DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 854

Uyen Dao-Xuan Nguyen

3D Pedestrian Tracking Using Neighbourhood Constraints

München 2020

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5266-6

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover
ISSN 0174-1454, Nr. 358, Hannover 2020

Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 854

3D Pedestrian Tracking Using Neighbourhood Constraints

Von der Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

Vorgelegt von

Dipl.-Ing. Uyen Dao-Xuan Nguyen

Geboren am 08.12.1989 in Tien Giang, Vietnam

München 2020

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5266-6

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover
ISSN 0174-1454, Nr. 358, Hannover 2020

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759

E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender: Prof. Dr.-Ing. Franz Rottensteiner

Referent: Prof. Dr.-Ing. habil. Christian Heipke

Korreferenten: Prof. Dr.-Ing. habil. Monika Sester
Prof. Dr.-Ing. Michael Yang (Twente, Netherlands)

Tag der mündlichen Prüfung: 29.05.2020

ISSN 0065-5325 ISBN 978-3-7696-5266-6

Abstract

Tracking pedestrians based on visual sensors has many diverse applications, among them autonomous driving. Through tracking, not only the position of pedestrians but also their temporal movement can be obtained. This information helps vehicles and robots to sense their surrounding environment and thus they can plan appropriate reactions. In addition to obtaining a high recall, maintaining the consistency of tracked trajectories during data association is one of the most crucial issues of any tracker.

Following the tracking-by-detection paradigm, a new method called 3D-TLSR (3D pedestrian tracking using local structure refinement) is presented in this thesis aiming at improving the accuracy, reliability, and consistency of tracked trajectories. The contributions of this work are fourfold. First, a framework combining both, 2D image and 3D object space information, to track multiple pedestrians in 3D object space is presented, in which tracking, detection, and prediction are all considered and improved to enhance tracking results in terms of completeness, correctness, and reliability. Second, a hierarchical association approach is introduced to improve the consistency of trajectories by utilising geometry cues, which is carried out in two steps: (1) targets whose assignments have a high probability of correctness are selected as anchors and (2) prior knowledge about the geometry changes of the anchors is used to correct unreliable assignments of detections with their nearby trajectories in 3D space. Additionally, the tracking-to-confirm-detection (TCD) approach is introduced to address low-quality detection results so that both, completeness and correctness of trajectories, can be improved during tracking. Third, a simple approach to estimate and correct the velocity of a tracked person is proposed based on the relationship of moving targets, which allows missed detections to be better retrieved. Fourth, a new dataset called MuVi, consisting of image sequences of pedestrians from three different viewpoints with a large overlapping has been acquired, which can be employed for either single view or multi-view collaborative tracking. The new dataset offers additional data for the community to promote research achievements theoretically and practically.

Experiments on different datasets are carried out to illustrate the advantages and weaknesses of the proposed tracking method and its individual component. Experimental results on the well known KITTI tracking benchmark, the ETHMS dataset, as well as a self-generated MuVi dataset show that the proposed tracker yields comparable results to other state-of-the-art methods and gives the best online result among all investigated approaches. On the ETHMS dataset, our approach obtains the best results with large margins for most tracking metrics. These findings confirm the effectiveness and generalization potential of the proposed tracking method.

Keywords 3D pedestrian tracking, tracking-confirm-detection, online association, linear programming, local structure constraints, missed detection recovery

Zusammenfassung

Die auf visuellen Sensoren basierende Fußgängerverfolgung findet in vielen verschiedenen Bereichen Anwendung, einschließlich dem des autonomen Fahrens. Die Verfolgung liefert dabei nicht nur die Position der Fußgänger, sondern auch deren Bewegung über die Zeit. Diese Informationen helfen Fahrzeugen und Robotern, ihre Umgebung zu erfassen und ermöglichen ihnen so, notwendige Reaktionen zu planen. Neben einer hohen Detektionsrate ist die Wahrung der Konsistenz nachverfolgter Trajektorien während der Datenzuordnung eines der Hauptprobleme für jede Methode zur Fußgängerverfolgung.

Dem Tracking-by-Detection-Paradigma folgend wird in dieser Arbeit unter dem Namen 3D-TLSR (3D-Fußgängerverfolgung mit lokaler Strukturverfeinerung) eine neue Methode vorgestellt, die darauf abzielt, Genauigkeit, Zuverlässigkeit und Konsistenz von nachverfolgten Trajektorien zu verbessern. Die vorliegende Arbeit beinhaltet dafür vier verschiedene Beiträge. Zunächst wird ein Framework vorgeschlagen, das sowohl 2D-Bild- als auch 3D-Objektrauminformationen kombiniert, um mehrere Fußgänger im 3D-Objektraum zu verfolgen. Dabei werden Verfolgung, Erkennung und Vorhersage berücksichtigt und optimiert, um die Ergebnisse im Sinne von Vollständigkeit, Korrektheit und Zuverlässigkeit zu verbessern. Zweitens wird ein Ansatz zur hierarchischen Zuordnung eingeführt, um die Konsistenz von Trajektorien durch Verwendung von geometrischen Hinweisen zu verbessern. Dies erfolgt in zwei Schritten: (1) Ziele, deren Zuordnungen mit hoher Wahrscheinlichkeit korrekt sind, werden als Anker ausgewählt und (2) Vorkenntnisse hinsichtlich geometrischer Änderungen dieser Anker werden verwendet, um unzuverlässige Zuordnungen von Detektionen zu benachbarten Trajektorien im 3D-Raum zu korrigieren. Darüber hinaus wird ein TCD-Ansatz (Tracking-to-Confirm-Detection) eingeführt, um dem Problem entgegenzuwirken, welches aus qualitativ schlechten Erkennungen resultiert. Damit kann sowohl die Vollständigkeit als auch die Korrektheit der Trajektorien während der Verfolgung verbessert werden. Drittens wird ein einfacher Ansatz zur Schätzung und Korrektur der Geschwindigkeit einer nachverfolgten Person vorgeschlagen, welcher auf der Beziehung zwischen bewegten Zielen basiert und fehlende Detektionen ausgleicht. Viertens werden Bildsequenzen von Fußgängern aus drei verschiedenen Perspektiven mit großem Überlappungsbereich erfasst und in Form des MuVi-Datensatzes vorgestellt. Dieser neue Datensatz kann zur Nachverfolgung auf Basis einer einzelnen oder mehrerer verschiedener Perspektiven verwendet werden und soll die wissenschaftliche Gemeinschaft bei theoretischer wie praktischer Forschung unterstützen.

Durch Experimente auf unterschiedlichen Datensätzen werden die Vor- und Nachteile der vorgeschlagenen Methodik und ihrer einzelnen Komponenten veranschaulicht. Experimentelle Ergebnisse auf dem bekannten KITTI-Tracking-Benchmark, dem ETHMS-Datensatz, sowie auf dem selbst erstellten Datensatz MuVi zeigen, dass der vorgeschlagene Ansatz dem Stand der Technik entspricht und das beste Online-Ergebnis aller untersuchten Methoden liefert. Auf dem ETHMS-

Datensatz erzielt für die meisten Tracking-Metriken mit großem Abstand die besten Ergebnisse. Diese Resultate bestätigen die Wirksamkeit und Allgemeingültigkeit der vorgeschlagenen Methodik zur Fußgängerverfolgung.

Schlüsselwörter 3D-Fußgängerverfolgung, Nachverfolgung bestätigt Erkennung, Online-Zuordnung, lineare Programmierung, lokale Strukturbeschränkungen, Wiederherstellung fehlender Erkennungen

Symbols

General notations

$\mathbb{R}^n$	the n-dimensional Euclidean space
$\|\quad\|$	absolute value, number of elements in a set
$\|\|\quad\|\|_{L_2}$	L_2 norm
σ_x	standard deviation of x
Σ_{xx}	covariance matrix of vector x
$p(x)$	marginal probabilty of x
$p(x\|y)$	conditional probabilty of x given y
$\mathcal{N}$	normal distribution
μ_x	mean value of x
$E\langle . \rangle$	expected value
$\odot$	element-wise multiplication

Localization

(Ω)	ground plane
ξ	disparity map
P	foot position of a pedestrian in 3D object space
I	foot position of a pedestrian in image space
B	bounding box of a detection

ϱ	detection confidence value
u, v	image coordinates
d	disparity value
$\mathcal{M}$	binary mask
ϵ	threshold value
$\mathcal{H}$	histogram
Ped_H, Ped_W	average of pedestrian height and width
$\zeta_{\mathcal{M}_s}$	ratio between the number of pixels in an instance segmentation mask and its bounding box
ζ_B	ratio between the height and width of a bounding box
c_u, c_v	image principle point coordinates
f	camera focal length
$Base$	base line of a stereo system
Z	depth value calculated from 3D point cloud
$Z^+_{H/W}$	depth value predicted from height or width of a bounding box

Tracking

$\mathcal{D}$	set of detections
$\mathcal{T}$	set of trajectories
D	detection
S	state vector
S^+	predicted state vector
S^*	updated state vector
ψ	state transition matrix
τ	trajectory

A	coefficient matrix
c	indicator vector
$\Gamma_{\mathcal{A}}$	appearance similarity
$\Gamma_{\mathcal{G}}$	geometry similarity
W	vector of association weight
w_i^j	association weight between detection i and trajectory j
ρ, θ, ν	weights of different terms in association weight
$gate_{3D}$	3D association gate
$gate_{2D}$	2D association gate
$\mathcal{L}$	regression line
F	measurement model
J_A	Jacobian matrix of A
v_X, v_Z	velocity in X and Z direction of a tracked target
a_X, a_Z	acceleration in X and Z direction of a tracked target

Table of Contents

1 Introduction

The human visual system is capable of capturing information about interesting objects like position, type, and interaction, accurately within an extremely short time. In contrast, this task is highly challenging for computer vision systems. In such systems, cameras act as the eyes to capture images and software algorithms take responsibility for analysing and providing necessary information for further applications. With the support of these systems, human effort in processing huge amounts of images, which is expensive and less stable in the long run, can be reduced or completely avoided. Despite constant development and progress in the fields of photogrammetry and computer vision, the performance of a computer system still cannot reach the human ability. One of the problems is the perception of motion at the object-level over time (Rasouli et al., 2019; Huang et al., 2019).

Derived from the development of applications related to autonomous driving, traffic safety, robotics, etc., pedestrians are one of the most momentous objects to be tracked. Today, with advanced technologies of computational vision systems in terms of both hardware and software, pedestrians, in principle, can be localized and tracked automatically in image sequences with or without prior information about the captured scenes. Tracking allows vehicles and robots not only to know where pedestrians probably appear in the scene but also to anticipate their moving directions and behaviours, which are crucial factors for planning their moving paths and safe navigation (Rasouli and Tsotsos, 2019). Though a substantial amount of studies have been carried out to tackle the problem, tracking pedestrians correctly and robustly still requires extensive improvements to deal with difficulties coming from various sources. First, pedestrians cannot be considered as rigid bodies, they constantly carry out flexible and articulated movements. Second, the surrounding illumination conditions and the visible complicated background change over time. These factors result in incomplete, incorrect, and noisy detections as well as significant changes of pedestrians' appearance. Moreover, when pedestrians appear in crowds, their projections in images can be occluded by the others. This also poses problems to assign a pedestrian detection to its corresponding detections in other image frames. All the aforementioned challenges usually lead to two main problems in tracking: missed detections and identity switches. Last but not least, though accurate and reliable 3D geometry trajectories are required by many real-world applications, most of the existing literature is targeted at improving the completeness and consistency of 2D trajectories. In summary, applying detection and tracking results to practical applications

requires significant quality of generated trajectories, which is still far from what has been accomplished (Leal-Taixé et al., 2017). Motivated by these challenges, this thesis deals with tracking pedestrians in 3D object space with high reliability and accuracy using stereo images.

Besides the development of novel and advantageous algorithms, the provision of public datasets also contributes to significantly promote research achievements theoretically and practically. Data sets are means to evaluate the accuracy, robustness, as well as the generalization potential of approaches, which allows the strong and weak points of a suggested method to be thoroughly analysed. Thus, current difficulties and challenges can be emphasized and untangled by the research community. Encouraged by this fact, a 3D pedestrian tracking dataset named multi-views (MuVi) was created within the scope of this thesis, in which stereo cameras are utilised to acquire the movements of pedestrians. Furthermore, to enable collaborative tracking by fusing information from multiple camera systems, the scenes were captured from three different viewpoints of a junction. To the best of our knowledge, at the time of writing (18-Feb-2020), no similar dataset is publicly available.

1.1 Problem statement

Tracking-by-detection is a well-known and widely used remedy in the state-of-the-art tracking literature (Xu et al., 2019), in which the tracking task is decomposed into two separate stages: detection and data association. Most of the studies following this approach concentrate on concatenating detections across image frames to form consistent trajectories for interesting objects. The data association task can become extremely complicated in crowded groups, especially when the tracking is carried out in the 2D image domain, which suffers the problem of dimensionality reduction. Moreover, for autonomous driving applications, the 3D position is essential information for a vehicle to plan its path. Hence, tracking in the image domain is neither sufficient nor effective.

In this study, the problem of pedestrian tracking is investigated using stereo images acquired from moving cameras in a probabilistic manner. The ultimate aim is to obtain correct 3D trajectories with high localization accuracy and completeness by combining both 2D image and 3D stereoscopic information. The state of a target at each epoch is accompanied by its uncertainty, which accounts for the precision of the estimated trajectory in terms of localization. This uncertainty information is vital for real-world applications in making decisions and responding to events. The difficulties of tracking are exposed both in the detection and the association stage. In a detector result, together with an increase in recall also the number of false positives (FPs) rises up. Consequently, choosing only observations with a high probability of correctness results in losing true positives (TPs). In contrast, taking into account also incorrect detections as input for tracking

causes more complexity and difficulty for the association. Due to problems such as erroneous input results and ambiguities in appearance or position, the association can easily fail under non-optimal conditions. Last but not least, the behaviour of pedestrians is sometimes unpredictable, which makes modelling their motion difficult, especially when important information is neglected such as undetected nearby pedestrians.

Based on specific characteristics of pedestrians such as size and moving behaviour, a number of filters are developed to eliminate wrong detections, which help to increase the accuracy of tracking results and to reduce the complications due to incorrect and incomplete inputs for the later stages. In order to maintain accurate identities for tracked trajectories, local geometry constraints among pedestrians in groups are employed to enhance association results. This idea has been explored using 2D image information (Yoon et al., 2016), yet it cannot help to completely understand the real-world geometry in 3D space. Therefore, in this work, 3D point clouds obtained from stereoscopic images are employed to model the relationship among pedestrians, which enables the inference of geometry constrains between them both, in 2D image and 3D object space. Combining those constraints with appearance cues, the accuracy of association and tracking tasks can be improved. To this end, the motion of a person is modelled by taking advantage of the relationship between pedestrians. In this approach, the moving direction and speed of a tracked pedestrian can be corrected and updated according to his/her friends. With a correct motion model, missed detections of a target can be recovered so that not only the recall value is improved but also the fragmentation of tracked trajectories is reduced.

1.2 Research objectives and contributions

The primary goal of this study is to develop an online tracker that can accurately and robustly localize and track multiple pedestrians on the street level in 3D object space using stereo images, and which yields results at least on par with the scientific state-of-the-art. For this purpose, several crucial issues of tracking including improving the recall of tracked people, enhancing the accuracy and reliability of generated trajectories are endeavoured and developed.

To achieve these research objectives, several contributions have been made in this thesis:

- A multi-person tracking framework is introduced to track pedestrians in world coordinates by employing both, 2D images and 3D stereoscopic information. Using stereo images, methods are proposed to model the scene and estimate pedestrian positions in 3D object space. The appearance of pedestrians in image space is utilised for detection and spatio-temporal features comparison.

- A hierarchical association approach to improve the re-identification accuracy of tracked tar-

gets by employing relationships in 3D space among nearby pedestrians, which is divided into two steps: (1) determining trajectories whose assignments are strongly believed to be correct, which are called anchors and (2) using local geometry constraints between the anchors and their nearby trajectories in 3D space to correct unreliable assignments in the first step. Additionally, the tracking-confirm-detection (TCD) approach is suggested to cope with the problem of low quality detection results so that high recall and small false alarm values of detections during tracking can be obtained.

- A method to reliably estimate and assess the motion of pedestrians is explored. In addition, a so-called friend relationship to correct pedestrian velocity and improve trajectory prediction is defined, which endorses the interpretation of the motion model for tracked pedestrians. Consequently, detections missed by the detector can be retrieved through the prediction step.

- A dataset containing image sequences of pedestrians from three different stereo rigs with a large overlapping area is created. This dataset, therefore, can be used to carry out experiments either for mono view tracking or for collaboration and fusion of images in multi-view tracking.

1.3 Outline of the thesis

The rest of this thesis is arranged as follows. Following this introduction is the presentation of fundamental theories for the thesis in Chapter 2. Existing literature related to this work is reviewed in Chapter 3, covering four primary aspects of the tracking problem, namely general tracking approaches, object detection methods, tracking-by-detection, and motion modelling. The details of the proposed tracker are given in Chapter 4. Particularly, Section 4.1 presents the general pipeline of the developed tracking approach and defines the relationship between pedestrians and the transition state of a trajectory, followed by the explanation of the detection and post-processing methods in Section 4.2. The association optimization and its involved cues are illustrated in Section 4.3. Section 4.4 describes in detail the suggested velocity estimation and missed detection prediction methods. Section 4.5 provides an implementation of an extended Kalman filter to smooth trajectories. Extensive experimental results are reported in Chapter 5. This chapter focuses on analysing three subjects, consisting of component optimization, method evaluation, and performance of the tracking approach compared to state-of-the-art methods. These results and their implications are discussed in Chapter 6. Finally, this thesis is concluded by an outlook for future works in Chapter 6.

2 Basics

This chapter presents fundamental theories and methods which are utilized to develop the tracking approach in this dissertation. The basic formulation and solution of linear programming, which is commonly used in data association optimization is described in Section 2.1. The architecture of the mask R-CNN detector which is employed to detect pedestrians in images, is presented in Section 2.2, followed by the description of TriNet in Section 2.3 which is exploited as a feature extractor for pedestrian appearance. Section 2.4 provides the theory of the social force model. On the ground of this, various motion models are designed to predict behaviours of pedestrians while they are moving. Finally, the fundamentals of Kalman filtering are presented in Section 2.5. This filter is often used in an object tracking approach to smooth the resulting trajectories.

2.1 Linear programming

The term linear programming (LP) can be traced back to the late 1940s and was first introduced by Dantzig (1998). Until now, this set of algorithms has been widely adopted to optimize (finding the maximum or minimum) a linear function subject to a set of constraints which can be either linear equalities or inequalities. Following (Bazaraa et al., 2011), a basic formulation of this problem can be depicted as follows:

$$\text{Minimize :} \quad c_1x_1 + c_2x_2 + ... + c_nx_n \tag{2.1.1}$$

$$\begin{aligned} \text{Subject to :} \quad & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \geq b_2 \\ & \dots \\ & a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \geq b_m \\ & x_i \geq 0 \quad i = 1, \dots, n \end{aligned} \quad , \tag{2.1.2}$$

in which the row vector $c = [c_1, c_2, ..., c_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the *cost coefficient vector* and $x = [x_1, x_2, ..., x_n] \in \mathbb{R}^n$ is *decision vector*. x needs to be optimized to minimize the objective function in Equa-

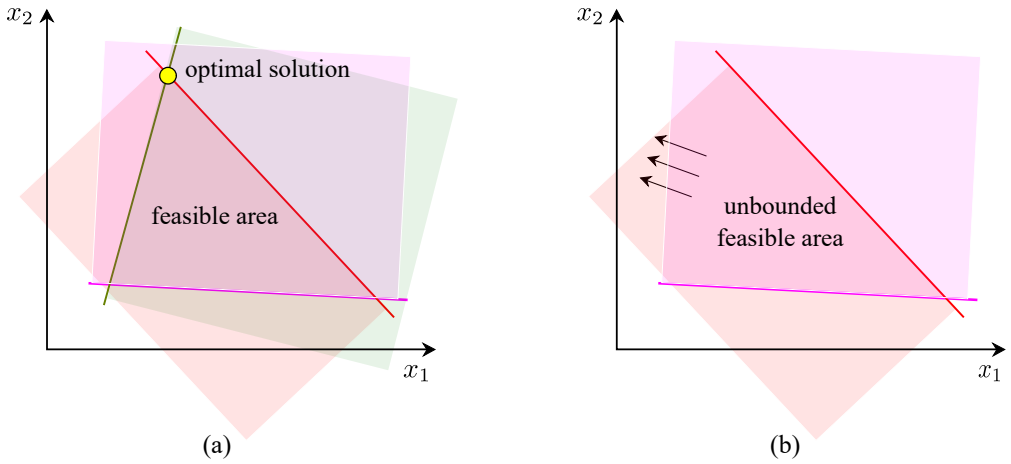


Figure 2.1: An example illustration of bounded (a) and unbounded (b) feasible area in 2-dimensional space, adapted from (Leal-Taixé, 2014).

tion (2.1.1) and satisfy constraints in Equation (2.1.2). $A \in \mathbb{R}^{mn}$ is the *constraint matrix*:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

and $b = [b_1, b_2, ..., b_m]$ is the *right-hand-side vector*. Each inequality constraint $a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \geq b_i$ is a half space in $\mathbb{R}^n$.

The linear programming can be expressed in short form as:

$$\min \quad \{cx : x \in \mathbb{R}^n, \quad Ax \geq b \quad x_i \geq 0 \quad i = 1, \dots, n\} \tag{2.1.3}$$

A solution $\hat{x} \in \mathbb{R}^n$ complying with the condition $A\hat{x} \geq b$ is called *feasible solution*. A problem is *feasible* if there is at least one *feasible solution* existing for it, otherwise, it is *infeasible*. The *feasible region* of an LP problem is formed by all feasible points and is a convex polytope as it is the intersection of half-spaces. If this region is finite and bounded, the problem is called *bounded*.

A *feasible* $x^* \in \mathbb{R}^n$ is *optimal* if $cx^* < c\hat{x}$ for all existing *feasible solutions* $\hat{x} \in \mathbb{R}^n$. It has been proven that if an LP is *feasible* and *bounded*, its *optimal solution* is one of the vertices of the *feasible area* (Bazaraa et al., 2011).

An inequality can be easily converted into an equality equation by simply adding non-negative surplus or slack variables. For instance, the constraint $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ is equivalent to $\sum_{j=1}^{n} a_{ij}x_j - x_{n+1} = b_i$ with $x_{n+1} \geq 0$. An LP is said to be in *standard* from if all constraints are equalities and all variables are non-negative. On the other hand, if all restrictions are constructed by inequality equations, the LP has *canonical* form. By transforming inequalities into equations, an LP problem can be converted from *canonical* to *standard* and vice versa.

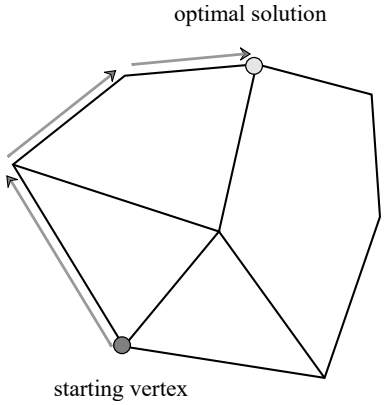


Figure 2.2: A visual exemplar of the simplex method, adapted from (Leal-Taixé, 2014).

Simplex method

Although it is known that the optimal solution of an LP problem lies on one of its feasible area vertices, exhaustively checking all of them is not an efficient way because usually the number of vertices in an LP problem is very large. In practice, the simplex method which was described in (Bazaraa et al., 2011) is extensively applied to solve this problem in standard form. The algorithm first starts with a vertex of the feasible region and moves along edges to another vertex until it reaches the optimal solution. The current solution only moves to one of its adjacent vertices if this makes the objective function improve its value so that the problem can converge. The two primary aspects which need to be inspected in this algorithm are how to evaluate whether a solution is optimal or not without checking the objective function value of other vertices and how to move to a better vertex so that the optimal solution can be obtained.

Consider an LP in standard form:

$$\min \quad \{cx : x \in \mathbb{R}^n, \quad Ax = b \quad x_i \geq 0 \quad i = 1, \ldots, n\}$$

Suppose that rank $(A, b) = $ rank $(A) = m$, B is an $m \times m$ invertible matrix, and N is an $m \times (n - m)$ matrix such that $A = [B, N]$.

Then, $x = \binom{x_B}{x_N}$, in which $x_B = B^{-1}b = \bar{b}$, $x_N = 0$ and satisfies the equation $Ax = b$ is called a *basic solution* of the LP. The component x_B contains *basic variables* and x_N includes *non-basic variables*.

If $B^{-1}b \geq 0$, then x is a *basic feasible solution*. The objective value z at x can be rewritten as:

$$\begin{aligned} z &= cx \\ &= (c_B \quad c_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} \\ &= c_B x_B + c_N x_N \\ &= c_B B^{-1} b \end{aligned} \tag{2.1.4}$$

It can be proven that the collection of *basic feasible solutions* are equivalent to a set of extreme points (i.e. vertices of a feasible area) (Dantzig, 1998) and the procedure of finding the optimal solution with an initial basic solution $x = \binom{x_B}{x_N}$ is carried out as follows:

1. Let:
$$\begin{aligned} z_k - c_k &= \max_{j \in J}(z_j - c_j) \\ z_j &= c_B B^{-1} a_j \end{aligned}, \tag{2.1.5}$$
in which j is an index of the *non-basic* variables in x_N whose $|x_N| = J$. a_j is the column j of matrix A. If $(z_k - c_k) \leq 0$, the current *basic feasible solution* x is the *optimal solution*. Otherwise, x_k is called the *entering variable* and the operation continues with step 2.

2. If $y_k = B^{-1}a_k \leq 0$, it is concluded that the optimal solution is unbounded.

3. r is the index of the *blocking variable* x_{B_r} based on the minimum ratio test:
$$\frac{\bar{b}_r}{y_{rk}} = \min_{1 \leq i \leq m} \{\frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0\} , \tag{2.1.6}$$
B is updated as a_{B_r} is replaced by a_k. Then repeat step 1.

In the worst case, the complexity of the simplex method can be exponential (Klee and Minty, 1972). Nevertheless, the simplex method often performs extremely well in practice. It is observed to usually converge within a number of iterations which linearly increases with the input dimensions. In other words, the simplex method has polynomial-time average-case complexity under various probability distributions. Moreover, the running time of this algorithm is assured to be sub-exponential $O(mn^2 + e^{O\sqrt{n \log n}})$ once some randomized pivot rules are applied (Matoušek et al., 1996).

Integer programming

In many practical applications, fractional solutions are not reasonable and acceptable. Thus, another variance of LP called integer programming (IP) are employed to optimize solely integer solutions x. A IP has similar form to an LP as follows:

$$\begin{aligned} \text{minimize} \quad & cx \\ \text{subject to} \quad & Ax \geq b, \\ & x \geq 0, \\ \text{with} \quad & x \in \mathbb{Z}^n . \end{aligned} \tag{2.1.7}$$

The computational complexity of IP is NP-hard and thus much higher than LP. While the simplex method can effectively solve LPs, it is not suitable for IP problems. Simply rounding the solution obtained by the simplex method may even not be a feasible of an IP (see Figure 2.3). Nevertheless, the solutions of an LP and its IP are observed to be highly correlated:

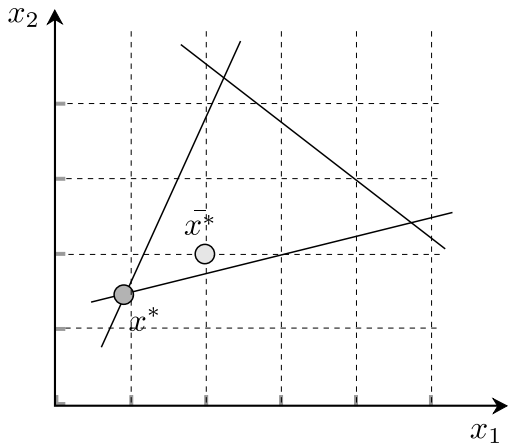


Figure 2.3: An example illustration of the optimal solutions x^* and $\bar{x^*}$ for an LP and the corresponding IP.

- The optimum objective value Z^* of an LP is the lower or upper bound for the objective of its corresponding IP, depending on whether the objective function is to be minimized or maximized.
- If an LP is feasible, so is its IP.

Several techniques have been proposed to solve the IP utilizing the above observations, two well-known approaches are branch-and-bound and cutting plane (Wolsey and Nemhauser, 1999). In the branch-and-bound method, the algorithm of finding an optimal solution is carried out in following steps:

- Find the solution for the corresponding LP using simplex method.
- Select a variable x_i that has fractional value x_i^* and divide the current problem into two sub-problems by adding one of the two constraints: $x_i < x_i^*$ and $x_i > x_i^*$ to the original problem. This procedure is called branching.
- Repeat step (1) and (2) for the sub-problems until either a branch is infeasible or an integer solution is obtained.

The branching routine is finished after a finite number of steps, yet requires a lot of computational effort. To reduce the number of branches, either the upper bound or lower bound Z^* can be used to terminate a branch if its objective value does not satisfy the bounding condition.

The cutting plane method literally adds additional constraints (i.e. *cuts*) into an LP to eliminate non-integer solutions in the feasible area. The cuttings are repeated until the optimal solution of the LP is integer. There are a number of algorithms for finding cuts, the one introduced by Gomory (Gomory, 1958) is one of the most common and prominent ones.

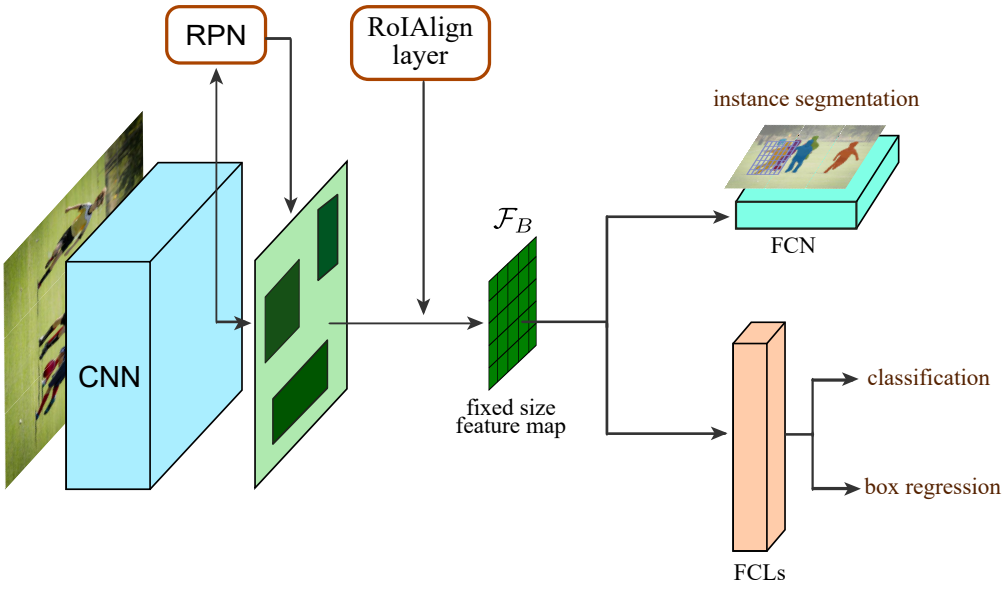


Figure 2.4: General network architecture of Mask R-CNN, adapted from (He et al., 2017).

In practice, the branch-and-bound algorithm usually works better than the cutting plane algorithm and converges fast. Nevertheless, in the worst case, the effort for convergence can grow exponentially with the problem size. Both, the branch-and-bound and the cutting plane methods are guaranteed to converge with in a number of finite steps.

2.2 Mask R-CNN

Mask R-CNN is a neural network introduced in (He et al., 2017) to simultaneously solve both, object detection and instance segmentation. The general architecture of this network is depicted in Figure 2.4. Mask R-CNN is trained in an end-to-end manner and has three main branches: region proposal, object classification and bounding box (BB) regression, and instance mask segmentation.

The region proposal network searches for all possible regions, i.e. a set of rectangles, in an image that can contain objects. First, a feature map of the whole input image is calculated using the convolution and pooling layers. At each position in the feature map, a sliding window is used to obtain n proposal BBs with different size and height-to-width ratio. Each BB has a score representing how likely it contains an object. For each proposal box, a fixed size feature map $\mathcal{F}_B$ is extracted employing the RoIAlign layer (see below). After obtaining the $\mathcal{F}_B$, mask R-CNN carries out three tasks at once as follows:

- The $\mathcal{F}_B$ is fed into a sequence of fully connected layers that are divided into two sibling output layers: one delivers a classification in term of discrete probability distribution ϱ over

$(k+1)$ object types including the background; the other layer outputs four BB coordinate offsets for each class (box regression in Figure 2.4).

- Another branch employs a fully convolutional networks (Long et al., 2015) to produce k instance binary masks $m \times m$, one for each proposal region. Then, the mask that matches with the predicted object type is scaled up to the region of interest (RoI) size. Since the instance mask needs a precise spatial layout to map between the feature map and the RoI in the original image, a RoIAlign layer is developed to preserve the explicit per-pixel spatial correspondence during the mask generation step.

During training, a multi-task loss is computed to train the whole network end-to-end:

$$L = L_{cls} + L_{box} + L_{mask} \ , \tag{2.2.1}$$

where L_{cls} is the classification loss, L_{box} is the bounding box loss, and L_{mask} is the loss for the instance segmentation mask, which are computed as follows:

$$L_{cls} = -\log(p_i) \ , \tag{2.2.2}$$

p_i is the classification probability for the ground truth (GT) class i (i.e. the detection confidence score ϱ) which is derived from the soft-max classification function.

$$\begin{gathered} L_{box}(t, t^*) = \sum_i (p_i^*) \sum_{q \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_q - t_q^*) \\ t_x = (x - x_a)/w_a \ , \quad t_y = (y - y_a)/h_a \\ t_w = \log(w/w_a) \ , \quad t_h = \log(h/h_a) \\ t_x^* = (x^* - x_a)/w_a \ , \quad t_y^* = (y^* - y_a)/h_a \\ t_w^* = \log(w^*/w_a) \ , \quad t_h^* = \log(h^*/h_a) \end{gathered} \ , \tag{2.2.3}$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{, otherwise} \end{cases} \tag{2.2.4}$$

where $\{x, y, w, h\}$ are the predicted BB coordinates and size which are returned by mask R-CNN for its proposal $\{x_a, y_a, w_a, h_a\}$ and $\{x^*, y^*, w^*, h^*\}$ is the GT. The term p_i^* illustrates that only predicted BB corresponding to the correct object class i are considered.

$$L_{mask} = -\frac{1}{m^2} \sum_{i=0}^{m} \sum_{j=0}^{m} \mathcal{M}_{ij} \log({\mathcal{M}^+}_{ij}) + (1 - \mathcal{M}_{ij}) \log(1 - {\mathcal{M}^+}_{ij}) \tag{2.2.5}$$

$\mathcal{M}_{ij}$ is a binary value of pixel (i, j) in the instance GT mask $\mathcal{M}$. The value of pixels in the predicted mask $\mathcal{M}^+$ range from 0.0 to 1.0. During the inference, $\mathcal{M}^+$ is binarized using the threshold of 0.5

2.3 TriNet

TriNet is introduced in (Hermans et al., 2017) to solve the problem of person re-identification (Re-Id) using a convolutional neural network (CNN) and triplet loss. For that purpose, the network is trained to learn an embedding function f_θ to extract person visual properties. In the embedding space, images of the same person should be closer to each other than those from different persons. Mathematically, if the picture of a person is represented as a data point in $\mathbb{R}^F$ and its appearance features are embedded as a vector in $\mathbb{R}^D$, the function f_θ maps semantically similar data points in $\mathbb{R}^F$ onto a metrically close point in $\mathbb{R}^D$. The function f_θ is parameterized by θ which are learned in the training phase of the CNN. The architecture of the TriNet and its exemplary results are shown in Figure 2.5.

