Utilization of Occluded Detections and Target Information in Multi-Person Tracking

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc.

Daniel Bernhard Stadler

aus Bruchsal

Tag der mündlichen Prüfung:	04.11.2024
Erster Gutachter:	Prof. Dr.-Ing. habil. Jürgen Beyerer
Zweiter Gutachter:	Prof. Dr.-Ing. Peter Eisert

Abstract

Multi-person tracking is a fundamental task in computer vision with various applications such as surveillance, autonomous driving, and sports analysis. The goal is to localize and identify all persons in each frame of a video sequence. This allows to track persons in safety-critical areas, predict the movement of pedestrians in road traffic, or calculate running statistics at soccer games. The majority of methods follows the tracking-by-detection paradigm dividing the tracking problem into the two subtasks detection and association. For the generated detections, motion and appearance cues are typically extracted to solve the association task of joining detections from the same targets to tracks. This thesis shows that existing tracking approaches from the literature exploit this available information in an insufficient way. Consequently, a novel tracking framework is introduced that improves both the utilization of available detections as well as the fusion mechanism of motion and appearance information in the association.

Most tracking errors occur in crowds, where missed detections due to occlusion complicate the association task. To improve the performance in such situations, an adapted non-maximum suppression is proposed, which allows to include detections under severe occlusion in the association process that were discarded by previous tracking approaches. Two different techniques are suggested to leverage the additional set of heavily-occluded detections. The first one integrates these in a second association stage, where they are matched to the remaining unassigned tracks from the first stage. The second approach utilizes the available track information to identify track clusters with missing detections and incorporates the heavily-occluded detections in these areas locally. Both techniques not only enhance detection recall under strong occlusion but also simplify the association task by reducing the number of missing detections and thus eliminating ambiguities in the assignment of detections to tracks.

Next to the usage of detections, the way of leveraging the available information on appearance and motion of targets plays a key role for the association accuracy. In this thesis, existing fusion mechanisms for motion and appearance information are evaluated within a common base framework, allowing a thorough and fair comparison for the first time, and weaknesses of the prevailing approaches are elaborated. Building on this, novel combined distance measures for a better utilization of motion and appearance information in the association are introduced that significantly outperform previous variants.

To prevent the start of ghost tracks from duplicate detections in crowded areas, an occlusion-aware initialization strategy is suggested. It derives knowledge about the neighborhood of unassigned detections from the available track information to identify and discard duplicates. Moreover, a lightweight model for compensating potential camera motion is presented, which is of great importance for applications with non-static cameras. The proposed modules are combined into a novel framework that surpasses the state of the art in established multi-person tracking benchmarks. This achievement is mainly due to a better use of available information in the tracking process, since the same models are adopted for detection as well as extraction of appearance and motion information as in the competing trackers.

Additionally, several optimizations are made to accelerate the computationally complex multi-person tracking framework, including the application of an efficient model for extracting appearance information, the use of a high-performance library for neural network inference, and parallelization. Without a significant loss of performance, the resulting system runs in real time while being capable of tracking hundreds of targets simultaneously. This is despite the fact that it includes all important components like appearance model or camera motion compensation, which are not used by many methods from the literature in order to achieve a low runtime.

Kurzfassung

Das Multi-Personen-Tracking ist eine grundlegende Aufgabe der Computer Vision mit verschiedenen Anwendungen wie etwa Überwachung, autonomes Fahren und Sportanalyse. Das Ziel ist es, alle Personen in jedem Bild einer Videosequenz zu lokalisieren und zu identifizieren. Dies ermöglicht es, Personen in sicherheitskritischen Bereichen zu verfolgen, die Bewegung von Fußgängern im Straßenverkehr vorherzusagen oder Laufstatistiken bei Fußballspielen zu berechnen. Die meisten Methoden folgen dem *Tracking-by-Detection*-Paradigma, bei dem das Trackingproblem in die beiden Teilaufgaben Detektion und Assoziation aufgeteilt wird. Für die generierten Detektionen werden typischerweise Bewegungs- und Erscheinungsmerkmale extrahiert, um die Assoziationsaufgabe zu lösen, die darin besteht, Detektionen von gleichen Objekten zu Tracks zusammenzufügen. Diese Arbeit zeigt, dass bestehende Trackingansätze aus der Literatur diese verfügbaren Informationen auf unzureichende Art und Weise ausnutzen. Daher wird ein neuartiges Trackingframework eingeführt, das sowohl die Nutzung der verfügbaren Detektionen als auch den Fusionsmechanismus von Bewegungs- und Erscheinungsinformationen bei der Assoziation verbessert.

Die meisten Trackingfehler treten in Menschenmengen auf, wo fehlende Detektionen aufgrund von Verdeckungen die Assoziationsaufgabe erschweren. Um die Performanz in solchen Situationen zu verbessern, wird eine angepasste *Non-Maximum Suppression* vorgeschlagen, die es ermöglicht, stark verdeckte Detektionen in den Assoziationsprozess einzubeziehen, die von bisherigen Trackingansätzen verworfen wurden. Zwei verschiedene Techniken werden eingeführt, um die zusätzliche Menge an stark verdeckten Detektionen zu nutzen. Die erste Methode integriert diese in einer zweiten Assoziationsstufe, in der sie mit den verbleibenden, ungematchten Tracks aus der ersten Stufe verglichen werden. Der zweite Ansatz verwendet die verfügbaren Trackinformationen, um Trackcluster mit fehlenden Detektionen zu identifizieren und die stark verdeckten Detektionen in diesen Bereichen lokal einzubeziehen. Beide Techniken erhöhen nicht nur die Detektionssensitivität bei starker Verdeckung, sondern vereinfachen auch die Assoziationsaufgabe indem sie die Anzahl der fehlenden Detektionen verringern und so Mehrdeutigkeiten bei der Zuordnung von Detektionen zu Tracks beseitigen.

