

A. Milan, K. Schindler, and S. Roth (2013). "Challenges of Ground Truth Evaluation of Multi-target Tracking." In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 735–742.

A. Milan, S. Rezatofighi, A. Dick, I. Reid, and K. Schindler (2017). "Online Multi-Target Tracking using Recurrent Neural Networks." In: *Proc. AAAI Conference on Artificial Intelligence (AAAI)*.

A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid (2015). "Joint Tracking and Segmentation of Multiple Targets." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler (2016). "MOT16: A Benchmark for Multi-Object Tracking." In: arXiv: 1603.00831v2 [cs.CV].

A. Milioto, L. Mandtler, and C. Stachniss (2019). "Fast Instance and Semantic Segmentation Exploiting Local Connectivity, Metric Learning, and One-Shot Detection for Robotics." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5481–5487. DOI: 10.1109/ICRA.2019.8793593.

D. Mitzel and B. Leibe (2012). "Close-Range Human Detection and Tracking for Head-Mounted Cameras." In: *Proc. British Machine Vision Conference (BMVC)*.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller (2013). “Playing Atari with Deep Reinforcement Learning.” In: *NIPS Deep Learning workshop*.

S. Molina, G. Cielniak, and T. Duckett (2019). “Go with the Flow: Exploration and Mapping of Pedestrian Flow Patterns from Partial Observations.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

M. Mononen (2009). *Recast Navigation*. https://github.com/recastnavigation/recastnavigation.

S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou (2017). “AgeDB: The First Manually Collected, In-The-Wild Age Database.” In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz (2010). “The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics.” In: *PLoS ONE* 5.4.

MuMMER project (2016–2020). *Project website of the EU H2020 publicly funded project MuMMER*. http://mummer-project.eu/. Last visited on November 6th, 2019.

M. Munaro, A. Basso, A. Fossati, L. Van Gool, and E. Menegatti (2014). “3D reconstruction of freely moving persons for re-identification with a depth sensor.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4512–4519. DOI: 10.1109/ICRA.2014.6907518.

M. Munaro, F. Basso, and E. Menegatti (2012). “Tracking people within groups with RGB-D data.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

M. Munaro, F. Basso, and E. Menegatti (2016). “OpenPTrack.” In: *Journal of Robotics & Autonomous Systems* 75.PB, pp. 525–538. DOI: 10.1016/j.robot.2015.10.004.

M. Munaro, A. Fossati, A. Basso, E. Menegatti, and L. Van Gool (2014). “One-Shot Person Re-identification with a Consumer Depth Camera.” In: *Person Re-Identification*. Advances in Computer Vision and Pattern Recognition. Springer.

M. Munaro, C. Lewis, D. Chambers, P. Hvass, and E. Menegatti (2016). “RGB-D Human Detection And Tracking For Industrial Environments.” In: *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*. Cham: Springer International Publishing, pp. 1655–1668. DOI: 10.1007/978-3-319-08338-4_119.

M. Munaro and E. Menegatti (2014). “Fast RGB-D people tracking for service robots.” In: *Autonomous Robots (AURO)* 37.3, pp. 227–242. DOI: 10.1007/s10514-014-9385-0.

C. B. Ng, Y. H. Tay, and B.-M. Goi (2013). “A Convolutional Neural Network for Pedestrian Gender Recognition.” In: *Int. Symposium on Neural Networks (ISNN)*. Vol. 7951. LNCS.

S. Oh, S. Russell, and S. Sastry (2009). “Markov Chain Monte Carlo Data Association for Multi-Target Tracking.” In: *IEEE Transactions on Automatic Control* 54.3, pp. 481–497. DOI: 10.1109/TAC.2009.2012975.

B. Okal and K. O. Arras (2016). “Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2889–2895. DOI: 10.1109/ICRA.2016.7487452.

B. Okal and K. O. Arras (2014). “Towards Group-Level Social Activity Recognition for Mobile Robots.” In: *IROS 2014 Workshop on Assistance and Service Robotics in a Human Environment*.

B. Okal, R. Triebel, and K. O. Arras (2016). “Real-time Social Activity Detection for Mobile Service Robots.” In: *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*.