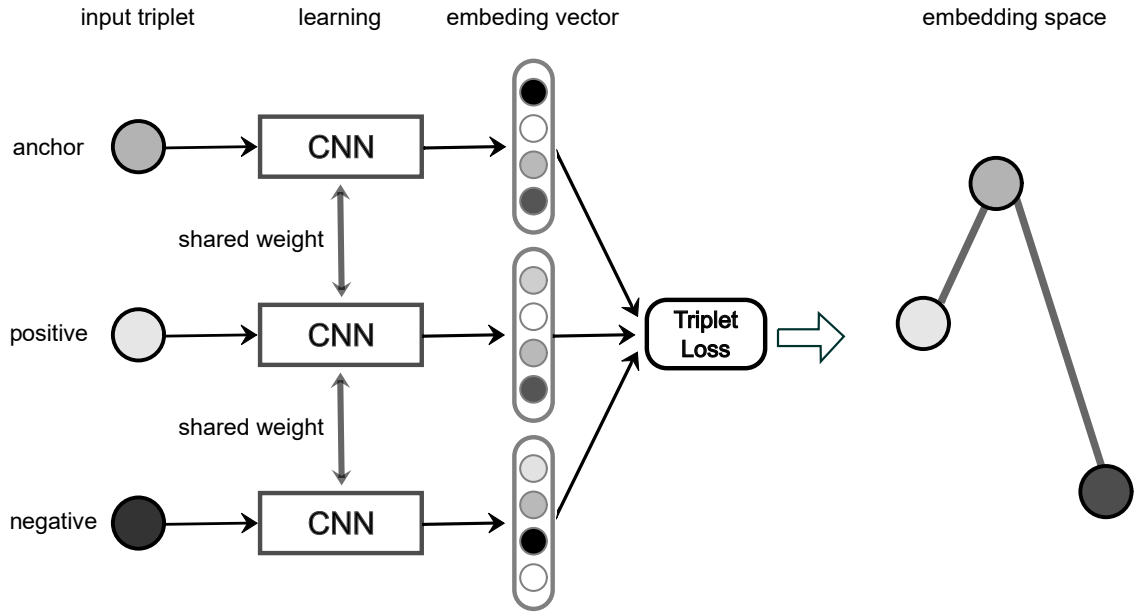


Figure 2.5: Overview of the TriNet architecture. The network takes three images as input, the positive has the same Id as the anchor, while the negative has different Id. The training strategy is that in embedding space, the distance between an anchor and its positive is smaller than the distance of the anchor and a negative by at least a margin m.

During training, TriNet takes three images as input, in which one is call anchor a, a positive image p contains the person of the same id with the one in the anchor, and a negative n is an image of another person. The feature vector y of these three images is extracted through a shared weights CNN as $y_i = f_\theta(i)$. The weights of this network are updated using the triplet loss as following:

$$\begin{aligned} L_{tri}(\theta) &= \sum_{a,p,n,y_a=y_p\neq y_n} \max(m + D_{a,p} - D_{a,n}, 0) \\ &= \sum_{a,p,n,y_a=y_p\neq y_n} [m + D_{a,p} - D_{a,n}]_+ \end{aligned} , \tag{2.3.1}$$

$[.]_+$ is a standard hinge function. $D_{i,j}$ is the distance metric between y_i and y_j. $D_{a,p}$ and $D_{a,n}$ are called pull and push term, respectively. And for a given training triplet a, p, n, the loss function $L_{tri}(\theta)$ is constructed to achieve at a situation where $D_{a,p}$ is smaller than $D_{a,n}$ by at least a margin m.

However, calculating the loss using pull and push term of all training samples not only time-consuming but also makes the network fail at non-trivial triplets. Therefore, moderate negative and positive data mining techniques are applied to help the network better generalize. The core idea is that batches are randomly sampled from P person identities. Each batch has PK samples, in which K images comes from each person. Then, for each anchor a in the batch, its hardest negative n_h and positive p_h are mined in the batch so that D_{a,p_h} has the biggest distance and D_{a,n_h} has the smallest. The loss function is modified to take the mining data strategy into account as follow:

$$L_{BH}(\theta; X) = \sum_{i=1}^{P} \sum_{a=1}^{K} [m + \max_{p=1...K} D(f_\theta(x_a^i), f_\theta(x_p^i)) - \min_{\substack{n=1...K \\ j=1...P \\ j \neq i}} D(f_\theta(x_a^i), f_\theta(x_n^j))]_+ \qquad (2.3.2)$$

There are also different methods to sample the hardest negative and positive samples in a batch depending on training strategies, which can lead to different performance of the network.

2.4 Social force model

The social force model (SFM) is suggested by Helbing and Molnar (1995) to explain the motion changes of pedestrians subject to social forces in most of situations and populations except complex scenarios. These models provide valuable clues to predict the walking trajectory of pedestrians so that vehicles and robots can plan their appropriate interactions in time. The force terms are reflected through intentions of people when they move including: a pedestrian planning to reach a desired place within a certain time on the most convenient path; a pedestrian always trying to keep a certain distance from other people and obstacle objects on streets like facades, traffic lights, vehicles; the attraction which can come from known persons or interesting events on streets. According to these terms, the process of behaviour changes depends on personal aims and perceptions about the surrounding environment of a person, which is depicted in Figure 2.6.

While the term *force* can be related as physical exertion on the pedestrian's body, the SFM describes the reactions of pedestrians in responding to their perception of the surrounding environment using quantity mathematics models.

Personal aims

A pedestrian α usually tries to reach a defined place $\vec{r}_\alpha$ as conveniently as possible by choosing

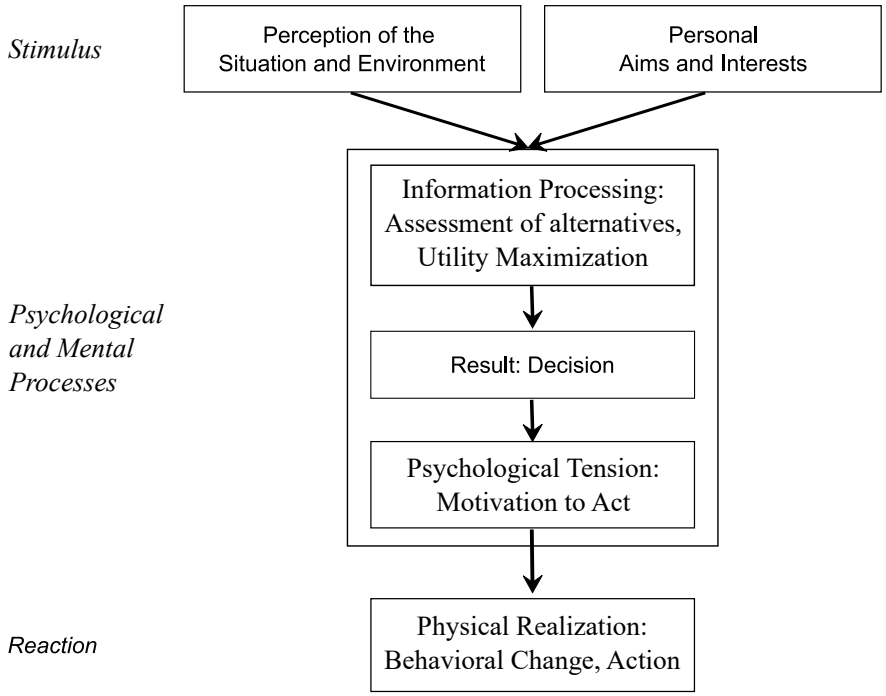


Figure 2.6: The diagram illustrates the procedure of pedestrian behaviour changing (Helbing and Molnar, 1995).

the shortest path. If he/she is not disturbed by the environment, he/she tends to keep a desired speed v_0 and direction $\vec{e}(t)$ when moving, $\vec{v}_\alpha(t) := v_0\vec{e_\alpha}(t)$. However, keeping with a constant velocity is nearly impossible in the real-world due to acceleration or deceleration processes. Given the real velocity $\vec{v_{\alpha,r}}$, a small time τ_α is needed to achieve the ideal one. This results in the acceleration term as follow:

$$\vec{F}^0_\alpha(\vec{v_{\alpha,r}}, \vec{v_\alpha}) := \frac{1}{\tau_\alpha}(\vec{v_\alpha} - \vec{v_{\alpha,r}}) \; . \tag{2.4.1}$$

Environment influences

People want to keep a certain distance from others while moving depending on the density level of objects around them and also their desired speed. This allows them to maintain their private and comfortable space, as well as to avoid collisions. Thus, the motion of a pedestrian is influenced by another pedestrian β, which results in repulsive effects represented as follows:

$$\vec{f}_{\alpha\beta}(\vec{r}_{\alpha\beta}) := -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta}[b(\vec{r}_{\alpha\beta})] \tag{2.4.2}$$

with

$$b := \frac{1}{2}\sqrt{(||\vec{r}_{\alpha\beta}|| + ||\vec{r}_{\alpha\beta} - v_\beta \triangle t\vec{e}_\beta||)^2 - (v_\beta \triangle t)^2} \; , \tag{2.4.3}$$

where $\vec{r}_{\alpha\beta} = \vec{r}_\alpha - \vec{r}_\beta$. $V_{\alpha\beta}(b)$ is a monotonically decreasing function of b. b is an equipotential line in elliptical form.

The same computation can be applied to calculate the repulsive effect between a pedestrian and

a static object B the on street:

$$\vec{F}_{\alpha B}(\vec{r}_{\alpha B}) := -\nabla_{\vec{r}_{\alpha B}} U_{\alpha B}(||\vec{r}_{\alpha B}||) \,, \tag{2.4.4}$$

where U is is a monotonically decreasing function. $\vec{r}_{\alpha B} = \vec{r}_\alpha - \vec{r}_B^{\,\alpha}$ and $\vec{r}_B^{\,\alpha}$ denotes the position of an object B which is nearest to the pedestrian α.

In contrast to repulsive forces, attractive effects are caused by interesting events, person, or object γ on the streets that make pedestrians want to get closer. Therefore, the attractive effects can be modelled similarly as in 2.4.2. However, different from the repulsive effect, the attractive effects usually decrease with time as pedestrian's interests are typically lost over time:

$$\vec{f}_{\alpha\gamma}(||\vec{r}_{\alpha\gamma}||, t) := \nabla_{\vec{r}_{\alpha\gamma}} W_{\alpha\gamma}(||\vec{r}_{\alpha\gamma}||, t), \tag{2.4.5}$$

$W_{\alpha\gamma}(b)$ is a monotonically increasing function.

Though the aforementioned effects are calculated for all objects that exist in the same area as a pedestrian, objects that appear in the perceiving direction of that person have more influence than others. Due to this fact, the motion direction of a pedestrian is taken into account (i.e. the effective angle 2φ) as weight value for different repulsive and attractive effects that can change the pedestrian velocity:

$$w(\vec{e}, \vec{f}) = \begin{cases} 1, \text{ if } \vec{e}\vec{f} > ||\vec{f}||\cos\varphi \\ c, \text{ otherwise} \quad (0 < c < 1) \end{cases} \,. \tag{2.4.6}$$

Both, the repulsive and the attractive effects lead to the acceleration or deceleration of pedestrian velocity, given by:

$$\begin{aligned} \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r_\alpha} - \vec{r_\beta}) &:= w(\vec{e}_\alpha, -\vec{f}_{\alpha\beta})\vec{f}_{\alpha\beta}(\vec{r_\alpha} - \vec{r_\beta}) \\ \vec{F}_{\alpha\gamma}(\vec{e}_\alpha, \vec{r_\alpha} - \vec{r_\gamma}, t) &:= w(\vec{e}_\alpha, \vec{f}_{\alpha\gamma})\vec{f}_{\alpha\gamma}(\vec{r_\alpha} - \vec{r_\gamma}, t) \end{aligned} \,. \tag{2.4.7}$$

Combine all the mentioned influential forces that a pedestrian can be subject to at time t results in its behaviour changes due to the social force $\vec{F}_\alpha(t)$ described as follows:

$$\begin{aligned} \vec{F}_\alpha(t) :=& \vec{F}_\alpha^0(\vec{v_\alpha}, \vec{v_{\alpha,r}}) + \sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r_\alpha} - \vec{r_\beta}) \\ &+ \sum_B \vec{F}_{\alpha B}(\vec{e}_\alpha, \vec{r_\alpha} - \vec{r_B^\alpha}) + \sum_\gamma \vec{F}_{\alpha\gamma}(\vec{e}_\alpha, \vec{r_\alpha} - \vec{r_\gamma}, t) \,. \end{aligned} \tag{2.4.8}$$

Let $\bar{v}_\alpha$ be the *preferred velocity* of α under the social force $\vec{F}_\alpha(t)$. Then, the change of $\bar{v}_\alpha$ over time is:

$$\frac{d\vec{v_\alpha}(t)}{dt} = \vec{F}_\alpha(t) + fluctuations \quad . \tag{2.4.9}$$

The fluctuation term is caused by random behaviour which stems from ambiguous situations and unusual motions. People move with a limited speed $v_\alpha^{\max}$. In many cases, the preferred velocity is

different from the actual movement $\vec{v}_{\alpha,r}$ they can make. Taking $v_\alpha^{\max}$ into account, $\vec{v}_{\alpha,r}$ is computed as follows:

$$\begin{aligned} \vec{v}_{\alpha,r}(t) &= \vec{v_\alpha}(t) g(\frac{v_\alpha^{\max}}{||\vec{v_\alpha}||}) \\ g(\frac{v_\alpha^{\max}}{||\vec{v_\alpha}||}) &= \begin{cases} 1, & \text{if } ||\vec{v_\alpha}|| \leq v_\alpha^{\max} \\ \frac{v_\alpha^{\max}}{||\vec{v_\alpha}||}, & \text{otherwise} \end{cases} \end{aligned} \quad . \tag{2.4.10}$$

2.5 Kalman filter

The Kalman filter is an estimator known to solve linear-quadratic problems (Grewal and Andrews, 2014). It is commonly used in statistic and control theory to recursively estimate an instantaneous state of a linear dynamic system. This filter works under the assumptions that both state and measurement variables of the system are affected by uncorrelated Gaussian noise and their relationship can be derived by a linear model.

The filter algorithm is carried out in two steps consisting of prediction and update. First, it predicts the current state of the dynamic system based on previous states which is called predicted state. Once the current measurement is observed, the state variables are updated as the weighted average of the measurement and the predicted state, in which less weight is given to estimates with higher uncertainty. By combining the information in the past and current time, the Kalman filter can deliver the state variable with smaller uncertainty.

System dynamic model

The system dynamic model describes the linear transition of state variables $S \in \mathbb{R}^n$ between two adjacent temporal epochs:

$$\begin{aligned} S_k &= \psi_{k-1} S_{k-1} + q_{k-1} \\ q_k &\sim \mathcal{N}(0, Q_k) \end{aligned} \quad , \tag{2.5.1}$$

where ψ_{k-1} is the *transition matrix*. $q_k \in \mathbb{R}^n$ is the *process noise* with zero mean ($E\langle q_k \rangle = 0$) and covariance $Q_k \in \mathbb{R}^{n \times n}$. It describes the deviations of the estimated state obtained from the linear model. Using the prior information about the previous state and the linear transition model, the current state S_k^+ can be predicted as follow:

$$\begin{aligned} E\langle S_k^+ \rangle =& \psi_{k-1} E\langle S_{k-1} \rangle + E\langle q_{k-1} \rangle \\ =& \psi_{k-1} E\langle S_{k-1} \rangle \ . \end{aligned} \tag{2.5.2}$$

The corresponding covariance Q_k^+ of the predicted state S_k^+ is derived as follow:

$$\begin{aligned}
Q_k^+ =& E\langle[S_k - E\langle S_k^+\rangle][S_k - E\langle S_k^+\rangle]^{\mathrm{T}}\rangle \\
=& E\langle[\psi_{k-1}[S_{k-1} - E\langle S_{k-1}\rangle] + q_{k-1}][\psi_{k-1}[S_{k-1} - E\langle S_{k-1}\rangle] + q_{k-1}]^{\mathrm{T}}\rangle \\
=& \psi_{k-1}\underbrace{E\langle[S_{k-1} - E\langle S_{k-1}\rangle][S_{k-1} - E\langle S_{k-1}\rangle]^{\mathrm{T}}\rangle}_{Q_{k-1}}\psi_{k-1}^{\mathrm{T}} + \psi_{k-1}E\langle[S_{k-1} - E\langle S_{k-1}\rangle]q_{k-1}^{\mathrm{T}}\rangle \\
&+ E\langle q_{k-1}[S_{k-1} - E\langle S_{k-1}\rangle]^{\mathrm{T}}\rangle\psi_{k-1} + \underbrace{E\langle q_{k-1}q_{k-1}^{\mathrm{T}}\rangle}_{P_{k-1}} \\
=& \psi_{k-1}Q_{k-1}\psi_{k-1}^{\mathrm{T}} + P_{k-1}\ .
\end{aligned} \tag{2.5.3}$$

Measurement model

let $Z_k \in \mathbb{R}^m$ be the measurement observed at time k, which is applied to estimate the state $S_k \in \mathbb{R}^n$ of a stochastic system. The linear relation between Z_k and S_k is represented through the *measurement sensitivity matrix* $H \in \mathbb{R}^{m\times n}$:

$$\begin{aligned}
Z_k &= H_k S_k + v_k\ , \\
v_k &\sim \mathcal{N}(0, R_k)
\end{aligned}\ , \tag{2.5.4}$$

where $v_k \in \mathbb{R}^m$ is the *measurement noise* with zero mean ($E\langle v_k\rangle = 0$) and covariance $R_k \in \mathbb{R}^{m\times m}$.

The optimal updated estimate state S_k^*, is a posterior value of S_k, which is computed based on the observation Z_k and the prior estimate state S_k^+:

$$S_k^* = K_k^1 S_k^+ + K_k Z_k\ , \tag{2.5.5}$$

K_k^1 and K_k are unknown and need to be determined so that S_k^* satisfies the orthogonality principle (Grewal and Andrews, 2014):

$$\begin{aligned}
&E\langle[S_k - S_k^*]Z_i^{\mathrm{T}}\rangle = 0\ , i = 1, 2, \cdots, k-1, \\
&E\langle[S_k - S_k^*]Z_k^{\mathrm{T}}\rangle = 0\ .
\end{aligned} \tag{2.5.6}$$

This equation can be rewritten as:

$$\begin{aligned}
& E[\langle\psi_{k-1}S_{k-1} + q_{k-1} - K_k^1 S_k^+ - K_k Z_k\rangle Z_i^{\mathrm{T}}] = 0\ , i = 1, 2, \cdots, k-1, \\
\Rightarrow\ & E[\langle\psi_{k-1}S_{k-1} - K_k^1 S_k^+ - K_k H_k S_k - K_k v_k\rangle Z_i^{\mathrm{T}}] = 0 \\
\Rightarrow\ & \psi_{k-1}E\langle S_{k-1}\rangle Z_i^{\mathrm{T}} - K_k^1 E\langle S_k^+\rangle Z_i^{\mathrm{T}} - K_k H_k \psi_{k-1} E\langle S_{k-1}\rangle Z_i^{\mathrm{T}} = 0 \\
\Rightarrow\ & E\langle S_k\rangle Z_i^{\mathrm{T}} - K_k^1 E\langle S_k^+\rangle Z_i^{\mathrm{T}} - K_k H_k E\langle S_k\rangle Z_i^{\mathrm{T}} + K_k^1 E\langle S_k\rangle Z_i^{\mathrm{T}} - K_k^1 E\langle S_k\rangle Z_i^{\mathrm{T}} = 0 \\
\Rightarrow\ & E\langle[S_k - K_k H_k S_k + K_k^1 S_k]\rangle Z_i^{\mathrm{T}} - K_k^1 E\langle[S_k^+ - S_k]\rangle Z_i^{\mathrm{T}} \\
\Rightarrow\ & [I - K_k H_k - K_k^1]E\langle S_k\rangle Z_i^{\mathrm{T}} = 0 \\
\Rightarrow\ & K_k^1 = I - K_k H_k \\
\Rightarrow\ & S_k^* = (I - K_k H_k)S_k^+ + K_k Z_k\ .
\end{aligned} \tag{2.5.7}$$

The optimum K_k is computed as follow:

$$K_k = Q_k^+ H_k^{\mathrm{T}} [H_k Q_k^+ H_k^{\mathrm{T}} + R_k]^{-1} , \tag{2.5.8}$$

K_k is called the Kalman gain. The covariance error Q_k^* of the updated state S_k^* is:

$$\begin{aligned} Q_k^* &= E\langle [S_k^* - S_k][S_k^* - S_k]^{\mathrm{T}} \rangle \\ &= E\langle [S_k^+ - K_k H_k S_k^+ + K_k Z_k - S_k][S_k^+ - K_k H_k S_k^+ + K_k Z_k - S_k]^{\mathrm{T}} \rangle \\ &= [I - K_k H_k] Q_k^+ . \end{aligned} \tag{2.5.9}$$

The updated state S_k^* is the optimum estimated state that can be obtained based on available information about the dynamic and measurement model. The best trade-off between the predicted S_k^+ and the measurement Z_k is computed based on their covariance matrices Q_k^+ and R_k and represented in the Kalman gain K_k.

Summary

The Kalman filter is carried out in two main steps consisting of prediction and update can be summarized as follow:

- *Prediction*:

$$\begin{aligned} S_k^+ &= \psi_{k-1} S_{k-1} \\ Q_k^+ &= \psi_{k-1} Q_{k-1} \psi_{k-1}^{\mathrm{T}} + P_{k-1} \end{aligned} \tag{2.5.10}$$

- *Update*:

$$\begin{aligned} S_k^* &= (I - K_k H_k) S_k^+ + K_k Z_k \\ Q_k^* &= [I - K_k H_k] Q_k^+ \\ K_k &= P_k^+ H_k^{\mathrm{T}} [H_k P_k^+ H_k^{\mathrm{T}} + R_k]^{-1} \end{aligned} \tag{2.5.11}$$

The Kalman filter is applied under the assumption that both the measurement and prediction models are linear. However, many problems of practical interest are non-linear, yet differentiable. For those cases, the extended Kalman filter algorithm can be used instead of the standard one. The non-linear functions are linearised using Taylor expansion (Grewal and Andrews, 2014).

3 Related Works

This chapter provides an overview of previous studies related to multi-object tracking (MOT) and pedestrian tracking to bring the contributions of this dissertation into a picture of reference w.r.t. state-of-the-art developments. Several significant aspects of MOT in general are thoroughly examined. Different available tracking approaches are summarized in Section 3.1, followed by a brief review of up-to-date object detection methods in Section 3.2. The tracking-by-detection method, which is employed by most successful trackers and also used as the framework of our approach, is discussed in Section 3.3. In this section, along with optimization methods of data association, feature cues that facilitate the person-reidentification during tracking are described. Motion models, which are explored by existing studies to anticipate reactions of tracked targets w.r.t. to their surrounding environment, are represented in Section 3.4. To close this chapter, strengths and weaknesses of previous works are summarized in Section 3.5, raising open questions of the tracking task and showing how our work can partially solve them.

3.1 Tracking approaches

Pedestrian tracking is the task of continuously localizing interesting targets over time consistently. This means a trajectory should be generated from the positions of one and solely one person. The task can come in the form of single or multiple-person tracking. Here, only the problem of tracking multiple pedestrians is investigated due to its significances for practical applications and because it includes single person tracking. One important condition that affects the formulation of the tracking task is the availability of input images at the time the tracker is executed, which enables it to be operated either in an online or offline manner. In addition, the considered domain, in which the tracking is carried out, also depends on the problems that it is applied to. Whereas some applications only require the trajectories of pedestrians in 2D image space (e.g. footage surveillance), others (e.g. autonomous vehicle) need the 3D positions of trajectories so that a scene in 3D space can be sensed.

Tracking methods

Most of the modern trackers employ the tracking-by-detection approach to constantly track pedestrians in image sequences (Pirsiavash et al., 2011; Zamir et al., 2012; Choi, 2015; Dehghan

et al., 2015a; Yoon et al., 2016; Klinger et al., 2017; Henschel et al., 2018). This method comes in two phases: (1) pedestrians are detected in each image separately, and (2) detections in consecutive frames are associated consistently to generate a set of trajectories. While detection is considered as an independent problem in the computer vision field and can be solved by employing state-of-the-art detectors (He et al., 2016; Zhang et al., 2016; He et al., 2017), most studies following this approach concentrate on concatenating observed objects across image frames accurately to form consistent trajectories. This task can become excessively challenging depending on the number of persons in a scene and their appearance scales, illuminations, etc. Previous studies usually try to handle the association task either by developing better optimization methods (Berclaz et al., 2011; Dehghan et al., 2015a) or by improving the appearance feature extractors (Leal-Taixé et al., 2016; Bae and Yoon, 2018). More insightful discussions on data association are presented later in Section 3.3. For the sake of smoothness, a filtering step can be added after the association stage, which is in charge of correcting a state variable consisting of the position and velocity of a pedestrian based on its previous states, motion model and current observation. One advantage of the tracking-by-detection approach is that the detection and association steps are separated. Thus, it is straightforward to develop and improve individual components as well as to analyse their performances independently. However, since the tracking-by-detection method heavily depends on the detection results to generate trajectories, it faces several severe challenges as follows:

- Missed detection: pedestrians may not be detected in some image frames because of illumination difficulties, occlusions, or scale. This can result in the fragmentation of trajectories and the decline of true positive (TP) results. If the detections of a person are missed for a long time, it is very hard to re-identify that person later because of changes in appearance or ambiguous position. Thus, its trajectory can be inconsistent or even completely lost.

- Low quality detection results: pedestrian detection is not a trivial task. Objects with similar looks can be incorrectly classified as a pedestrian or one pedestrian can have several detections that partly overlap in an image. Because of these false alarms, generated trajectories can contain false positive (FP) results. This leads to a reduction in the quality of trajectories in terms of consistency and accuracy.

- Appearance similarity: though the visual property is an important cue to link detections of the same person in consecutive frames, different pedestrians can look alike and also the same person may look differently depending on the viewing direction of the camera system, occlusion, and illumination conditions.

- Unpredictable behaviours: in most tracking systems, pedestrians are assumed to have smooth movements, which enables the employment of the geometry cue. However, this assumption does not always hold, people can change their walking intentions fast and unpredictably. Hence, it can be problematic to keep track of those people accurately.

With the explosion of deep learning techniques in recent years, CNNs have emerged as powerful tools for analysing time-series data and understanding images (Jin et al., 2013; Ma et al., 2015; Nam and Han, 2016). The general idea of these approaches is that the networks take various small image sequences as input to learn and encode semantic information of objects over time in the training phase. The learned features are then used to recognize and track moving objects in the testing phase. The drawback of those systems is that they need a huge amount of training data for the network to learn. Moreover, deep learning approaches alone do not allow to control the mutual interaction among tracked objects easily. Thus, their tracking results are intractable and it is really hard to predict when and why they fail to track a specific person.

Lately, two groups of tracking methods based on siamese architecture and correlation filters have attracted great attention in the tracking community due to their speed and favourable performance. First, siamese architecture networks are designed to track objects in an end-to-end fashion (Valmadre et al., 2017; Zhu et al., 2018; Wang et al., 2018; Li et al., 2018a; Wang et al., 2019; Li et al., 2019) by following the strategy of appearance similarity comparison, whose key element is metric learning. The ultimate goal of siamese trackers is to learn an embedding space to maximize the visual differences between various objects and minimize the intraclass dissimilarity for the same object.

Applying the same strategy of the siamese tracking method, the second group of trackers utilises CNNs to learn discriminative correlation filters so that an object can be discriminated from the background (Mueller et al., 2017; Valmadre et al., 2017; Zhang et al., 2017; Liu et al., 2018; Cheng et al., 2019). While siamese trackers can handle the task of multiple object tracking, the correlation filter-based approaches are usually suitable to track single objects only. However, since these two groups of trackers are developed based on a similar idea, they are subject to the same weaknesses: first and foremost, they heavily rely on only appearance features to search and match objects in consecutive images, while important geometry information is more or less neglected. Thus, they can hardly cope with occlusions, which is one of the most significant problems of tracking. Furthermore, these tracking approaches are developed based on very strong assumptions that desired objects must all appear in the first epoch and camera motion is completely smooth so that the positions of interesting targets in forwarding frames are known to be within a small region. Due to the mentioned drawbacks, these approaches are solely appropriate to track objects in specific scenarios, where prior knowledge about the scenes, sensor motions as well as object behaviours are guaranteed to satisfy the described assumptions. Hence, they are currently not suitable for tracking objects in highly dynamic scenes such as pedestrians in down-town areas or on streets.

It is critical that the detection results are accurate and reliable for both tracking-by-detection and CNN-based methods to yield good results. Different from all approaches mentioned earlier, tracking objects based on motion does not require the detection step to localize target subjects in

images. These trackers utilise optical flow or object scene flow to separate moving objects from the background, and the object type can be determined through analysing their shape, blob, or moving characteristics (Schwarz et al., 2012; Aslani and Mahdavi-Nasab, 2013; Leal-Taixe et al., 2014a). The disadvantage of motion-based tracking is that it only works when interesting objects perform movements continuously. Moreover, eliminating the ego-motion of moving cameras to separate static and dynamic objects is not easy. Therefore, this technique is usually applied to track large and fast-moving objects like vehicles rather than pedestrians at street level.

Instead of only focusing on the task of object tracking, many studies carry out both segmentation and tracking simultaneously, as these two tasks are highly correlated and can support each other. Similar to the optical flow tracking approach, many trackers following tracking-by-segmentation do not require bounding boxes of interesting targets as inputs (Xiao et al., 2015; Yeo et al., 2017; Lee et al., 2018). Instead, super pixels are employed to perform both, separating interesting objects from the background and tracking. However, while super pixels are helpful to deal with articulated and deformable objects, this approach becomes very complicated when the number of observations increases. Moreover, based on super pixels, global features about appearance and motion of a whole object are hard to obtain.

Online and offline trackers

In general, the object tracking task can be carried out either in a local (online) or a global (offline) manner. For the online approach, the pedestrian trajectories are calculated and updated at every epoch when the input images are fed to the tracker. Since only information about current and past frames are available, this method is vulnerable to wrong detections (Breitenstein et al., n.d.; Kim et al., 2012; Lenz et al., 2015; Xiang et al., 2015; Fagot-Bouquet et al., 2016; Kieritz et al., 2016). Global methods, on the other hand, generate tracklets or complete trajectories from a batch of frames or the whole image sequence. This enables global properties of target objects to be taken into account during the optimization. That is why most global matchers usually outperform local approaches (Zhang et al., 2008; Yang and Nevatia, 2014; Berclaz et al., 2011; Pirsiavash et al., 2011; Zamir et al., 2012; Dehghan et al., 2015a). Nevertheless, requiring the entire image sequence before tracking, global techniques can only be used for offline applications. In applications where instant responses are demanded, like autonomous driving or robot-human interaction, only online approaches are appropriate.

For the purpose of closing the gap between local and global approaches, a method called ”near online” is proposed in (Choi, 2015; Tang et al., 2015; Henschel et al., 2019). This algorithm follows the sliding window scheme in the way that tracklets of pedestrians are generated from a certain number of epochs in the past until the current one. On the ground of additional evidence from the past, this method can generate more accurate trajectories and mistakes in the present can be corrected later. However, many practical applications demand information from tracking to

make decisions in real-time which cannot be corrected later. Thus, this method certainly can boost the tracking results, yet only improvements of current and future frames are meaningful, not the past ones.

Tracking domain

Most state-of-the-art methods execute tracking in the 2D image space and concentrate on strengthening the identity (Id) accuracy of estimated trajectories (Breitenstein et al., n.d.; Fagot-Bouquet et al., 2016; Kieritz et al., 2016; Leal-Taixé et al., 2017). Using only 2D information from images, the trackers belonging to this paradigm usually make strong hypotheses about the movement of camera systems. This assumption enables the inference of pedestrian motions from the 2D image domain to 3D object space so that geometry cues can be introduced. Unfortunately, mobile systems in the real-world can violate the hypothesis, which can cause failures of trackers.

Positions and moving directions of pedestrians in 3D object space are essential prerequisites for vehicles to automatically manage their movements. For this reason, several systems do tracking in 3D based on stereo or RGB-D cameras or sensors based on structured light. Although widely used for indoor tracking studies (Jafari et al., 2014; Linder et al., 2016), RGB-D devices are not appropriate for the outdoor environment due to illumination problems and complicated surfaces. To cope with outdoor scenes in autonomous driving applications, either a stereo rig or a combination of camera and LiDAR sensor are usually employed to track people on streets (Mitzel et al., 2010; Schindler et al., 2010; Ošep et al., 2017; Dimitrievski et al., 2019). The 3D geometric position of a pedestrian is estimated by inspecting the detected bounding box or intersecting the image space detection with the ground plane. Estimating the foot positions of pedestrians on the ground plane allows to reduce pedestrians' movement in 3D space from three to two dimensions, which is easier to handle.

3.2 Object detection

Pedestrian detection is a specific case of object detection, which is one of the most active domains in computer vision today. In principle, two types of detectors have been developed for this task. A detector in the first group manually extracts defined features of RoIs and then a classifier takes responsibility for object classification (Benenson et al., 2013; Zhang et al., 2014; Dollár et al., 2014; Zhang et al., 2015). Some famous and representative detectors of this group are the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005) and the integral channel features (ICF) (Dollár et al., 2009). Various deviations of these approaches were proposed to improve the detection outcomes and achieved significant results (Felzenszwalb et al., 2008; Satpathy et al., 2014; Hoang et al., 2014; Paisitkriangkrai et al., 2014). The second group of detectors employs CNNs, which are able to handle feature extraction directly from raw pixel values through

forward and backward propagation (Krizhevsky et al., 2012; Girshick et al., 2014; Redmon et al., 2016; He et al., 2017). Even though these two types of object detectors deliver promising results, currently CNNs-based approaches outperform other methods and achieve the best performance in object classification (Zhu et al., 2019). This is because CNNs are able to extract not only low-level features as the handcrafted methods do but also high-level abstract features.

While a detector performs detection for a particular area where an interesting object is supposed to lie in, the purpose of object detection is to search the locations of target objects in the whole image. Hence, it is necessary to determine all instances of interesting objects in the image, so that they can be delineated, e.g. with a bounding box, in order to subsequently feed them to the detector for classification. To do so, the most common practice is to utilise exhaustive search over the image with sliding windows of multiple scales. This method is simple and easy to implement but time-consuming, especially when using it along with a CNN detector. An alternative approach is to determine possible locations in an image where desired objects can appear using region proposal methods. The aim of proposal generators is to determine all instances of interesting objects in an image with as few false alarms as possible, which allows more sophisticated and accurate classifiers like CNNs to be adopted to detect the objects. Currently, several paradigms are widely applied to the region proposal task including objectness scoring (Alexe et al., 2012; Zitnick and Dollár, 2014), super pixel grouping (Uijlings et al., 2013; Arbeláez et al., 2014), CNN (Ren et al., 2015; Li et al., 2018b), and 3D object proposal (Chen et al., 2015; Nguyen et al., 2018).

Hand-crafted detectors

The core of hand-crafted detectors is their feature extraction components. The broadly used HOG descriptor is the basic of a large numbers of hand-crafted detectors. HOG was first introduced by Dalal and Triggs (2005) to detect humans. The main idea behind this approach is to exploit the distribution of gradient directions to describe the local appearance of an object. The HOG descriptor is obtained by combining local 1-D histograms of all local small cells into a feature vector. The histogram of one cell is the accumulation of edge orientation of all pixels in it. Given that HOG combines the whole appearance of an object into one feature vector, its performances can be easily damaged by occlusions or deformations. Built on top of HOG, the deformable part model (DPM) is proposed by Felzenszwalb et al. (2008) to overcome disadvantages of rigid templates and improve the generic object detection performance. In DPM, an object is decomposed into multiple deformation parts whose relative positions to the centre of the object are defined according to a spatial model. Different HOG descriptors are trained for the whole object which is called root filter, and for each part which is referred to as a part model. The detection is accomplished by taking into account the root filter score, part models scores and the displacement of those parts in the spatial model. Taking another direction, Dollár et al. (2009) suggest the integral channel features (ICF), relying on multiple registered image channels to integrate and obtain richness and diversity information of an image. These channels can be either linear or non-linear transformations from

the source one. Features such as Gabor filters, local sums, or the histogram of a local rectangular region extracted over various channels can be part of the ICF.

Convolutional neural network detectors

Instead of extracting features based on prior knowledge, CNNs use convolution filters to explore meaningful feature spaces that facilitate object classification. Given a labelled training dataset, CNNs carry out the tasks of feature extraction and classification simultaneously through back-propagation using gradient descend to minimize a loss function which represents the task objective. While the CNN detector is the backbone for a CNN-based object detection approach, the task of generating object proposals can be separated from the network by using an existing proposal method (Ren et al., 2015; He et al., 2015; Dai et al., 2016). In contrast, many CNN detectors are designed to integrate a region proposal network as an additional branch or using the BB regression paradigm so that the detection can be trained end-to-end (He et al., 2017; Zhang et al., 2018; Liu et al., 2016). Though currently, CNN detectors achieve the best performance for object detection, they are excessively data hungry. This makes them extremely expensive to train in a supervised way, which is applied in most of the modern detectors. Due to this limitation, various studies have been investigated on either unsupervised or semi-supervised CNNs to reduce the labelling efforts as well as to make use of huge unlabelled data (Yang et al., 2013; Gao et al., 2019; Tang et al., 2016b). Another direction that has attracted the community is designing CNN detectors to employ not only visual properties from images but also 3D point clouds information (Chen et al., 2017; Mousavian et al., 2017; Tang and Lee, 2019).