Neben der Verwendung von Detektionen spielt die Art und Weise, wie die verfügbaren Informationen über Erscheinung und Bewegung von Personen genutzt werden, eine Schlüsselrolle für die Assoziationsgenauigkeit. In dieser Arbeit werden existierende Fusionsmechanismen für Bewegungs- und Erscheinungsformationen innerhalb eines gemeinsamen Basisverfahrens evaluiert, was einen umfassenden und fairen Vergleich erstmalig ermöglicht, und Schwächen der vorherrschenden Ansätze werden herausgearbeitet. Darauf aufbauend werden neuartige kombinierte Distanzmaße zur besseren Ausnutzung von Bewegungs- und Erscheinungsinformationen in der Assoziation vorgestellt, die bisherige Varianten deutlich übertreffen.

Um die Entstehung von Geistertracks durch Mehrfachdetektionen in Bereichen hoher Personendichte zu verhindern, wird eine verdeckungsbewusste Initialisierungsstrategie vorgeschlagen. Sie leitet Wissen über die Nachbarschaft von ungematchten Detektionen aus den verfügbaren Trackinformationen ab, um Mehrfachdetektionen zu identifizieren und zu verwerfen. Darüber hinaus wird ein leichtgewichtiges Modell zur Kompensation möglicher Kamerabewegungen vorgestellt, das für Anwendungen mit nicht statischen Kameras von großer Bedeutung ist. Die eingeführten Module werden zu einem neuartigen System kombiniert, das den Stand der Technik in etablierten Benchmarks des Multi-Personen-Trackings übertrifft. Diese Leistung ist vor allem auf eine bessere Nutzung der verfügbaren Informationen im Trackingprozess zurückzuführen, da die gleichen Modelle für die Detektion sowie die Extraktion von Erscheinungs- und Bewegungsinformationen wie in den konkurrierenden Trackern verwendet werden.

Außerdem werden mehrere Optimierungen vorgenommen, um das rechenintensive Framework für das Multi-Personen-Tracking zu beschleunigen. Dazu gehören die Anwendung eines effizienten Modells für die Extraktion von Erscheinungsinformationen, die Verwendung einer leistungsstarken Bibliothek für die Inferenz neuronaler Netze und Parallelisierung. Das resultierende System läuft ohne nennenswerte Leistungseinbußen in Echtzeit und kann hunderte von Personen gleichzeitig tracken. Und das, obwohl es alle wichtigen Komponenten wie Erscheinungsmodell oder Kamerabewegungskompensation enthält, die von vielen Methoden aus der Literatur nicht verwendet werden, um eine geringe Laufzeit zu erreichen.

Contents

1 Introduction

This thesis aims at the development of a real-time-capable framework for tracking multiple persons in videos. Derived from an exhaustive study of applicable modules, the focus lies on leveraging available information that has been overlooked by previous approaches or used in an insufficient way. This includes, for instance, the utilization of severely-occluded detections in the tracking process and an improved fusion of motion and appearance information of targets.

The conducted research is motivated in Section 1.1, where various applications of multi-person tracking (MPT) are presented. Moreover, deficiencies of the prevailing approaches and how these are addressed in this thesis are outlined. Section 1.2 gives a comprehensive overview of typical challenges that emerge when developing an MPT system. After that, the main contributions of this thesis are summarized in Section 1.3, and finally, Section 1.4 describes the structure for the remainder of this thesis.

1.1 Motivation

Tracking multiple persons in videos has a wide variety of practical applications. In recent years, the increasing availability of cameras in both public and private spaces has lead to a growing demand for efficient MPT solutions. Next to the major fields of surveillance [Elh21] and autonomous driving [Yur20], MPT plays an important role in smart city applications [Luc21], robot navigation [Fai19], human computer interaction [Laz17], customer behavior analysis [Qui16], and sports analysis [Cui23].

In the surveillance context, the main goal of tracking is to help ensuring the safety of people. For instance, crowd behavior analysis in mass events allows to identify gatherings with critical person densities and makes early interventions possible [Kok16]. Moreover, person tracking can support security personnel and law enforcement authorities with the prevention, mitigation, or prosecution of suspects in cases of break-in, theft, vandalism, assault, or other crimes, and can also help to find missing people [Cyb23]. During a pandemic, social distancing could be enforced by tracking pedestrians in public areas [Pun21]. Furthermore, video surveillance is also applied frequently in private spaces. Companies use cameras to guard their premises or to make sure that certain sections are only entered by authorized personnel. In addition, employees at safety-critical workstations or construction sites can be monitored to detect accidents quickly [Tei09].

The safety of people has also a high priority in automated driving. Besides other traffic participants, pedestrians have to be detected and tracked robustly such that the self-driving vehicle can react to persons crossing the street or children jumping onto the road. MPT is also a key component for collision avoidance and path planning in other robotic domains. Areas in which robots already automatically navigate through spaces with people include production facilities, where driverless transport systems are used to carry material from one location to another [BMW24], or museums, where assistance robots are employed to guide visitors [Tya21]. In the future, autonomous robots will be employed more frequently in our daily lives [Tra22] and most require the ability to recognize and track the persons in their surroundings.

The growing number of available surveillance cameras is not only beneficial for safety and security reasons but also allows a variety of smart city applications that improve the quality of living and sustainability. For example, by analyzing the traffic flow of vehicles and pedestrians, traffic light switching can be optimized and emissions reduced [Cho19]. Traffic statistics can further be leveraged for infrastructure planning or improving bus timetables. Moreover, smart cameras are able to identify traffic rule violations [Pat22] or people trespassing on restricted footpaths and railways [Zha22b].

Retail is another area in which the analysis of human behavior is relevant. Customer flows in stores reveal information on where products should be strategically placed in order to increase sales. Additionally, tracking individual customers together with the products they select paves the way for cashier-less checkout systems [Pol18] and thus has the potential to save manpower and costs in the future.