K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan (2018). “Localization Recall Precision (LRP): A New Performance Metric for Object Detection.” In: *Proc. European Conference on Computer Vision (ECCV)*.

P. Ondruska, J. Dequaire, D. Zeng Wang, and I. Posner (2016). “End-to-End Tracking and Semantic Segmentation Using Recurrent Neural Networks.” In: *Robotics Science and Systems (RSS), Workshop on Limits and Potentials of Deep Learning in Robotics*.

T. Ophoff, K. Van Beeck, and T. Goedemé (2019). “Exploring RGB+Depth Fusion for Real-Time Object Detection.” In: *Sensors* 19.4. DOI: 10.3390/s19040866.

A. Ošep, W. Mehner, P. Voigtlaender, and B. Leibe (2018). “Track, then Decide: Category-Agnostic Vision-based Multi-Object Tracking.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

L. Palmieri and K. O. Arras (2014). “A novel RRT extend function for efficient and smooth mobile robot motion planning.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 205–211. DOI: 10.1109/IROS.2014.6942562.

L. Palmieri, A. Rudenko, and K. O. Arras (2017). “A Fast Random Walk Approach to Find Diverse Paths for Robot Navigation.” In: *IEEE Robotics and Automation Letters (RA-L)* 2.1, pp. 269–276. DOI: 10.1109/LRA.2016.2602240.

L. Palmieri (2018). “Efficient and smooth motion planning techniques for nonholonomic wheeled robots.” PhD thesis. Germany: University of Freiburg. DOI: 10.6094/UNIFR/16223.

A. K. Pandey and R. Gelin (2018). “A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind.” In: *IEEE Robotics & Automation Magazine* 25.3, pp. 40–48. DOI: 10.1109/MRA.2018.2833157.

C. Pantofaru (2010). *ROS leg_detector package*. https://wiki.ros.org/leg_detector.

D. J. Papageorgiou and M. R. Salpukas (2009). “The Maximum Weight Independent Set Problem for Data Association in Multiple Hypothesis Tracking.” In: *Optimization and Cooperative Control Strategies*. Ed. by M. J. Hirsch, C. W. Commander, P. M. Pardalos, and R. Murphey. Springer Berlin Heidelberg, pp. 235–255.

S. Pellegrini, A. Ess, and L. van Gool (2010). “Improving data association by joint modeling of pedestrian trajectories and groupings.” In: *Proc. European Conference on Computer Vision (ECCV)*.

J. C. Platt (1999). “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.” In: *Advances in Large-Margin Classifiers*. MIT Press, pp. 61–74.

E. Prassler, J. Scholz, and A. Elfes (2000). “Tracking Multiple Moving Objects for Real-Time Robot Navigation.” In: *Autonomous Robots (AURO)* 8.2, pp. 105–116. DOI: 10.1023/A:1008997110534.

G. W. Pulford (2005). “Taxonomy of multiple target tracking methods.” In: *IEE Proceedings - Radar, Sonar and Navigation* 152.5, pp. 291–304. DOI: 10.1049/ip-rsn:20045064.

C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas (2018). “Frustum PointNets for 3D Object Detection from RGB-D Data.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas (2017). “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas (2017). “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.” In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 5099–5108.

Z. Qin and C. R. Shelton (2012). “Improving Multi-target Tracking via Social Grouping.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

U. Rafi, B. Leibe, J. Gall, and I. Kostrikov (2016). “An Efficient Convolutional Network for Human Pose Estimation.” In: *Proc. British Machine Vision Conference (BMVC)*.

O. A. I. Ramírez, H. Khambhaita, R. Chatila, M. Chetouani, and R. Alami (2016). “Robots learning how and where to approach people.” In: *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 347–353. DOI: 10.1109/ROMAN.2016.7745154.

O. A. I. Ramírez, G. Varni, M. Andries, M. Chetouani, and R. Chatila (2016). “Modeling the dynamics of individual behaviors for group detection in crowds using low-level features.” In: *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1104–1111. DOI: 10.1109/ROMAN.2016.7745246.

J. Redmon and A. Farhadi (2017). “YOLO9000: Better, Faster, Stronger.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690.