3.3 Tracking-based-detection

Currently, the tracking-based-detection is the most effective approach for multi-object tracking in general, which includes the detection and association stages. Association can be considered as the most critical and challenging issue for any tracker following the tracking-by-detection approach. This task is done by concatenating detections in consecutive frames together to form trajectories for tracked objects under the hypothesis that visual appearance and positions of pedestrians only vary smoothly over time. Visual properties, geometry, and motion are common cues that are usually exploited to solve the data association task. These similarity cues are then combined together in a weight or loss function, which represents the probability that two detections belong to the same person. Association is then optimized with the objective of maximizing/minimizing the total weight/loss values. In the past, numerous trackers were investigated to improve the accuracy of the assignment optimization using methods such as k-shortest path, conditional random fields, Bayesian networks, and network flow (Zhang et al., 2008; Berclaz et al., 2011; Milan et al., 2013b; Yang and Nevatia, 2014; Choi, 2015; Klinger et al., 2017). As the performance of optimizers seems

to saturate in recent years, many studies focus on improving the appearance feature extractors, so that linking detections across frames, which relies on these features, is also improved (Leal-Taixé et al., 2016; Kieritz et al., 2016; Bae and Yoon, 2018).

Thus, the two elements that primarily influence the performance of an association-method are the optimization approach and similarity features. Existing research, which is relevant to these problems, is reviewed in the next sub-sections.

Data association optimization methods

The concatenating of observations across images can be considered as an optimization problem, in which the global solution is desired. The optimization is then formulated using different methods such as network flow optimization (Zhang et al., 2008; Butt and Collins, 2013; Tang et al., 2017), Markov Chain Monte Carlo sampling (Oh et al., 2009; Shitrit et al., 2013; Lee et al., 2016), Markov Decision Process (Xiang et al., 2015), multi-hypothesis (Kim et al., 2015), which all achieve impressive results.

As a network flow problem, the relationships between hypothesis detections are represented by a directed graph $G = (V, E)$. Vertices V are a set of detections or tracklets of a video sequence. Edges E connect observations within and across images frames. Cost or reward values are estimated for each edge depending on the similarity of two vertices linked by it. The network flow solution can be found efficiently by using the minimum-cost maximum-flow algorithms, in which a network is decomposed into sub-graphs. Each of them represents a hypothesis trajectory. Various tracking studies investigated network flow optimization solutions for MOT. Zhang et al. (2008) solve the global association for the whole image sequence using a min-cost flow algorithm. An explicit occlusion model is also suggested to handle both short and long term occlusions of existing tracked targets by adding an occluded object hypothesis in the graph. The weakness of this study is that it can be only operated offline. Butt and Collins (2013) propose to exploit higher-order-constraints through replacing matching pairs of observed objects in two consecutive images as nodes in G. Consequently, the cost of each edge encodes the cost of detection in three epochs. Trajectory solutions are found efficiently using Lagrange relaxation and min-cost flow. However, the convergence of this method is not guaranteed. Shitrit et al. (2013) introduce multi-commodity network flow to include image appearance similarity between groups of objects. In other words, the single graph G is duplicated for each appearance-group. Multiple networks are solved in parallel using linear programming. As the suggested approach yields much larger network than the single-commodity solution, the complexity is also much higher. Wang et al. (2014) use a modified version of network flow where the nodes in the graph encode different orientations and locations of detections, which is capable of tracking objects of different interacting types. The network solutions are found by mixed integer programming. A target identity-aware network flow (TINF) is proposed in (Dehghan et al., 2015b) to simultaneously solve the detection and association exploit-

ing online discriminative learning. The difference of this network when compared to the common ones is that each candidate location is represented with a pair of nodes, which are linked by K observation edges. In that way, TINF can better deal with missed detections and occlusions. This network flow obtains high-quality trajectories using Lagrangian relaxation. A major downside of this method is that it requires manual annotations to initialise every target entering the test scene. To deal with long-term occlusion, the data association in (Tang et al., 2017) is formulated as a minimum cost lifted multi-cut problem. It is similar to the normal graph, in which each node describes a detection, yet a special type of edge named *lifted* is introduced to connect detections far in time but have similar appearance. These lifted edges imply the hypothesis that those detection usually belong to the same person. However, as the core of the algorithm is to take into account long-range connection of detections far in time, this approach cannot be operated online. Carrying out association as bipartite matching, Klinger et al. (2017) first estimate the cost or the posterior function, which represents how likely detections belong to a person. Then, the solution is obtained directly by using linear programming without formulating the matching as a network problem. This approach is straightforward and also enables incorporating tracking constraints easily.

Another direction, that has been investigated to solve the problem of association, is to apply Markov Chain Monte Carlo (MCMC) sampling to generate trajectory solutions for a set of detections. The distribution probability of hypotheses is defined by a likelihood (Benfold and Reid, 2011), posterior or energy function (Fagot-Bouquet et al., 2016) based on affinity cues such as visual, motion, and BBs intersection over union (IoU). Those functions are called objective functions. A sampling method like Metropolis-Hastings is then adopted to explore the space of hypotheses and find an approximate configuration subject to minimize or maximize the defined objective function. Though an exact solution is not guaranteed, the advantage of MCMC sampling methods is that the interaction and dynamic model can be easily adopted. Moreover, the random nature of MCMC helps to prevent the search from becoming stuck at local maxima or minima of the objective function. Two different versions named single-scan and multi-scan are developed in (Oh et al., 2009) to track pedestrians in case the number of interesting targets is fixed and unknown, respectively. In (Benfold and Reid, 2011), the head BBs of pedestrians are employed for tracking instead of the whole body as usual and the motion cue representing the affinity between observations is derived from the Kanade-Lucas-Tomasi corner features (Tomasi and Kanade, 1991). This is the primary clue to design their likelihood function. Milan et al. (2013a) integrate appearance features together with multiple physical constraints such as mutual exclusion and track persistence into a non-convex energy function. Lee et al. (2016) formulate the problem of multi-class multi-object tracking as a Bayesian filter, which is handled by MCMC. In this approach, object detector responses are combined with a motion detector estimated by a changing points detection model to build the likelihood function. Xiang et al. (2015) solve the assignment optimization of hypothesis detections as decision making based on an Markov decision process (MDP). In their framework, the lifetime of a target is explained by a MDP and the

similarity between observations is learned through a reinforcement method, which is considered equivalent to learning a policy of the MDP.

An alternative approach for high-order information data association is multiple hypothesis tracking (MHT), in which a number of track hypotheses are generated from observations of a certain number of frames. The hypothesis representations are similar to hierarchical tree structures. The quality of the hypotheses are evaluated using an objective function, which allows obtaining final global trajectory solutions (Kim et al., 2015; Yoo et al., 2016). In MHT, all hypotheses in the current epoch can be propagated to future time steps to create new hypotheses. This allows the whole hypothesis space to be thoroughly explored, yet the space grows exponentially over time as the number of detections increases, which makes the optimization for a global solution become highly complicated. Therefore, in MHT a pruning step is usually utilised to reduce the search space by choosing a subset of trees with the highest probability of representing true trajectories. A main advantage of the MHT is that hypotheses are kept active and passed to the future until ambiguous associations are solved. However, at the same time this is one of its negative aspects that does not allow the method to be operated online as the association decisions are postponed for a certain number of epochs.

Appearance model

The visual properties such as colour, shape, texture are important features to link observations of targets together over time. However, comparing appearances between pedestrians is not a trivial task because the appearance of an object can change dramatically under different circumstances such as lighting, occlusion or view point changes of the camera system. Moreover, pedestrians may look very similar due to their fashions, gestures, etc., which can result in indistinguishable visual appearance. Therefore, a lot of recent tracking studies focus on improving appearance modelling so that the association can be enhanced (Kim et al., 2015; Leal-Taixé et al., 2016; Tang et al., 2017).

The visual property of an object can be characterized by very simple and basic descriptors at the pixel level like BB colour histograms (Mitzel et al., 2010; Mitzel and Leibe, 2011; Dehghan et al., 2015a). Klinger et al. (2017) divide a BB into multiple horizontal stripes. In every strip, weighted mean and standard deviation values are calculated for each input channel, which are then concatenated to form the feature vector. To employ not only raw colour information but also shape, texture, etc., Kuo et al. (2010); Choi and Savarese (2012); Zhang and van der Maaten (2013) combine several types of feature extractors such as gradient-based histogram and the covariance matrix of image features to discriminate interesting objects. Also, gradient-based and pixel-level features are used, Hu et al. (2012); Dicle et al. (2013) enhance the efficiency of those features by modelling their distributions at different regions of the image and represent the appearance of interesting targets through those distributions. Adopting a similar scheme, Kieritz et al. (2016),

additionally, improve the robustness of extracted features using multiple-instance learning to update the appearance of objects online at every time step. After computing the appearance model for desired objects, the classification or comparison method such as histogram intersection (Mitzel and Leibe, 2011; Dehghan et al., 2015a), metric learning (Mitzel and Leibe, 2011; Dehghan et al., 2015a), support vector machine (Mitzel and Leibe, 2011; Dehghan et al., 2015a), or online random forest (Klinger et al., 2017) can be adopted to distinguish the appearance of different pedestrians. While these handcrafted features work reasonably well, they can solely represent low and mid-level features, which may not be sufficiently robust to various challenging environments. In some cases, to compare the differences between those extracted features, simple measurements like Euclidean or absolute distance are not sufficient. A learning step may be required to transform the feature vectors into another space so that their power of discrimination can be exploited better.

To avoid designing hand-crafted features which highly depend on prior knowledge about interesting targets and to extract more abstract features for appearance representation, CNNs have been adapted broadly to efficiently extract appearance features of desired objects. Kim et al. (2015) separate the task of feature extraction and classification into two separate steps. First, they employ a CNN to calculate feature vectors of observations. Then, they train and update simultaneously multiple linear regressors for the classification w.r.t. person Ids. Instead of separating the feature extraction from the classification, various studies simultaneously carry out both of the tasks using end-to-end learning. On the ground of siamese architecture, Leal-Taixé et al. (2016) train a network that takes two detected BBs as inputs and produces the label positive if the two inputs belong to the same person and the negative label if not. Different siamese topologies were explored to find the best way to combine the input information of BBs. Similar to (Leal-Taixé et al., 2016), Tang et al. (2017) also use the siamese network architecture to discriminate the appearance of observations. To boost the performance of their network, they include the prior information about body part localization of BB into the network as additional inputs. Besides methods that have been explored to effectively represent the appearance of objects in tracking scenarios, a completely independent research field also involving the task of people discrimination under arbitrary conditions based on visual properties, is people re-identification, which has been extensively studied and has achieved impressive results (Matsukawa et al., 2016; Hermans et al., 2017; Blott et al., 2018; Xia et al., 2019). Thus, various powerful approaches developed in this field can also be adopted for a data association task.

Estimated from pixel intensity values, optical flow is utilised as local appearance features in the form of interesting points (IPs) and their trajectories or histograms, which can represent both local appearance features and relative movements of targets in consecutive images. This information is then used to model the likelihood of matching interesting objects over time. In (Izadinia et al., 2012), the intersection between optical flow histograms of two observations is regarded as appearance similarity. Combining interesting point trajectories computed from an optical flow method with a median filter, Izadinia et al. (2012) solve the task of object tracking by performing

the tracking in both forward and backward directions. Choi (2015) proposes to aggregate IPs of a detected BB by a local flow descriptor that illustrates relative motions of two BBs in different epochs. The authors claim that while individual IPs can contain an error, their aggregation is a strong cue to differentiate detections. Not using any additional descriptor, the association cost between observations in (Tang et al., 2016a) is estimated based on the number of intersection and union of IPs in their BBs. Though optical flow is useful for comparing the appearance of objects and capable of providing motion cues at the same time, it only represents local points relevant to a target. Consequently, many global properties of the whole object appearance are ignored.

Geometry affinity

Though appearance cues are important and useful to discriminate objects, this is typically not sufficient for obtaining a robust and accurate tracking system. Geometry cues regarding the position of objects either in 2D or 3D space further facilitates tracking of pedestrians in image sequences. The authority of this feature stems from the prior knowledge about movements of pedestrians: their speed is limited and there can be only one object occupying a place in object space at a specific time. Milan et al. (2013b) adopt the inter-object exclusion to constrain the unique assigning between observations and targets and also to eliminate the co-occurrence of trajectories. Nevertheless, most existing trackers perform tracking in 2D image space, the distance between two detections is often formulated as their BBs IoU with a motion model to anticipate the movements of observed targets (Zamir et al., 2012; Dehghan et al., 2015a; Fagot-Bouquet et al., 2016; Kieritz et al., 2016). However, this can lead to incorrect interpretation during self occlusion of pedestrians and once the camera system does not move smoothly. To deal with this problem, Yoon et al. (2016) propose a method named structural constraint. They first randomly choose different assignment pairs as anchors. The costs of assigning the other detections are computed based on the 2D geometry changes of anchor pairs with the assumption that their 2D geometry changes are similar. However, this hypothesis is guaranteed only if the anchors and observed objects are near each other and have similar moving characteristics. In contrast, relying on 3D point clouds, the geometry affinity calculation does not suffer the ill-pose problem of using low dimension (2D image space) data to represent the position in a higher domain (3D object space) (Mitzel et al., 2010; Schindler et al., 2010; Ošep et al., 2017; Dimitrievski et al., 2019). Instead of using additional sensors to obtain 3D information, Klinger et al. (2017) infer pedestrian positions in object space by assuming that their heights are constant, which enables calculating the 3D distance between the predicted position of a trajectory and position of a detection.

3.4 Motion model

Pedestrian tracking is a special case of object tracking, in which movements are usually predictable and follow certain characteristics. Employing this prior knowledge, many motion models have been studied to describe the dynamic behaviour of pedestrians, which can be either embedded in the geometry similarity (Klinger et al., 2017) or used as an additional cue in estimating association cost (Yang et al., 2011; Zamir et al., 2012; Leal-Taixé and Rosenhahn, 2013). Moreover, during tracking, the position of a trajectory at a specific epoch can be corrected based on its previous states and an accurate motion model. This not only improves the smoothness of trajectories but also enhances the accuracy of pedestrian localization, which enables the identity consistency of estimated trajectories by reducing the search space during the association optimization. By far, a linear motion model is the most popular approach due to its simplicity. In this model, the velocity of a tracked pedestrian is considered to be smooth and constant for a certain time (Breitenstein et al., 2009; Xing et al., 2009; Yang et al., 2011; Qin and Shelton, 2012; Zamir et al., 2012). The smoothness of object movement can be calculated from total differences of velocities between epochs or modelled through a distribution.

Though the linear motion method is common and simple to apply, it ignores reactions of objects corresponding to their surrounding environment. Therefore, several non-linear motion models have been pursued to explain the movement of interesting targets in the real world. In (Yang and Nevatia, 2012; Maksai et al., 2017), moving patterns are first learned from a set of training data, which are called motion or pattern maps. The motion similarity of observations in a tracklet is, then, scored according to how good their movements can be explained by the motion map. This approach enables any free moving and reaction models, in which no hand-crafted physical rule needs to be defined or known beforehand, but estimating the similarity of a hypothesis trajectory with the motion map requires the whole sequence to be available.

The social force model (SFM) is suggested in (Helbing and Molnar, 1995), reflecting physical constraints that are usually observed while people are moving, is adopted by many tracking approaches (Pellegrini et al., 2009; Luber et al., 2010; Yamaguchi et al., 2011; Leal-Taixé et al., 2011) to design their motion models. Pellegrini et al. (2009); Luber et al. (2010) predict future positions of tracked persons considering the social forces as well as social contexts defined by the SFM. Luber et al. (2010) incorporate their motion model in a Kalman filter to predict peoples' actions. Pellegrini et al. (2009) combine the interactions of pedestrians with the surrounding environment like obstacle avoidance together with their destinations in the scene to model dynamic social behaviours of pedestrians. The level of influence of one pedestrian trajectory on another is assessed using their spatial distance and angular displacement of their moving directions. Also following the movement rules of SFM, but not only taking into account individual reactions, Pellegrini et al. (2010); Yamaguchi et al. (2011), additionally, consider reactions of groups in a scene

while predicting the velocity of a tracked target. The motivation is that a single person and a group of pedestrians usually show different reactions to a certain event. However, since grouping results in only binary decisions, many potential influences of pedestrians in different groups can be neglected. Taking a further step, the moving intention of pedestrians in (Leal-Taixé et al., 2011) is also modelled using the SFM and group behaviour, but in a global manner. In that approach, information in the past and future are utilised to more accurately interpret the movement of tracked objects, assuming that people have a tendency to plan their moving trajectories in advance. Certainly, modelling the motion of pedestrians in the global way results in better performance, but it is only suitable for offline applications. Zhang and van der Maaten (2013) suggest predicting the position of a pedestrian by observing the movements of its neighbours. Similarly, also applying a grouping model, Klinger et al. (2017) improve this method by weighting the effect of each neighbour based on an angular displacement of its moving directions compared to the current person. A Gaussian process regression is adopted to model the change of pedestrian velocity according to physical constraints.On top of this, the defined motion model is combined with a Kalman filter so that the beliefs about pedestrian positions and velocities can be updated at run-time. In (Leal-Taixé et al., 2014b), interaction feature strings, encoding the velocity of observed pedestrians w.r.t. their local scene are extracted from optical flow information. Then, a random forest framework taking these feature strings as input is trained to predict the velocity of desired targets. The advantage of this method is that it does not rely on hand-crafted physical constraints like SFM, thus missed detections do not affect the predicted velocity of interesting objects. However, this approach cannot be applied to generic applications because the random forest needs to be trained beforehand and heavily depends on training data. In (Yoon et al., 2015) and (Yoon et al., 2016), the 2D spatial distance of a target is estimated based on its 2D history trajectory and the relative displacement of nearby persons in image space. However, using 2D image information to infer the non-linear movement in 3D object space is solely correct if pedestrians are near to each other and have similar velocities. Furthermore, the proposed method anticipates the states of a target based on the history of all observed trajectories, including the movement of irrelevant people, which might affect the results.

3.5 Discussion

To close this chapter and provide an insight into the motivation behind developments of this dissertation, this section briefly summaries limitations and open questions of current state-of-the-art works with respect to the research objective of this study. Based on open issues, proposed approaches to close those gaps are discussed.

Multi pedestrians tracking approach

Available tracking approaches can be categorized according to three main major characteristics, which significantly contribute to the total performance of a tracker. First, a tracking approach can either be based on the tracking-by-detection approach or employ CNNs to end-to-end train a tracker without a detector. Though CNN-based trackers can automatically track objects without the demand of designing various modules to handle detection, association, occlusion, etc., they expose several disadvantages. Such deep tracking networks usually assume that the objects' appearance in the next frame is more or less at the same place in image space w.r.t. to the previous image. This hypothesis typically holds once camera movements are smooth. As a consequence, object appearing in the scene in the later stage will be ignored. Moreover, it is hard to integrate other cues such as motion, position, and interactions in those networks. The second property needed to be considered when designing a tracking framework is offline vs. online. Certainly, with richer information, offline approaches usually outperform the online ones. However, at the same time, they require more computations and are naturally not suitable for applications demanding instant responses. Finally, whereas conventional tracking in the 2D image domain is more convenient compared to 3D object space, since there is no need of additional depth information, many assumptions in 2D tracking rely heavily on the smoothness of camera and pedestrians movement, which strictly limit the flexibility of a tracker to deal with complicated and dynamic scenes. Moreover, many aspects of tracking such as the accuracy of localization and state estimation can only be thoroughly explored in 3D space. All aforementioned aspects of a tracking approach need to be taken into account when developing a tracking framework, which not only depends on the accuracy to be achieved but also on other properties of an application.

Motivated by autonomous driving applications, the proposed tracking method is designed to be flexible so that it can track multi-pedestrian at street-level without restrictions on the movement of sensors and to be capable of operating online (i.e. local association approach). For that purpose, the tracking is carried out in 3D space using stereo images and follows the tracking-by-detection approach. Bipartite matching is applied to associate interesting objects in adjacent frames. However, instead of using only information of two contiguous epochs that might contain high uncertainties and errors, the information from a certain number of previous epochs is aggregated to increase the accuracy of data association. Employing the depth value, the localization accuracy and movements of estimated trajectories are explored and improved. To this end, an extended Kalman filter is applied to recursively update the state of observed objects.

Observation processing

In the detection-based tracking approach, detection results serve an important role in the final tracking performance. They provide instance appearance of interesting objects in image space. It has been demonstrated in current literatures that deep CNNs are much more advanced than hand-crafted object detectors (Zhang et al., 2016). Nevertheless, they still have the problem of increasing the number of FPs when recall is being increased. Many trackers try to cope with

this problem by finding multiple detections in an image corresponding to one object using non-maximum-suppression or data association (Tang et al., 2017). This helps to reduce FPs but usually complicated algorithms and sometimes additional cues such as depth, texture, motion must be dealt with. Hence, obtaining a high number of TPs, but still keeping FPs at a low rate is one of research goals of this work. It is achieved by modifying the association step of the tracking pipeline, which connects results of consecutive frames: in this step, while employing all detections of the current frame as input for the assignment, solely highly accurately detected pedestrians are used to create new trajectories, a strategy called tracking-confirm-detection (TCD).

In addition, a number of properties related to pedestrians such as height and BB ratio are exploited to eliminate incorrect detections in a pre-processing step. To this end, while most of the current approaches focus on improving the identity consistency for tracked objects and consider that as their primary problem, the correctness level of geometry is significant for applications that need 3D information for their interaction with interesting objects. Therefore, in this study, prior knowledge about pedestrian height and width is combined with the reconstructed scene and 3D point clouds to precisely determine the positions of detected objects and also provide the uncertainty for those computations.

Online data association

A majority of existing trackers cast the problem of data association as network flow or graph optimization, which can be solved efficiently by linear programming. Our tracker also follows this paradigm to optimize the assignments of detections in consecutive images. Since our framework is developed for online applications, the association is carried out using bipartite matching. In this case, a global solution is guaranteed and the running time is polynomial. While most state-of-the-art works primarily count on appearance features to estimate the association cost (i.e. observation affinity), and it is apparent that though appearance features are an important and powerful cue for tracking, problems can still occur in challenging situations involving scale, occlusion, and illumination differences. As a result, visual features can become indistinguishable. Similar to (Yoon et al., 2015), therefore, in this thesis, the advantages of geometry cues are examined to improve the association results. However, employing solely positions of pedestrians in 2D image space as in (Yoon et al., 2015) is usually not enough to robustly infer correct movements of targets in 3D object space. Therefore, taking a further step, both 2D and 3D information are used together in this work to leverage relationships among pedestrians and refine the local structure among nearby tracked targets. In addition, strong association events (anchors) are determined before the local structure refinement (LSR) is applied for remaining detections. This makes the local structure refinement (LSR) more robust and less prone to errors.

Besides geometry and motion, visual properties play an important role in distinguishing observations of different tracked identities. Apart from algorithms directly solve the to tracking problem,

a substantial amount of research has been introduced for people Re-Id in general. These methods are usually more effective, because images of the same person can be taken from various arbitrary viewpoints. Therefore, to obtain robust appearance features of observations under different conditions, the TriNet (Hermans et al., 2017) which is originally proposed to solve the problem of people Re-Id, is employed in this thesis to extract visual properties of detections. Although there are a lot of handcrafted and CNN methods that have been proposed to improve possibility to distinguish the appearance between pedestrians, this task becomes more and more difficult in the case of occlusions and clustered background. To reduce the effect of these problems, instead of directly feeding a BB to the network as input, an object is first isolated from the background, and then the background pixels are replaced with random values. This random noise prevents a network from using background information for feature extraction. Thus, the results of the visual comparison can be enhanced.

Motion model

Following state-of-the-art research, a non-linear motion model is designed in this work to predict movements of tracked pedestrians. Nevertheless, different from previous studies, it is argued that estimating reactions of a trajectory with respect to all other people in the scene is not necessary. Instead, such estimation should only rely on the ones that move in the same group with the target of interest. There are several explanations for this argument. First, a group of people usually have similar reactions to a certain event and maintain a similar velocity. Second, in a group, there are always some persons that are more clearly visible than the others and their trajectories are updated continuously, which results in high reliability of the trajectories estimated for those pedestrians. Thus, the movements of other people in the group can be modelled w.r.t. those trajectories, which already contains the interaction of a group to the other pedestrians in a scene. Finally, relying on people not moving in the same groups as in the SFM requires all nearby objects to be detected, which is hard to achieve in a dynamic and complex scene. In addition, anticipating behaviours of a target can lead to even worse results if velocities of objects included in a motion model are incorrectly calculated. While this is a critical problem that needs to be considered in anticipating behaviours of a target, most of the existing studies assume that the velocity of individual pedestrians is correctly computed, which is difficult to achieve in practice. Hence, efforts are made in this study to estimate and evaluate the accuracy and reliability of velocities for tracked targets. Moreover, since a group of people moves with more or less the same velocity, noisy velocities of a pedestrian can be corrected by their neighbours whose motions are reliably computed. To this end, based on the correct velocity, missed detections of tracked pedestrians are retrieved as well.

4 Multi-pedestrian Tracking in 3D Object Space

This chapter represents the rationale as well as the mathematical formulation for a new tracking approach using bipartite matching and local structure refinement to track multiple pedestrians in both, 2D and 3D object space. The general pipeline of the tracking approach is introduced in Section 4.1, in which the connections between primary components and their tasks are illustrated. Section 4.2 illustrates how the localization of pedestrians in image and 3D space is accomplished. The concatenating of detected pedestrians in consecutive images to generate consistent trajectories is detailed in Section 4.3. This data association step is developed using a bipartite technique such that trajectories gradually evolve once new input images arrive. Based on information about previous positions and velocities of trajectories, missed detections can be retrieved both in 2D and 3D space, which are introduced in Section 4.4. Trajectories are not simply extended but also smoothed and corrected employing an extended Kalman filter, see Section 4.5. Finally, this chapter is concluded by discussions on theoretical advantages and limitations of the proposed tracking approach in Section 4.6.

4.1 Problem statement and the general pipeline

Aiming at tracking multiple pedestrians in 3D object space at street level for autonomous driving and robotic related applications, our tracking approach, called 3D-TLSR (3D pedestrian tracking using local structure refinement), is developed to track people based on images acquired by a stereo camera pair mounted on a mobile platform. The tracker takes calibrated and normalised stereo image pairs, i.e. pairs with known interior and relative orientation as well as scale rectified to epipolar geometry, as input and provides 3D trajectories of pedestrians as output. Following the tracking-by-detection paradigm, our tracking pipeline is decomposed into three primary phases:

- First, detection takes responsibility to search for areas that people appear in image space and to delineate them with rectangular bounding boxes (BBs). Then, their positions are localized in object space using 2D image detections and 3D stereo information.
- Second, detections in adjacent epochs are linked together in the hierarchical data association stage, in which the most significant need to be fulfilled is maintaining correct identities (Ids) for the generated trajectories.

- Finally, trajectories of tracked pedestrians are smoothed in the prediction and filtering step. This phase also retrieves pedestrians in images that the detector method did not recognize due to difficulties such as occlusions or adverse illumination conditions, which is called missed detection recovery.

Besides the two main stages, the tracking pipeline also includes scene modelling based on 3D point clouds, which supports the positioning of detected pedestrians. A general overview of our tracker is shown in Figure 4.1.

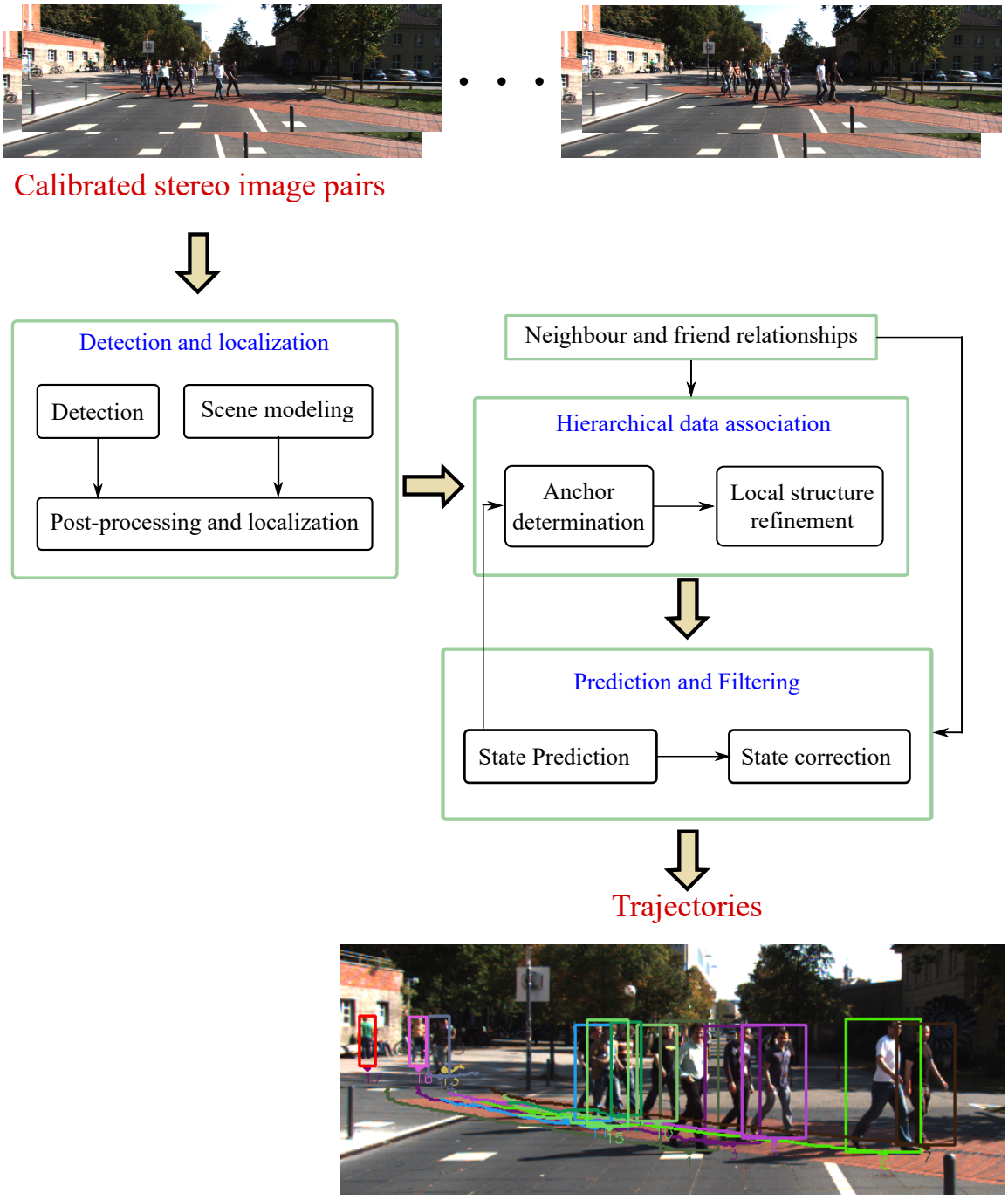


Figure 4.1: The overview framework of the proposed tracker.

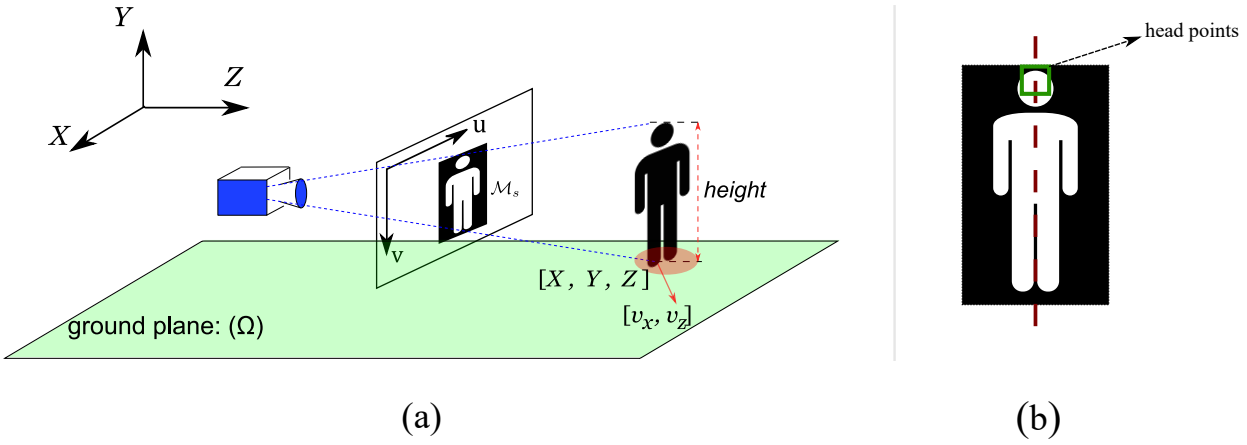


Figure 4.2: Our 3D coordinate system, in which the Z axis points in viewing direction. The 3D position and height of a detection are computed using the reconstructed the ground plane (Ω), the segmentation mask $\mathcal{M}_s$ and 3D information from the stereo rig. For the sake of simplicity, the stereo camera is reduced to only the left camera (a). The green box is used to select head points for estimating the height of a detection in 3D object space (b).

In this tracking approach, it is assumed that pedestrians only move on a ground plane (Ω) and the world coordinate system is defined as in Figure 4.2. The position of a pedestrian in 3D object space is considered to be its foot position. Therefore, tracked pedestrians only show movements in X and Z directions.

Let $\mathcal{D} = \{D_{1,t}, ..., D_{n,t}\}$ be n detections and $\mathcal{T} = \{\tau_{1,t}, ..., \tau_{m,t}\}$ be m tracked trajectories at epoch t. A detected object $D_{i,t}$ at epoch t includes its positions of the foot point in both stereo images, $I = [u, v, d]$, in which u and v are image coordinates and d is disparity value, and in 3D space, $P = [X, Y, Z]$. Each position I in image space is associated with an uncertainty σ_I. Apart from the positions, a detection $D_{i,t} = \{I, \sigma_I, P, \varrho, B\}$ also contains a detection confidence ϱ representing how likely the detection is TP and the 2D BB B. A trajectory $\tau_{j,t} = \{S_{j,k}, ..., S_{j,t-1}\}$ contains previous states of a tracked object. A state vector $S_{j,k} = [X, Y, Z, v_x, v_z]^T$ consists of 3D position and velocity. Note that people are assumed to move on the ground plane, so there is no movement in Y direction and v_y is ignored in the state vector. The detection confidence and the BB of a detection that assigned to τ_j at epoch t are denoted as $\varrho_{\tau_j,t}$ and $B_{\tau_j,t}$.

During tracking, a trajectory has one of three different attributes (see Figure 4.3): (1) if there is a detection assigned to the target, it is $active$; (2) once a trajectory is not assigned to any detection, it is called $inactive$ and its positions can be further predicted for a number of epochs; and (3) after a while, positions of an inactive target are not inferred any longer, because the predictions can be inaccurate, it then becomes $invalid$ and will be deleted after a few further epochs.

Two different relationships among trajectories are also defined. Observed targets are considered

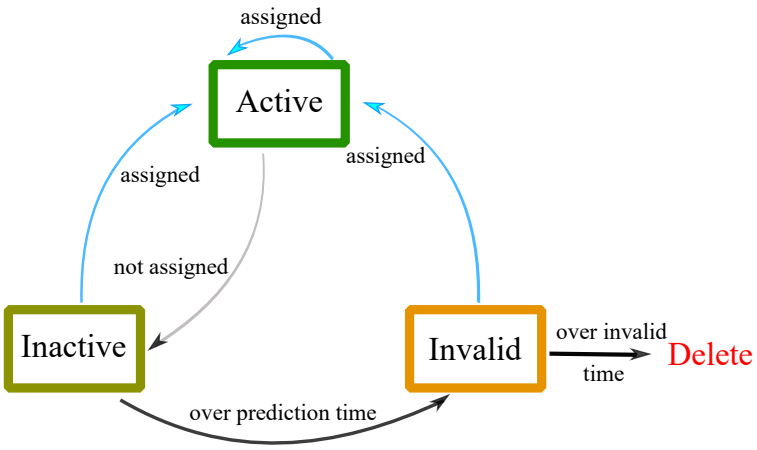


Figure 4.3: The three possible states of a trajectory and the transitions between them.

as $neighbours$ if their 3D distance is small. If trajectories are neighbours for a long time, they become $friends$. Their friendship ends when they are not neighbours any more for a certain number of epochs. Friends are supposed to have similar velocities.

4.2 Detection and localization

In this section, several steps are described to detect and estimate positions of pedestrians appearing in input images. Using the 3D stereoscopic information, the ground plane (Ω) in each stereo image pair is reconstructed. At the same time, pedestrians are detected in 2D input images individually based on their visual properties such as texture and shape using a detection method. Then, their 3D positions are computed by projecting their conjugate point pairs onto (Ω). Moreover, geometric constraints corresponding to pedestrians having a certain sizes are employed to filter out FP detections.

Given a stereo image pair with known orientation parameters, the disparity map ξ of all pixels with respect to the left image is first estimated using a state-of-the-art dense matching approach (Yamaguchi et al., 2014). Then, 3D point clouds are computed using the disparity d values.