MPT is also commercially used for sports analysis. Tracking all players as well as the match ball in football, basketball, volleyball, etc. can be used to assess running distances, team formations, or pass statistics [Cui23]. With an ongoing increasing commercialization of sports, automatic tactical analyses become more and more important for coaches, scouts, and the reporting media.

While such analyses can be performed offline without critical time constraints, the majority of applications requires to process video streams in real time. Especially in the surveillance domain—to which this thesis pays particular attention—a fast response to safety-critical events is essential. However, conventional video surveillance systems typically require human operators who monitor multiple streams simultaneously, which is both costly and prone to fatigue-related errors [Rob19]. Moreover, the complexity of some scenes, for example, with a high person density, cannot be overlooked in detail solely by a human. Therefore, automatic and efficient solutions to assist human operators are required. Note that an assistive system can also improve privacy, for instance, if it automatically processes the captured data and only shows relevant incidents to the human operator [Ros23].

To meet the growing demand for intelligent surveillance systems, this thesis aims at the development of a real-time-capable MPT framework. A strong baseline method with well-established tracking components for detection, re-identification (REID), and motion modeling from the literature is built. Following the widely used *tracking-by-detection* (TBD) paradigm [Aha22, Bew16, Cao23, Woj17, Zha22c]—where detections are generated independently in each frame of the video and then associated to tracks based on target cues—the base framework provides a high flexibility and is a good starting point for further developments.

A comprehensive analysis of all tracking components performed in this thesis shows that prevailing approaches do not utilize important information or only in an ineffective manner. For instance, detections with severe overlaps are typically filtered by the standard non-maximum suppression (NMS) technique and not further used in the tracking process, which makes the association task under strong occlusion very difficult. To address this issue, an adapted NMS method is introduced that allows to use such heavily-occluded detections for the first time in an MPT framework. Building on this, two different association strategies are proposed that leverage the additional detections to resolve ambiguities in the detection-to-track assignments under severe occlusion, where usually most tracking errors occur.

In challenging scenarios, an effective utilization of available target information is key for a high tracking accuracy. Many methods from the literature combine motion and appearance cues to enhance the association performance [Aha22, Wan20, Woj17]. However, a direct comparison of these fusion approaches is not reasonable as they are applied within various tracking frameworks. Moreover, a detailed understanding of their working mechanism is often missing due to the lack of ablative experiments. To close this gap, an in-depth investigation of existing fusion techniques is conducted within the base framework of this thesis, which enables a fair comparison for the first time and reveals several weaknesses of the examined techniques. Based on the findings, new distance measures for a combined motion- and appearance-based association are proposed that significantly outperform previous fusion approaches on multiple datasets.

After the association, unassigned detections are used to initialize tracks. While the decision whether a remaining detection belongs to a newly arrived target or is a false positive (FP) is usually based on the confidence or continuity of the detection in consecutive frames, available context information is not considered. Differently, a novel initialization strategy is suggested that takes information about already tracked targets in the neighborhood of unassigned detections into account, which improves the initialization accuracy under occlusion and prevents the start of *ghost* tracks.

The introduced tracking modules are combined to a sophisticated MPT framework, which also comprises a method for camera motion compensation (CMC). Since the most frequently applied CMC method in the MPT literature [Eva08] is computationally too expensive for a real-time application, an alternative approach is suggested that achieves equally good results while being about 30 times faster. The proposed MPT framework sets a new state of the art (SOTA) on the standard benchmarks, and its generalization capabilities are demonstrated through extensive experiments on several datasets.

Containing multiple modules, e.g., for detection, modelling of target motion and appearance, or CMC, an MPT framework has a high computational complexity. Many top-performing methods are thus not real-time capable, and the runtime has been widely overlooked in the MPT literature. Therefore, a detailed runtime analysis of the proposed framework is conducted, and bottlenecks that prevent a fast execution are identified. Based on that, several optimizations are suggested, for instance, employing a more efficient model for appearance extraction and using a library for accelerated neural network inference. The optimized MPT system is up to 40 times faster compared to the baseline without sacrificing its performance. To the best of the author's knowledge, the proposed framework is the first MPT approach that achieves SOTA results and contains all important tracking components, while being able to track hundreds of targets in real time on standard hardware.

1.2 Challenges

MPT is a difficult task due to several aspects. Challenges arise from various scene characteristics, different camera positions and properties, environmental conditions, or hardware constraints, to name a few. In Section 1.2.1, challenges resulting from the image acquisition are described. After that, MPT-specific difficulties are discussed in Section 1.2.1 and lastly, challenges of real-world applications are exploited in Section 1.2.3. The discussion is oriented on a surveillance scenario but a multitude of aspects is also valid for other application domains.

1.2.1 Challenges Resulting from Image Acquisition

The position of cameras, their characteristics, and also environmental conditions have influence on the success of an MPT system. Some of the challenges that typically emerge from the imaging point of view are presented below.

Camera placement: Depending on the area to be monitored, the camera's field of view, and infrastructural constraints, the position and orientation of the installed camera can differ significantly as displayed in Figure 1.1. Mostly, images are acquired from an oblique view, which leads to several challenges, in particular, a high variability in object size, frequent obstacle–person or person–person occlusions, and small object sizes in the background. Beyond that, if pan–tilt–zoom (PTZ) cameras are used or the camera is moving (most commonly in non-surveillance applications), the introduced camera motion has to be compensated in the tracking algorithm, which can be difficult and computationally expensive. Thus, one part of this thesis deals with an efficient compensation of camera motion.