— (2018). *YOLOv3: An Incremental Improvement*. arXiv:1804.02767. arXiv: 1804.02767 [cs.CV].

D. B. Reid (1979). “An algorithm for tracking multiple targets.” In: *IEEE Transactions on Automatic Control* 24.6.

S. Ren, K. He, R. Girshick, and J. Sun (2017). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 39.6, pp. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

S. Reuter, B. Vo, B. Vo, and K. Dietmayer (2014). “The Labeled Multi-Bernoulli Filter.” In: *IEEE Transactions on Signal Processing* 62.12, pp. 3246–3260. DOI: 10.1109/TSP.2014.2323064.

S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid (2015). “Joint Probabilistic Data Association Revisited.” In: *Proc. IEEE International Conference on Computer Vision (ICCV)*.

S. R. Richter, V. Vineet, S. Roth, and V. Koltun (2016). “Playing for Data: Ground Truth from Computer Games.” In: *Proc. European Conference on Computer Vision (ECCV)*.

B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo (2011). “A Metric for Performance Evaluation of Multi-Target Tracking Algorithms.” In: *Signal Processing, IEEE Transactions on* 59.7, pp. 3452–3457.

I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodriguez, and E. Jauregi (2019). “Video Activity Recognition: State-of-the-Art.” In: *Sensors* 19.14. DOI: 10.3390/s19143160.

O. Ronneberger, P. Fischer, and T. Brox (2015). “U-Net: Convolutional Networks for Biomedical Image Segmentation.” In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, pp. 234–241.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez (2016). “The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

R. Rothe, R. Timofte, and L. Van Gool (2018). “Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks.” In: *Int. Journal of Computer Vision* 126.2, pp. 144–157. DOI: 10.1007/s11263-016-0940-3.

A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras (2020). “Human motion trajectory prediction: a survey.” In: *International Journal of Robotics Research (IJRR)* 39.8, pp. 895–935. DOI: 10.1177/0278364920917446.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei (2015). “ImageNet Large Scale Visual Recognition Challenge.” In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

R. B. Rusu, N. Blodow, and M. Beetz (2009). “Fast Point Feature Histograms (FPFH) for 3D registration.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3212–3217. DOI: 10.1109/ROBOT.2009.5152473.

S. Šabanović, C. Bennett, W. Chang, and L. Huber (2013). “PARO robot affects diverse interaction modalities in group sensory therapy for older adults with dementia.” In: *IEEE Int Conf Rehabil Robot*. DOI: 10.1109/ICORR.2013.6650427.

N. Sarafianos, X. Xu, and I. A. Kakadiaris (2018). “Deep Imbalanced Attribute Classification Using Visual Attention Aggregation.” In: *Proc. European Conference on Computer Vision (ECCV)*, pp. 708–725. DOI: 10.1007/978-3-030-01252-6_42.

I. Sárándi, T. Linder, K. O. Arras, and B. Leibe (2018a). “How Robust is 3D Human Pose Estimation to Occlusion?” In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’18) – Workshop on Robotic Co-workers 4.0: Human Safety and Comfort in Human-Robot Interactive Social Environments*.

— (2018b). “Synthetic Occlusion Augmentation with Volumetric Heatmaps for the 2018 ECCV PoseTrack Challenge on 3D Human Pose Estimation.” In: *European Conference on Computer Vision Workshops (ECCVW)*. Winner of ECCV 2018 PoseTrack 3D Challenge.

K. Schindler, U. James, and H. Wang (2006). “Perspective n-view Multibody Structure-and-Motion through Model Selection.” In: *Proc. European Conference on Computer Vision (ECCV)*.

K. Schindler, A. Ess, B. Leibe, and L. Van Gool (2010). “Automatic Detection and Tracking of Pedestrians from a Moving Stereo Rig.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 65.6.

D. Schulz, W. Burgard, D. Fox, and A. Cremers (2001). “Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

D. Schulz, W. Burgard, D. Fox, and A. B. Cremers (2003). “People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters.” In: *International Journal of Robotics Research (IJRR)* 22.2, pp. 99–116.