4.2.1 Scene modelling

It is supposed that a scene is mainly composed of a more or less horizontal ground plane (e.g. a road), vertical planes (e.g. building facades) and the sky (which is not considered further). In addition, other objects such as pedestrians are presented, which are the objects of interest. Many objects in an urban scene can be considered as vertical planar surfaces supported by the ground plane. Reconstructing the ground plane in object space provides additional evidence for

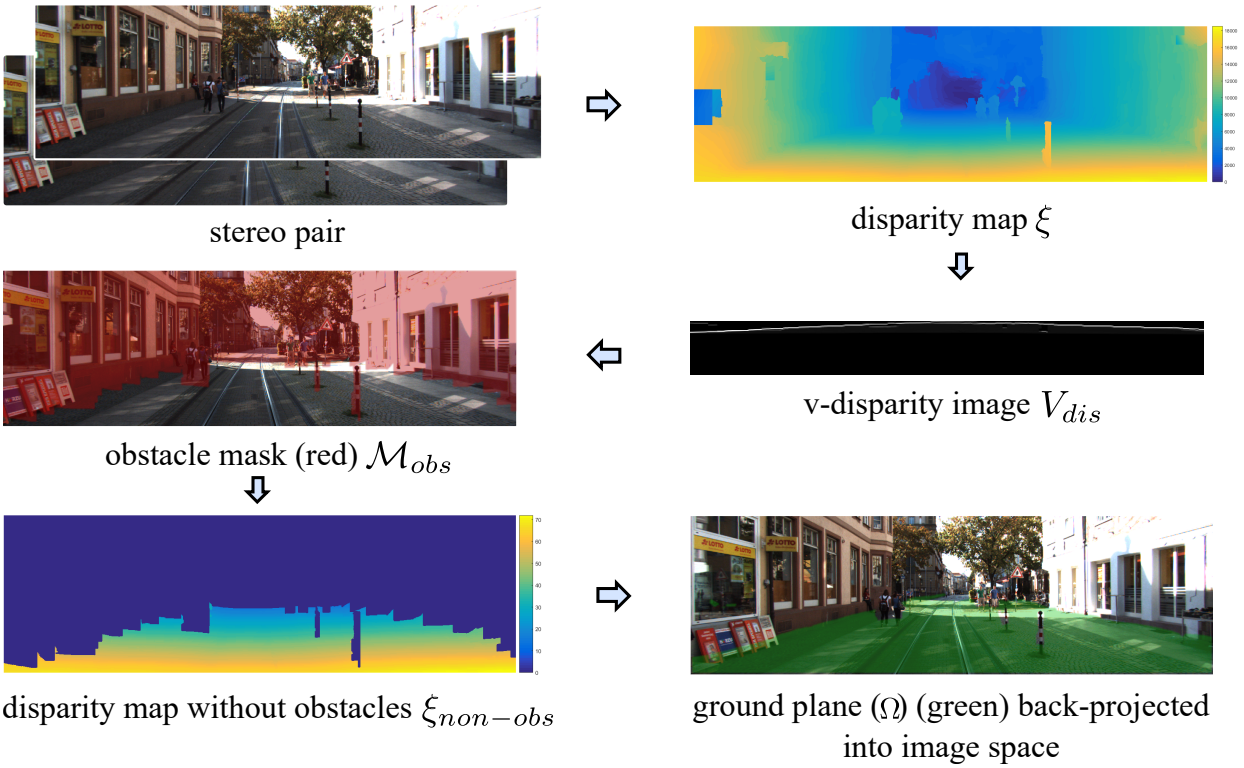


Figure 4.4: The procedure of modeling a scene from a stereo image input to reconstruct the ground plane (Ω).

pedestrian detection and localization, which involves the two steps of obstacle determination and plane estimation.

Potential obstacles

Potential obstacles, i.e. non-pedestrian objects, are defined as regions in an image which have a normal vector parallel to the ground plane. Under this definition, a number of pixels in each image column that have the same disparity value belong to an obstacle if images are taken with horizontal and parallel optical axes. A binary obstacle mask for each input image is estimated using the following steps proposed by Hu and Uchimura (2005):

1. From the disparity map ξ , the vertical or v-disparity image V_{dis} is computed such that each column in V_{dis} is a disparity histogram of the corresponding column in ξ with bin size of 5 px.

2. The intensity of a pixel (u, v) in V_{dis} represents the number of pixels in column v of ξ that have approximately disparity u. Hence, the binary obstacle mask $\mathcal{M}_{obs}$ can be built by finding pixels in the disparity map ξ, which correspond to pixels in V_{dis} with an entry larger than a threshold value. Those pixels are considered as obstacle regions in the obstacle mask $\mathcal{M}_{obs}$, for which $\mathcal{M}_{obs} = 0$ and the other remaining pixels are set to 1.

3. Morphological closing is used to join small obstacle regions together. Then, all non-obstacle regions smaller than a threshold are considered to be caused by errors and, thus, are included as obstacle areas.

Ground plane extraction

Unlike vertical object pixels, ground plane pixels should have similar depths per row. Using this assumption, the ground plane is estimated as follows:

1. The obstacle mask $\mathcal{M}_{obs}$ is used to eliminate pixels related to vertical objects in the disparity map, so that the remaining pixels in a new disparity map,

$$\xi_{non-obs} = \mathcal{M}_{obs} \odot \xi \ , \tag{4.2.1}$$

   mostly belong to the ground plane. In Equation (4.2.1), $\odot$ denotes a pixel-wise multiplication of the grey values.

2. Ground pixels in $\xi_{non-obs}$ are determined in a similar way as the obstacle mask. However, instead of using the v-disparity image, the disparity histogram of each row in $\xi_{non-obs}$ is computed to generate a horizontal or h-disparity image. As a result, ground pixels in $\xi_{non-obs}$ with their 3D positions are collected in a set $\mathcal{P}_{ground}$.

3. Any 3-D point (x, y, z) lying on the ground plane (Ω) must ideally satisfy the planar equation

$$(\Omega) : ax + by + cz + d = 0 \ ,$$

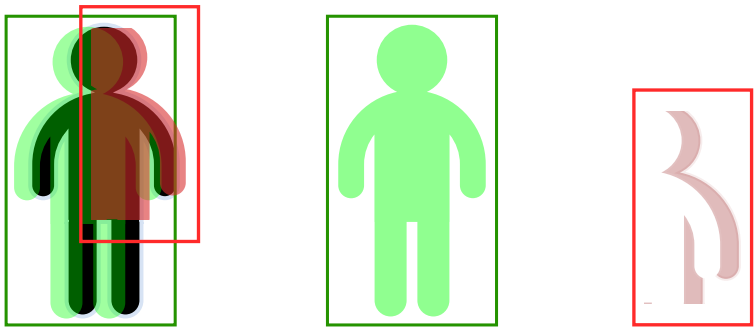
   where (a, b, c) is the normal vector with length 1 and d is the distance from the origin to (Ω). The ground plane (Ω) is determined using the 3-D ground points $\mathcal{P}_{ground}$, together with RANSAC to remove outliers.

4.2.2 Observations

The state-of-the-art detection method mask R-CNN (He et al., 2017) is used to detect pedestrians in the left image of each stereo pair individually. Though the detection can be performed on both, the left and right image, it is observed that this does not help to significantly boost the accuracy of the detection stage because the differences between the two results are not large. The pre-train mask R-CNN provided by (He et al., 2017) is directly adopted by the proposed tracker without re-training.

For each detected object, mask R-CNN provides:

- A BB $B = \{r, c, w, h\}$ closely covering the detection, where r and c represent the top left corner position, and w and h the size of the bounding box.



(a) (b) (c)

Figure 4.5: Two overlapping BBs corresponding to a pedestrian (a) and their mask after the processing to separate the overlapping area, (b) and (c). In this example the ϱ of the green BB is larger than the one of red BB.

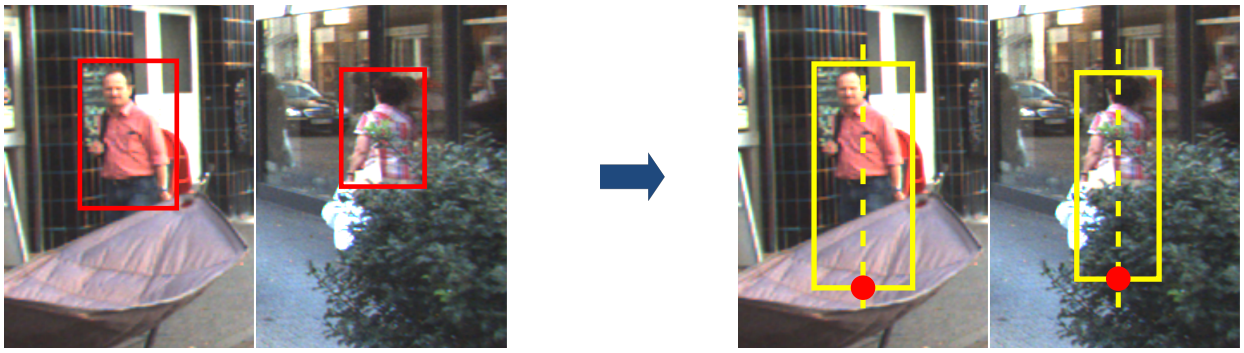
- An instance segmentation mask $\mathcal{M}_s$, separating the foreground from the background in the bounding box B.
- A confidence score ϱ for the probability that a detection is a TP.

In addition to the high accuracy, the instance segmentation mask $\mathcal{M}_s$ is a big advantage of mask R-CNN. This mask simplifies the estimation of the position and height of a target in object space. All detections classified as pedestrians and having a confidence value ϱ larger than a threshold $\varepsilon_{\varrho 1}$ are considered for post-processing. A detection with $\varrho < \varepsilon_{\varrho 1}$ is regarded as an FP.

Similar to other object detectors, mask R-CNN usually outputs multiple detections for only one pedestrian. Each of these detections is associated with an instance segmentation mask, yet they cover the same person as illustrated in Figure 4.5. This means that there can exist pixels belonging to that person which are included in $\mathcal{M}_s$ of several detections, which is not reasonable. Hence, here the masks are processed so that one pixel in an image can belong to at most one detection, which is selected to be the one with highest confidence belief ϱ.

After detecting a pedestrian in 2D image space, his/her foot position in 3D is computed using the segmentation mask $\mathcal{M}_s$ and the 3D point clouds. The point clouds obtained from matching are usually noisy, which leads to incorrect estimation of pedestrian positions in 3D object space. Therefore, it is essential to eliminate 3D points not belonging to a detected object. First, for each mask $\mathcal{M}_s$, a morphological erosion is utilised to shrink $\mathcal{M}_s$, this helps to deal with blunders. Then, for all 3D points corresponding to pixels in $\mathcal{M}_s$, a histogram of all depth values is estimated. The bin with the highest count has a Z value range of $[Z_1, Z_2]$. All 3D points of the mask $\mathcal{M}_s$ for which Z does not lie in that range are considered as noise and are not used for further computation.

To localize a pedestrian in 3D object space, all valid 3D points are then projected onto (Ω) and averaged to obtain the foot point $P = [X, Y, Z]$ of the pedestrian. Next, P is back-projected into



(a) (b)

Figure 4.6: The detected 2D bounding box (a) is corrected using the back-projected foot point from 3D (b).

image space to obtain the foot point in the stereo images $I = [u, v, d]$, where u and v are the image coordinates of the left image, and d is the disparity value. This procedure often allows to compute the 3D position and recover the entire body of an observed object in the input image even if only parts are visible as shown in Figure 4.6. The difference of BB height before and after the correction is denoted by ΔB_h, which is used to determine the uncertainty of v.

It is assumed that a number of points in the mask $\mathcal{M}_s$ having the smallest v value are head points of a detected object (see again Figure 4.2 for the definition of the images coordinate system and how the head points are selected). The rectangle is used to select head points is a shrinkage of the detected BB. From those head points in images and the point clouds, the head position of interesting objects in 3D $P_{head} = [X_{head}, Y_{head}, Z_{head}]$ is estimated, which is then used together with the foot point position to compute the object heights: $height = Y_{head} - Y$.

False alarm recognition

It is clear that a BB contains an object only if the number of pixels $|\mathcal{M}_s|$ in $\mathcal{M}_s$ is large enough, which is illustrated through the ratio between $|\mathcal{M}_s|$ and its BB size:

$$\zeta_{\mathcal{M}_s} = \frac{|\mathcal{M}_s|}{B_w B_h} .$$

In addition, a pedestrian is a special type of object in which height and width are limited to a certain range. However, as the mask R-CNN is pre-trained to detect various object types, the specific characteristics related to human size are ignored, and thus the algorithm yields a number of false alarms. These can partly be detected and eliminated by utilising additional properties as follows:

- Pedestrian heights ($height$) are limited in a certain range $[\varepsilon_{H_1}, \varepsilon_{H_2}]$.
- A pedestrian detection must be covered by a BB whose ratio between height and width $\zeta_B = \frac{B_h}{B_w}$ is bounded by $[\varepsilon_{Br_1}, \varepsilon_{Br_2}]$.

- A BB is a TP if its instance segmentation mask is large enough $\zeta_{\mathcal{M}_s} > \varepsilon_{\mathcal{M}_s}$.

Detected objects that do not satisfy these three constraints are not further considered in the tracking phase.

Position uncertainties

Together with the image position I, its uncertainties $\sigma_I = [\sigma_u, \sigma_v, \sigma_d]$ is also estimated. This uncertainty vector provides essential information to determine possible areas that can contain the position of a detected pedestrian in image space, which is important for smoothing the trajectories of tracked objects in the filtering step.

The values of σ_u and σ_v are approximated heuristically based on the detection confidence score ϱ and the BB size as follows (note that $0.0 < \varrho \leq 1.0$):

$$\begin{aligned} \sigma_u &= \max(0.05 B_r/\varrho, \eta_u) \\ \sigma_v &= \max(\Delta B_h/\varrho, \eta_v) \end{aligned} , \tag{4.2.2}$$

where η_u and η_v are minimum values for σ_u and σ_v.

In addition, in this tracking framework, the uncertainty of d for each detected pedestrian is required. To approximate σ_d, two depth values are predicted directly from the BB size of a detection, which theoretically should be close to the true depth value. The uncertainty of the depth estimated from the 3D point clouds is assessed based on these predicted depths.

Assume that the mid-point of the BB with the lowest v corresponds to the head point B_{head} of the pedestrian in the image and the BB mid-point with the largest v is the foot point B_{foot}. The conversion between image space and 3D object space of those points are given by:

$$\begin{aligned} & B_{head} = c_u - \frac{fY_{head}}{Z}, \quad B_{foot} = c_u - \frac{fY_{foot}}{Z} \\ \Rightarrow & B_h = B_{foot} - B_{head} = \frac{f(Y_{head} - Y_{foot})}{Z} \\ \Rightarrow & B_h = \frac{fPed_H}{Z} , \end{aligned} \tag{4.2.3}$$

in which f is the focal length of the camera, c_u and c_v are the image coordinates of the principal point, Ped_H is the height of the observed pedestrian, Z axis points in the viewing direction of the camera.

Similarly, the width B_w of a BB can be computed from the real width Ped_W in object space of a pedestrian:

$$B_w = \frac{fPed_W}{Z} . \tag{4.2.4}$$

Employing Equation (4.2.3) and Equation (4.2.4) and using standard values for Ped_H and Ped_W, values for the z-coordinate of the foot point Z_H^+ based on the BB height B_h and the foot point Z_W^+

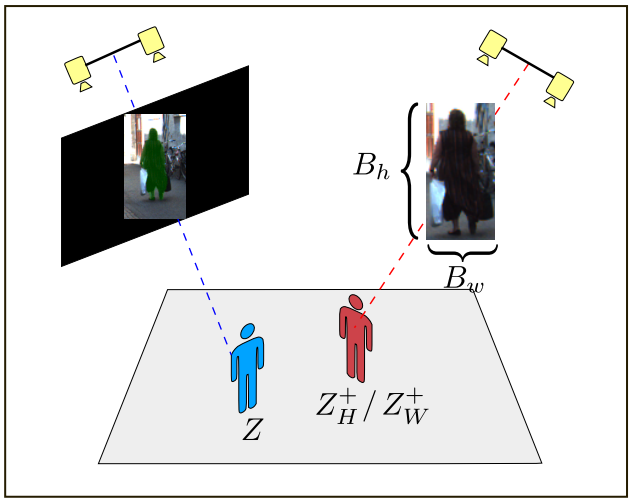


Figure 4.7: The depth Z in the estimated position P which is computed using 3D point clouds. $Z_{W,H}^+$ is predicted based on BB size and camera parameters. The difference between Z and $Z_{W,H}^+$ is a clue to determine the uncertainty σ_Z of Z.

using the BB width B_w can be predicted as follows:

$$Z_H^+ = \frac{fPed_H}{B_h}\,, \quad Z_W^+ = \frac{fPed_W}{B_w}\,, \tag{4.2.5}$$

These depth values can then be compared to the depth Z derived from the 3D point clouds. The uncertainty of Z is calculated using the following equation:

$$\begin{aligned} a &= min(|Z_H^+ - Z|, |Z_W^+ - Z|) \\ \sigma_Z &= \begin{cases} a, \text{ if } a \geq \varepsilon_Z \\ \eta_{\sigma_Z}, \text{ otherwise} \end{cases}\,, \end{aligned} \tag{4.2.6}$$

where η_{σ_Z} is a constant. The threshold ε_Z needs to be selected large enough to ensure that the difference between Z and $Z_{W,H}^+$ is the result of the matching uncertainty, and does not stem from the fact that people can have slightly different height Ped_H and width Ped_W. σ_d is then calculated based on σ_Z through error propagation as follows:

$$\sigma_d = \frac{fBase}{Z^2}\sigma_Z\,, \tag{4.2.7}$$

where f is the camera focal length and $Base$ is the base length of the stereo system.

4.3 Hierarchical data association

The detections in two consecutive images are linked in an online manner using a hierarchical data association approach called local structure refinement (LSR). The quality of association can be

considered as the most significant problem of every tracker, in which the identities (Ids) of tracked pedestrians should be consistently maintained. To determine whether two detections belong to the same person, the two cues geometry and appearance are utilised. While the geometry cues are derived from the fact that at the same time, there can be at maximum one object occupying a specific place in 3D space, the appearance cue is obtained by hypothesising that the visual properties of a pedestrian remain similar in a small period of time. Though these two cues can help to re-identify the pedestrian in two adjacent images, problems can arise in complicated situations, where both geometry and appearance become ambiguous and indistinguishable. Therefore, to improve the accuracy of data association, the proposed approach is carried out in two steps: anchor determination and LSR. Trajectories that are assigned to detections with high probability of correctness are defined as $anchors$, these are matched first. Assigning of less reliable detections is then supported based on the geometric adjustments of the anchors in the LSR step. While the introduced association approach can be proceeded into two steps, the LSR is only applied if at least one anchor is found in the first step. Otherwise, the assignment results are directly obtained in one step.

Figure 4.8: Association result without (left) and with (right) the use of anchors (green boxes). The detections in the previous frame are denoted in dashed lines, the current detections are shown in solid lines.

4.3.1 Anchor determination

In this step, a number of trajectories matched to detections with a high degree of accuracy are determined. This includes the calculation of similarity (i.e. association weight) between a detection and an existing target and the global optimization to find the optimal assignment results.

Association weight

This weight describes the likelihood that an observation to be assigned to a target, which is primarily explained by its visual appearance $\Gamma_{\mathcal{A}}$ and spatial distance $\Gamma_{\mathcal{G}}$ similarity. Beyond that, a high confidence detection is preferred to be allocated to existing trajectories over one with low

confidence. The association weight is computed as follows:

$$w_i^j = \rho\Gamma_{\mathcal{G}}(D_{i,t}, \tau_{j,t}) + \theta\Gamma_{\mathcal{A}}(D_{i,t}, \tau_{j,t}) + \nu\varrho_{D_{i,t}} \,, \tag{4.3.1}$$

where ρ, θ, and ν are parameters used to define the impact of each criterion on the association weight value and $\rho+\theta+\nu = 1$. The component $\Gamma_{\mathcal{G}}$ and $\Gamma_{\mathcal{A}}$ are defined in the following paragraphs.

Geometry similarity

This value is related to the 3D spatial distance of an object and its potential target. Let $S_{j,t}^+$ be a predicted state of $\tau_{j,t}$ at an epoch t, which is estimated by the Kalman filter (see Equation (4.5.1)). The Mahalanobis distance $\phi_{\mathcal{G}}$ is computed in 3D space between the predicted position $S_{j,t}^+$ at t of $\tau_{j,t}$ and the 3D position $P_{D_{i,t}}$ of $D_{i,t}$ as their geometry affinity. Using this distance, both the position and the uncertainty of the prediction state are taken into account:

$$\phi_{\mathcal{G}}(D_{i,t}, \tau_{j,t}) = \sqrt{(S_{j,t}^+ - P_{D_{i,t}})^T(\Sigma_{SS,t}^+)^{-1}(S_{j,t}^+ - P_{D_{i,t}})} \,, \tag{4.3.2}$$

where $\Sigma_{SS,t}^+$ is the predicted variance of $S_{j,t}^+$ (see Equation (4.5.2)). In the above calculations, only the position entries $[X, Y, Z]$ of $S_{j,t}^+$ is used while the velocity elements are disregarded.

$\phi_{\mathcal{G}}$ is then mapped to a value range of 0.0–1.0 by an exponential function to obtain the criteria $\Gamma_{\mathcal{G}}$:

$$\Gamma_{\mathcal{G}}(D_{i,t}, \tau_{j,t}) = e^{-\frac{\phi_{\mathcal{G}}(D_{i,t}, \tau_{j,t})}{\eta_{\mathcal{G}}}} \,, \tag{4.3.3}$$

where $\eta_{\mathcal{G}}$ is a free parameter.

Appearance similarity

The appearance similarity accounts for the resemblance between two objects in image space in terms of texture, color, shape, etc. Besides the geometric similarity, this is a significant cue to distinguish between different persons. The visual properties of a detection are represented by a feature vector f. TriNet (Hermans et al., 2017) is employed to extract the appearance feature vector f of an interesting object based on its BB. However, instead of directly feeding a BB to the network as input, the segmentation mask $\mathcal{M}_s$ is used to isolate an object from the background first and then the background pixels are replaced with random values as shown in Figure 4.9. The random noise prevents TriNet from using background information for feature extraction. Thus, the results of the visual comparison can be enhanced.

At time t, the feature vector of a trajectory $\tau_{j,t}$ is the average of its appearance vectors from a certain number of previous epochs, which can account for visual properties of a trajectory within a temporal window. The appearance similarity $\Gamma_{\mathcal{A}}$ between $D_{i,t}$ and τ_j is computed as:

$$\begin{aligned} \phi_{\mathcal{A}}(D_{i,t}, \tau_{j,t}) &= \|f_{\tau_{j,t}} - f_{D_{i,t}}\|_{L_2} \\ \Gamma_{\mathcal{A}}(D_{i,t}, \tau_{j,t}) &= e^{-\frac{\phi_{\mathcal{A}}(D_{i,t}, \tau_{j,t})}{\eta_{\mathcal{A}}}} \end{aligned} \,, \tag{4.3.4}$$

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

Figure 4.9: (a) and (b) show the detection results of two pedestrians. Note that one pedestrian occludes the other. (c) and (d) depict the results after separation and using random pixels as background.

where $\eta_{\mathcal{A}}$ is a free parameter, and $\|f_{\tau_{j,t}} - f_{D_{i,t}}\|_{L_2}$ is the Euclidian distance between the two feature vectors $f_{\tau_{j,t}}$ of the trajectory and $f_{D_{i,t}}$ of the detection.

Association gates

Since there is at maximum only one person can occupy a spot in 3D object space at a specific time, the distance between a detection and its corresponding target must be small in both, image and object space. Exploiting this property, two geometric gates are generated, which indicate whether a detection can be assigned to a target or not. The first gate is used to restrict detections and trajectories that are distant in 3D object space, which is called 3D gate:

$$\text{gate}_{3D}(D_{i,t}, \tau_{j,t}) = \begin{cases} 1, \text{ if } ||P_{D_{i,t}}, S^{+}_{\tau_{j,t}}|| < \varepsilon_{3D-gate} \\ 0, \text{ otherwise} \end{cases} . \tag{4.3.5}$$

The second gate, named 2D gate, guarantees that the BBs IoU in image space of a detection and a target that belong to same pedestrian at epoch t must be larger than a threshold:

$$\text{gate}_{2D}(D_{i,t}, \tau_{j,t}) = \begin{cases} 1, \text{ if IoU}(B_{D_{i,t}}, B^{+}_{\tau_{j,t}}) > \varepsilon_{2D-gate} \\ 0, \text{ otherwise} \end{cases} , \tag{4.3.6}$$

where $B^{+}_{\tau_{j,t}}$ is the predicted BB of trajectory $\tau_{j,t}$ at epoch t (see Section 4.4). While the 3D gate reduces the confusion of pedestrians at spares level, the 2D gate helps to increase grouping.

These gates compensate for indistinguishable appearance between tracked pedestrians to avoid incorrect associations in case different pedestrians look similar. In addition, they help to reduce the complication of the optimization problem as the number of hypothesis assignments become smaller. These gating results are directly included in the assignment optimization using linear programming by modifying the association weight value as follow:

$$w^{j}_{i} = w^{j}_{i} \, \text{gate}_{3D}(D_{i,t}, \tau_{j,t}) \, \text{gate}_{2D}(D_{i,t}, \tau_{j,t}) \, . \tag{4.3.7}$$

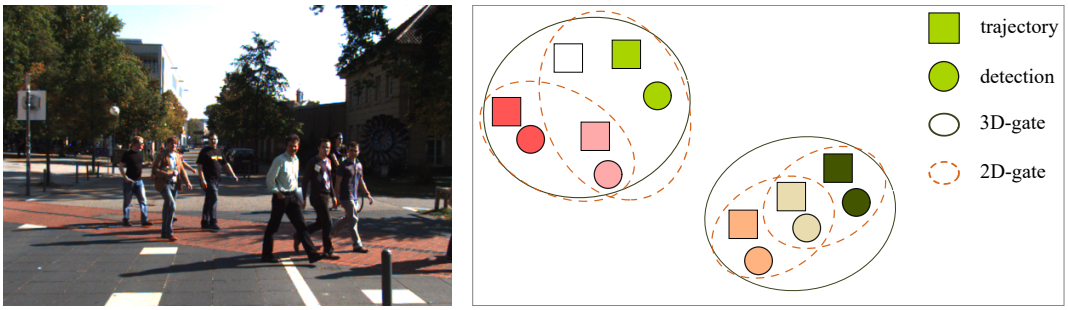


Figure 4.10: The position of detections in the current image (left) and existing trajectories in the previous time step are clustered in groups using 3D and 2D gates. An unions between two gates do not necessarily empty.

Tracking-confirm-detection

Since detected pedestrian results can be noisy, using a single detection confidence threshold (DCT) is usually hard to achieve high recall and low false alarm at the same time. Considering observed objects with a low score as TPs can result in inaccurate trajectories which contain FPs and also make the association become incorrect. On the other hand, using only detections with high confidence scores can lead to less tracked pedestrians or increasing fragmentation of trajectories.

To mitigate this problem, in the proposed tracking-confirm-detection (TCD) approach, two predefined DCTs are utilised: a low $\varepsilon_{\varrho 1}$ and a high $\varepsilon_{\varrho 2}$. All detections with a confidence value larger than $\varepsilon_{\varrho 1}$ are considered during assignment optimization. The reason is that a trajectory can be used to confirm the presence of a TP detection even if its confidence value is low. However, when a new trajectory is created, there is no additional evidence to confirm its correctness other than its detection confidence. Hence, at a specific epoch, a detection that is not assigned to any existing target initializes a new trajectory if its confidence value is larger than $\varepsilon_{\varrho 2}$.

Assignment optimization

The problem of assigning n detections in $\mathcal{D}$ to m targets in $\mathcal{T}$ is solved using a binary integer program. However, since a detection may not belong to any existing target, a dummy trajectory representing a potential new trajectory is assigned to every observation with a defined weight value. The assignment objective is to maximize the sum of association weight, while still maintain a set of constraints as follows:

$$\begin{cases} \text{maximize} & c^T w \\ \text{subject to} & (Ac)_k \leq 1,\, k = 0, ..., (n+m) \end{cases}, \tag{4.3.8}$$

where c is an $(nm+n)$ indicator vector. For $c_i^j = 1$ the detection $D_i \in \mathcal{D}$ and trajectory $\tau_j \in \mathcal{T}$ are associated with each other, otherwise, $c_i^j = 0$; τ^* is a dummy variable, which means that a new trajectory is created. The association weight $w_i^j \in w = \{w_i^j, ..., w_n^*\}$ describes how likely D_i and τ_j belong to one and the same person; w_i^* is set to a constant value. A is a $(n+m) \times (nm+n)$

design matrix and has the effect that one detection is assigned to at most one trajectory and vice versa:

$$A = \underbrace{\begin{bmatrix} \underbrace{1\dots1}_{n} & \dots & \underbrace{0\dots0}_{n} \\ \vdots & \ddots & \vdots \\ 0\dots0 & \dots & 1\dots1 \\ 10\dots0 & \dots & 10\dots0 \\ \vdots & \ddots & \vdots \\ 0\dots01 & \dots & 0\dots01 \end{bmatrix}}_{mn+n} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} m \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n \end{matrix} \tag{4.3.9}$$

and

$$c = \begin{bmatrix} \underbrace{c_0^0 \dots c_n^0}_{n} & \dots & \underbrace{c_0^m \dots c_n^m}_{n} & \underbrace{c_0^* \dots c_n^*}_{n} \end{bmatrix} . \tag{4.3.10}$$

After the optimization using IP, the anchor can be chosen by two different strategies:

- A trajectory has an assignment with an association weight larger than a threshold ε_{an1} , it is then considered as an anchor. This way, the chosen anchors are guaranteed to be correct at a certain level. However, at an epoch, there may be no anchor.
- A certain percentage ε_{an2} of trajectories with the highest association weight are anchors. Obviously, in this scheme, always at least one anchor is determined. However, some anchors can be unreliable, which may result in unstable and incorrect prior information for LSR.

4.3.2 Local structure refinement

The assignment results obtained from the anchor determination step are usually accurate in case pedestrians appear clearly in image space. In crowded groups, where occlusion can happen often and pedestrians also move very near to each other, preserving a correct Id for tracked targets becomes much more difficult. To improve the association accuracy in these situations, the geometry changes of anchors can be employed as additional information to find the correct assignments for other trajectories.

Since the movement of the camera system results in only global changes in image space, and pedestrians do not move fast, in adjacent epochs the geometry changes in image space of two nearby trajectories τ_1 and τ_2 are similar. This assumption can be expressed through the IoU as follows:

$$IoU(B_{\tau_1,t-1}, B_{\tau_1,t}) \approx IoU(B_{\tau_2,t-1}, B_{\tau_2,t}) , \tag{4.3.11}$$

where $B_{\tau_1,t-1}$, $B_{\tau_2,t-1}$, $B_{\tau_1,t}$, and $B_{\tau_2,t}$ are BBs corresponding to τ_1 and τ_2 at $t-1$ and t, respectively.

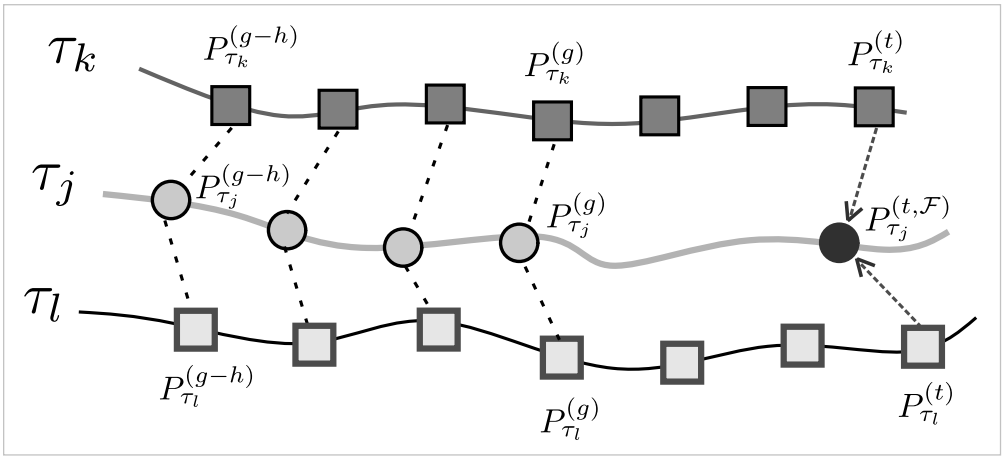


Figure 4.11: The position $P_{\tau_j}^{(t,\mathcal{F})}$ at epoch t of invalid trajectory τ_j is inferred using its anchor friends τ_l and τ_k. The relative positions between τ_j and its friends are estimated in h epochs before it starts to be inactive at epoch g.

Let a trajectory $\tau_{j,t}$ have l neighbours which are anchors. $\mathcal{I} = \{iou_1, ..., iou_l\}$ are the IoUs between those anchor neighbours and their assigned detections. Assume that $\mathcal{I}$ has normal distribution with mean $\mu_{\mathcal{I}}$ and standard deviation $\sigma_{\mathcal{I}}$. Let $B_{\tau_j,t-1}$ be the BB at epoch $(t-1)$ of the tracked pedestrian corresponding to trajectory τ_j and $D_{i,t}$ the correct candidate detection assigned to τ_j at t. Then, the IoU iou_j^i between $B_{\tau_j,t-1}$ and $D_{i,t}$ should have a similar distribution as $\mathcal{I}$, which is modelled through the normalised maximum likelihood function $\Gamma_{\mathcal{G}^*}(D_{i,t}, \tau_{j,t}, \mathcal{I})$ as in Equation (4.3.12). The association weight is modified to take the local structure constraint of neighbours reflecting in Equation (4.3.11) into account as follows:

$$\begin{aligned} w_i^j &= \rho\Gamma_{\mathcal{G}^*}(D_{i,t}, \tau_{j,t}, \mathcal{I}) + \theta\Gamma_{\mathcal{A}}(D_{i,t}, \tau_{j,t}) + \nu\varrho_{D_{i,t}} \\ \Gamma_{\mathcal{G}^*}(D_{i,t}, \tau_{j,t}, \mathcal{I}) &= e^{-\frac{(iou_j^i - \mu_{\mathcal{I}})^2}{2\sigma_{\mathcal{I}}^2}} \end{aligned}, \tag{4.3.12}$$

where ρ, θ and ν are again free parameters, and again $\rho + \theta + \nu = 1$. Note, the same symbols are used as in Equation (4.3.1) (and also in eq. Equation (4.3.14) below), although the values of those free parameters are not necessarily the same. In this system, however, the same numerical values are used in the experiments.

Here, only neighbours which are anchors as well are utilised, so that our LSR is less prone to error due to incorrect matching in the first step. This LSR step is only applied to active and inactive targets and detections which are not considered as anchors in the first step.

Invalid trajectory handling

Once a target becomes invalid, it means that its predicted positions are not very trustworthy any more and its appearance can also become ambiguous. This is especially hard to handle when the tracked pedestrian also moves in a crowd, where there are a lot of candidate detections close to it and occlusions regularly occur. However, while in a group, a target τ_j typically walks together

with other tracked objects, which are considered as friends. Friends usually move in a similar way and cover a similar 2D and 3D distance in image and object space, respectively. Therefore, the position $P_{\tau_j}^{(t,\mathcal{F})}$ at epoch t of τ_j can be estimated according to its anchor friends $\mathcal{F} = \{\tau_l, ..., \tau_k\}$ as follows:

$$P_{\tau_j}^{(t,\mathcal{F})} = \frac{1}{k-l+1}\sum_{i=l}^{k}(P_{\tau_i}^{(t)} + \frac{1}{h+1}\sum_{q=g-h}^{g} P_{\tau_j}^{(q)} - P_{\tau_i}^{(q)}) \,, \tag{4.3.13}$$

where g is the epoch that τ_j starts to become invalid and the difference between positions of τ_j and its friends τ_i are estimated in h epochs before g. Moreover, the average $\mu_{dis_\mathcal{F}}$ and standard deviation $\sigma_{dis_\mathcal{F}}$ of 2D distances of all trajectories of $\mathcal{F}$ between epochs g and t are accounted as a threshold to restrict the possible area that a correct detection of $\tau_{j,t}$ can appear in image space. The association weight between $\tau_{j,t}$ and a detection $D_{j,t}$ is then computed as follows:

$$w_i^j = \begin{cases} \rho\Gamma_\mathcal{G}(D_{i,t}, P_{\tau_j}^{(t,\mathcal{F})}) + \theta\Gamma_\mathcal{A}(D_{i,t}, \tau_{j,t}) + \nu\varrho_{D_{i,t}} \\ 0, \text{ if } \quad |dis(D_{i,t}, \tau_{j,g}) - \mu_{dis_\mathcal{F}}| > \eta_\mathcal{F}\sigma_{dis_\mathcal{F}} \end{cases} . \tag{4.3.14}$$

After the LSR step and recomputing the association weights for invalid trajectories, the global optimum association results are obtained by using linear programming as presented in Equation (4.3.8).