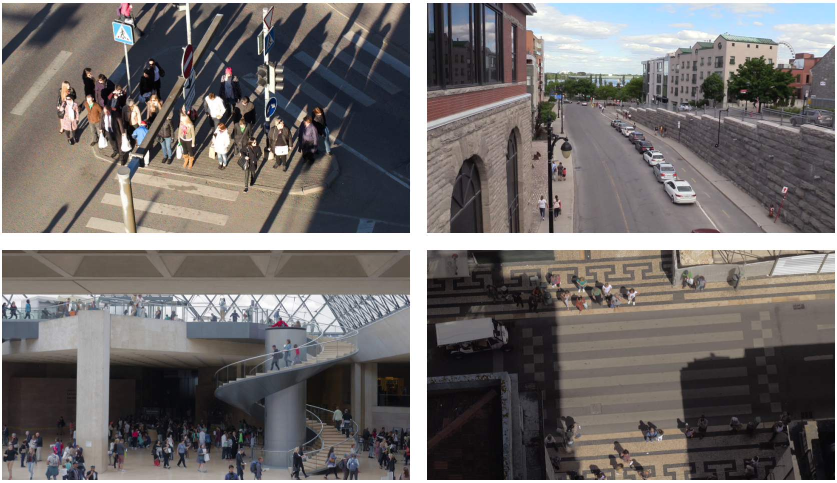


Figure 1.1: A large diversity in camera views leads to a high variety in person size and appearance on the image. Moreover, an oblique view results in frequent occlusions of targets. The images are taken from the PersonPath22 dataset [Shu22].

Light and weather conditions: The appearance of an object in a camera image depends not only on the viewing angle and the pose of the object, but also on environmental factors that often cannot be controlled. Different illuminations in indoor scenes and various times of day as well as weather conditions (sunny, cloudy, rainy, snowy, etc.) outdoors lead to a high variability in the appearance of captured images and persons as shown in Figure 1.2. This particularly renders the development of robust models for person detection and REID challenging.

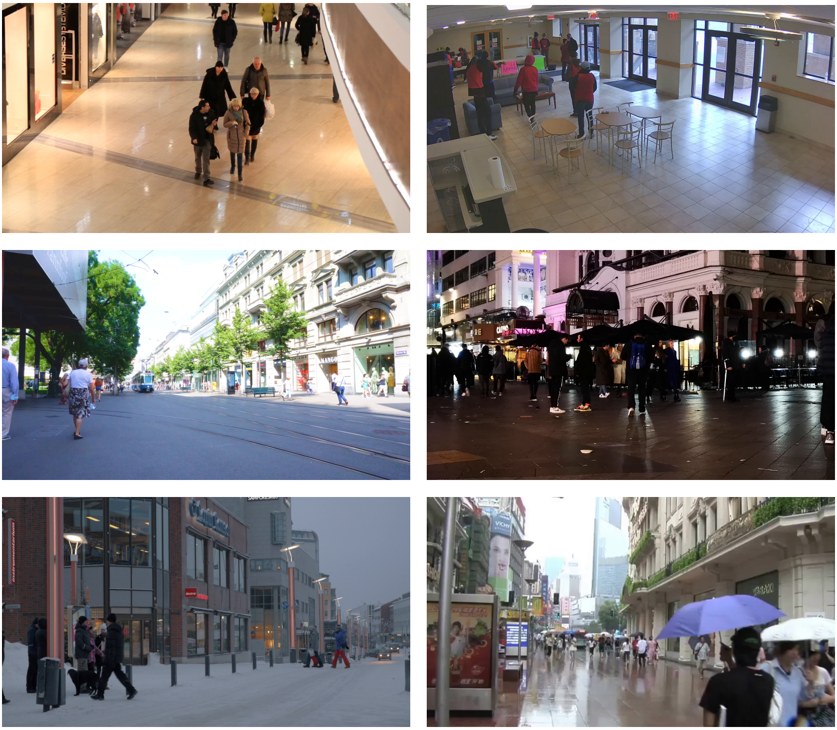
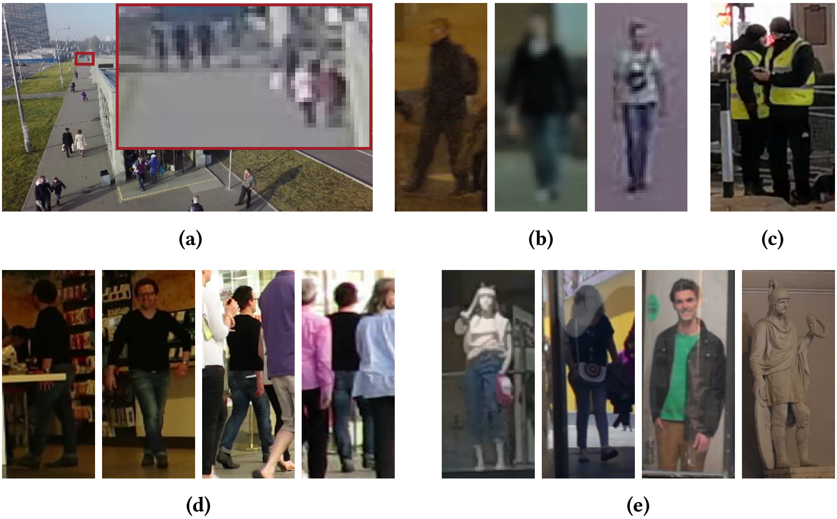


Figure 1.2: Different lighting indoors and various times of day as well as weather conditions outdoors make the development of robust MPT methods difficult. The images are taken from the PersonPath22 dataset [Shu22].

Low image resolution: To save costs, cheap cameras with a small spatial resolution are often employed. Furthermore, infrared cameras, which can be leveraged to allow surveillance under poor lighting conditions or during night time, usually have a lower resolution than RGB cameras that capture the visual spectrum. But even with high-resolution cameras, the oblique view leads to small objects in the image background being only depicted by a few pixels as illustrated in Figure 1.3a. Detecting and tracking such targets with limited image information is a hard task in MPT.


(a) (b) (c)

(d) (e)

Figure 1.3: Various challenges arising in MPT include (a) low resolution (in the background), (b) image degradation like noise, blur, or compression artifacts, (c) similar looking persons, (d) appearance changes of targets, and (e) distractions such as mannequins or reflections, posters, and statues of persons. The images are taken from the MOT17 [Mil16], PersonPath22 [Shu22], and SOMPT22 [Sim23] dataset.