J. Sell and P. O’Connor (2014). “The Xbox One System on a Chip and Kinect Sensor.” In: *IEEE Micro* 34.2, pp. 44–53. DOI: 10.1109/MM.2014.9.

F. Setti and M. Cristani (2019). “Evaluating the Group Detection Performance: The GRODE Metrics.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 41.3, pp. 566–580. DOI: 10.1109/TPAMI.2018.2806970.

S. Shah, D. Dey, C. Lovett, and A. Kapoor (2017). “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles.” In: *Field and Service Robotics (FSR)*.

S. Shi, X. Wang, and H. Li (2019). “PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

J. Shotton, A. Fitzgibbon, A. Blake, A. Kipman, M. Finocchio, B. Moore, and T. Sharp (2011). “Real-Time Human Pose Recognition in Parts from a Single Depth Image.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

T. Shu, M. S. Ryoo, and S.-C. Zhu (2016). “Learning Social Affordance for Human-robot Interaction.” In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI’16. New York, New York, USA: AAAI Press, pp. 3454–3461.

R. Siegwart, K. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet, G. Ramel, G. Terrien, and N. Tomatis (2003). “Robox at Expo.02: A Large-Scale Installation of Personal Robots.” In: *Journal of Robotics & Autonomous Systems* 42.3-4.

M. Simon, S. Milz, K. Amende, and H.-M. Gross (2019). “Complex-YOLO: An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds.” In: *European Conference on Computer Vision Workshops (ECCVW)*. Ed. by L. Leal-Taixé and S. Roth, pp. 197–209.

R. Socher, B. Huval, B. P. Bath, C. D. Manning, and A. Y. Ng (2012). “Convolutional-Recursive Deep Learning for 3D Object Classification.” In: *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*.

F. Solera, S. Calderara, and R. Cucchiara (2016). “Socially Constrained Structural Learning for Groups Detection in Crowd.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.5, pp. 995–1008. DOI: 10.1109/TPAMI.2015.2470658.

SPENCER project (2013–2016). *Project website of the EU FP7 publicly funded project SPENCER*. https://www.spencer.eu. Last visited on November 6th, 2019.

L. Spinello, R. Triebel, and R. Siegwart (2010). “Multiclass Multimodal Detection and Tracking in Urban Environments.” In: *International Journal of Robotics Research (IJRR)* 29.12, pp. 1498–1515. DOI: 10.1177/0278364910377533.

L. Spinello and K. O. Arras (2011). “People Detection in RGB-D Data.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

L. Spinello, M. Luber, and K. O. Arras (2011). "Tracking People in 3D Using a Bottom-Up Top-Down People Detector." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

L. Spinello and R. Siegwart (2008). "Human Detection using Multimodal and Multidimensional Features." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

STRANDS project (2013–2017). *Project website of the EU FP7 publicly funded project STRANDS*. http://strands.acin.tuwien.ac.at. Last visited on November 6th, 2019.

C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian (2016). "Deep Attributes Driven Multi-camera Person Re-identification." In: *Proc. European Conference on Computer Vision (ECCV)*, pp. 475–491.

P. Sudowe and B. Leibe (2011). "Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video." In: *Computer Vision Systems*.

P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov (2019). *Scalability in Perception for Autonomous Driving: An Open Dataset Benchmark*. arXiv: 1912.04838 [cs.CV].

Q. Sun, B. Schiele, and M. Fritz (2017). "A Domain Based Approach to Social Relation Recognition." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel (2018). "Implicit 3D Orientation Learning for 6D Object Detection from RGB Images." In: *Proc. European Conference on Computer Vision (ECCV)*.

C. S. Swaminathan, T. P. Kucner, M. Magnusson, L. Palmieri, and A. Lilienthal (2018). "Down the CLiFF: Flow-Aware Trajectory Planning under Motion Pattern Uncertainty." In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7403–7409.

R. Szeliski (2010). *Computer Vision: Algorithms and Applications*. 1st. Berlin, Heidelberg: Springer-Verlag.

J. Tang, X. Liu, H. Cheng, and K. Robinette (2011). "Gender Recognition Using 3-D Human Body Shapes." In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 41.6.