4.4 Motion correction and position prediction

Detection is a difficult task, in which challenges can come, e.g., from illumination, scale, occlusion, and unusual shape of pedestrians. Hence, during tracking, some interesting objects can be missed, which is severe in online applications, because instance responses are demanded at every epoch. Therefore, retrieving missed detections is an important task of a tracking system, which not only improves the tracking results by increasing the number of TPs but also reduces the fragmentation of tracked targets. Employing the trajectory information, the positions of missed detections can be recovered through prediction. However, this inference can also create more FPs as soon as predicted positions drift away from the true ones or the prediction is applied to trajectories that do not represent pedestrians. Therefore, it is important to assess how long the prediction should last and evaluate the correctness of predicted positions. In order to answer these questions, several concerns have been investigated including computing velocity and its correctness of interesting targets, using relationships among pedestrians to correct velocity of a desired target, and termination conditions, which are detailed in next sub-sections.

4.4.1 Velocity calculation and correction

Due to uncertainties in detection and association, the position of a target also contains uncertainties, which can result in incorrect velocities. A simple but efficient approach to estimate the 3D velocity of a tracked target and evaluate its correctness is proposed. As people typically do not change their speed and direction of movement significantly within a short time interval, it is assumed that their velocities in several epochs are similar. Therefore, the more consistent the velocities during those epochs, the higher the probability that they are reliable. Let $\mathcal{V}_X = \{v_{X,t-k+1}, ..., v_{X,t}\}$ be velocities in direction of the X-axis of a target, calculated from its 3D positions for the most recent k epochs. We estimate the histogram of $\mathcal{V}_X$, all $v_{X,*}$ that fall in the bin with the highest count a are averaged to obtain v_X. The reliability of v_X is assessed by the frequency of the highest bin: $p(v_X) = a/k$. The same calculations are applied to compute v_Z and $p(v_Z)$.

To add more credit to the correctness of the estimated velocity $\overrightarrow{v} = [v_X, v_Z]$, the least square method is applied to fit the positions of the observed trajectory in the k most recent epochs to a straight line $\mathcal{L}$ with standard deviation being the slope $\sigma_{\mathcal{L}}$. Based on the movement of the target, the direction of $\mathcal{L}$ is determined. $\mathcal{L}$ is afterwards transformed to vector form $\overrightarrow{\mathcal{L}}$. The posterior probability of $\overrightarrow{v}$ is updated as follows:

$$\begin{aligned} p(\overrightarrow{v}\,|\,\overrightarrow{\mathcal{L}}) &\approx p(\overrightarrow{\mathcal{L}}\,|\,\overrightarrow{v})p(\overrightarrow{v}), \quad p(\overrightarrow{v}) = p(v_X)p(v_Z) \\ p(\overrightarrow{\mathcal{L}}\,|\,\overrightarrow{v}) &= \frac{1}{\sqrt{2\pi\sigma_{\mathcal{L}}^2}} e^{-\frac{\alpha^2}{2\sigma_{\mathcal{L}}^2}} \end{aligned} \quad , \tag{4.4.1}$$

where α is the angle between $\overrightarrow{v}$ and $\overrightarrow{\mathcal{L}}$. For the rest of the paper, the notation $p(v)$ is used instead of $p(\overrightarrow{\mathcal{L}}\,|\,\overrightarrow{v})$ for simplicity.

Motion correction

While a social force model (SFM) is employed by many trackers to model the behaviour of pedestrians w.r.t. their surrounding environment, this requires all objects which can affect the movement of a pedestrian to be detected first. Moreover, evaluating the relationship between a person and his/her nearby objects may also be needed such as in case of repulsive forces for interesting events or other objects, such as friends. Thus, using a SFM is only efficient if there is no missed detection and prior knowledge about pedestrians is available. These requirements are hard to fulfil in real world and highly dynamic scenes. Based on similar force terms to predict the movement changes of tracked pedestrians, the proposed method indirectly explains the observed changes through movement of neighbouring pedestrians. Friend trajectories are supposed to have similar velocities and reactions to their surrounding environment. In addition, as image sequences are usually captured at a high frequency, the velocity of a pedestrian between two epochs should only vary slowly. Therefore, it is beneficial to predict motion tendencies of inactive trajectories w.r.t. to their friends, but not the active ones. On the other hand, the velocity of people moving in

a group is also corrected. In order to avoid adding errors to corrected velocities, only targets with highly reliable velocities can be used to correct the estimated movement of their friends whose velocities are more uncertain.

Let τ_j be a trajectory with an unreliable velocity which has a set of friend trajectories $\mathcal{F} = \{\tau_l, ..., \tau_k\}$. The movements of friends and the own moving properties of a target are assumed to have equal effects on updating the velocity of that target. Thus, the velocity of τ_j can be updated as follows, where the influence of the target and that of its friends have been set equal:

$$\begin{aligned} v_X^{\tau_j} &= 0.5\frac{\sum_{i=l}^{k} p(v_{\tau_i}) v_X^{\tau_i}}{\sum_{i=l}^{k} p(v_{\tau_i})} + 0.5 v_X^{\tau_j} \\ p(v_{\tau_j}) &= 0.5\frac{1}{k-l+1}\sum_{i=l}^{k} p(v_{\tau_i}) + 0.5 p(v_{\tau_j}) \end{aligned} \quad . \tag{4.4.2}$$

The same calculations are carried out for $v_Z^{\tau_j}$.

4.4.2 Missed detections retrieval

There are two critical criteria that need to be fulfilled to obtain good predictions: (a) the last active state S_t at frame t is highly accurate, which means that both, position and velocity are reliable, and (b) the last detection which is assigned to the trajectory should be a TP. The accuracy of S_t is evaluated using the difference between S_t and its predicted position S_t^+ (see Equation (4.5.1)). Based on the listed cues, the number of epochs that for $\tau_{j,t}$ inference can take place is estimated as follows:

$$\mathcal{N}_{\tau_{j,t}} = \varrho p(v) e^{-\frac{||S_t^+ - S_t||_{L2}}{\eta_{\mathcal{N}}}} \varepsilon_{\mathcal{N}} \,, \tag{4.4.3}$$

where ϱ is the confidence of the detection assigned to $\tau_{j,t}$ at epoch t; $\varepsilon_{\mathcal{N}}$ is the maximum number of inactive epochs a trajectory can have; and $\eta_{\mathcal{N}}$ is a constant. The 3D positions are predicted using the Kalman filter as in Equation (4.5.1), which are subsequently back projected into image space to obtain the 2D foot point. The BB is moved to the new foot point position and its size is updated according to the change of the distance between the object and the camera system.

Let S_{t+1}^+ and I_{t+1}^+ be predicted positions in object and image space of a inactive trajectory at $(t+1)$. The inferred BB B_{t+1}^+ is determined by moving its previous BB B_t to a new position such that I_{t+1}^+ lies in the middle of the bottom edge of B_{t+1}^+ (see Figure 4.12). The predicted BB is then examined whether it contains the tracked pedestrian or not based on its percentage of pixels that have 3D positions similar to S_{t+1}^+. If most of the 3D points in B_{t+1}^+ lie further away from the camera than the 3D predicted position S_{t+1}^+, it is assumed that there is no object in B_{t+1}^+. In the case of a large portion of 3D points nearer to camera than S_{t+1}^+, it is assumed that the object is occluded.

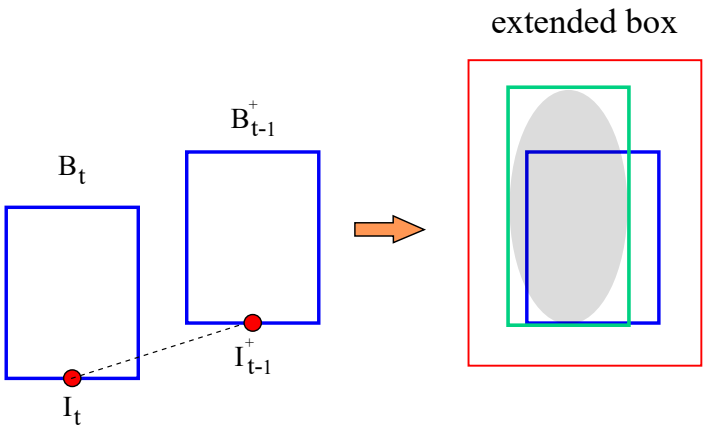


Figure 4.12: 2D bounding box prediction and correction. The corrected box (green) tightly covers the inactive target object (gray).

Once the presence of an object in a predicted BB is assumed, the BB is first enlarged by a fixed amount, then all pixels in the extended BB with similar depth are assumed to belong to the object in question. The predicted BB is adjusted to cover all those points as shown in Figure 4.12.

4.5 Filtering

As a trajectory evolves over time, pedestrian states consisting of positions and velocities close in time are correlated. Therefore, the state of the trajectory at a specific epoch can be predicted from its previous states. This predicted state is employed to correct the current measurement using an extended Kalman filter (Gelb, 1974).

Prediction

Let the state vector of a trajectory at $(t-1)$ be $S = [X_{t-1}, Y_{t-1}, Z_{t-1}, v_{x,t-1}, v_{z,t-1}]^T$, its predicted state vector has the form of $S^+ = [X_t^+, Y_t^+, Z_t^+, v_{x,t}^+, v_{x,t}^+]^T$. While the position of X_t^+ and Z_t^+ are computed based on the velocities and position of last epoch, Y_t^+ is considered to be not changed. It is assumed that there is no acceleration between a small period time of two epochs.

$$S_t^+ = \psi S_{t-1}$$
$$\psi = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.5.1}$$

where ψ is the transition matrix, which transforms a state vector S_{t-1} to the current epoch and Δt is the time interval between two epochs.

The covariance matrix Σ^{+}_{SS} of the predicted state S^{+} is estimated based on the covariance matrix of previous epoch and process noise Q_{pn}:

$$\Sigma^{+}_{SS,t} = \psi \Sigma_{SS,t-1} \psi^{T} + Q_{pn} \, . \tag{4.5.2}$$

The process noise accounts for unexpected events happening while pedestrians move such as accelerations a_x and a_z. Those changes can violate the assumption that v_x, v_z, and Y are constant, which is described in form of $u = [a_x, a_z, v_y]$. Since u is caused by unforeseen incidents, it is assumed to have white noise with mean of 0 and covariance $\Sigma_{uu} = diag(\sigma^2_{a_x}, \sigma^2_{a_z}, \sigma^2_{v_y})$. The uncertainty of u affects the predicted state and is taken into account through the process noise as follows:

$$Q = G \Sigma_{uu} G^{T} = \begin{bmatrix} \frac{\sigma^2_{a_x} \Delta t^4}{4} & 0 & 0 & \frac{\sigma^2_{a_x} \Delta t^3}{2} & 0 \\ 0 & \sigma^2_{v_y} \Delta t^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sigma^2_{a_z} \Delta t^4}{4} & 0 & \frac{\sigma^2_{a_z} \Delta t^3}{2} \\ \frac{\sigma^2_{a_x} \Delta t^3}{2} & 0 & 0 & \sigma^2_{a_x} \Delta t^2 & 0 \\ 0 & 0 & \frac{\sigma^2_{a_z} \Delta t^3}{2} & 0 & \sigma^2_{a_z} \Delta t^2 \end{bmatrix} , \tag{4.5.3}$$

in which G is defined as:

$$G = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & 0 & \Delta t \end{bmatrix} . \tag{4.5.4}$$

Update

The measurement $I_t = [u_t, v_t, d_t]$ of the current time is obtained with the measurement noise V_F . The noise is assumed to have Gaussian distribution of $\mathcal{N}(0, \Sigma_{II})$. The uncertainties of I is described by its variance matrix $\Sigma_{II} = diag(\sigma^2_u, \sigma^2_v, \sigma^2_d)$. The conversion from a state vector to its measurement values is done as follows:

$$I_t = F(S_t) + V_F \, , \tag{4.5.5}$$

where F is the measurement model, which back-projects the 3D position of a pedestrian into images space. In this project, the transformation is computed via triangulation for stereo based on the camera principle point (c_u, c_v), focal length f, and the base line $Base$ between two stereo cameras:

$$\begin{aligned} u &= c_u - X \frac{f}{Z} \\ v &= c_v - Y \frac{f}{Z} \\ d &= f \frac{Base}{Z} \end{aligned} \quad . \tag{4.5.6}$$

Let I_t be the position in image space of a detected object which is assigned to τ_j at (t). The state vector S_t is then updated as follows:

$$\begin{aligned} S_t^* &= S_t^+ + K(I_t - F(S_t^+)) \\ \Sigma_{SS,t}^* &= \Sigma_{SS,t}^+ - K J_F \Sigma_{SS,t}^+ \end{aligned} \quad , \tag{4.5.7}$$

where K is the Kalman gain matrix; J_F is Jacobian matrix of F w.r.t. the state parameters:

$$J_F = \begin{bmatrix} \frac{\partial u_F}{\partial X} & \frac{\partial u_F}{\partial Y} & \frac{\partial u_F}{\partial Z} & 0 & 0 \\ \frac{\partial v_F}{\partial X} & \frac{\partial v_F}{\partial Y} & \frac{\partial v_F}{\partial Z} & 0 & 0 \\ \frac{\partial d_F}{\partial X} & \frac{\partial d_F}{\partial Y} & \frac{\partial d_F}{\partial Z} & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{-f}{Z} & 0 & \frac{fX}{Z^2} & 0 & 0 \\ 0 & \frac{-f}{Z} & \frac{fY}{Z^2} & 0 & 0 \\ 0 & 0 & \frac{-fBase}{Z^2} & 0 & 0 \end{bmatrix} \quad . \tag{4.5.8}$$

In the Jacobian matrix J_F, only position variables of the state vector are taken into account and velocity variables v_x and v_z are ignored. This is because only positions can be derived by measurement, but the velocity comes from the temporal modelling.

4.6 Discussion

Based on detailed descriptions of the methodology and mathematics modelling of the suggested tracking approach, several advantages and weak-points of the tracker are envisioned and analysed in this section.

4.6.1 Probabilistic pedestrian tracking

In the proposed approach, the uncertainties of both, detection and localization are taken into account, which leads to results represented in a probabilistic way rather than an absolute one. Consequently, an application can include those uncertainties while making reactions to any observed targets with higher certainty. Several filter steps are implemented to refine the positions in object space so that the approach can be directly applied to different sensor types which can provide 3D and visual information such as a fusion of a lidar with a mono camera. Moreover, specific features related to pedestrians are exploited to reduce false positive detections and predict the uncertainties of measurements. This results in improving the accuracy of tracked trajectories in terms of geometry and Id consistency.

The data association is carried out hierarchically to obtain prior information about geometry changes of trajectories in two adjacent epochs. On top of that, including the relationships between pedestrians strengthens the geometry constraints among trajectories in terms of local structure constraints: assignments with low confidence are supported by those with high belief of correctness. This is expected to increase the association accuracy in difficult scenarios. Together with LSR,

appearance is a significant cue that affects assignment optimization results. Instead of directly feeding detected BBs to TriNet, pixels belonging to the background of a BB are filled with random values before the feature extraction step. In this way, the temporal visual similarity computation should be more robust against occlusions and background effects, which is expected to increase the Id accuracy of tracked targets. In contrast to existing trackers, which usually rely only on 2D position, the proposed approach employs both, 2D and 3D information to compare the geometry of trajectories. While the 2D geometry can help to distinguish pedestrians in image space even when their distance in 3D space is small, occlusions pose a major problem. On the other hand, the 3D position is not useful to distinguish targets that are close to each other because it is affected by the typically reduced resolution of point clouds. Hence, by combining both types of geometry comparisons in forms of 3D distances and IoU, the association weight can be more effectively computed and should lead to more accurate association.

The proposed TCD method allows observations with different belief confidences to be taken into account. This is done by considering the relationship between trajectories and detections in the way that detections create trajectories and trajectories can endorse the correctness of detections. This enables the trackers to better deal with noisy detection inputs without additional steps like non-maximum-suppression to retrieve the most number of TPs, but not FPs.

The velocity estimation employs the position information in different techniques including line regression and histogram calculations which can provide not only more accurate velocity values but also their correctness in the form of the posterior probability. The uncertainty of the velocity is an important cue for the tracker to decide how long a trajectory can stay in the inactive state. Moreover, it enables the suggested motion model in which the movement of a pedestrian can be corrected and predicted based on its neighbours to work more effectively. This is because only targets with well-estimated velocities can contribute to the model. Hence, the model should work robustly against the problem of incorrect trajectory generation and localization.

The prediction of trajectory state vectors affords the recovery of detections that are missed so that the trajectories are better completed and less fragmented. The prediction is carried out considering position and velocity accuracy and trajectory consistency so that the predicted positions should better correspond to the correct ones. Also, the predicted positions provide for a 3D geometry comparison between trajectories and detections so that correct assignments should be obtained.

The proposed tracking approach requires both the interior and exterior orientation parameters of the stereo cameras. This could be a restriction in some practical cases, but in general, these can be easily obtained. Since the platform is not restricted to be static, the relative pose of the cameras between epochs can be acquired either from additional sensors such as a global navigation satellite system (GNSS) or an inertial measurement unit (IMU). Alternatively, visual odometry can be applied to self-localize the sensor. This does not require additional devices but usually

suffers from drift. However, since our tracker works in an online way, it solely demands accurate relative translation and rotation of the camera in two consecutive epochs, which is not so difficult to achieve.

4.6.2 Assumptions

The tracking method is developed under the assumption that the ground is a plane, especially when in computing foot points of pedestrians in 3D space. Hence, violations of this hypothesis may lead to incorrect localization results. However, in practice, the stereo platform mounted on a vehicle is capable of delivering 3D information within a certain range, where the surface of the ground usually does not change dramatically. Also, the viewing direction of the camera system is assumed to be more or less parallel to the ground plane and the projection centre high enough to capture the full body of pedestrians. Since vehicles certainly move on a street, this assumption can be easily satisfied by appropriately placing the sensor on the vehicle.

The interesting targets tracked by the proposed tracker are considered to have limited sizes. This postulate supports the elimination of FP detections depending on their heights. At the same time, this restriction is employed to compute the uncertainty of depth values. Extreme cases, in which the size of pedestrians is different from the defined range, cannot be handled by the proposed tracker.

To calculate and correct the velocities of tracked targets, it is supposed that the pedestrians need time to accelerate or decelerate their movements. Additionally, pedestrians who move together for a long time are assumed to have similar movements. These expected behaviours hold for most cases. Yet, pedestrians can behave in an unforeseeable way and without additional information, sudden changes cannot be captured by information in the past solely. Certainly, this affects the association weight due to an incorrect geometry term and the recovery of missed detections can be unsuccessful. Nevertheless, the appearance can help to overcome the ill-judged geometry similarity so that the association optimization accuracy should be maintained. While FPs will increase owing to drifting prediction, the checking step whether a predicted BB contains an object assists in detecting these failures.

In the developed tracker, several free parameters that need to be determined. Some parameters are independently set without prior knowledge about the dataset, instead, they are based on characteristics related to pedestrians. The other groups of free parameters are determined using a small training dataset. Therefore, once the distribution of a testing dataset changes, those parameters also need to be tuned to achieve the best performance. While some parameters can be sensitive and have a high impact on the performance of the trackers, some are less important.

5 Experiments and Results

In this chapter, the proposed tracking approach is evaluated including the effectiveness of each component in the framework as well as the performance of the tracker w.r.t. the accuracy of localizing and tracking multiple pedestrians in 3D object space. The tracking results are compared to other state-of-the-art methods to better evaluate advantages and also disadvantages of the developed method. Besides assessing the quality of the introduced approach through tracking evaluation metrics, the sensitivity with respect to free parameters of the suggested methods is examined as well. The results contain hints to analyse the generalization ability of the tracking approach. The datasets and evaluation metrics used for experiments in this chapter are introduced in Section 5.1. The free parameter settings as well as how they can affect the tracker performance are illustrated in Section 5.2. An evaluation of the contribution of individual components of the proposed methods is reported in Section 5.3. The accuracy of localization is discussed in Section 5.4, followed by the comparison of the suggested tracking approach with other state-of-the-art methods in Section 5.5.

5.1 Data and evaluation metrics

Two principal components of an experiment are datasets and evaluation metrics. While the data provides means to check the capability of the proposed approach in the aspect of handling various situations and extreme cases, the evaluation metrics support to comprehend the effectiveness of the tracker in terms of solving the multi-pedestrians tracking problem and also enable the comparison of the developed tracker to other existing works. In this section, these two important elements are described.

5.1.1 Data

In the experiments, three datasets are employed: the KITTI benchmark (Geiger et al., 2012), the ETH mobile scene dataset (ETHMS) (Ess et al., 2008), and a dataset called multi-views (MuVi) (`https://doi.org/10.25835/0082741`) acquired by ourselves. These datasets contain various scenarios of pedestrians walking on streets, mostly in complicated down-town ar-

eas. Hence, they cover the movements of pedestrians with different behaviours, lots of mutual occlusions, and at diverse depths with respect to camera position and thus, the size of pedestrians appearing in images also varies a lot. During the experiments, the global coordinate system is defined as the position of the left camera in the first image frame of a sequence. The relative translation and rotation of the camera between two adjacent frames are derived from visual odometry (Geiger et al., 2011).

KITTI

The KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) object tracking benchmark is a dataset of the KITTI vision benchmark (Geiger et al., 2012). It provides RGB images captured by stereo cameras with a baseline of approximately 0.54 m mounted on a moving vehicle with a height above from the ground of around 1.6 m. Since the cameras are placed at a quite high position with respect to the ground and have a long baseline, they can capture scenes with a relatively large range of depths. With the purpose of investigating autonomous driving applications, images of KITTI are obtained on the street level with a frequency of 10 Hz, so that the movement of pedestrians can be recorded continuously.

The tracking benchmark has two separate datasets, one for training and one for testing. The training includes 21 sequences with labels for 8 object types, in which only 5 sequences contain pedestrians, namely sequences no. 13, 15, 16, 17, and 19. Ground truth (GT) is provided in both, 2D BB coordinates and 3D position relative to the left camera for each time step. Each pedestrian of a sequence is assigned to a unique ID. The level of occlusion and truncation are also annotated; these are cues to define the level of difficulty of those objects. The testing dataset has 29 sequences, but without GT, in which the evaluation is carried out by the KITTI team independently. Therefore, the test dataset can be solely used to evaluate the performance of the complete tracking framework. The assessment of each component of the tracker as well as parameter learning are investigated using the training dataset.

According to KITTI evaluation criteria, only pedestrians appearing in image space with the height of at least 25 px are counted as TPs. Detected objects belonging to neighbouring classes of pedestrians, such as sitting people or cyclists, are not considered as FPs. The occlusion of an object is specified by three levels of visibility: completely, partly, and hardly visible. If the pedestrians leave the image of scene, it is illustrated by the truncation level of 1; otherwise, it is 0. Based on how challenging it is to detect and track pedestrians due to their BB height, occlusion, and truncation, TP objects are classified into three types: easy, medium, and hard. Missed detections of hard objects are not included as false negatives (FNs).

ETH mobile scene

The ETH mobile scene dataset (ETH mobile scene (ETHMS)) (Ess et al., 2008) offers seven

Dataset	no. seqs.	no. seqs. containing peds.	no. of peds.	no. of ground truth	easy (%)	moderate (%)	hard (%)	avg. no. peds. per image	max. peds. per image
KITTI Training	21	5	143	10551	63.3	19.7	17.0	5.6	13
KITTI Testing	29	22	292	23731	-	-	-	-	-
ETHMS (Bahnhof and Sunnyday)	2	2	201	7273	-	-	-	5.4	18
MuVi (first view)	8	8	215	17604	48.4	37.3	14.3	12	23

Table 5.1: Statistics related to the KITTI, ETHMS, and MuVi dataset. Unknown values are specified by "-".

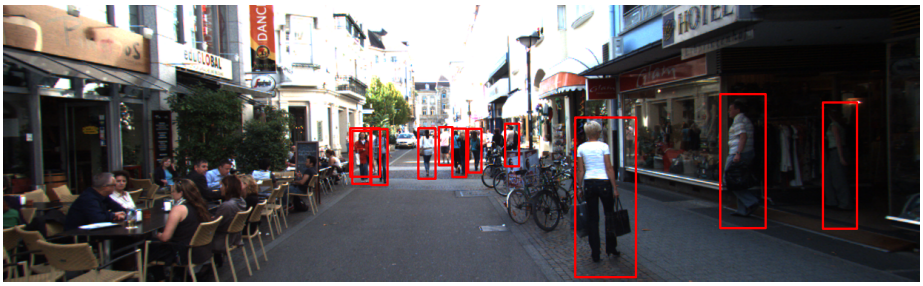
stereo videos containing images of pedestrians in down-town areas captured from mobile platforms with a frequency of 13–14 Hz. As the stereo system used to capture these sequences was mounted on a kid stroller, its height above the ground is small and thus a number of occlusions exists in the images. This dataset only provides detection annotations with 2D BB positions and sizes, Ids of pedestrians are not available. For the two sequences called Bahnhof and Sunnyday, which are widely employed to test algorithm performance in the literature, 2D tracking labels are provided by MOT15[1]. In total, these two sequences contain 9571 GT objects in 253 trajectories. The annotation from MOT15 has a flag named invalid for difficult objects and they can be ignored in evaluations, resulting in 7273 GT objects and 201 trajectories. All valid objects have a minimum height of 49 px in image space which is quite large compare to smallest object in KITTI with a minimum height of 25 px.

Multi-views (MuVi)

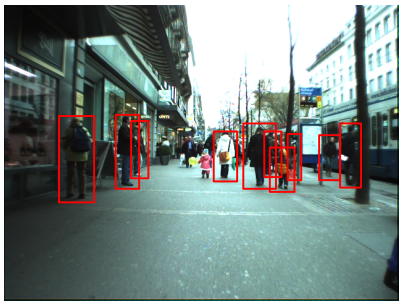
While there are several published benchmarks for 2D pedestrian tracking, only KITTI and ETHMS offer stereo images. Hence, to increase the diversity of pedestrian movements in crowded places for 3D pedestrian tracking, a measurement campaign was carried by the team of the research training group i.c.sens (Schön et al., 2018), in which the author took primary responsibility (`https://doi.org/10.25835/0082741`). Different from existing benchmarks, pedestrians are observed from various viewpoints by three stereo systems, the dataset is thus called Multi-views (MuVi). The MuVi dataset can be used to assess tracking approaches. It also facilitates future work related to collaborative 3D tracking by fusing the information from multiple viewpoints. As far as the author knows, there is currently no other publicly available tracking dataset for that purpose.

Besides capturing unknown people moving in the experimental area, a number of i.c.sens colleagues also took part in the measurement as pedestrians walking along pre-designed paths to

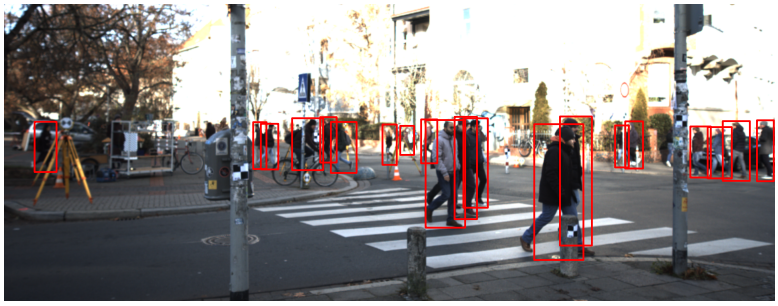
[1] `https://motchallenge.net/data/2D_MOT_2015/`



(a)



(b)



(c)

Figure 5.1: Example image with annotated GTs (red boxes) of the KITTI (a), ETHMS (b), and MuVi (c) dataset.

create complicated scenarios. In this way, not only the number of people appearing in images but also the complexity of the whole scenes was partly controlled. It can be seen from Table 5.1 that the average number of pedestrians appearing in an image is much higher than in both KITTI training and ETHMS dataset. Hence, the MuVi dataset contains images that are much more complicated than the KITTI and ETHMS ones. Besides the difficulties caused by crowded scenes, the challenges of MuVi also come from illumination conditions, and image blur, which can occur when capturing image sequences in the real-world. Consequently, MuVi can be employed to test and analyse the generalization capabilities of tracking approaches and to improve their robustness and accuracy.

Similar to KITTI, the MuVi dataset is captured at 10 Hz. There are approximately 1500 images in 8 sequences obtained from each viewpoint. The position of pedestrians is manually annotated in 2D images with BBs. Each GT detection is classified into five levels of visibility corresponding to the percentage of occlusion that a BB suffers including 0 %, 25 %, 50 %, 75 %, and 100 %, in which the occlusion of 0 % is considered as easy, 25 % and 50 % are moderate, 75 % and 100 % are hard. These complexity levels are not exactly the same as for KITTI, but similar. Each pedestrian is assigned to a distinctive identity (Id) number in a sequence and across camera perspectives. More detailed statistics of the KITTI and MuVi datasets are provided in Table 5.1.

5.1.2 Evaluation metrics

According to the KITTI benchmark evaluation criteria, a detection is considered as a TP if its IoU with the GT is equal or larger than 0.5, which is also applied to MuVi. Following the detection evaluation standard, detection results are reported based on the number of TPs, FPs, and FNs, which are used to compute two detection metrics, namely completeness and correctness (also called recall and precision, respectively):

$$\begin{aligned} recall/completeness &= \frac{TP}{TP+FN} \\ precision/correctness &= \frac{TP}{TP+FP} \end{aligned} . \tag{5.1.1}$$

The performance of a tracker is analysed using the CLEAR MOT metrics (Bernardin and Stiefelhagen, 2008) consisting of multi object tracking accuracy (MOTA) and multi object tracking precision (MOTP). MOTA represents the number of TPs, FPs, and Id switches (IDs) over all image frames n, computed as follows:

$$MOTA = 1 - \frac{\sum_n (FP + FN + IDs)}{\sum_n GT} . \tag{5.1.2}$$

The performance of different trackers are ranked based on this metric.

MOTP describes how well a tracker can localize TPs, which can be either computed in 2D image or 3D object space. While 2D-MOTP is the average IoU between TP detections and their GTs:

$$MOTP_{2D} = \frac{\sum_n IoU(B, B_{ref})}{n} , \tag{5.1.3}$$

3D-MOTP expresses the percentage of well-localized TPs, i.e. detections having 3D Euclidean distance $dist$ to their corresponding GTs smaller than a threshold $\varepsilon_{3D-MOTP}$:

$$MOTP_{3D} = \frac{\sum_n I(dist(S, S_{ref}), \varepsilon_{3D-MOTP})}{\sum_n TP} , \tag{5.1.4}$$

in which I is an indicator function with the value of 1 if $dist \geq \varepsilon_{3D-MOTP}$; otherwise, it is 0.

Moreover, the performance of a tracker is also assessed utilising four additional metrics suggested in (Li et al., 2009), namely the percentage of mostly tracked (MT) and mostly lost (ML) targets, fragmentation (FG), and the number of identity switches (IDs). Trajectory that covers at least 80 % of its GT are counted as MT, while MLs consist of less than 20 % of GT. IDs is the number of switched Ids between two trajectories plus the case where a new Id is assigned an existing trajectory. The continuity of generated trajectories is represented by the number of fragmentations FG.

Symbol	Description	Value	Unit
$[\varepsilon_{H_1}, \varepsilon_{H_2}]$	pedestrian height range	1.2–2.5	m
$[\varepsilon_{Br_1}, \varepsilon_{Br_2}]$	ratio range between BB height and width of a TP	1.5–5.0	-
$\varepsilon_{\mathcal{M}_s}$	percentage of pixels in the segmentation mask w.r.t. the BB of a TP	15	%
ε_u, ε_v	minimum of σ_u and σ_v, Equation (4.2.2)	2.0, 2.0	px
η_{σ_Z}	default uncertainty of Z when it is well estimated from 3D point clouds, Equation (4.2.6)	0.3	m
ε_Z	threshold to evaluate the accuracy of depth values calculated from 3D point clouds, Equation (4.2.6)	3	m
ρ, θ, ν	weights for different cues in association weight, Equation (4.3.1), Equation (4.3.12), Equation (4.3.14)	$0.45, 0.45, 0.1$	-
$\eta_{\mathcal{G}}$	constant to normalize geometry similarity value, Equation (4.3.3)	12	-
$\eta_{\mathcal{A}}$	constant to normalize appearance similarity value, Equation (4.3.4)	22	-
$\varepsilon_{3D-gate}$	threshold for the 3D association gate, Equation (4.3.5)	2	m
$\varepsilon_{2D-gate}$	threshold for the 2D association gate, Equation (4.3.5)	0.05	-
$\varepsilon_{\varrho_1}, \varepsilon_{\varrho_2}$	confidence (cfd) values used for tracking-confirm-detection	0.2, 0.85	-
$\eta_{\mathcal{F}}$	constant to calculate bad association for invalid trajectories, Equation (4.3.14)	1.3	-
ε_{an1}	association weight threshold to choose anchors	0.6	-
$\eta_{\mathcal{N}}$	constant to normalise the correctness of a state vector of a trajectory, Equation (4.4.3)	2.0	m
$\varepsilon_{\mathcal{N}}$	maximum number of inactivate states, Equation (4.4.3)	10	epoch

Table 5.2: Setting of free parameters of our tracking system, which are separated into three groups corresponding to three components of the tracker consisting of detection, association, and prediction.

5.2 Component optimization

As mentioned in Chapter 4, the performance of the proposed tracking approach is affected by a number of free parameters. In general, it is difficult to directly find optimum values for these parameters using gradient descent, since there is usually no available objective functions to link the error metrics of a specific setting with the GT of training data. Thus, to find values which yield a strong optimum, the direct search algorithm is applied. Specifically, each parameter is initialized with a range of values obtained from either the relevant physical facts it represents or from statistical training data. Various values in this range are then sampled and the results are compared using accuracy metrics and learning data. To keep the results tractable and easier to analyse, normally only one parameter is changed at a time, unless the parameters in the same equation are highly correlated. In this case, they are examined simultaneously. Moreover, to assess the impact of a parameter on the results as easily as possible, the simplest version of the related components is used. In this section, the setting of those parameters is described in detail according to the components of the framework that they are associated with, and their sensitivity is studied as well. This does not applied to ε_u, ε_v, and $\eta_{\mathcal{F}}$.

The free parameters are learned by employing sequence 13, 16, 17 of the KITTI tracking training dataset.

5.2.1 Detection and post-processing

The post-processing of detection to eliminate false alarms involves a number of parameters restricting the height in 3D object space $[\varepsilon_{H_1}, \varepsilon_{H_1}]$ and the BB ratio $[\varepsilon_{Br_1}, \varepsilon_{Br_2}]$ of pedestrians. In addition, the percentage of pixels belonging to the instance segmentation mask $\varepsilon_{\mathcal{M}_s}$ over all pixels of a BB is utilised to filter out bad quality detections. The height, BB ratio between the BB height and width, and the percentage of pixels in the segmentation mask of TP and FP detections are shown in Figure 5.2. By selecting thresholds related to these features, the number of FPs is reduced, while recall is still maintained. The priority is to select thresholds that cover most of TPs which are not easy to retrieve later during the tracking. On the other hand, the FPs not filtered in this step can still be recognized in later stages.

Based on the statistics in Figure 5.2 (a), the height values of TPs are concentrated in the range of 1.0–2.5 m, which also reflects the possible height of pedestrians in the real world. There are also some detected pedestrians with an unreasonable estimated height, i.e. larger than 2.5 m or less than 0.5 m, whose average detection confidence scores are approximately 0.5. Hence, those unrealistic heights are due to either incorrect 3D point clouds corresponding to those detections or a bad quality of the detection results. To encourage the proposed method to deal with pedestrians in the real world, instead of just overfitting the learning dataset, the height range threshold is set to 1.2–2.5 m (m) to generalize common cases in reality. This restriction helps to reduce 8.8 % of FPs, but at the same time, a small percentage (1.5 %) of TPs is lost. Certainly, there may be pedestrians whose heights are less than 1.2 m, however in this study, the most common situations are the focus, while other cases are counted as outliers.

Deriving from Figure 5.2 (b) and (c), a detection is only considered as a TP if its BB ratio is within the range of 1.5–5.0 and its mask ratio is larger than 15 %. Since the whole BB of a detection can contain only foreground pixels, there is no need to have an upper-bound for $\varepsilon_{\mathcal{M}_s}$, even though the results show that the mask ratio of TPs is lower than 0.9. While the BB ratio limitation allows recognizing 4.2 % of FPs, this value is 27.4 % in the case of the mask ratio. However, a number of TPs are also deleted due to these criteria, which are 0.4 % and 0.3 % for BB ratio and segmentation mask ratio thresholds, respectively. Combining all the aforementioned restrictions related to height, BB, and mask ratio, the total reduction of FPs is 36.7 %, which clearly dominates the loss of 2.2 % of TPs (note that these results are not necessarily independent of each other).