Limited frame rate: The average frame rate of surveillance cameras lies at around 15 Hz [IPV21], but cameras with lower rates are also in operation. A small frame rate is unfavorable for MPT, since the positions of persons on the image can change considerably between consecutive frames, especially when people move fast. In such scenarios, the predicted positions by motion models can be imprecise so that further target cues like appearance might be necessary to achieve a high tracking accuracy. Yielding complementary information, much emphasis is put on an effective fusion of motion and appearance cues in this thesis.

Image degradation: The quality of a captured image can be deteriorated by multiple factors such as noise, blur, or compression. Image noise manifests in random changes of pixel colors or intensities and mainly originates from disturbances in the sensor. Blurred images or regions lead to a loss of detail with reduced sharpness, for instance. There are several causes for blur as motion of camera or targets and atmospheric effects. The compression of images to save memory or bandwidth can also lead to degradation, e.g., undesirable artifacts. Examples of degraded person images are depicted in Figure 1.3b.

1.2.2 Multi-Person Tracking-Specific Challenges

While many challenges of the previous section apply to most computer vision tasks, this section focuses on difficulties that are more specific to the MPT problem, i.e., occlusion, similar looking persons, appearance changes, irregular motions, distractions, and small objects.

Occlusion: One of the most severe challenges in MPT is occlusion. As mentioned before, persons are frequently concealed fully or partially by obstacles or other targets due to the oblique view of the camera (Figures 1.1 and 1.2), which easily leads to missing detections. In crowded scenes, duplicate detections also occur because it is difficult for the detector to reason about the boundaries of the targets. Both false negatives (FNs) and FPs complicate the association task of assigning detections to tracks. Moreover, imprecise bounding boxes under occlusion can deteriorate the motion states of the targets and lead to inaccurate motion predictions. Besides that, extracted appearance information from occluded bounding boxes can be misleading as a significant amount of image information might belong to another object. All these aspects contribute to a high risk for identity switches (IDSWs) when tracking through occlusions. Therefore, specific strategies for handling occlusion are required, which is one of the main focuses of this thesis.

Similar looking persons: People with similar appearance can quickly be confused by the tracking algorithm, especially when they are in close proximity so that positional information might also be ambiguous as in Figure 1.3c. Such situations occur frequently in public spaces with a high person density including pedestrian zones or shopping malls. The problem becomes even more severe at mass events like football games, for example, where fans wear similar clothes making it very hard to distinguish them by appearance information. Thus, leveraging accurate models for the motion prediction of targets is of high importance, which is also considered in this thesis.

Appearance changes: On top of a potentially small inter-class appearance variability considering each individual as a single class, a large intra-class variability can further complicate the MPT task. Concretely, the appearance of a person moving through the camera's field of view alters on the image due to changing distance and angle to the camera, shadows or other illumination factors, and occlusion by obstacles or other targets. This makes identity preservation over a long time period challenging. An example of a person with severe appearance changes due to different illuminations, perspectives, and occlusions is depicted in Figure 1.3d.

Irregular motions: Next to tracking persons in sports [Cui23] or while dancing [Sun22], complex motion patterns can also be observed in common surveillance scenarios. Pedestrians react to oncoming persons to avoid collisions and typically keep a social distance to others. Besides complicated target motions, potential camera movements can introduce further irregularities, which makes the modeling of motion in MPT a non-trivial task.

Distractions: Objects or entities that look similar to persons but are no targets to be tracked are referred to as *distractions*. Examples are mannequins in clothing stores, reflections of people in shop windows, advertising posters of persons, and statues, which are depicted in Figure 1.3e. Such distractions often lead to FP detections that impede the association task and thus can lead to further tracking errors.

Small objects: In contrast to distractions that cause FPs, small objects are easily overlooked by the detection model leading to FNs that also harm the tracking performance. As already noticed, small persons in the image background occur frequently in MPT due to the oblique view of the camera (Figure 1.3a). Even if the detector has recognized such a small person, the corresponding bounding box might contain only a few pixels so that the extraction of valuable appearance information for the association task is problematic. Moreover, tracking small targets becomes especially challenging under severe camera motion, as the image regions of the same person in consecutive frames might not overlap, even if the camera frame rate is high.

1.2.3 Real-World Challenges

In addition to the so-far treated aspects, challenges arise when applying an MPT system in a practical application including real-time processing, hardware limitations, and requirements on the generalization ability.

Real-time processing: To solve the MPT task, different subproblems as the detection of targets, motion modeling, and association of detections to tracks have to be tackled. Since not all subproblems can be treated simultaneously, multiple modules have to be executed one after another. Furthermore, additional models for extracting important appearance information and for compensating potential camera motion are often employed to improve the tracking accuracy. As a consequence, the resulting MPT system is computationally expensive, which makes a real-time processing challenging. In fact, many SOTA methods in well-established MPT benchmarks [Den20, Mil16] are not real-time capable and compromises between runtime and accuracy have to be made in order to enable an application in the real world. In this thesis, a multitude of runtime optimizations is performed to achieve real-time capability while maintaining the high accuracy of the proposed MPT framework.

Hardware limitations: The hardware used in a real MPT system is limited in several respects, mainly due to budget restrictions. For instance, cheap cameras might have a low image resolution or poor sensor quality. Moreover, the available compute power of employed central processing units (CPUs) and graphics processing units (GPUs) imposes constraints on the computational complexity of MPT algorithms. This holds true particularly if execution should be performed on edge devices that often have very limited resources.

Generalization ability: Current MPT frameworks usually comprise deep learning models for person detection or appearance feature extraction. The performance of such data-driven approaches in practice highly depends on the extent of *domain shift* between training and inference, i.e., how much the data distribution in the real-world application differs from the data the models have been trained on. Due to the large diversity in image appearance and tracking scenarios, the development of a robust real-world MPT system is a challenging task.