A. Teichman and S. Thrun (2011). "Tracking-Based Semi-Supervised Learning." In: *Robotics Science and Systems (RSS)*. Los Angeles, CA, USA. DOI: 10.15607/RSS.2011.VII.042.

S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz (1999). "MINERVA: A Tour-Guide Robot that Learns." In: *KI-99: Advances in Artificial Intelligence*. Ed. by W. Burgard, A. B. Cremers, and T. Cristaller. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 14–26. DOI: 10.1007/3-540-48238-5_2.

J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel (2017). "Domain randomization for transferring deep neural networks from simulation to the real world." In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30. DOI: 10.1109/IROS.2017.8202133.

W. van Toll, R. Triesscheijn, M. Kallmann, R. Oliva, N. Pelechano, J. Pettré, and R. Geraerts (2016). "A Comparative Study of Navigation Meshes." In: *Proceedings of the 9th International Conference on Motion in Games*. MIG '16. Burlingame, California: ACM, pp. 91–100. DOI: 10.1145/2994258.2994262.

N. Tomatis (2011). "BlueBotics: Navigation for the Clever Robot [Entrepreneur]." In: *IEEE Robotics & Automation Magazine* 18.2, pp. 14–16. DOI: 10.1109/MRA.2011.941629.

N. Tomatis, G. Terrien, R. Piguet, D. Burnier, S. Bouabdallah, K. O. Arras, and R. Siegwart (2003). "Designing a secure and robust mobile interacting robot for the long term." In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 3, 4246–4251 vol.3. DOI: 10.1109/ROBOT.2003.1242256.

E. Topp and H. Christensen (2005). "Tracking for Following and Passing Persons." In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Alberta, Canada.

P. Trahanias, W. Burgard, A. Argyros, D. Hahnel, H. Baltzakis, P. Pfaff, and C. Stachniss (2005). "TOURBOT and WebFAIR: Web-operated mobile robots for tele-presence in populated exhibitions." In: *IEEE Robotics Automation Magazine* 12.2, pp. 77–89. DOI: 10.1109/MRA.2005.1458329.

R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, H. Hung, O. A. I. Ramírez, M. Joosse, H. Khambhaita, T. Kucner, B. Leibe, A. J. Lilienthal, T. Linder, M. Lohse, M. Magnusson, B. Okal, L. Palmieri, U. Rafi, M. van Rooij, and L. Zhang (2016). "SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports." In: *Field and Service Robotics: Results of the 10th International Conference*. Ed. by D. S. Wettergreen and T. D. Barfoot. Cham: Springer International Publishing, pp. 607–622. DOI: 10.1007/978-3-319-27702-8_40.

N. A. Tsokas and K. J. Kyriakopoulos (2012). "Multi-robot multiple hypothesis tracking for pedestrian tracking." In: *Autonomous Robots (AURO)* 32.1, pp. 63–79. DOI: 10.1007/s10514-011-9259-7.

S. Tzafestas (2015). *Sociorobot World: A Guided Tour for All*. 1st edition. Springer Publishing.

J. Uhrig, E. Rehder, B. Fröhlich, U. Franke, and T. Brox (2018). "Box2Pix: Single-Shot Instance Segmentation by Assigning Pixels to Object Boxes." In: *IEEE Intelligent Vehicles Symposium (IV)*.

S. Vahora and N. Chauhan (2019). "Deep neural network model for group activity recognition using contextual relationship." In: *Engineering Science and Technology, an International Journal* 22.1, pp. 47–54. DOI: https://doi.org/10.1016/j.jestch.2018.08.010.

G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid (2017). "Learning from Synthetic Humans." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

S. Vascon and L. Bazzani (2017). "Chapter 3 - Group Detection and Tracking Using Sociological Features." In: *Group and Crowd Behavior for Computer Vision*. Ed. by V. Murino, M. Cristani, S. Shah, and S. Savarese. Academic Press, pp. 29–66. DOI: https://doi.org/10.1016/B978-0-12-809276-7.00004-7.

S. Vascon, E. Z. Mequanint, M. Cristani, H. Hung, M. Pelillo, and V. Murino (2016). "Detecting conversational groups in images and sequences: A robust game-theoretic approach." In: *Computer Vision and Image Understanding* 143, pp. 11–24. DOI: https://doi.org/10.1016/j.cviu.2015.09.012.