Localization uncertainty

In this experiment, free parameters required for estimating depth uncertainty are explored. As hypothesized in Equation (4.2.5), the position Z in the object space of an observation can be predicted from the camera parameters and its BB height Z_H^+ or width Z_W^+. This is confirmed by

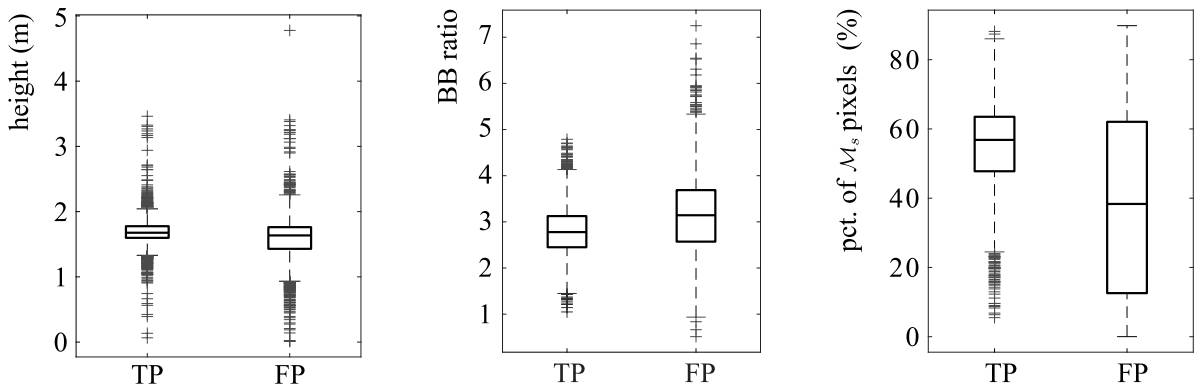


Figure 5.2: Estimated height (meter), the ratio between BB height and width, and the percentage of segmentation mask pixels in a detected BB for TP and FP detections delivered by mask R-CNN.

the experimental results in Figure 5.3 (a) and (b). By repeating the prediction of Z^+ with different values of height Ped_H and width Ped_W, it is revealed that the best-predicted depths that are closest most to their GTs are obtained with the assumption that pedestrians have an average height Ped_H of 1.7 m and width Ped_W of 0.55 m.

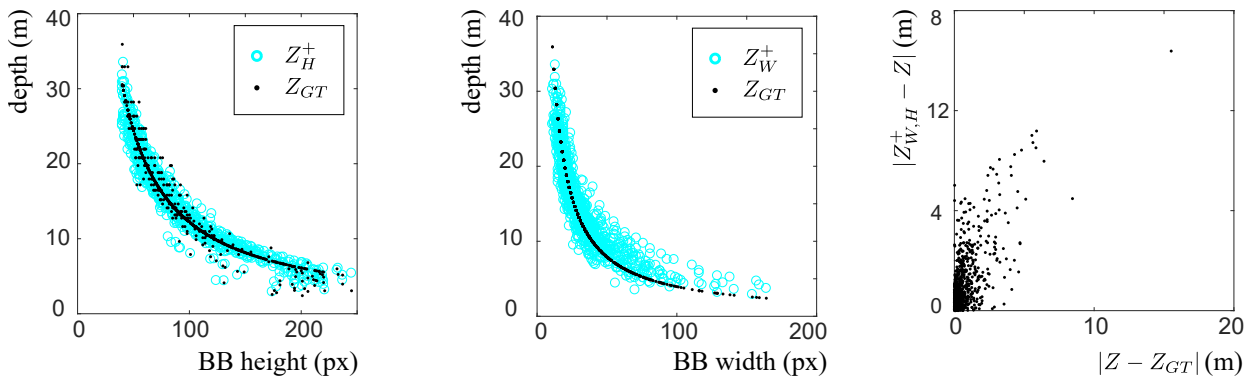
It is shown in Figure 5.3 (c) that once the absolute differences between predicted depths $Z^+_{W,H}$ and their corresponding values Z computed from the 3D point clouds are less than 3.0 m, the errors of Z, which are the distances to their GTs Z_{GT}, are also small with a mean value of 0.3 m. On the other hand, when $|Z^+_{W,H} - Z|$ becomes larger, the error $|Z - Z_{GT}|$ of Z increases with $|Z^+_{W,H} - Z|$. Based on these observations, the threshold ε_Z of 3.0 m is used to decide whether the depth estimated from the 3D point clouds is highly incorrect. The default uncertainty η_{σ_Z} of Z is fixed to 0.3 m in case Z is considered to be well estimated.

5.2.2 Data association

The quality of the proposed association approach depends on the capability to distinguish different persons based on weight values and the effectiveness of both, the association gates and the LSR step. In this section, all free parameters related to these methods are discussed.

Appearance model

The TriNet only delivers feature vectors for detected BBs, the classification result for an image pair whether it is positive (i.e. the image pair represents the same person) or negative (i.e. the image pair is captured from different persons) is assessed through the Euclidean distance between feature vectors and a distance threshold. It is shown in Figure 5.4 that the distribution of positive



(a) (b) (c)

Figure 5.3: The depth value Z of a detection calculated from the 3D point clouds and predicted depth inferred from either BB height Z_H^+ (a) or BB width Z_W^+ (b) are highly close. The difference between the depth computed from the point clouds and the predicted depth $|Z_{W,H}^+ - Z|$ corresponding to the depth error $|Z - Z_{GT}|$ (c).

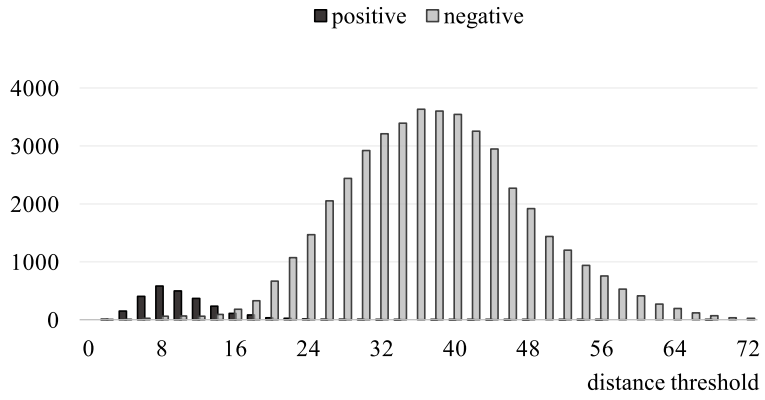


Figure 5.4: The distribution of Euclidean distance of feature vectors extracted by TriNet for positive and negative pairs.

and negative samples mostly overlap together in the distance range of 10–25. In this experiment, the Re-Id is evaluated for detections of two adjacent image frames.

To enhance the performance of TriNet, the random noise masks are employed to reduce the effects of occlusion and background. The comparison between the two versions of TriNet in terms of the precision metric is shown in Figure 5.5. In this experiment, the distance thresholds are selected within the range of $[10, 25]$. The performance of the network is evaluated based on the $F1$ score:

$$F1 = (2 \times \frac{precision \times recall}{precision + recall}) \,.$$

By using the masks, the average classification precision and F1 are slightly improved, with an average of 1.6 % and 0.9 %, respectively, while the recall nearly does not change. Although the increase rate is quite small, since the number of negative and positive samples is huge, these improvements actually represent large numbers of correct classifications that the mask version can

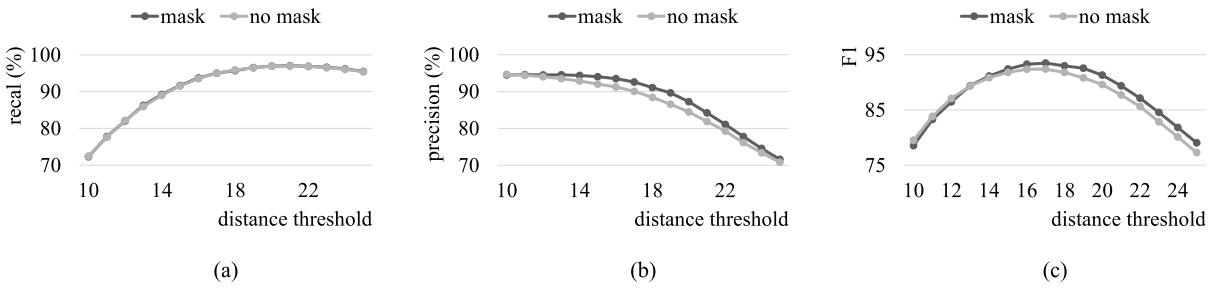


Figure 5.5: Person Re-Id results of TriNet with and without mask in terms of average classification recall (a), precision (b), and F1 (c) at different distance threshold.

achieve compared to the one without the mask, which can lead to the reduction of IDs.

The best F1 score is achieved at the distance threshold of 16. Hence, a pair of images having a distance larger than 16 can be considered as negative pair and should have appearance similarity $\mathcal{A}$ smaller 0.5. Based on these criteria, $\eta_{\mathcal{A}}$ is set to 22 (see Equation (4.3.4)).

Figure 5.6: Some exemplary positive pairs of BBs with mask (a), (b) and without the random noise mask (c), (d). The masks significantly assist to reduce the appearance distance between two images in one pair which belong to the same person.

Association weight

The association weight is constructed by three cues, namely geometry, appearance, and detection confidence. The confidence term ϱ does not represent any similarity between two detections but serves as a clue for an existing target to find its most suitable detections among those that are similar to it as described in the TCD method. Therefore, the parameter ν, which illustrates the impact of ϱ, should be lower than ρ and θ of the geometry and apparent cues. In this experiment, the value of ν is varied in the range of $[0.05, 0.35]$, while the impact of ρ and θ are thoroughly examined in the broad spectrum of $[0, 1 - \nu]$. Since $\rho + \theta + \nu = 1$, it is enough to inspect the variations of θ and ν. In this experiment, the simplest version of the association component is

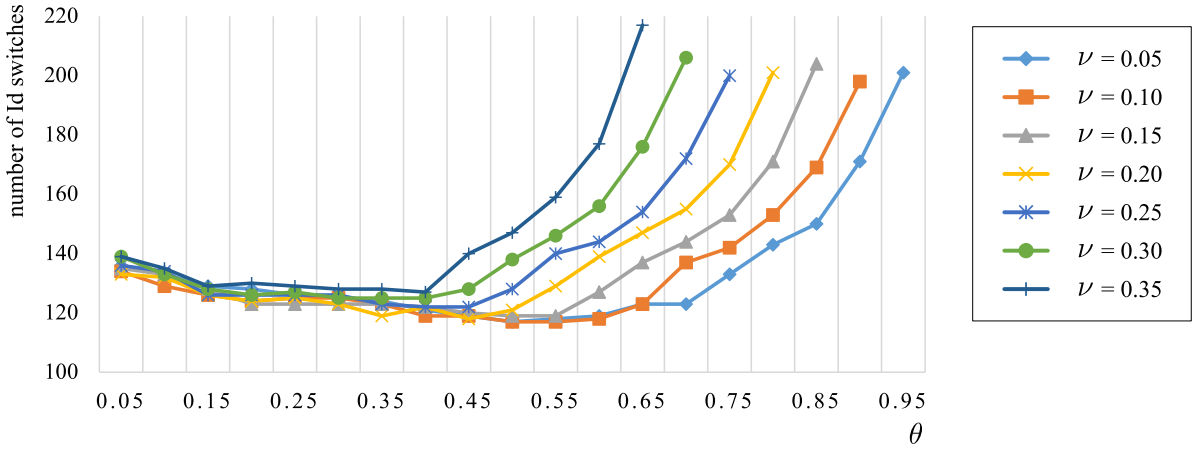


Figure 5.7: The number of IDs with respect to different values of ν and θ.

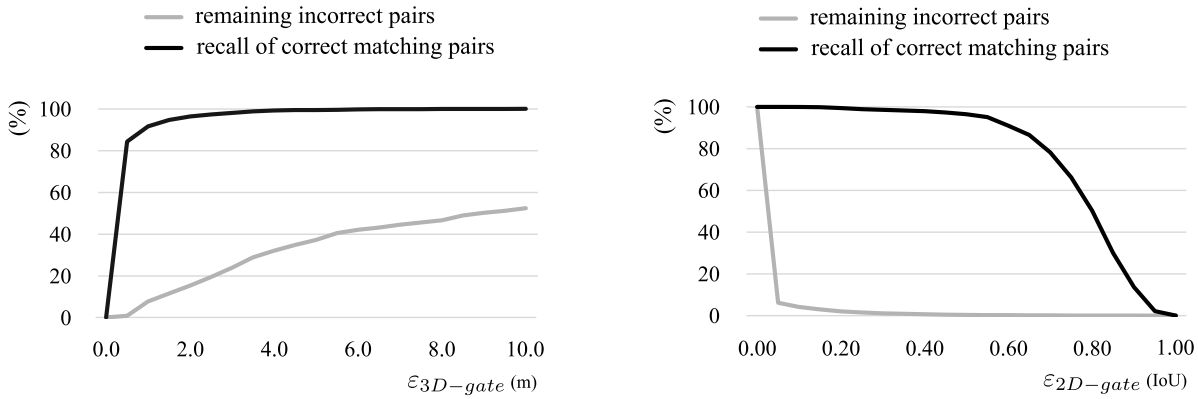
utilised, i.e. the association gates and LSR are omitted. This way allows to study the impacts of ν, ρ, and θ on the tracking results. The optimum of each setting is evaluated through the number of IDs, the lower this number, the more favourable the setting is.

The experiment results are shown in Figure 5.7. It is interesting that completely ignoring the geometry cue ($\rho = 0.0$) leads to much worse results than excluding appearance similarity ($\theta = 0.0$) from the association weight. The optimum result of IDs is achieved by combining both cues. The parameter ν also affects the accuracy of data association, in which the number of IDs increases together with the value of ρ, its impacts are especially evident when θ dominates ρ.The optimal result of IDs is achieved when ν falls in the range of $[0.05, 0.15]$ and the values of θ and ρ are similar. Specifically, the setting of $\nu = 0.1$, $\theta = 0.45$, and $\rho = 0.45$ is the most favourable according to the experimental results. While changing the parameters ρ, θ, and ν obviously leads to fluctuations in the accuracy of person Re-Id during tracking, it is noticeable that as long as θ and ρ lie in a certain range, the association results are more or less stable. However, when one of the terms, geometry or appearance, loses its contribution to the association weight, the number of IDs is very sensitive and can seriously change in response to the variation of free parameter settings.

Association gates

The association gate efficiency is defined by the number of incorrect matching pairs that can directly be eliminated without affinity calculations. The association gates involve the 3D distance $\varepsilon_{3D-gate}$ and the 2D IoU $\varepsilon_{2D-gate}$ threshold to recognize whether a tracking target and a detection in two adjacent epochs are likely to have the same Id. While reducing unreasonable associations, these gates must be able to recover most correct pairs, which are determined in the later assignment optimization step. The recall of positive association pairs at various threshold values of the 3D gate $\varepsilon_{3D-gate}$ and the 2D gate $\varepsilon_{2D-gate}$ are shown in Figure 5.8. In the case of the 3D gate, selecting a distance threshold of less than 2.0 m results in the loss of a lot of correct assignments. In contrast, once the threshold is larger than 5.0 m, the 3D gate becomes much less important as it filters out

less incorrect associations. The same phenomena occurs for the 2D gate. According to the results in Figure 5.8, the optimum IoU threshold should lie within the range of $[0.05, 0.3]$. These observed boundaries are then employed as searching ranges to find the best setting for the free parameters of the association gates.



(a) (b)

Figure 5.8: The recall of correct matching pairs and the percentage of incorrect pairs that cannot be filtered out by the 3D-gate (a) and 2D-gate (b) at different threshold settings.

Since the 3D and 2D gates work independently, their thresholds, thus, are determined separately in turn and the order of experiments does not affect the results. The number of IDs is used to assess the performance of the association gates at different threshold values. It is shown in Figure 5.9 that $\varepsilon_{3D-gate} = 2.0$ (m) yields the optimum results that helps to reduce approximately 30 % of IDs. Though a detection and a target belong to a same person cannot have a distance of 2.0 m, this value reflects the uncertainty of position estimation in 3D object space rather than the ideal situation in reality. The reduction is boosted to even higher level of 45 % when the 2D-gate with the threshold of either $\varepsilon_{2D-gate} = 0.05$ or $\varepsilon_{2D-gate} = 0.1$ is applied. Here, the lower value $\varepsilon_{2D-gate} = 0.05$ is adopted for the 2D-gate, as association gates are binary decisions. Hence, a more relaxed restriction is preferred over a tight one even if its performance is a little inferior. Though the Id accuracy significantly changes as $\varepsilon_{2D-gate}$ fluctuates around its determined optimum point, the number of IDs is not too sensitive to the value of $\varepsilon_{3D-gate}$. This means that while the association gates work effectively, their free parameters need to be learned and sampled carefully so that the gates are able to maximize the performances of the whole method.

Tracking-confirm-detection

The recall and precision of detection results using mask R-CNN on the KITTI training dataset are shown in Figure 5.10. While the recall gradually decreases, the precision sharply increases as the confidence (cfd) score rises. Because new trajectories should be initialized with high correctness, the threshold ε_{ϱ_2} is inspected within the bounds of $[0.7, 0.95]$. In contrast, ε_{ϱ_1} is set to a small value within the range of $[0.0, 0.5]$ so that most TP detections are considered during the association

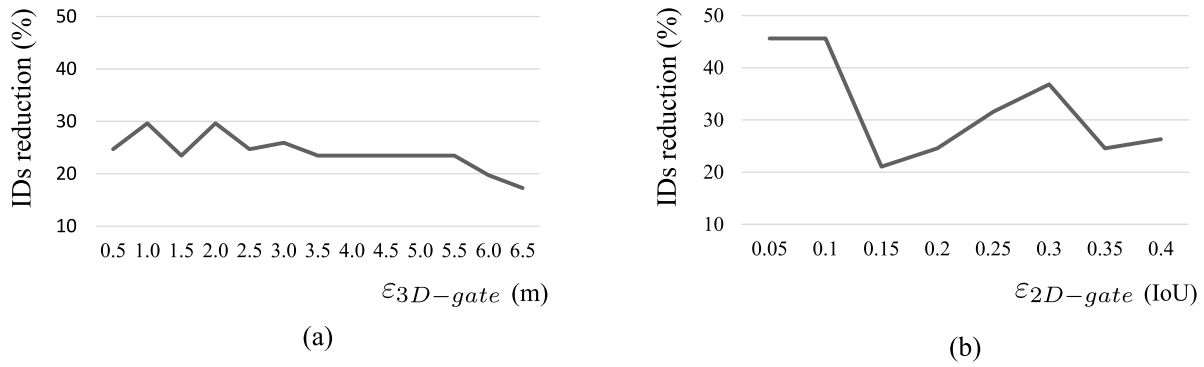


Figure 5.9: The reduction of IDs as the 3D-gate is adopted with various threshold values (a). On the top of using the best 3D-gate, the 2D-gates are also applied (b).

optimization. The MOTA results are used as the criteria for assessing the validity of parameter configurations. In addition, MT and ML are also computed for reference. Since the impact of $\varepsilon_{\varrho 2}$ on the tracking accuracy is not affected by the value of $\varepsilon_{\varrho 1}$, it is first sampled with $\varepsilon_{\varrho 1} = 0.0$ and $\varepsilon_{\varrho 1}$ is determined afterwards. The reason for this is that the correctness of existing trajectories is significant to support TP detections with low cfd scores during the association.

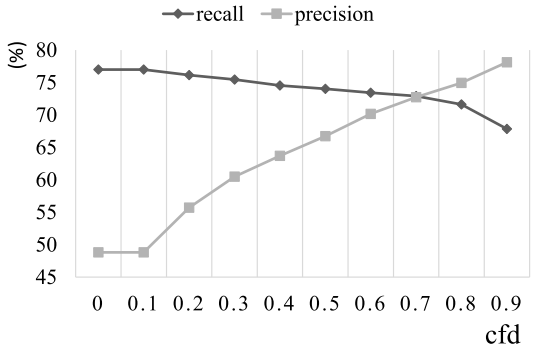


Figure 5.10: The detection results in terms of recall and precision at various cfd values.

The experimental results in Figure 5.11 (a) show that MOTA gradually increases together with the value of $\varepsilon_{\varrho 2}$. This is because when $\varepsilon_{\varrho 2}$ is becoming larger, less FPs are utilised to create trajectories, while the number of TPs only drops slightly. However, for $\varepsilon_{\varrho 2}$ larger than 0.85, the decline of TPs starts to dominate the improvement of FPs. This results in a slight decrease of MOTA and a significant reduction of MT. While the value of $\varepsilon_{\varrho 2}$ clearly shows an influence on MOTA and MT, $\varepsilon_{\varrho 1}$ has insignificant effects on these tracking metrics as long as it remains small enough (see Figure 5.11 (b)). The setting $\varepsilon_{\varrho 2} = 0.85$ and $\varepsilon_{\varrho 2} = 0.2$ were determined to yield the best results in terms of both MOTA and recall.

Anchor selection

Anchor pairs provide important prior knowledge for enhancing association results of difficult scenarios. However, incorrect anchors, offering untruthful information, can damage the perfor-

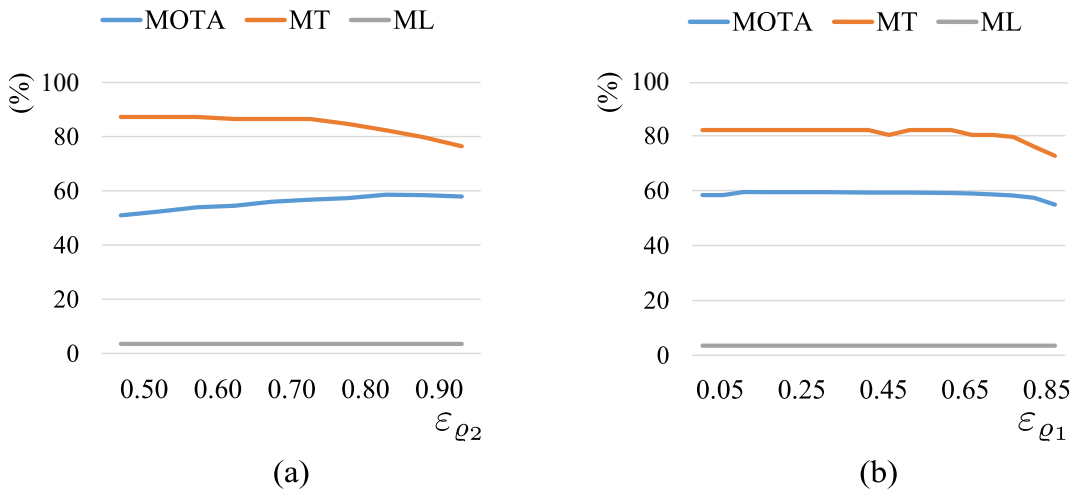


Figure 5.11: The tracking results including MOTA, MT, and ML according to various thresholds of $\varepsilon_{\varrho 2}$ (a) and $\varepsilon_{\varrho 1}$ (b).

mance of the association stage. In this experiment, two issues are evaluated, namely free parameter setting for anchor determination and the consequences of wrong anchors selections on the tracking consistency in terms of IDs. There are two ways to select anchors from the first step of the assignment optimization. In the first way, they are selected from matching pairs with association weights larger than a threshold ε_{an1}. The result is illustrated in Figure 5.12. The percentage of epochs having at least one anchor drops dramatically as soon as ε_{an} is larger than 0.8. The percentage of incorrect anchors, representing how much the prior knowledge is untruthful, is almost unchanged with a value of 24 % and becomes smaller once ε_{an1} rises above 0.55. In the second way, to ensure that there is always at least one anchor at each epoch, a percentage ε_{an2} of association pairs having the highest weights are chosen as anchors. Similar to the first experiment, the percentage of incorrect anchors is more or less stable. The reason for this is that the association gates already filtered out a lot of unreasonable matching pairs and the rest can be partly recognized with extreme restrictions such as ε_{an1} larger than 0.6 or ε_{an2} smaller than 20 %. Since both approaches used to choose anchors yield similar results, it is sufficient to investigate the sensitivity of the local structure refinement (LSR) method using the threshold ε_{an1} to determine anchors.

The impact of selected anchors on the accuracy of the data association stage can be seen in Figure 5.13. Using a small ε_{an1}, a lot of incorrect matchings are considered as anchors, the LSR, therefore, only affects inactive trajectories which are always considered in the LSR step. This helps to decrease the number of IDs by about 11 %. Interestingly, at the point that ε_{an1} reaches the value of 0.5, the number of IDs suddenly increases again. Setting $\varepsilon_{an1} = 0.6$ yields the best enhancement, where the IDs are reduced by 37 %, and thus this value is employed to select anchors in the first stage of the association. When ε_{an1} becomes larger than 0.6, the IDs reduction decreases to 33 %. These fluctuations can be explained as follows: when the threshold changes, many trajectories with incorrect associations in the first step cannot find any nearby anchor to use

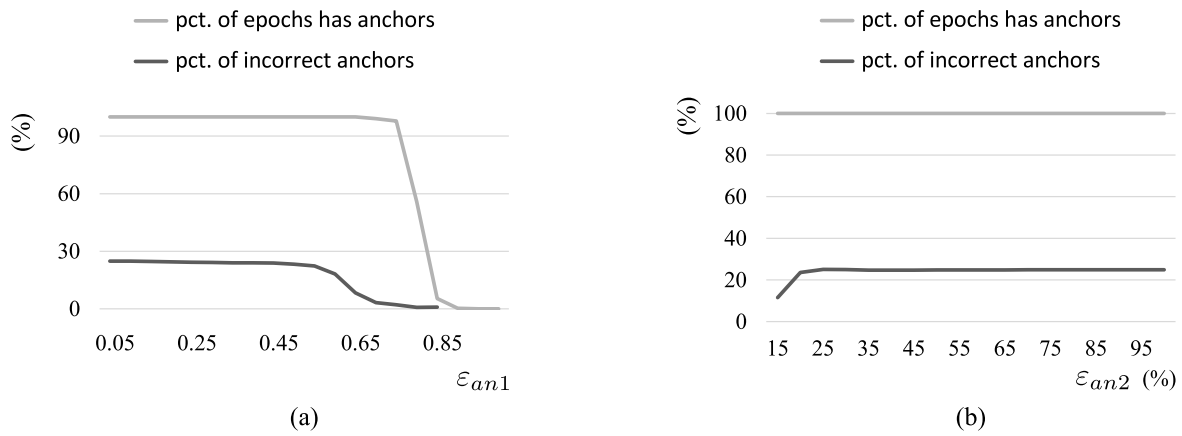


Figure 5.12: The percentage of epochs having at least one anchor and the percentage of incorrect anchors at various thresholds of ε_{an1} (a) and ε_{an2} (b).

for the correction in the LSR step. In general, it is clear that the benefit of the LSR depends on the quality of selected anchors, which are controlled by ε_{an1}. Nevertheless, while selecting only anchors that can satisfy the severe restriction posed by ε_{an1} helps to eliminate errors, it does not allow to maximize the performance of the LSR step. The oscillation of ε_{an1} does not lead to a dramatic difference in the efficiency of the LSR step if ε_{an1} is large enough.

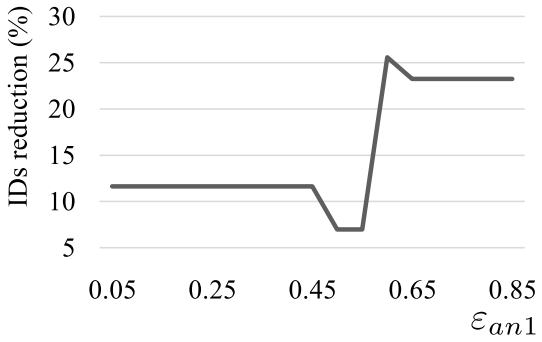


Figure 5.13: The accuracy of the data association stage measuring by the number of IDs according to different values of ε_{an1}.

5.2.3 Missed detections recovery

Prediction allows the restoration of detections that mask R-CNN failed to detect by employing temporal information about movement of pedestrian. This stage is only beneficial for the proposed tracker if the number of recovered TPs is higher than the number of false alarms. In order to maximize the advantage of this component, its free parameters, namely $\varepsilon_{\mathcal{N}}$ and $\eta_{\mathcal{N}}$, involved in specifying the number of epochs $\mathcal{P}$ that a trajectory can be elongated by prediction, need to be determined. $\varepsilon_{\mathcal{N}}$ defines the maximum number of epochs the prediction can happen. Since pedestrians may change their movement intention suddenly, $\varepsilon_{\mathcal{N}}$ should not be too large. In this experiment, $\varepsilon_{\mathcal{N}}$

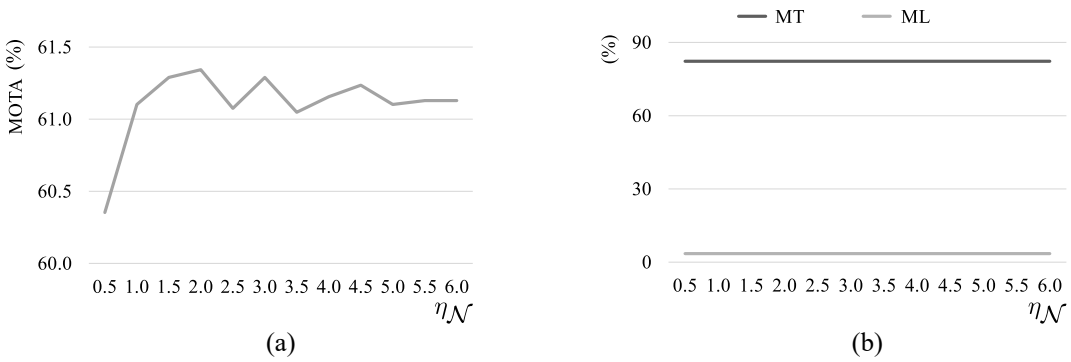


Figure 5.14: MOTA (a), MT, and ML (b) of tracking results w.r.t. $\eta_{\mathcal{N}}$.

is set to 10 epochs corresponding to 1 s. This is usually the maximum time that a trajectory can be well extended through prediction. $\eta_{\mathcal{N}}$ is used to convert the correctness of a state vector into the range value of 0.0–1.0. A large value of $\eta_{\mathcal{N}}$ allows trajectories to be extended longer, which can lead to more drifts. On the other hand, selecting $\eta_{\mathcal{N}}$ to small value, trajectories are predicted for only a short time and thus many missed detections cannot be retrieved. In this experiment, $\eta_{\mathcal{N}}$ is increased until the number of FPs caused by prediction surpasses the recall.

As the prediction more or less only affects the metrics MOTA, MT, and ML, these are employed to determine the optimum values for $\eta_{\mathcal{N}}$; $\eta_{\mathcal{N}} = 2.0$ (m) yields the best results (see Figure 5.14). As $\eta_{\mathcal{N}}$ becomes larger, the percentage of recall that the prediction can retrieve also increases. However, after a certain point (2.0 m), the increase in recall begins to slow down because almost detections of existing targets are usually lost for solely a short time. In contrast, false alarm rises together with $\eta_{\mathcal{N}}$ and gradually dominates the recall. Consequently, MOTA is improved first and then drops. The effect of prediction on MT and ML is not clear in this experiment even though recall is apparently boosted. This is because the temporal information of ML trajectories are usually not well estimated and therefore, their missed detections cannot be recovered well.

It is illustrated in Figure 5.15 that when $\mathcal{P}$ is restricted by a small value of $\eta_{\mathcal{N}}$, the percentage of easy and moderate detections recovered by the prediction are quite far from the hard object type. As $\eta_{\mathcal{N}}$ is more relaxed, the recalls obtained from the prediction at different detection difficulty levels are on par with each other. This implies that the quality of a trajectory in terms of position precision is highly correlated with the difficulty of its missed detections, which also explains the improvement of recall and the non-alteration in ML.

5.3 Component evaluation

In the previous sections, a number of experiments has been presented to determine the optimum values for free parameters based on the training dataset. In addition, the performance of individ-

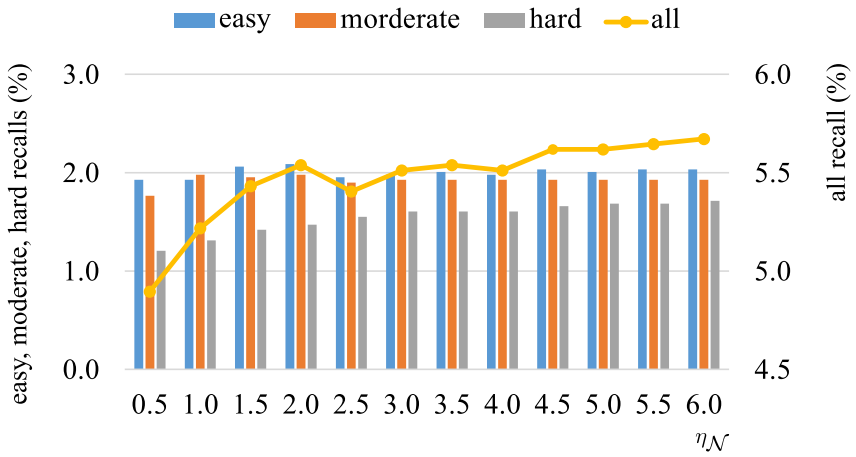


Figure 5.15: Recall of easy, moderate, and hard object types that are recovered by the prediction w.r.t. $\eta_{\mathcal{N}}$.

ual components were independently analysed corresponding to the goals that it was designed for. Thus, it is important now to clarify the influences of the proposed methods on the whole tracking framework in this section. The impact of a component is assessed by the evaluation of all tracking metrics described in Section 5.1.2 after it is omitted. Finally, the difficult cases that even the best model fails to handle are analysed.

Six variants of the proposed model are investigated. The version (a), i.e. *full model*, is used as a standard to evaluate the others. The post-processing step of the detection stage, responsible for eliminating FPs, is not used in the variant (b), i.e. *no post-processing*. This makes the detection input for the association phase much noisier. Hence, it poses more challenges for the association and prediction stages. Methods related to the association stage, consisting of tracking-confirm-detection (*TCD*), *association gate*, and local structure refinement (*LSR*) are omitted in versions (c), (d), and (f), respectively. As TCD is carried out by combining two different thresholds, in version (c), only a single threshold is employed to select the detection input for the tracker. For this, the version (c1), i.e. *no TCD-low*, uses $\varepsilon_{\varrho 1}$ while the version (c2), i.e. *no TCD-high*, utilises $\varepsilon_{\varrho 2}$. These variants reveal how these components affect association accuracy (i.e. IDs) and overall tracking results. In the last experiment (f), i.e. *no prediction*, missed detections are not retrieved. This means that no TPs or FPs are created by the prediction step, yet tracked targets still have their inactive and invalid states.

According to the KITTI evaluation criteria, a detection is counted as a TP if its IoU with its GT is larger than 0.5. However, with IoU lower than 0.5, interesting objects can still be localized and tracked well in 3D object space. Thus, in this experiment, several thresholds of IoU are used to determine correct detections and the comparison between the models is carried out by observing the changes at different values of IoU. Results of the different experiments are presented

IoU	Model	Recall (%)	FP (%)	MT (%)	ML (%)	IDs	FG	MOTA	2D-MOTP
0.5	(a) full model	73.1	17.6	50.7	13.6	43	316	55.9	70.8
	(b) no post-processing	74.0	19.2	52.3	13.6	46	314	55.0	70.8
	(c1) no TCD-low	74.7	41.3	55.5	13.6	84	322	29.1	70.8
	(c2) no TCD-high	71.0	17.2	45.3	18.2	38	326	55.85	70.6
	(d) no association gates	73.4	19.4	46.9	13.6	102	264	53.2	71.0
	(e) no LSR	73.4	19.0	49.9	13.6	53	346	53.6	70.6
	(f) no prediction	69.3	16.6	32.2	18.2	35	382	52.6	71.1

Table 5.3: Tracking results of all models on the KITTI training dataset.

in Table 5.3 and Figure 5.16.

Similar to the results in Section 5.2.1, it is again confirmed that without post-processing, the percentage of both FPs and recall increases compared to the *full model*. Thus, the number of MT is also slightly higher when the IoU threshold is less than 0.5. However, since the increase of false alarms is larger than the improvement of recall, MOTA is reduced by approximately 1.0 %. In addition, TPs that are not eliminated in the post-processing step usually have bad quality, which brings more confusion to the association step due to ambiguities coming from either geometry or appearance. Consequently, the identity accuracy of trajectories generated by this tracking model is impaired compared to the full one. The differences between the *no post-processing* tracking model and the full framework are consistent for all examined IoU.

By using all detections even with high uncertainty of correctness, the version (c1), i.e. *no TCD-low*, can achieve the best recall compared to all other models regardless of IoU threshold values. The average differences over all IoU values w.r.t. the full model is 1.5 % of TPs and 2.5 % of MTs. Nevertheless, this tracker version also has to deal with a lot more FPs, causing a big drop in both MOTA which is less than the *full model* by 27 % on average. In terms of IDs, this model has a huge gap to the performance of the *full model*, IDs are higher by 95 %. In contrast to *no TCD-low*, the model (c2), i.e. *no TCD-high*, only supports detections with high cfd scores. Thus, both recall and incorrect detections are cut down with a noticeable percentage, which leads to worse results for MT. With respect to the *full model*, the percentage of MT is reduced by around 6 %. The data association stage of variant (c2) performs better not only because it must cope with less noisy input detections but also because there are fewer trajectories that it can track.