1.3 Contributions

The major goal of this thesis is the development of a real-time-capable MPT framework with focus on real-world surveillance scenarios. In the following, the main contributions are summarized.

- A baseline method with well-established components from the MPT literature is built [Sta23a] including strong modules for detection, REID, and motion modeling [Sta22b]. This base framework is not only representative for many existing MPT approaches, but is also a good starting point for further developments. A thorough analysis of all modules conducted in this thesis reveals several shortcomings of prevailing MPT methods: the non-utilization of occluded detections, a poor fusion of motion and appearance information, and a track initialization that does not take the presence of already tracked targets into account.

- An adapted NMS [Sta21c] is proposed that keeps a set of severely-occluded detections next to the standard set and allows to leverage the additional detections in the association for the first time. Building on that, two different approaches to utilize the occluded detections are introduced in this thesis. First, a two-stage association strategy termed BYTEv2 [Sta23c] matches them with the unassigned tracks from the first association stage, while preventing FP detections from starting incorrect tracks. Second, the tracking with clusters (TWC) approach [Sta21c] leverages information about track positions to identify regions with missing detections, in which occluded detections are integrated. Both methods simplify the association task by reducing the number of missing detections under occlusion, where naturally most tracking errors occur.

- Since a sophisticated combination of motion and appearance information is important for a high association accuracy [Sta20, Sta21a], popular fusion strategies are integrated into a common base framework in this thesis, which allows a fair comparison for the first time. Furthermore, an in-depth analysis of their working mechanisms is conducted, and several weaknesses of the existing methods are identified. Based on the findings, novel distance measures for a combined motion- and appearance-based association are introduced [Sta23b, Sta23d] that significantly outperform previous fusion approaches.
- Several works have shown that a dedicated treatment of occlusion is beneficial in MPT [Spe21, Sta21b]. To improve the track initialization process under occlusion, an occlusion-aware initialization (OAI) is proposed [Sta23b]. In contrast to existing initialization techniques that rely mostly on the detection confidence to assess whether an unassigned detection belongs to a newly arrived target, the OAI additionally takes the surroundings of a detection into account by leveraging information from already tracked targets. This prevents the start of ghost tracks under occlusion and thus avoids further tracking errors.
- A fast CMC approach based on the matching of image keypoint descriptors is introduced for the use in MPT [Sta23c]. Extensive experiments with various keypoint detectors and descriptor extractors are conducted. The resulting method performs slightly better than the most frequently used CMC method in the MPT literature while being up to 30 times as fast and thus allows real-time processing.

- Both the single components and the overall tracking framework of this thesis are evaluated on multiple MPT datasets under various settings. The superiority compared to previous methods and a strong generalization ability are shown through comprehensive experiments. Moreover, the proposed framework sets a new SOTA on the standard MPT benchmarks, which indicates a good suitability for real-world applications.
- A detailed runtime analysis of the proposed tracking framework is carried out, and bottlenecks that hinder a fast computation are identified. Consequently, several optimizations are introduced such as leveraging a more efficient REID model, executing modules in parallel, and utilizing a high-performance library for neural network inference. The optimized framework runs up to 40 times faster while maintaining SOTA performance. To the best of the author's knowledge, the optimized framework of this thesis is the first SOTA approach for MPT that comprises all important modules as REID or CMC and is able to track hundreds of persons in real-time on standard hardware.

1.4 Thesis Outline

The rest of this thesis is structured as follows. Related literature is thoroughly reviewed in Chapter 2. A categorization of MPT approaches is made shedding light on various aspects of the task, and many popular works are discussed. Moreover, research areas closely related to MPT are briefly treated. After that, the general concept of the proposed framework of this thesis is introduced in Chapter 3. Then, Chapter 4 describes the experimental setup for assessing the performance of tracking methods including a description of the utilized datasets, evaluation measures, and protocols. Chapter 5 is devoted to the base framework that is built using a multitude of established modules from the MPT literature. A detailed analysis is conducted revealing several shortcomings of existing MPT approaches, which is the basis for further developments covered in Chapter 6. First, the focus is put on an improved utilization of detections and tracks under occlusion. This is achieved by two different approaches that integrate occluded detections filtered by an adapted NMS into the tracking process and the OAI that leverages positional information to improve the track initialization in crowded scenes. The second part of Chapter 6 focuses on various fusion approaches for motion and appearance cues, and novel distance measures for an improved combination of the two information sources are suggested. Together with a new method for CMC, the proposed modules are combined to a sophisticated tracking framework, which is evaluated and compared with the SOTA. Afterwards, Chapter 7 deals with the runtime optimization of the proposed MPT framework. It is demonstrated that, despite the high complexity of the overall system, some modifications lead to an optimized version that can run in real-time while maintaining the high accuracy. Finally, Chapter 8 draws conclusions from this thesis and outlines ideas for future work.

2 Related Work

The goal of this thesis is to develop an MPT framework that focuses on an improved utilization of available information in the tracking process. Diverse approaches to leverage the useful information for solving the MPT task can be found in the literature and are presented in this chapter. First, Section 2.1 gives a comprehensive overview of existing MPT methods. After that, research areas closely related to MPT are briefly discussed in Section 2.2.

2.1 Multi-Person Tracking

An MPT system typically comprises multiple modules that are responsible for solving different subtasks, e.g., detection, motion modeling, appearance extraction, affinity computation, and track management. Thus, several approaches have evolved that focus on various aspects of the MPT problem. Possibilities to categorize MPT methods are given in Section 2.1.1, before the most common types of approaches are thoroughly presented in Section 2.1.2. Finally, a short summary is given, and the proposed framework of this thesis is put in context with existing approaches in Section 2.1.3.