A. Vasquez, M. Kollmitz, A. Eitel, and W. Burgard (2017). "Deep Detection of People and their Mobility Aids for a Hospital Robot." In: *Proc. European Conference on Mobile Robotics (ECMR)*.

B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. I. Osborne, R. Mahler, and B.-T. Vo (2015). "Multitarget Tracking." In: *Wiley Encyclopedia*, pp. 1–25. DOI: 10.1002/047134608X.W8275.

P. Voigtlaender, M. Krause, A. Ošep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe (2019). "MOTS: Multi-Object Tracking and Segmentation." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Volkhardt, C. Weinrich, and H. M. Gross (2013). "Multi-modal people tracking on a mobile companion robot." In: *Proc. European Conference on Mobile Robotics (ECMR)*, pp. 288–293. DOI: 10.1109/ECMR.2013.6698856.

M. Vrigkas, C. Nikou, and I. A. Kakadiaris (2015). "A Review of Human Activity Recognition Methods." In: *Frontiers in Robotics and AI* 2, p. 28. DOI: 10.3389/frobt.2015.00028.

D. Z. Wang, I. Posner, and P. Newman (2015). "Model-free detection and tracking of dynamic objects with 2D lidar." In: *International Journal of Robotics Research (IJRR)* 34.7, pp. 1039–1063. DOI: 10.1177/0278364914562237.

G. Wang, A. Gallagher, J. Luo, and D. Forsyth (2010). "Seeing people in social context: recognizing people and social relationships." In: *Proc. European Conference on Computer Vision (ECCV)*.

H.-J. Wang, Y.-L. Lin, C.-Y. Huang, Y.-L. Hou, and W. Hsu (2013). "Full body human attribute detection in indoor surveillance environment using color-depth information." In: *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pp. 383–388.

X. Wang, S. Zheng, R. Yang, B. Luo, and J. Tang (2019). *Pedestrian Attribute Recognition: A Survey*. arXiv: 1901.07474 [cs.CV].

Z. Wang and K. Jia (2019). “Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin (2018). “Deep Reasoning with Knowledge Graph for Social Relationship Understanding.” In: *Int. Conf. on Artificial Intelligence (IJCAI)*. AAAI Press, pp. 1021–1028.

O. Wasenmüller and D. Stricker (2017). “Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision.” In: *Asian Conference on Computer Vision (ACCV) Workshops*. Ed. by C.-S. Chen, J. Lu, and K.-K. Ma. Cham: Springer International Publishing, pp. 34–45.

M. Weber, J. Luiten, and B. Leibe (2019). *Single-Shot Panoptic Segmentation*. arXiv: 1911.00764 [cs.CV].

C. Weinrich, T. Wengefeld, C. Schröter, and H.-M. Gross (2014). “People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features.” In: *Proc. 23rd IEEE Int. Symp. Robot and Human Interactive Communication (RO-MAN)*, pp. 767–773. DOI: 10.1109/ROMAN.2014.6926346.

X. Weng and K. Kitani (2020). “3D Multi-Object Tracking: A Baseline and New Evaluation Metrics.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

T. Wengefeld, B. Lewandowski, D. Seichter, L. Pfennig, and H. Gross (2019). “Real-time Person Orientation Estimation using Colored Pointclouds.” In: *Proc. European Conference on Mobile Robotics (ECMR)*. DOI: 10.1109/ECMR.2019.8870914.

T. Wengefeld, S. Mueller, B. Lewandowski, and H. M. Gross (2019). “A Multi Modal People Tracker for Real Time Human Robot Interaction.” In: *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*.

T. Wengefeld, M. Eisenbach, T. Q. Trinh, and H.-M. Gross (2016). “May I be your Personal Coach? Bringing Together Person Tracking and Visual Re-identification on a Mobile Robot.” In: *ISR 2016 - 47st International Symposium on Robotics*.

T. Wiedemeyer (2014). *IAI Kinect2*. https://github.com/code-iai/iai_kinect2. University Bremen: Institute for Artificial Intelligence.