Excluding the association gates component leads to a significant increase in IDs, which is more than 140 % the corresponding value when it is employed. Applying this ((d) *no association gates*) tracking variant, a target can be more often assigned to an observation with incorrect Id, which

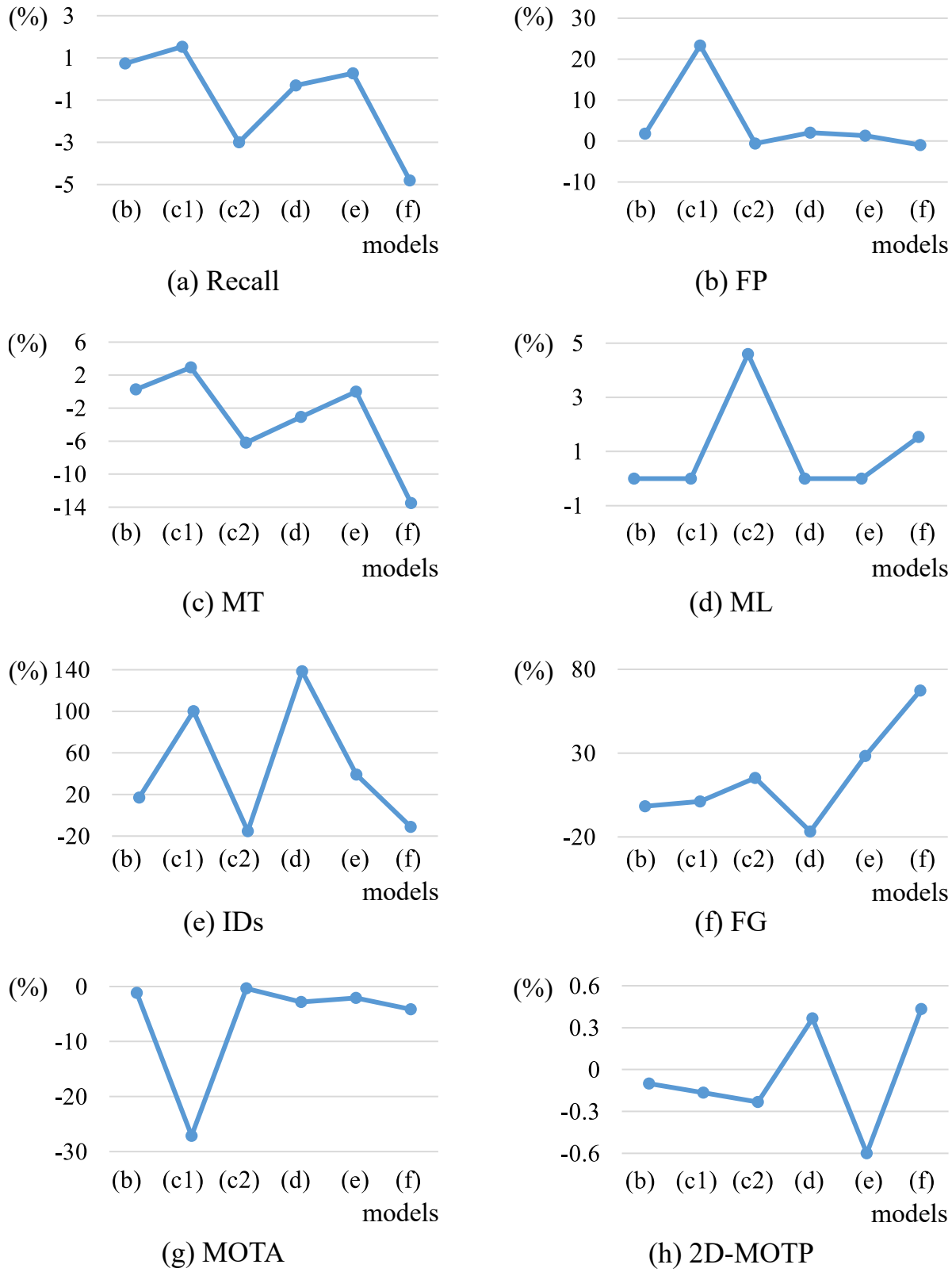


Figure 5.16: The average tracking results (IoU values are 0.3, 0.4, and 0.5) of all models: (b) no post-processing, (c1) no TCD-low, (c2) no TCD-high, (d) no association gates, (e) no LSR, (f) no prediction, in comparison with the variant (a) full model for different tracking metrics.

results in not only less fragmentation but also a slight increase in recall, FP, MT, and 2D-MOTP. While the association gates are proposed to filter out incorrect matching pairs to improve the accuracy of trajectory identity, it is clear that they also help to enhance MOTA a little bit.

The LSR component is proposed with the purpose of improving the Id accuracy for targets that often suffer from occlusions due to its neighbour. As this component is designed for special crowded scenarios, its impacts on the tracker are not as compelling as the association gates, but the reduction of IDs when the method is included in the tracking framework is obvious with an average of 28 % for all examined IoU values. Similar to association gates, the LSR also enables the elimination of false alarms which leads to the improvements of both MOTA and 2D-MOTP.

While the retrieval of missed detection is not applied in the model (f), recall, MOTA, MT, and fragmentation become worse. In comparison to the full variant tracker, this model looses an average 13.7 % MT which results in smaller number of IDs. In contrast, without prediction, no false alarms are generated due to drift and therefore, FP is improved compared to the full model. However, this does not lead to better performance in term of MOTA due to the diminishing number of TP.

In summary, by omitting each suggested method from the full framework in turn, their influences on the performance of the proposed tracker are clearly revealed. While a particular model in Table 5.3 may have better results on some criteria than the *full model*, it does not really perform better but is biased towards a specific metric such as recall or IDs, worse results are obtained for the others. Considering MOTA as a metric to rank the performance of a tracker, the full framework achieves the best results among all variants. Moreover, examining the value of one metric in relevance with all other metrics, the *full model* offers the best balanced tracking results.

Challenge issues

Though the full model has been proven to be the most promising tracker model compared to the other variants, it still exposes some limitations. In this part, problems of the proposed tracker including FPs, missed detections, and IDs are inspected.

It is illustrated in Figure 5.17 (a) that more than 75 % of false alarms directly come from detection results and 25 % is due to the prediction. If $IoU = 0.5$ is used, more than 48 % of these FPs actually overlap with at least one FN. This means they are not truly false alarms but rather detections that cannot cover the whole appearance in image space of desired objects well. Thus, depending on the selected IoU threshold, the number of FPs can also vary significantly as illustrated in Table 5.3. This at the same time allows the increase of recall. Nevertheless, there are still nearly 50 % of false alarms that do not cover any GT, which should be eliminated by using better detection methods or FP eliminate mechanisms.

Regardless of the fact that the prediction step allows recovering a number of missed detections,

the FN rate still needs to be further improved. The histogram of FNs w.r.t. their depth to the camera in 3D object space are shown in Figure 5.17 (b). Missed detections happen at every depth range, however, concentrating mostly at distances of 5–30 m, because this is the range that most pedestrians appear. It can be observed from the experimental results that the percentage of FNs that are occluded or truncated in each depth range increases as their distances to the camera decreases. This well explains the problem that these missed objects are hard to handle because they are too close to the camera and only partly captured in images. Approximately 40 % of FNs happens after their corresponding trajectories are already tracked. Hence, there is a chance to recover them through the prediction step, yet a crucial challenge remains, because pedestrians can change their behaviours suddenly.

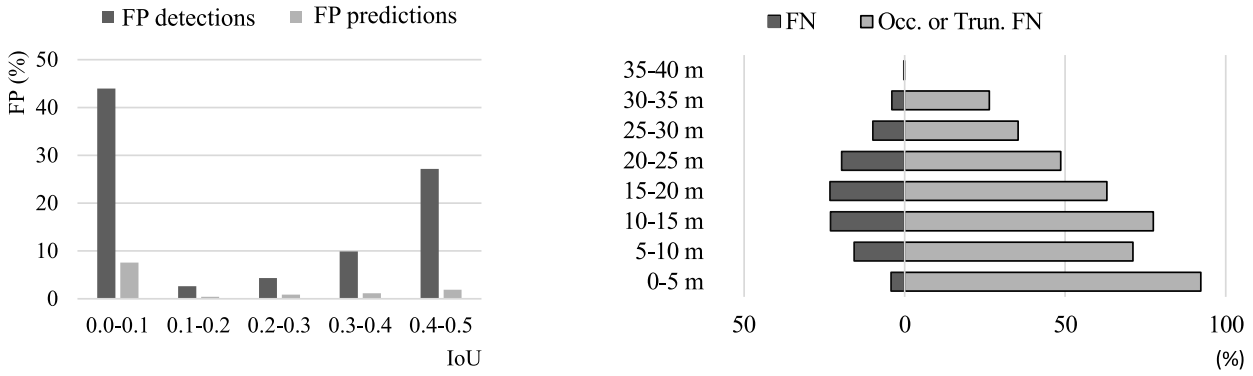
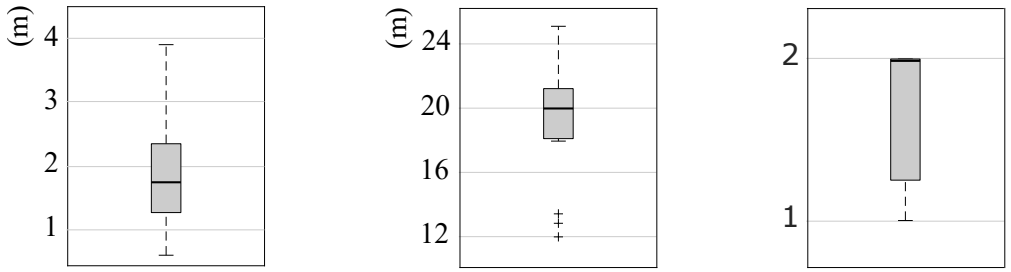


Figure 5.17: Histogram of FPs caused by detection inputs and prediction drifting at different IoU intervals smaller than 0.5 (a). Histogram of FNs with IoU smaller than 0.5 at various distances, whose corresponding percentage of either occlusion or truncation level is also reported (b).

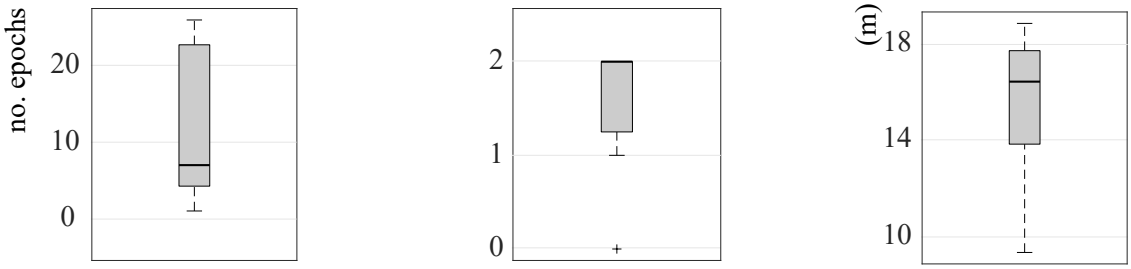
By applying the suggested association gates and the LSR method, the Id accuracy of generated trajectories is greatly enhanced. Nevertheless, a number of challenging situations still causes IDs in the current association approach. The IDs can happen either due to exchange of Ids between targets, accounting for more than 75% of the errors, or a target is assigned to a new Id. It is shown in Figure 5.18 that two targets swapping their Id have quite small spatial distance which is 1.8 m on average. Since these targets are usually further away from the camera with an average of 20 m in depth, their appearances in image space are small and hard to be distinguished by visual features. In addition, there is usually at least one of the two targets which is not clearly visible in image space due to occlusions. All aforementioned difficulties make both of geometry and appearance cues become ambiguous and inaccurate, which then can result in IDs. In another circumstance, a trajectory can be assigned to a new Id, which is usually the consequence of being inactive, on average for 5 epochs, as illustrated in Figure 5.19 (a). Falling into inactive state prevents a trajectory from updating its positions and appearance precisely due to lack of observations. Thus, the trajectory is very hard to be re-matched again with its corresponding detections appearing

later, regardless of how far the target is from camera (see Figure 5.19 (c)). Furthermore, as the inactivation of a target is commonly caused by occlusions, its observations arriving in latter epochs may suffer from this problem with average occlusion level of 1.9 (see Figure 5.19 (b)), which can add more uncertainties to their 3D position and visual features.



(a) dist between targets (b) dist to camera (c) occlusion

Figure 5.18: Statistics of target pairs that exchange their Ids w.r.t. their spatial distance (a), the maximum occlusion or truncation level of two targets in a pair (b), and their average distance to the camera (c).



(a) no. of deactivated states (b) occlusion level of observations (c) dist to camera

Figure 5.19: Statistics of targets that are assigned to new Ids due to unsuccessful Re-Id w.r.t. the number of inactive epochs that the targets fall into (a), the occlusion level of observations (b), the distance from targets to the camera (c).

5.4 Localization accuracy in 3D object space

As mentioned before, for many applications, it is significant to estimate the positions of tracked objects in 3D space precisely. Therefore, evaluating whether a detection is a TP or a FP solely based on the IoU does not reflect the actual capability of the proposed tracker. In this section, the localization accuracy of tracked trajectories in 3D object space is exhaustively examined. Unfortunately, due to the lack of GT for 3D position, only the training dataset of the KITTI tracking benchmark is employed in the experiments. TPs are supposed to be detections having IoU with their corresponding GT larger than 0.0 so that detections that are not well localized in image space

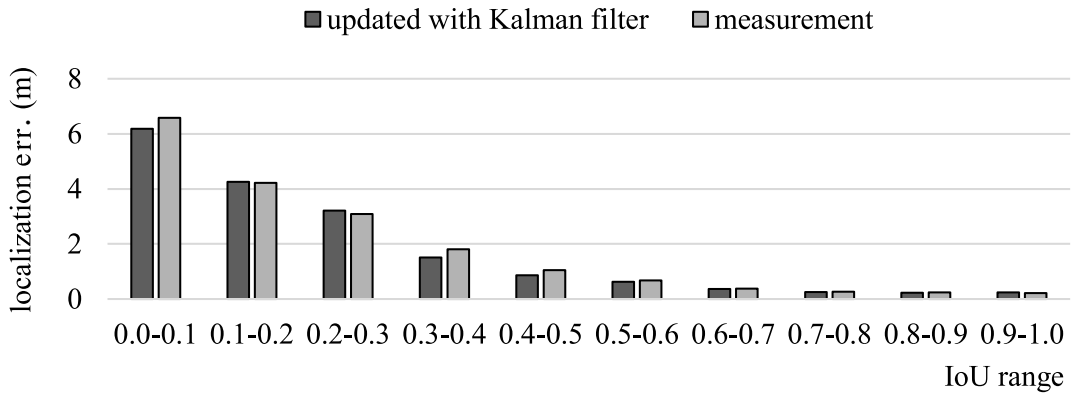


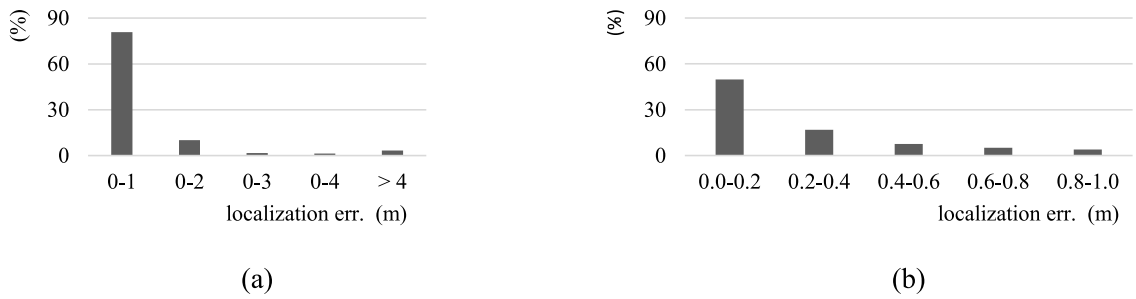
Figure 5.20: The average of localization error at various range of IoU when the 3D position is computed directly from measurements (i.e. detections) and when the Kalman filter is applied to smoothness tracked trajectories.

are also examined. This allows to inspect the localization accuracy of all detections and important factors affecting this issue can be observed.

The average errors of 3D positioning calculated directly from measurements (i.e. detections) and smoothed using the Kalman filter is shown in Figure 5.20. In both cases, either with or without the Kalman filter, the localization error significantly decreases from approximately 6 m to 0.3 m when the IoU increases from 0.0 to 0.6 and then remains nearly stable. The impact of the Kalman filter on the accuracy of 3D position is largest when the IoU is in the range of 0.0–0.1 (bad quality detections), in which the localization accuracy is improved by approximately 0.4 m. On the other hand, as the IoU gets larger (> 0.5), the effect of the Kalman filter on the smoothness of 3D trajectories is reduced. Nevertheless, the advantage of the Kalman filter on improving the accuracy of 3D positioning is obvious, which also verifies the validity of the proposed motion model.

To evaluate the ability of the proposed tracker on localizing pedestrians in 3D object space, the 3D positions updated by the Kalman filter are used in the next experiments. The histogram of localization errors in Figure 5.21 (a) shows that more than 80 % of recalls are localized within 1 m from their reference and approximately 10 % has positioning errors within the range of 1.0–2.0 m. At a closer look, a percentage of 49.2 % of the observed objects are well-localized with the spatial distance to their GT less than 0.2 m (see Figure 5.21 (b)). As the localization error increases from 0.2 m to 1.0 m, the recall gradually decreases from 16.2 % to 3.2 %.

It can be seen in Figure 5.22 that as the IoU increases, the accuracy of estimated positions of interesting objects is enhanced as well. When the IoU with GT is less than 0.3, more than 45 % of recalls are localized more than 4 m away from their reference and less than 15 % are positioned within an error of 1 m. Once the appearances of pedestrians are well-retrieved in images with IoU ≥ 0.3, their positions in 3D object space are computed with higher accuracy, more than 40 % have positioning errors within 1 m and 70 % have errors not larger than 2 m. Generally, as



(a) (b)

Figure 5.21: The histogram of localization error at coarse (a) and fine (b) scales.

the IoU rises, the percentage of recalls that is well localized also increases. The distribution of the localization errors clearly changes from the majority of inaccurate estimations to most of the computed positioning errors being less than 2 m at the point that IoU $= 0.3$. Hence, depending on positioning accuracy requirement, different IoU thresholds can be applied to determined TPs, it is not necessary to always choose $IoU = 0.5$ as in the KITTI benchmark.

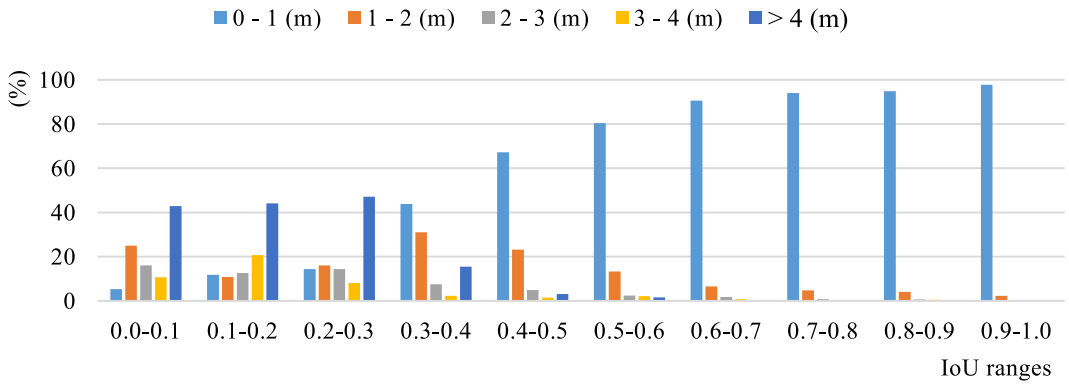


Figure 5.22: The histogram of localization errors w.r.t. various IoU ranges between detected objects and their GTs.

The correlation between the distances of interesting objects to the camera in depth direction and their positioning accuracy are illustrated in Figure 5.23. For objects appearing no more than 20 m from the camera, positions are usually estimated within a localization error of 1 m. The most promising area to obtain well-estimated 3D geometry for desired objects is in the depth range of 5–30 m. When pedestrians are too close to the sensor, part of their body may not be well captured which leads to a lack of information to obtain correct disparity and thus, their 3D geometry is incorrectly estimated. On the other hand, the inaccuracy of positioning clearly increases with depths because a same disparity error results in a larger error of 3D points as depth increases. This is well explained through the error propagation of the stereo triangulation formula. Moreover, since the size of pedestrians in image space linearly decreases with depth, from a certain distance onwards, it is difficult to obtain their BBs with high IoU, which also damages the accuracy of their 3D position as shown in Figure 5.22.

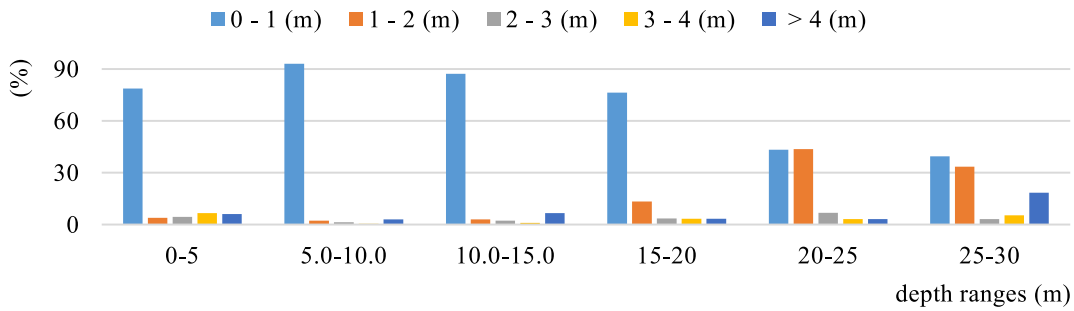


Figure 5.23: The histogram of localization errors corresponding to the depth ranges of tracked objects.

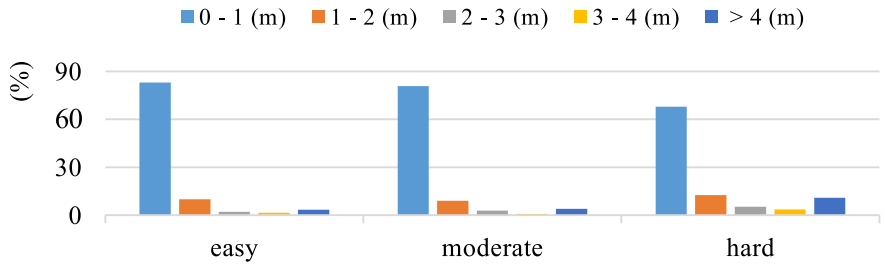


Figure 5.24: The histogram of localization errors corresponding to different object difficulty levels.

Figure 5.24 shows that complexity levels of detected objects, which are classified based on their occlusion and truncation degree in image space, influence the quality of their computed 3D positions. The localization results for the easy and moderate object types are comparable, however, the easy one is a little better. 83 % of objects belonging to the easy group have positioning error of no more than 1 m and 80 % in case of the moderates. When it comes to the hard group, the percentage of observations inaccurately localized (i.e. distance to references over 4 m) significantly increases to approximately 10 %. In contrast, the number of objects having spacial distance error within 1 m largely drops down to around 68 %. These results are well associated to the correlations between localization error and either IoU or depth ranges. Firstly, the more difficult an object, the harder it is to recover its whole appearance in image space. Thus, IoU with its reference BB is usually not high enough to yield well computed 3D geometry. In another aspect, a subset of the hard type observations appear quite far from the cameras, which is also a reason for their inaccurate 3D positions.

To this end, the positioning accuracy depends on a number of factors including how well objects are visible in images and the capability of the sensors. These issues can be partly solved by the proposed filtering methods, but cannot be completely discarded. The limitations of the sensor come from the pixel size and the baseline length of the stereo set up. For KITTI, the pixel size is around 0.007–0.4 (m) (focal length is approximately 720 px) at the depth range of 5–30 (m) (in KITTI, most of pedestrians appear in this range) and the baseline is approximately 0.54 m. As the dense matching results are not perfect and usually contain errors (around 0.9 px on KITTI),

which results in the depth error of approximately 0.007–2.1 (m) in the depth range of 5–30 (m). This can be improved by either using a stereo pair with larger baseline or better dense matching. In addition, pedestrians are non-rigid objects and thus their positions, represented by only one-foot points, are hard to be precisely annotated. Depending on the posture of a pedestrian, the centre mass point in the ground plane can be different from the foot point. When the uncertainties of disparity value is taken into account, the localization error in 3D object space is only smaller than 1.0 m when pedestrians are not away from the camera further than 20 m. Consequently, in general, the localization error of 0.2 m is not only a good result on KITTI but also for other mobile platforms.

5.5 Comparison with state-of-the-art trackers

In order to evaluate the performance of the proposed tracker w.r.t. other state-of-the-art methods, the results obtained on the testing tracking benchmark of KITTI and ETHMS dataset are presented in this section. Besides evaluation metrics described in Section 5.1.2, three additional features related to a tracker, i.e. association manner (online/ near online/ offline), tracking space (2D image/ 3D object space), and detection methods, are reported. The tracking approaches of the other state-of-the-art methods are presented in Chapter 3. In this section, they are discussed again with respect to the performance of the proposed tracking method. Additionally, the tracking results of the suggested tracker on the MuVi benchmark are presented, which allows an assessment of the generalization of the proposed tracking approach.

KITTI testing tracking benchmark

The performance of the proposed tracking framework, 3D-TLSR (3D pedestrian tracking using local structure refinement), is compared to all state-of-the-art trackers that are published on the testing tracking benchmark of KITTI[2]. The experimental results are summarized in Table 5.4 which is sorted according the metric MOTA.

As the first observation, the proposed tracker achieves comparable performance to TuSimple of Choi (2015) in most of the metrics, only MOTA is lower with approximately 4.2 %. While the suggested tracking method employs 3D information to track pedestrians in an online manner, TuSimple uses additional optical flow features in a ”near online” approach. However, TuSimple has much more IDs than 3D-TLSR, the difference is 28 %, though both methods have comparable MT (30.6 % and 29.6 %) and ML (24.1 % and 23.7 %) .

The 3D-TLSR tracker outperforms CAT (Nguyen et al., 2019), Be-track (Dimitrievski et al., 2019), MCMOT-CPD (Lee et al., 2016), RMOT (Yoon et al., 2015), and CIWT (Ošep et al.,

[2] `http://www.cvlibs.net/datasets/kitti/eval_tracking.php`, accessed on 22.03.2020

2017) in most metrics, partly with noticeable margins. Compared to its previous version CAT, 3D-TLSR employs post-processing more intensively and also applies LSR, which enables 3D-TLSR to reduce IDs by a factor of 2.0. Besides IDs, 3D-TLSR also enhances the tracking results in all metrics compared to CAT, except for a drop in the percentage of MT of around 5 %. Similar to 3D-TLSR, Be-Track and CIWT also employ 3D information to track pedestrians, yet their results are remarkably worse than 3D-TLSR in all metrics. Especially, while 3D-TLSR produces significantly larger MOTA (+2.7 % and +10.6 %), MT (+8.6 % and +15.8 %), and smaller ML (-7.6 % and -11.0 %) than Be-Track and CIWT, it is still capable of achieving a much lower number of IDs (-15.2 % and -10.7 %) and fragmentation (FR) (-1.5 % and -7.3 %). Though that MCMOT-CPD carries out the tracking offline, its performance in terms of tracking accuracy and Id consistency is not on par with 3D-TLSR whose association is performed in an online manner. Despite the fact that improvements on MOTA, MT, and ML of 3D-TLSR certainly stem from differences in performance of the employed detection methods, the proposed prediction method makes an important contribution in increasing the number of TPs, which results in good performance of the tracker in terms of recall and accuracy.

MDP (Xiang et al., 2015), SCEA (Yoon et al., 2016), JCSTD (Tian et al., 2019), and LP-SSVM (Wang and Fowlkes, 2017) are trackers having less IDs than 3D-TLSR. MDP formalizes the problem of tracking as a Markov decision process and the similarity function for data association is learned using reinforcement learning, which helps to produce a low number of IDs. Also employing structural constraints between pedestrians to improve the association results, SCEA and JCSTD carry out tracking in image domain, instead of in 3D object space like 3D-TLSR. LP-SSVM casts the tracking problem as a network flow approach and solves it offline once the whole image sequences are available. While these methods have a lower number of IDs than 3D-TLSR, it does not necessarily mean that they can perform the data association better. The important reason for their better IDs values is their low number of tracked pedestrians which are represented by the smaller MOTA (MDP: -6.7 %, SCEA: -9.8 %, JCSTD: -10.1 %, and LP-SSVM: -10.3 %), MT (MDP: -5.5 %, SCEA: -13.6 %, JCSTD: -13.4 %, and LP-SSVM: -8.9 %), and larger ML (MDP: -4.1 %, SCEA: -10.0 %, JCSTD: -19.6 %, and LP-SSVM: -10.7 %) compared to 3D-TLSR. It is obvious that the more pedestrians are tracked, the harder it is to maintain consistency and continuity of their trajectories. Thus, IDs and FR can happen more often. However, with the suggested tracking framework, even when handling a high number of targets with MOTA of 54 %, MT of 29.6 %, and ML of 23.7 %, the number of IDs is still kept at 100, which can be considered reasonably low. Hence, we argue that 3D-TLSR can handle the Id accuracy of pedestrians at least on par with other state-of-the-art trackers.

Among all up-to-date methods in Table 5.4, the 3D-TLSR tracker obtains the highest result for MOTP (73.0 %). It is illustrated in Section 5.4 that the localization error in 3D is highly correlated with the results of MOTP. Hence, this means that 3D-TLSR not only capable of tracking pedestrians with high accuracy but also can localize interesting objects both in image space and

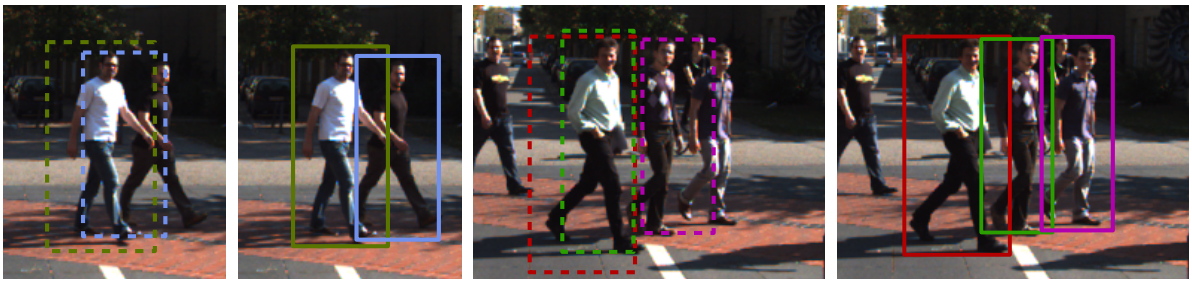


Figure 5.25: Two examples for inaccurate GT annotations of KITTI dataset (dashed boxes) and our detections (solid boxes).

object space with better precision than the other trackers.

ETHMS dataset

The performance of the 3D-TLSR tracker in comparison to the other tracking approaches is shown in Table 5.5. The proposed tracker achieves the best results on most metrics including recall, precision, and MT with large margins. Although 3D-TLSR operates online and is capable of tracking a large number of pedestrians with MT of 77.3 % and a recall of 89.3 %, a good Id accuracy of tracked trajectories is still achieved with IDs of 32. Except for SCEA (Yoon et al., 2019) and OnlineCRF (Yang and Nevatia, 2014), this number of IDs is noticeably lower than that of the other state-of-the-art methods, which work either online or offline. Taking into account the IDs and FG together with recall and MT, it is fair to say that the consistency of trajectories obtained by 3D-TLSR is at least as good as that of SCEA and OnlineCRF.

Comparing between ETHMS and KITTI, the performance of the proposed tracker on the ETHMS dataset is much better than on KITTI, though free parameters are learned and optimized using the KITTI training dataset. There are a number of reasons for this. First, the number of images and sequences in KITTI is much larger than in ETHMS, which also means that challenging situations can happen more often. Second, while KITTI takes into account objects with high difficult level and small appearance in image space (minimum height is 25 px), ETHMS with annotations of MOT15 only considers pedestrians with a BB height not smaller than 49 px. With larger appearance in image space, the visual feature of an object can be observed better and thus, not only it can be detected easier and more accurately but also their appearance similarity comparison is improved. Nevertheless, these results still imply that the proposed tracker is not over-fitted on a particular dataset and is able to perform properly on a new dataset that it has never seen before.

MuVi tracking benchmark

The tracking results on the MuVi benchmark are shown in Table 5.6, in which all tracking metrics are computed according to different thresholds of IoU in the range of 0.3–0.5. As the IoU becomes smaller, the recall and its related metrics including FP, MOTA, MT, ML improve. How-

ever, the improvements are not as significant as can be observed for the training dataset of KITTI (see Table 5.3). This is partly because the GT of MuVi are annotated with better quality. The problem with GT of KITTI is that pedestrians are annotated on 3D point clouds and the 2D BB is computed by back-projecting the 3D BB to image space. As soon as pedestrians are near to each other, the labelling in 3D usually is low quality which can result in incorrectly annotated BBs (see Figure 5.25).

In order to have an unbiased comparison with KITTI, the tracking results at $IoU = 0.5$ are employed for analysing the validity of the developed tracking method on both benchmarks. While free parameters are not re-learned for MuVi, the tracker still obtains 57.5 % for MOTA and 76.9 % for MOTP. Both values are thus considerably better than for KITTI. The differences are even larger for MT and ML, in which the values are increased by 27.6 % for MT and decreased by 20 % for ML. These results evidently validate the generic ability of the tracker to yield comparable results in diverse situations. Moreover, the free parameters can apparently also be transferred from one to another dataset without decreasing the performance of the tracker. While the number of IDs and FG are quite high for MuVi, considering the challenges of the image sequences in MuVi due to larger amounts of occlusion associated with the more crowed scene, the poorer image quality and the more difficult illumination conditions, the quality of extracted trajectories in MuVi can be seen to correspond to that in KITTI.

Tracker	domain	online	detector	MOTA	2D-MOTP	MT (%)	ML (%)	IDs	FR
TuSimple (Choi, 2015)	2D	near	Resnet (He et al., 2016)	**58.2**	71.9	30.6	24.1	138	818
3D-TLSR	3D	✓	Mask R-CNN (He et al., 2017)	54.0	**73.0**	29.6	**23.7**	100	835
CAT (Nguyen et al., 2019)	3D	✓	Mask R-CNN (He et al., 2017)	52.4	71.6	**34.4**	**23.7**	206	804
Be-Track (Dimitrievski et al., 2019)	3D	✓	SubCNN (Xiang et al., 2017)	51.3	72.7	21.0	31.3	118	848
MDP (Xiang et al., 2015)	2D	✓	SubCNN (Xiang et al., 2017)	47.2	70.4	24.1	27.8	87	825
MCMOT-CPD (Lee et al., 2016)	2D		Faster R-CNN (Ren et al., 2015)	46.0	72.4	20.6	34.4	143	764
JCSTD (Tian et al., 2019)	2D	✓	Regionlet	44.2	72.1	16.5	33.7	**53**	917
SCEA (Yoon et al., 2016)	2D	✓	Regionlet	43.9	71.9	16.2	43.3	56	**641**
RMOT (Yoon et al., 2015)	2D	✓	Regionlet	43.8	71.0	19.6	41.2	153	748
LP-SSVM (Wang and Fowlkes, 2017)	2D		Regionlet	43.8	70.5	20.6	34.4	73	809
CIWT (Ošep et al., 2017)	3D	✓	Regionlet	43.4	71.1	13.8	34.7	112	901

Table 5.4: Evaluation results on the Kitti dataset of the proposed tracking method 3D-TLSR (in bold) and other state-of-the-art methods. Regionlet detection results are provided by Kitti. Best value of each metric is in bold.

Tracker	domain	online	Recall (%)	Pre (%)	MT (%)	ML (%)	IDs	FG	MOTA	2D-MOTP
3D-TLSR	3D	✓	**89.3**	**97.7**	**77.3**	9.0	32	34	87.1	74.7
SCEA (Yoon et al., 2019)	2D	✓	82.5	89.6	71.1	5.6	24	32	-	-
RMOT (Yoon et al., 2015)	2D	✓	81.5	86.3	67.7	**4.8**	38	40	-	-
OnlineCRF (Yang and Nevatia, 2014)	2D		79.0	90.4	68.0	7.2	**11**	19	-	-
MotiCon (Leal-Taixé et al., 2014b)	2D		83.8	79.7	72.0	4.7	71	85	-	-
CRFT (Milan et al., 2013b)	2D		77.3	87.2	66.4	8.2	57	69	-	-
MOT-TBD (Poiesi et al., 2013)	2D	✓	78.7	85.5	62.4	8.0	69	45	-	-
StructMOT (Kim et al., 2012)	2D	✓	78.4	84.1	62.7	7.7	72	**5**	-	-
LPSFM (Leal-Taixé et al., 2011)	3D		74.1	75.3	55.1	7.9	131	184	-	-
KalmanSFM (Pellegrini et al., 2009)	3D		72.3	84.1	51.6	5.6	77	206	-	-

Table 5.5: Evaluation results of the proposed tracking method on the ETHMS dataset. While the results of KalmanSFM (Pellegrini et al., 2009) and LPSFM (Leal-Taixé et al., 2011) come from MotiCon (Leal-Taixé et al., 2014b), the others are extracted by original publications. The best results are in bold. Unknown values are specified by "-".