2.1.1 Categorization

As differences cannot only be observed within specific tracking modules, but the whole structure and the working mechanism of an MPT system can vary significantly, several categorizations of existing approaches are feasible. This can also be seen when studying multiple survey papers that classify MPT methods differently [Agr24, Bas22, Cia20, Du24, Luo21]. Moreover, the focus of MPT researchers has changed over the years. For instance, it is found by analysis of the SOTA in [Lea17] that before 2015, the attention lied on optimizing the data association problem, whereby the task of linking detections to tracks was often solved with graph-based methods. After 2015, the focus shifted towards the design of strong appearance cues for assessing the similarity of different objects. As predicted by the authors of the study, powerful deep learning-based methods have been used more frequently for extracting appearance features in the following years, replacing traditional hand-crafted solutions. Furthermore, new tracking paradigms have evolved with the development of new deep learning architectures such as transformers [Vas17].

Before the most common approaches are presented in the next section—grouped mainly based on the employed architecture—the following characterization of MPT approaches is made on a more abstract level. One can categorize MPT methods based on

- whether they work *online* or *offline*,
- the *tracking paradigm* they follow,
- which *tracking cues* (motion, appearance, etc.) are leveraged,
- whether they use *hand-crafted* solutions or rely on *deep learning*, or
- whether tracking is preformed in *2D* or *3D*.

Note that this list does not claim to be complete but gives a good overview of several aspects of MPT. In the following, the mentioned distinctions are briefly discussed, and advantages as well as disadvantages of the respective categories are elaborated.

Online vs. Offline

Offline trackers [Cet23, Lar24, Tan17, Wan22a, Zha08], also referred to as *batch* methods, process the video as a whole, whereas online trackers [Ber19, Bew16, Woj17, Zha21, Zha22c] treat each frame of a video sequentially, having only access to the information up to the current time step. Thus, online methods are also termed *causal*. Being able to look into the future of a specific time step, offline methods are more robust to occlusions and theoretically can achieve better performance because more information is available. However, they suffer from a high computational complexity, which can grow exponentially with the length of the processed video. Offline methods typically build a graph, in which the nodes and edges represent detections and possible links between detections, respectively. To solve the MPT task, different graph optimization strategies have been proposed including min-cost flow algorithms [Ber11, Pir11, Zha08], multicut approaches [Tan15, Tan16, Tan17], or generalized maximum multi-clique [Deh15]. Despite offline trackers having the potential to achieve higher accuracy, the current SOTA is dominated by online methods. One reason for this is that offline trackers are not suitable for real-time applications, which is why the focus of this thesis lies on online approaches.

Tracking Paradigm

Most MPT methods follow the TBD paradigm performing detection and association independently [Aha22, Bew16, Cao23, Woj17, Zha22c]. A person detector is applied on each frame of the video and afterwards, the detections are linked to build tracks. The TBD paradigm can be used both in online and offline approaches and provides a high flexibility. For instance, different cues such as motion [Bew16, Qin23, Zha22c] or appearance [Aha22, Du23, Woj17] can be leveraged in the association. Furthermore, it is easily possible to change individual components of a TBD-based system, for example, detector or motion model, to adapt to different application domains or runtime constraints.

Another popular paradigm termed *tracking-by-regression* (TBR) was first introduced in [Ber19] and adopted by following works [Bra20, Hor20, Liu20, Xu20]. Tracked boxes are taken as input to the regression network head of the two-stage detector Faster R-CNN [Ren17] in consecutive frames, updating the position and size of tracked targets continuously. Thus, the association task is solved implicitly and the design of specific measures to assess the similarity of person detections becomes obsolete. While this simplifies the association procedure, it precludes the inclusion of meaningful cues such as appearance features. TBR is not limited to Faster R-CNN or other two-stage detectors. In [Zho20], the single-stage point-based detector CenterNet [Zho19b] is extended by a network branch that is trained to predict the motion of targets from two consecutive input images, supported by a heatmap that encodes the track center positions from the previous time step.

Besides TBR, the *tracking-by-attention* (TBA) [Mei22, Sun21a, Zen22, Zhu23] paradigm aims at integrating the detection and tracking task more tightly. The transformer architecture [Vas17]—originally introduced to solve natural language processing tasks but meanwhile adopted for many computer vision problems—is used to leverage the concept of *attention* on global frame-level features. Some TBA approaches are trained end-to-end and learn to reason about track initialization, identity, and spatio-temporal trajectories purely from image data. Exemplary TBA methods will be presented in more detail in the next section. While learning the whole MPT task in an end-to-end manner removes the need for separate appearance or motion extraction modules and association strategies, the high flexibility of TBD-based methods is lost. Consequently, integrating specific knowledge is difficult in TBA approaches. As this thesis strives for improving the available information in MPT, the flexible TBD paradigm is used as a basis for the proposed tracking framework. This design choice is supported by the observation that the vast majority of current SOTA trackers still follows the well-established TBD paradigm.

Used Tracking Cues

Different object cues can be leveraged for assessing the similarity of detections to form tracks, and one can classify MPT methods based on what kind of cues they apply. A large amount of trackers is motion-based, i.e., target positions in the next frame are predicted by a motion model and are compared with the positions of detections in that frame. Typically, a Kalman filter [Kal60] is employed and either the Mahalanobis distance [Wan20, Woj17, Yi24] or intersection over union (IoU) [Aha22, Bew16, Zha22c] is used as similarity measure. Further examples for motion cues can be found in [Bra20], where geometric features are designed based on position and size information, and in [Pan20], where even the motion direction is considered. Two reasons for the widespread use of motion information in MPT are the generally fast computation and the usually high frame rates of videos. This property ensures that the change of target positions between consecutive frames is limited, leading to quite accurate motion predictions even with simple linear motion models.

Another highly important tracking cue is the appearance of targets. While earlier works used hand-crafted features to describe the appearance of detected persons [Ben11, Bre09, Cho15, Oku04], nowadays, deep learning-based models are applied [Du23, Sun21b, Wan21, Woj17, Zha21]. A common strategy is to adopt networks from the person REID field and use them as module for extracting appearance features within an MPT framework [Aha22, Du23, Woj17]. As these networks can be computationally expensive, it is hard for appearance-based MPT approaches to achieve real-time capability.