N. Wojke and D. Paulus (2016). “Global data association for the Probability Hypothesis Density filter using network flows.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 567–572. DOI: 10.1109/ICRA.2016.7487180.

N. Wojke, A. Bewley, and D. Paulus (2017). “Simple Online and Realtime Tracking with a Deep Association Metric.” In: *IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.

N. Wojke, R. Memmesheimer, and D. Paulus (2017). “Joint operator detection and tracking for person following from mobile platforms.” In: *Proc. International Conference on Information Fusion (FUSION)*, pp. 1–8. DOI: 10.23919/ICIF.2017.8009746.

Y. Xu, X. Zhou, S. Chen, and F. Li (2019). “Deep learning for multiple object tracking: a survey.” In: *IET Computer Vision* 13.4, pp. 355–368. DOI: 10.1049/iet-cvi.2018.5598.

Y. Xu, A. Osep, Y. Ban, R. Horaud, L. Leal-Taixe, and X. Alameda-Pineda (2020). “How To Train Your Deep Multi-Object Tracker.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Y. Yan, Y. Mao, and B. Li (2018). “SECOND: Sparsely Embedded Convolutional Detection.” In: *Sensors* 18.10. DOI: 10.3390/s18103337.

Z. Yan, T. Duckett, and N. Bellotto (2017). “Online Learning for Human Classification in 3D LiDAR-based Tracking.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Z. Yan, L. Sun, T. Duckett, and N. Bellotto (2018). “Multisensor Online Transfer Learning for 3D LiDAR-based Human Detection with a Mobile Robot.” In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain.

A. Yilmaz, O. Javed, and M. Shah (2006). “Object Tracking: A Survey.” In: *ACM Comput. Surv.* 38.4. DOI: 10.1145/1177352.1177355.

T. Yu, S. N. Lim, K. A. Patwardhan, and N. Krahnstoever (2009). “Monitoring, recognizing and discovering social networks.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

B. Yuan, A. Wu, and W. Zheng (2018). “Does A Body Image Tell Age?” In: *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2142–2147. DOI: 10.1109/ICPR.2018.8545590.

X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli (2018). “A LiDAR Point Cloud Generator: From a Virtual World to Autonomous Driving.” In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ICMR ’18. Yokohama, Japan: ACM, pp. 458–464. DOI: 10.1145/3206025.3206080.

H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, and D.-S. Chen (2019). “A Comprehensive Survey of Vision-Based Human Action Recognition Methods.” In: *Sensors* 19.5. DOI: 10.3390/s19051005.

J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang (2016). “RGB-D-based action recognition datasets: A survey.” In: *Pattern Recognition* 60, pp. 86–105. DOI: https://doi.org/10.1016/j.patcog.2016.05.019.

L. Zhang, Y. Li, and R. Nevatia (2008). “Global data association for multi-object tracking using network flows.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: 10.1109/CVPR.2008.4587584.

N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev (2014). “PANDA: Pose Aligned Networks for Deep Attribute Modeling.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

S. Zhang, R. Benenson, and B. Schiele (2017). “CityPersons: A Diverse Dataset for Pedestrian Detection.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li (2018). “Single-Shot Refinement Neural Network for Object Detection.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li (2019). *Bag of Freebies for Training Object Detection Neural Networks*. arXiv:1902.04103.

Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen (2019). “Seeing Isn’t Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors.” In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’19. London, United Kingdom: ACM, pp. 1989–2004. DOI: 10.1145/3319535.3354259.

Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu (2018). *Object Detection with Deep Learning: A Review*. arXiv: 1807.05511 [cs.CV].

Y. Zhou and O. Tuzel (2018). “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection.” In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Ziaeefard and R. Bergevin (2015). “Semantic human activity recognition: A literature review.” In: *Pattern Recognition* 48.8, pp. 2329–2345. DOI: https://doi.org/10.1016/j.patcog.2015.03.006.

C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox (2018). “3D Human Pose Estimation in RGBD Images for Robotic Task Learning.” In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.

Z. Zou, Z. Shi, Y. Guo, and J. Ye (2019). *Object Detection in 20 Years: A Survey*. arXiv: 1905.05055 [cs.CV].

All links were last followed in April 2020, unless noted otherwise.