IoU	Recall (%)	FP (%)	MT (%)	ML (%)	IDs	FG	MOTA	2D-MOTP
0.50	70.7	13.2	57.6	9.5	351	998	57.5	**76.9**
0.45	71.7	12.1	62.1	9.5	**346**	**976**	59.5	76.4
0.40	72.4	11.4	65.6	8.9	354	989	61.0	76.0
0.35	72.9	10.7	67.5	**7.8**	364	1017	62.1	75.7
0.30	**73.2**	**10.2**	**68.9**	**7.8**	377	1043	**63.0**	75.3

Table 5.6: Evaluation results on the MuVi dataset of the proposed tracking method with various IoU thresholds. Best value of each metric is in bold.

6 Discussion

In this chapter, the experimental results reported in Chapter 5 are discussed in detail. In Section 6.1, the advantages and disadvantages of the proposed approach and the individual component are analysed, the impact on the final tracking results and the sensitivity of the free parameters are presented. Afterwards, an overall evaluation of the whole framework w.r.t. the accuracy and precision of the generated trajectories is given.

6.1 Proposed components

In order to make this section compatible with experiments in Chapter 5, the proposed methods are represented in the same order as the tested models in Section 5.3.

Detection post-processing

The suggested post-processing step aims at removing FP detections so that the inputs of the association stage are less noisy. The prior information about pedestrians that are employed includes: their height in object space is limited and the ratio between the height and width should lie in a certain range. In addition, the percentage of pixels belonging to the instance segmentation masks provided by mask R-CNN is utilised to eliminate bad quality detections which usually cover solely a small part of pedestrians. The results in Section 5.2.1 illustrate that combining all the aforementioned cues helps to eliminate in total roughly 40 % of FPs directly after the detection step. However, at the same time, a small percentage of 2.2 % of TPs is also incorrectly determined as false alarms. The free parameters of this approach have been defined by inspecting the statistic of detection results. By the way of eliminating a huge number of incorrect detections, this component significantly contributes to the improvement of the tracker performance not only in terms of FPs and MOTA but also as far as the number of IDs is concerned, which are illustrated in Table 5.3.

Association gates

The association gates are designed to use geometry cues in both image and object space to eliminate false matching pairs. The motivation behind this suggested method is that people walk at a limited speed and at the same time a pedestrian cannot stay at two different spots. By filtering out

association pairs that are far away in object space and do not overlap enough in image space, the association optimization becomes less complicated and more accurate. It is demonstrated in Figure 5.8 that the association gates help to eliminate more than 45 % of IDs, significantly improving the consistency of generated trajectories. Without taking these gates into account, the tracker cannot maintain the Id accuracy for tracked objects, which results in more than double the number of IDs (see Table 5.3). Since the gates make binary decisions about whether a detection can be assigned to a trajectory or not, which cannot be corrected later, their thresholds need to be determined with care. Though the free parameters of the association gates are slightly sensitive, they can be learned from the training data and prior knowledge about pedestrian movements.

Tracking confirm detection

The TCD approach is introduced to take advantage of detections with high probability of correctness, while not ignoring detected pedestrians with low cfd scores. The influences of this method on the tracking approach are shown in Table 5.3. While using only detections with high certainty of correctness leads to a reduction in the recall, MOTA, and MT, employing also detections with low scores to initialize new trajectories helps to increase TP but at the same time more FPs are generated, which results in a large decrease in MOTA. It is important to carefully select the optimum threshold for creating new trajectories as this can lead to the varying tracking accuracy. In contrast, a second threshold, which is responsible for selecting, which detections should be considered during the association, does not cause significant change in MOTA, MT, and ML.

Local structure refinement

In order to improve the association accuracy in complicated, often crowded situations with many occlusions, the LSR is proposed to obtain geometry changes of trajectories with a high probability of correctness as preliminary information. However, it is not easy to identify whether an anchor is a correct match or not. Thus, a number of false information is always contained in the selected anchors as shown in Figure 5.12. The experimental results presented in Section 5.2.2 reveal that many incorrect anchors (here 24 %) does not lead to good results. By employing only associations with a high certainty of correctness as anchors, a lot of problems are eliminated. This also does not help to maximize the performance of LSR, due to the loss of many correct anchors. The important point is that there should be enough correct anchors near to where the IDs happen. A good balance between the number of anchors and the percentage of incorrect results gives the best performance for the LSR. The importance of this method in the whole tracker is confirmed by experimental results in Section 5.3. By applying the LSR, the number of IDs is reduced by a noticeable percentage. Besides Id accuracy, other tracking metrics including MOTA, MOTP, and FG are also improved.

Missed detection recovery

Through the prediction step, pedestrians missed by the detector are retrieved using spatio-temporal information of their trajectories. This enables an increase in recall, but can add more false alarms due to a divergence between a correct position and its predicted one. By selecting proper values for free parameters, the drift can be constrained by examining the number of epochs that a trajectory can stay in the inactive state. The experiments in Section 5.3 show that the prediction allows enhancing tracking results in all metrics with noteworthy margins, except for FPs and IDs. However, arguably the number of IDs increases only, because more and longer trajectories are tracked. It does not necessarily mean that the Id accuracy of tracked targets is reduced.

6.2 Performance of the proposed tracker

The tracker developed in this work aims at improving both, accuracy and precision of multiple person tracking. The obtained improvements are presented and evaluated through practical experiments in Chapter 5.

The tracking completeness and correctness of the proposed approach are investigated in Section 5.5. For the testing KITTI dataset, the tracking method achieves 54 % of MOTA, which is the second best result among the published trackers listed in the benchmark and 73 % for MOTP, the best overall results. More than 29 % of the pedestrians appearing in the testing sequences are regularly tracked (i.e. MT). Meanwhile, approximately 24 % of interesting objects are usually lost during tracking (i.e. ML). This means that some pedestrians are detected and tracked, but their appearance in most of epochs are fully recovered. One way to overcome this problem is to use a more powerful detector. The results on the KITTI testing benchmark prove that the suggested tracking approach performs on par with other state-of-the-art methods. For ETHMS, the tracking approach achieves the best recall, precision, MT, and ML among other up-to-date tracking approaches. In all three datasets used for the experiments, the 3D-TLSR tracker obtains the best results on the ETHMS dataset, though the tracker's components are optimized using KITTI dataset. However, this also because the objects in KITTI are more challenging to track than in ETHMS. On the MuVi dataset, the tracker even yields better results with 57 % of MOTA, 57 % of MOTP, 23.6 % of MT, and 25 % of ML, though the sequences in this dataset are much more challenging than KITTI. The comparable performance on KITTI, ETHMS, and MuVi demonstrates the generalization capability of the proposed tracker, which means that it is not over-fitted on one dataset, but can cope with various scenarios.

The consistency of tracked trajectories is measured by the number of IDs; improving this metric which is one of the main objectives of this work. The trajectory consistency is primarily accomplished in the association stage, in which the suggested association gates and LSR methods enable an improved assignment. In addition, the other components in the framework including post-

processing and TCD also support achieving a low number of IDs. The capability of generating trajectories with high Id accuracy is reported in Chapter 5. However, since IDs need to be considered together with the number of tracked objects, it is hard to compare tracking results on this metric between various approaches. The lower the percentage of ML, the harder it is to maintain accurate Ids due to the large number of tracked objects that a tracker needs to handle. In this sense, the proposed tracking framework can consistently track pedestrians with a low number of IDs on the KITTI benchmark in comparison with other state-of-the-art approaches. For the ETHMS, the tracing approach can handle the association well with a large number of tracked objects. Though using online association approach, the 3D-TLSR tracker has a lower number of IDs than other methods whose association is carried out offline. For the MuVi, while the number of IDs is quite large, putting the number into the perspective with respect to the challenges of image sequences in MuVi and the number of tracked objects, it is fair to say that the suggested association approach performs comparably on both benchmarks.

With the purpose of tracking pedestrians in 3D object space, a high localization precision is one of the goals that the developed tracking approach aims at. The positioning accuracy of tracked trajectories is explored in Section 5.4 using the training KITTI dataset. If all detections have IoU larger than 0.0 are considered as TP, more than 80 % of recalls are localized within 1 m of displacement compared to GT and 10 % of the tracked objects have localization error larger than 2 m . Using the evaluation criteria of KITTI with $IoU = 0.5$, these results change to 87 % and 4 %, respectively. This demonstrates that the tracking approach is able to track pedestrians in 3D object space with high precision under many challenges such as moving sensors, changing illuminations and noisy disparity values. Though 2D-MOTP is not the focus of this work, on this metric, the proposed approach still obtains the best performance on the KITTI score board and similar results on MuVi.

Generally, compared to other state-of-the-art tracking methods, the performance of the suggested tracker is comparable on all metrics. Although the tracking is carried out in an online manner, the consistency of generated trajectories is still preserved and no less competitive than the methods using offline association approaches. In addition, not using any prior information about the scene enables the tracker to be able to deal with highly dynamic scenes. By using additional 3D information from stereo cameras, the tracker can avoid strong assumptions on the movement of sensor systems the other approaches do. The tests on MuVi demonstrate the generalization ability of the proposed tracking method. While a number of free parameters are required to be determined so that the tracker can perform well, they can be learned from a small subset of training data. One important assumption, which needs to hold so that the tracker can work, is that the ground is a plane instead of terrains. However, this is not the drawback of the tracker, since the problem can be solved by fitting multiple planes for the ground or exploiting a digital terrain model.

7 Conclusion

Motivated by applications related to autonomous driving, the presented thesis investigated the problem of multiple pedestrian tracking in 3D object space using stereo cameras. Following the state-of-the-art tracking-by-detection paradigm, several issues have been investigated and developed in this work to improve both, tracking accuracy and precision. In this section, conclusions are drawn for the methods proposed in the tracking framework and an outlook for extensions in future work is discussed.

By employing specific characteristics associated with the appearance of pedestrian and instance segmentation masks provided by mask R-CNN, a number of FPs are filtered out. Consequently, these methods enable not only lowering the number of false alarms but also reducing the complication of assignment optimization, which has proven to yield significant improvements in tracking accuracy. While a large number of FPs can be eliminated in the post-processing stage, the recall heavily depends on the detector capability. Thus, one way to improve the results is a collaborative set up using in multiple viewpoints so that a person who cannot be detected in one sensor can be well-captured in the others. Additionally, redundant observations observed from different sensors allow the enhancement of positioning precision. Especially, this direction is also important in the future for autonomous driving, in which cars can cooperate to understand their common dynamic surrounding by exchanging sensor information.

The Id accuracy of generated trajectories is improved in the association step, in which geometry and visual features are combined into association weights to express how likely an observation belongs to an existing target. The assignments are then globally optimized using linear programming. Association gates and LSR are introduced to improve the performance of the association stage by utilising prior knowledge about the movements of pedestrians. The advantage of this approach is that it is straightforward to add more constraints or cues to the association weight, if needed. However, as the geometry is compared only in a local manner between a detection and an interesting target, one interesting cue, i.e., the geometry change of all pedestrians in the scene, is neglected. As pedestrians move smoothly for a small period of time, the graph formed by their 3D position should maintain a similar topology in two epochs. One possible research line for the association is to directly incorporate the geometry of all targets in a scene into a step of calculating the weights using graph matching. In this way, the geometry changes of all targets can be used as additional cue for assignment optimization.

The prediction step is introduced to retrieve missed detection. The core of this step is to evaluate when the position prediction should be stopped for a particular trajectory. Though the suggested method works well and facilitates the increase of recall, several aspects still need to be advanced further in the future. In the current approach, the interaction of pedestrians is taken into account by the motion model, which allows to anticipate the movement of the desired target. The suggested model is developed based on the assumption that groups of pedestrians have similar movement intentions. Instead of using assumptions about the movement of people in general, a neural network can directly predict the motion of a pedestrian based on training data. This trend of research can also be advanced by including segmentation information so that not only pedestrians are taken into account during the prediction but also other objects in a scene can be considered as well.

While there are several existing metrics to evaluate the performance of a tracking approach in terms of accuracy and precision, these terms pose some limitations on comparing state-of-the-art methods. The MOTA value more or less only reflects the number of recall and false alarm, which is much more dominant than the number of IDs. Thus, the consistency of the tracked target hardly to be analysed by this metrics. On the other hand, using only IDs to compare the association accuracy of various trackers is not sufficient, as this value needs to be put into the context with the number of MT, ML, and recall. Such metrics are not available yet and should be investigated in the future for more efficient evaluations and comparisons.

To this end, this work has presented a new framework to track multiple pedestrians in 3D object space. With a focus on automotive applications, the tracker has been developed to carry out the task in a flexible manner, in which no prior information about the scene is used and the association is implemented in an online manner. The experimental results on KITTI, ETHMS, and MuVi demonstrate that the suggested tracking approach is currently the best online 3D tracker in terms of accuracy and the other criteria commonly applied in multiple pedestrian tracking.

Bibliography

Alexe, B., Deselaers, T. and Ferrari, V., 2012. Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 34(11), pp. 2189–2202.

Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F. and Malik, J., 2014. Multiscale Combinatorial Grouping. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 328–335.

Aslani, S. and Mahdavi-Nasab, H., 2013. Optical Flow Based Moving Object Detection and Tracking for Traffic Surveillance. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 7(9), pp. 1252–1256.

Bae, S.-H. and Yoon, K.-J., 2018. Confidence-Based Data Association and Discriminative Deep Appearance Learning for Robust Online Multi-Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 40(3), pp. 595–610.

Bazaraa, M. S., Jarvis, J. J. and Sherali, H. D., 2011. *Linear Programming and Network Flows*. John Wiley & Sons.

Benenson, R., Mathias, M., Tuytelaars, T. and Van Gool, L., 2013. Seeking the Strongest Rigid Detector. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3666–3673.

Benfold, B. and Reid, I., 2011. Stable Multi-Target Tracking in Real-Time Surveillance Video. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3457–3464.

Berclaz, J., Fleuret, F., Turetken, E. and Fua, P., 2011. Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 33(9), pp. 1806–1819.

Bernardin, K. and Stiefelhagen, R., 2008. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*.

Blott, G., Yu, J. and Heipke, C., 2018. View-Aware Person Re-identification. In: *German Conference on Pattern Recognition (GCPR)*, pp. 46–59.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E. and Van Gool, L., 2009. Robust Tracking-by-Detection using a Detector Confidence Particle Filter. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1515–1522.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E. and Van Gool, L., n.d. Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 33(9), pp. 1820–1833.

Butt, A. A. and Collins, R. T., 2013. Multi-target Tracking by Lagrangian Relaxation to Min-cost Network Flow. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1846–1853.

Chen, X., Kundu, K., Zhu, Y., Berneshawi, A. G., Ma, H., Fidler, S. and Urtasun, R., 2015. 3D Object Proposals for Accurate Object Class Detection. In: *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, pp. 424–432.

Chen, X., Ma, H., Wan, J., Li, B. and Xia, T., 2017. Multi-View 3D Object Detection Network for Autonomous Driving. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1907–1915.

Cheng, X., Zhang, Y., Zhou, L. and Zheng, Y., 2019. Visual Tracking via Auto-Encoder Pair Correlation Filter. *IEEE Transactions on Industrial Electronics* 67(4), pp. 3288–3297.

Choi, W., 2015. Near-Online Multi-Target Tracking With Aggregated Local Flow Descriptor. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3029–3037.

Choi, W. and Savarese, S., 2012. A Unified Framework for Multi-target Tracking and Collective Activity Recognition. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 215–230.

Dai, J., Li, Y., He, K. and Sun, J., 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In: *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. 379–387.

Dalal, N. and Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893.

Dantzig, G. B., 1998. *Linear Programming and Extensions*. Vol. 48, Princeton University Press.

Dehghan, A., Modiri Assari, S. and Shah, M., 2015a. GMMCP Tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4091–4099.

Dehghan, A., Tian, Y., Torr, P. H. and Shah, M., 2015b. Target Identity-Aware Network Flow for Online Multiple Target Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1146–1154.

Dicle, C., Camps, O. I. and Sznaier, M., 2013. The Way They Move: Tracking Multiple Targets with Similar Appearance. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2304–2311.

Dimitrievski, M., Veelaert, P. and Philips, W., 2019. Behavioral Pedestrian Tracking Using a Camera and LiDAR Sensors on a Moving Vehicle. *Sensors* 19(2), pp. 391.

Dollár, P., Appel, R., Belongie, S. and Perona, P., 2014. Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 36(8), pp. 1532–1545.

Dollár, P., Tu, Z., Perona, P. and Belongie, S., 2009. Integral Channel Features. Proc. of the British Machine Vision Conference (BMVC), pp. 91.1–91.11.

Ess, A., Leibe, B., Schindler, K. and Van Gool, L., 2008. A Mobile Vision System for Robust Multi-Person Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 1–8.

Fagot-Bouquet, L., Audigier, R., Dhome, Y. and Lerasle, F., 2016. Improving Multi-frame Data Association with Sparse Representations for Robust Near-online Multi-object Tracking. In: *Proc. of the European Conference on Computer Vision (ECCV)*, Springer, pp. 774–790.

Felzenszwalb, P., McAllester, D. and Ramanan, D., 2008. A Discriminatively Trained, Multiscale, Deformable Part Model. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Gao, J., Wang, J., Dai, S., Li, L.-J. and Nevatia, R., 2019. NOTE-RCNN: NOise Tolerant Ensemble RCNN for Semi-Supervised Object Detection. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 9508–9517.

Geiger, A., Lenz, P. and Urtasun, R., 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361.

Geiger, A., Ziegler, J. and Stiller, C., 2011. StereoScan: Dense 3d Reconstruction in Real-time. In: *Intelligent Vehicles Symposium*, pp. 963–968.

Gelb, A., 1974. *Applied Optimal Estimation*. MIT press.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587.

Gomory, R. E., 1958. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society* 64, pp. 275–278.

Grewal, M. S. and Andrews, A. P., 2014. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons.

He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask R-CNN. In: *Proc. of the IEEE Conference on Computer Vision (ICCV)*, pp. 2961–2969.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9), pp. 1904–1916.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep Residual Learning for Image Recognition. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Helbing, D. and Molnar, P., 1995. Social force model for pedestrian dynamics. *Physical Review E* 51(5), pp. 4282–4286.

Henschel, R., Leal-Taixé, L., Cremers, D. and Rosenhahn, B., 2018. Fusion of Head and Full-Body Detectors for Multi-Object Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Workshops)*, pp. 1428–1437.

Henschel, R., Zou, Y. and Rosenhahn, B., 2019. Multiple People Tracking Using Body and Joint Detections. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Workshops)*.

Hermans, A., Beyer, L. and Leibe, B., 2017. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*.

Hoang, V.-D., Le, M.-H. and Jo, K.-H., 2014. Hybrid cascade boosting machine using variant scale blocks based HOGfeatures for pedestrian detection. *Neurocomputing* 135, pp. 357–366.

Hu, W., Li, X., Luo, W., Zhang, X., Maybank, S. and Zhang, Z., 2012. Single and Multiple Object Tracking Using Log-Euclidean Riemannian Subspace and Block-Division Appearance Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(12), pp. 2420–2440.

Hu, Z. and Uchimura, K., 2005. U-V-Disparity: An efficient algorithm for Stereovision Based Scene Analysis. In: *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 48–54.

Huang, Y., Bi, H., Li, Z., Mao, T. and Wang, Z., 2019. STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 6272–6281.

Izadinia, H., Saleemi, I., Li, W. and Shah, M., 2012. (MP)2T: Multiple People Multiple Parts Tracker. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 100–114.

Jafari, O. H., Mitzel, D. and Leibe, B., 2014. Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras. In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5636–5643.

Jin, J., Dundar, A., Bates, J., Farabet, C. and Culurciello, E., 2013. Tracking with Deep Neural Networks. In: *Proc. of the 47th Annual Conference on Information Sciences and Systems (CISS)*, IEEE, pp. 1–5.

Kieritz, H., Becker, S., Hübner, W. and Arens, M., 2016. Online multi-person tracking using Integral Channel Features. In: *Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 122–130.

Kim, C., Li, F., Ciptadi, A. and Rehg, J. M., 2015. Multiple Hypothesis Tracking Revisited. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4696–4704.

Kim, S., Kwak, S., Feyereisl, J. and Han, B., 2012. Online Multi-target Tracking by Large Margin Structured Learning. In: *Proc. of the Asian Conference on Computer Vision (ACCV)*, pp. 98–111.

Klee, V. and Minty, G. J., 1972. How Good is the Simplex Algorithm? *Inequalities* 3(3), pp. 159–175.

Klinger, T., Rottensteiner, F. and Heipke, C., 2017. Probabilistic multi-person localisation and tracking in image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing* 127, pp. 73–88.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105.

Kuo, C.-H., Huang, C. and Nevatia, R., 2010. Multi-Target Tracking by On-Line Learned Discriminative Appearance Models. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 685–692.

Leal-Taixé, L. and Rosenhahn, B., 2013. Pedestrian Interaction in Tracking: The Social Force Model and Global Optimization Methods. In: *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective*, Springer, pp. 267–294.

Leal-Taixé, L., 2014. Multiple object tracking with context awareness. *arXiv preprint arXiv:1411.7935*.

Leal-Taixé, L., Canton-Ferrer, C. and Schindler, K., 2016. Learning by Tracking: Siamese CNN for Robust Target Association. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Workshops)*, pp. 33–40.

Leal-Taixe, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B. and Savarese, S., 2014a. Learning an Image-based Motion Context for Multiple People Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3542–3549.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B. and Savarese, S., 2014b. Learning an Image-based Motion Context for Multiple People Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3542–3549.

Leal-Taixé, L., Milan, A., Schindler, K., Cremers, D., Reid, I. and Roth, S., 2017. Tracking the Trackers: An Analysis of the State of the Art in Multiple Object Tracking. *arXiv preprint arXiv:1704.02781*.

Leal-Taixé, L., Pons-Moll, G. and Rosenhahn, B., 2011. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV) (Workshops)*, pp. 120–127.

Lee, B., Erdenee, E., Jin, S., Nam, M. Y., Jung, Y. G. and Rhee, P. K., 2016. Multi-class Multi-object Tracking Using Changing Point Detection. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 68–83.

Lee, S.-H., Jang, W.-D. and Kim, C.-S., 2018. Tracking-by-segmentation using superpixel-wise neural network. *IEEE Access* 6, pp. 54982–54993.

Lenz, P., Geiger, A. and Urtasun, R., 2015. FollowMe: Efficient Online Min-Cost Flow Tracking With Bounded Memory and Computation. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4364–4372.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J. and Yan, J., 2019. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4282–4291.

Li, B., Yan, J., Wu, W., Zhu, Z. and Hu, X., 2018a. High Performance Visual Tracking With Siamese Region Proposal Network. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8971–8980.

Li, B., Yan, J., Wu, W., Zhu, Z. and Hu, X., 2018b. High Performance Visual Tracking With Siamese Region Proposal Network. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8971–8980.

Li, Y., Huang, C. and Nevatia, R., 2009. Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2953–2960.

Linder, T., Breuers, S., Leibe, B. and Arras, K. O., 2016. On Multi-Modal People Tracking from Mobile Platforms in Very Crowded and Dynamic Environments. In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5512–5519.

Liu, F., Gong, C., Huang, X., Zhou, T., Yang, J. and Tao, D., 2018. Robust Visual Tracking Revisited: From Correlation Filter to Template Matching. *IEEE Transactions on Image Processing* 27(6), pp. 2777–2790.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C., 2016. SSD: Single Shot MultiBox Detector. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 21–37.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully Convolutional Networks for Semantic Segmentation. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440.

Luber, M., Stork, J. A., Tipaldi, G. D. and Arras, K. O., 2010. People Tracking with Human Motion Predictions from Social Forces. In: *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 464–469.

Ma, C., Huang, J.-B., Yang, X. and Yang, M.-H., 2015. Hierarchical Convolutional Features for Visual Tracking. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3074–3082.

Maksai, A., Wang, X., Fleuret, F. and Fua, P., 2017. Non-Markovian Globally Consistent Multi-Object Tracking. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2544–2554.

Matoušek, J., Sharir, M. and Welzl, E., 1996. A Subexponential Bound for Linear Programming. *Algorithmica* 16, pp. 498–516.

Matsukawa, T., Okabe, T., Suzuki, E. and Sato, Y., 2016. Hierarchical Gaussian Descriptor for Person Re-Identification. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1363–1372.

Milan, A., Roth, S. and Schindler, K., 2013a. Continuous Energy Minimization for Multitarget Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(1), pp. 58–72.

Milan, A., Schindler, K. and Roth, S., 2013b. Detection- and Trajectory-Level Exclusion in Multiple Object Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3682–3689.

Mitzel, D. and Leibe, B., 2011. Real-Time Multi-Person Tracking with Detector Assisted Structure Propagation. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV) (Workshops)*, pp. 974–981.

Mitzel, D., Horbert, E., Ess, A. and Leibe, B., 2010. Multi-person Tracking with Sparse Detection and Continuous Segmentation. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 397–410.

Mousavian, A., Anguelov, D., Flynn, J. and Kosecka, J., 2017. 3D Bounding Box Estimation Using Deep Learning and Geometry. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7074–7082.

Mueller, M., Smith, N. and Ghanem, B., 2017. Context-Aware Correlation Filter Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1396–1404.

Nam, H. and Han, B., 2016. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4293–4302.

Nguyen, U., Rotteinsteiner, F. and Heipke, C., 2018. Object Proposals for Pedestrian Detection in Stereo Images. *38. Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung in München* Band 27(9), pp. 611–623.

Nguyen, U., Rottensteiner, F. and Heipke, C., 2019. CONFIDENCE-AWARE PEDESTRIAN TRACKING USING A STEREO CAMERA. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W5, pp. 53–60.

Oh, S., Russell, S. and Sastry, S., 2009. Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Transactions on Automatic Control* 54(3), pp. 481–497.

Ošep, A., Mehner, W., Mathias, M. and Leibe, B., 2017. Combined Image- and World-Space Tracking in Traffic Scenes. In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1988–1995.

Paisitkriangkrai, S., Shen, C. and Van Den Hengel, A., 2014. Strengthening the Effectiveness of Pedestrian Detection with Spatially Pooled Features. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 546–561.

Pellegrini, S., Ess, A. and Van Gool, L., 2010. Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 452–465.

Pellegrini, S., Ess, A., Schindler, K. and Van Gool, L., 2009. You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 261–268.

Pirsiavash, H., Ramanan, D. and Fowlkes, C. C., 2011. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1201–1208.

Poiesi, F., Mazzon, R. and Cavallaro, A., 2013. Multi-target tracking on confidence maps: An application to people tracking. *Computer Vision and Image Understanding (CVIU)* 117(10), pp. 1257–1272.

Qin, Z. and Shelton, C. R., 2012. Improving Multi-target Tracking via Social Grouping. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1972–1978.

Rasouli, A. and Tsotsos, J. K., 2019. Autonomous Vehicles That Interact With Pedestrians: A Survey of Theory and Practice. *IEEE Transactions on Intelligent Transportation Systems* 21(3), pp. 900–918.

Rasouli, A., Kotseruba, I., Kunic, T. and Tsotsos, J. K., 2019. PIE: A Large-Scale Dataset and Models for Pedestrian Intention Estimation and Trajectory Prediction. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 6262–6271.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *Advances in Neural Information Processing Systems 28 (NIPS)*, pp. 91–99.

Satpathy, A., Jiang, X. and Eng, H.-L., 2014. Human Detection by Quadratic Classification on Subspace of Extended Histogram of Gradients. *IEEE Transactions on Image Processing* 23(1), pp. 287–297.

Schindler, K., Ess, A., Leibe, B. and Van Gool, L., 2010. Automatic detection and tracking of pedestrians from a moving stereo rig. *ISPRS Journal of Photogrammetry and Remote Sensing* 65(6), pp. 523–537.

Schön, S., Brenner, C., Alkhatib, H., Coenen, M., Dbouk, H., Garcia-Fernandez, N., Fischer, C., Heipke, C., Lohmann, K., Neumann, I. et al., 2018. Integrity and Collaboration in Dynamic Sensor Networks. *Sensors* 18(7), pp. 2400.

Schwarz, L. A., Mkhitaryan, A., Mateus, D. and Navab, N., 2012. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing* 30(3), pp. 217–226.

Shitrit, H. B., Berclaz, J., Fleuret, F. and Fua, P., 2013. Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(8), pp. 1614–1627.

Tang, S., Andres, B., Andriluka, M. and Schiele, B., 2015. Subgraph Decomposition for Multi-Target Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5033–5041.

Tang, S., Andres, B., Andriluka, M. and Schiele, B., 2016a. Multi-person Tracking by Multicut and Deep Matching. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 100–111.

Tang, S., Andriluka, M., Andres, B. and Schiele, B., 2017. Multiple People Tracking by Lifted Multicut and Person Re-Identification. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3539–3548.

Tang, Y. S. and Lee, G. H., 2019. Transferable Semi-Supervised 3D Object Detection From RGB-D Data. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1931–1940.

Tang, Y., Wang, J., Gao, B., Dellandréa, E., Gaizauskas, R. and Chen, L., 2016b. Large Scale Semi-Supervised Object Detection Using Visual and Semantic Knowledge Transfer. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2119–2128.

Tian, W., Lauer, M. and Chen, L., 2019. Online Multi-Object Tracking Using Joint Domain Information in Traffic Scenarios. *IEEE Transactions on Intelligent Transportation Systems* 21(1), pp. 374–384.

Tomasi, C. and Kanade, T., 1991. Detection and tracking of point features.

Uijlings, J. R., Van De Sande, K. E., Gevers, T. and Smeulders, A. W., 2013. Selective Search for Object Recognition. *International Journal of Computer Vision (IJCV)* 104(2), pp. 154–171.

Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A. and Torr, P. H., 2017. End-To-End Representation Learning for Correlation Filter Based Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2805–2813.

Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W. and Maybank, S., 2018. Learning Attentions: Residual Attentional Siamese Network for High Performance Online Visual Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4854–4863.

Wang, Q., Zhang, L., Bertinetto, L., Hu, W. and Torr, P. H., 2019. Fast Online Object Tracking and Segmentation: A Unifying Approach. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1328–1338.

Wang, S. and Fowlkes, C. C., 2017. Learning Optimal Parameters for Multi-target Tracking with Contextual Interactions. *International Journal of Computer Vision* 122, pp. 484–501.

Wang, X., Türetken, E., Fleuret, F. and Fua, P., 2014. Tracking Interacting Objects Optimally Using Integer Programming. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 17–32.

Wolsey, L. A. and Nemhauser, G. L., 1999. *Integer and combinatorial optimization*. Vol. 55, John Wiley & Sons.

Xia, B. N., Gong, Y., Zhang, Y. and Poellabauer, C., 2019. Second-Order Non-Local Attention Networks for Person Re-Identification. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3760–3769.

Xiang, Y., Alahi, A. and Savarese, S., 2015. Learning to Track: Online Multi-Object Tracking by Decision Making. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4705–4713.

Xiang, Y., Choi, W., Lin, Y. and Savarese, S., 2017. Subcategory-Aware Convolutional Neural Networks for Object Proposals and Detection. In: *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 924–933.

Xiao, J., Stolkin, R. and Leonardis, A., 2015. Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4978–4987.

Xing, J., Ai, H. and Lao, S., 2009. Multi-Object Tracking through Occlusions by Local Tracklets Filtering and Global Tracklets Association with Detection Responses. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1200–1207.

Xu, J., Cao, Y., Zhang, Z. and Hu, H., 2019. Spatial-Temporal Relation Networks for Multi-Object Tracking. In: *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3988–3998.

Yamaguchi, K., Berg, A. C., Ortiz, L. E. and Berg, T. L., 2011. Who are you with and Where are you going? In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1345–1352.

Yamaguchi, K., McAllester, D. and Urtasun, R., 2014. Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. In: *Proc. of the European Conference on Computer Vision (ECCV)*, Springer, pp. 756–771.

Yang, B. and Nevatia, R., 2012. Multi-Target Tracking by Online Learning of Non-linear Motion Patterns and Robust Appearance Models. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1918–1925.

Yang, B. and Nevatia, R., 2014. Multi-Target Tracking by Online Learning a CRF Model of Appearance and Motion Patterns. *International Journal of Computer Vision (IJCV)* 107, pp. 203–217.

Yang, B., Huang, C. and Nevatia, R., 2011. Learning Affinities and Dependencies for Multi-Target Tracking using a CRF Model. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1233–1240.

Yang, Y., Shu, G. and Shah, M., 2013. Semi-supervised Learning of Feature Hierarchies for Object Detection in a Video. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1650–1657.

Yeo, D., Son, J., Han, B. and Hee Han, J., 2017. Superpixel-based tracking-by-segmentation using markov chains. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1812–1821.

Yoo, H., Kim, K., Byeon, M., Jeon, Y. and Choi, J. Y., 2016. Online Scheme for Multiple Camera Multiple Target Tracking Based on Multiple Hypothesis Tracking. *IEEE Transactions on Circuits and Systems for Video Technology* 27(3), pp. 454–469.

Yoon, H. J., Lee, C.-R., Yang, M.-H. and Yoon, K.-J., 2016. Online Multi-Object Tracking via Structural Constraint Event Aggregation. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1392–1400.

Yoon, J. H., Lee, C.-R., Yang, M.-H. and Yoon, K.-J., 2019. Structural Constraint Data Association for Online Multi-object Tracking. *International Journal of Computer Vision (IJCV)* 127(1), pp. 1–21.

Yoon, J. H., Yang, M.-H., Lim, J. and Yoon, K.-J., 2015. Bayesian Multi-object Tracking Using Motion Context from Multiple Objects. In: *Proc. of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 33–40.

Zamir, A. R., Dehghan, A. and Shah, M., 2012. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. In: *Proc. of the European Conference on Computer Vision (ECCV)*, Springer, pp. 343–356.

Zhang, L. and van der Maaten, L., 2013. Structure Preserving Object Tracking. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1838–1845.

Zhang, L., Li, Y. and Nevatia, R., 2008. Global Data Association for Multi-Object Tracking Using Network Flows. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Zhang, S., Bauckhage, C. and Cremers, A. B., 2014. Informed Haar-like Features Improve Pedestrian Detection. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 947–954.

Zhang, S., Benenson, R. and Schiele, B., 2015. Filtered Channel Features for Pedestrian Detection. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1751–1760.

Zhang, S., Benenson, R., Omran, M., Hosang, J. and Schiele, B., 2016. How Far Are We From Solving Pedestrian Detection? In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1259–1267.

Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S. Z., 2018. Single-Shot Refinement Neural Network for Object Detection. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4203–4212.

Zhang, T., Liu, S., Xu, C., Liu, B. and Yang, M.-H., 2017. Correlation Particle Filter for Visual Tracking. *IEEE Transactions on Image Processing* 27(6), pp. 2676–2687.

Zhu, X., Pang, J., Yang, C., Shi, J. and Lin, D., 2019. Adapting Object Detectors via Selective Cross-Domain Alignment. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 687–696.

Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J. and Hu, W., 2018. Distractor-aware Siamese Networks for Visual Object Tracking. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 101–117.

Zitnick, C. L. and Dollár, P., 2014. Edge Boxes: Locating Object Proposals from Edges. In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 391–405.

Acknowledgement

Since this dissertation is successfully finished with the supports of many people in one way or another, I would like to express my warmest appreciation to all of the people who accompanied me during my study.

First and foremost, I would like to express my deepest gratitude to my advisor Christian Heipke for his careful guidance and continuous supports. While still giving me the freedom to develop my own ideas, he always gave me thorough advice and encouragement during my working at Institute of Photogrammetry and GeoInformation (IPI) in Hannover.

I would like to thank my second advisor Franz Rottensteiner, who also acted as chair of the examination committee. I received a lot of valuable comments and suggestions from discussions with him. Furthermore, I also want to thank Prof. Dr.-Ing. habil. Monika Sester and Prof. Dr.-Ing. Michael Yang for being my referee.

I thank all my colleagues from the Institute of Photogrammetry and GeoInformation and the i.c.sens for creating a friendly and professional working environment. Especially, I enjoyed the fruitful working, discussing, and joking time with my colleagues in our i.c.sens office.

Finally, my gratitude goes to my family for their unwavering encouragement, love, and support throughout my life. I particularly thank my dear husband, Huy, for his understanding and love during the eventful time.

Curriculum Vitae

Personal Information

Name	Uyen D-X, Nguyen
Date of Birth	08-December-1989

Work Experience

December 2016 - May 2020	Institute of Photogrammetry and GeoInformation Leibniz Universität Hannover *Research Assistant*
June 2015 - September 2015	Machine Vision and Pattern Recognition Laboratory Lappeenranta University of Technology, Finland *Research Assistant*
March 2012 - July 2014	Robert Bosch Engineering and Business Solutions Vietnam *Software Developer*

Education

September 2014 - May 2016	Intelligent Computing Lappeenranta University of Technology *Master of Science*
September 2007 - April 2012	Computer Science and Engineering University of Technology, Vietnam *Bachelor*
September 2004 - May 2007	High School