Several works use both motion and appearance information, and different fusion strategies have been proposed [Aha22, Du23, Wan20]. This thesis also leverages both information sources and strives at combining them in the best possible way [Sta23b, Sta23d]. Further cues exploited in MPT are human pose information [Bao21, Tan17] and features that model the relation or interaction of targets [Liu20, Sad17, Wen16].

Hand-Crafted vs. Deep Learning-Based

The distinction of approaches in hand-crafted solutions vs. deep learning-based techniques can be made for various subtasks of the MPT problem. As mentioned earlier, traditional cues like optical flow [Cho15, Iza12, Rod09], color histograms [Ben11, Iza12, Oku04], and histogram of oriented gradients [Bre09, Cho12, Iza12] have been largely replaced by appearance features generated from neural networks [Aha22, Du23, Woj17]. Likewise, traditional detectors used in MPT [Dol14, Fel04, Yan14] have been substituted by deep learning models [Ge21, Ren17, Zho19b]. While some methods also try to learn the association task from data [Chu19, Xu20] including transformer-based approaches that are able to train the whole MPT pipeline end-to-end [Mei22, Sun21a, Zen22], the majority of current SOTA methods relies on hand-crafted association and track management strategies [Aha22, Cao23, Jun24, Yan23, Zha22c]. The same holds true for the motion modeling task that is still most often solved using the traditional Kalman filter [Du23, Liu23, Men23, Yi24, Zha22c]. This thesis also combines deep learning models with hand-crafted solutions allowing an easy integration of specific knowledge and to apply established heuristics, which is difficult in end-to-end-trainable approaches.

2D vs. 3D

Another possibility of differentiating MPT methods is whether tracking is performed in image space (using only the projected 2D data) or in world space (when reliable 3D information is available). In the following, these approaches are denoted by 2D vs. 3D based on the originally available information. Note, however, that even with available 3D data, tracking may be performed on a 2D ground plane—yet, in this case, Euclidean world coordinates can be applied in contrast to the perspective-projected image coordinates in cases when only 2D information is available.

The decision whether 2D or 3D trackers are applied is closely related to the types of available input data and thus also to the application area. In the surveillance domain, standard monocular RGB cameras are typically employed leading mostly to 2D solutions. Contrarily, in autonomous driving, the availability of various kinds of sensors including stereo cameras, LiDAR, and radar allows for robust tracking in three dimensions [Fro18, Lei08, Zha19]. Another type of 3D approaches are such based on RGB-D cameras (with depth information), which are surveyed in [Cam17]. Moreover, multi-camera networks with an overlapping field of view make the exploitation of 3D information possible [Bri19]. There also exist works that extract 3D cues like depth from 2D monocular images [Den22, Khu21, Sha18b] to support the association task. While gaining additional information when considering tracking in 3D—which surely can be beneficial for the overall tracking performance—several challenges arise including a higher computational complexity and a more expensive acquisition of labeled data compared to 2D methods. Therefore, this thesis develops an MPT framework that works on 2D images of monocular RGB cameras.

2.1.2 Common Approaches

The previous section already gave a high-level classification of existing MPT methods and has discussed advantages and disadvantages of different tracking paradigms. Next, the related literature is reviewed from a more technical perspective. Over the years, various groups of approaches have emerged for several reasons. For instance, joint detection and embedding (JDE) methods have been developed as efficient alternatives for previous separate detection and embedding (SDE) approaches, which suffer from a high computational complexity due to the consecutive execution of two networks. Moreover, newer deep learning architectures such as transformers or graph neural networks (GNNs) have influenced the field of MPT. In the following, some representatives of different groups of MPT approaches are presented. This includes works with *focus on motion modeling*, *SDE* approaches, *JDE* networks, as well as methods with *GNNs* and *transformers*. Other MPT approaches with less relevance in recent years (e.g., probabilistic methods) are only discussed briefly.

Focus on Motion Modeling

The by far most common approach to model the motion of targets in MPT is to apply a linear Kalman filter [Kal60] using a constant velocity assumption [Aha22, Bew16, Cao23, Du23, Gao24, Han22, Mag23, Woj17, Yi24, Zha22c]. Different formulations for the motion state vector are employed. For example, SORT [Bew16] models the motion state as a 7-tuple comprising x- and y-coordinates, the bounding box size (area) and aspect ratio as well as the derivatives of x, y, and size. In DeepSORT [Woj17], the motion vector is eight-dimensional including x, y, aspect ratio and height of bounding box as well as the respective derivatives. Instead of considering the aspect ratio or size, box width and height are directly modeled in [Aha22]. Based on the classical Kalman filter formulation, several modifications have been proposed in the literature. The Noise Scale Adaptive (NSA) Kalman filter [Du21] leverages the confidence score of a detection to dynamically adapt the measurement covariance matrix in the Kalman filter update step to account for the uncertainty of a detection. In [Gao24], a learned localization score is used instead of the classification-based confidence to scale the covariance matrix. ConfTrack [Jun24] recognizes that the NSA formulation can only reduce the measurement noise in case of confident detections and introduces an amplifying factor that allows also to increase the noise when confidence scores are low. Moreover, ConfTrack keeps the box size fixed during prediction to prevent the effect of unstable box size variation of occluded targets. This is inspired by the height preservation (HP) module—which was introduced in a previous work from the author of this thesis [Sta22b]—that enforces a constant bounding box height in the Kalman filter prediction step. In [Cao23], it is found that the linear motion assumption results in a square-order error accumulation w.r.t. time when observations are missing in consecutive frames due to occlusion. The authors suggest to focus more on the high-quality observations of modern detectors and propose an observation-centric Kalman filter.

As long as precise detections to update the target states and a high camera frame rate are available, Kalman filter-based motion modeling is very accurate in *static* cameras. However, one encounters *moving* cameras in some applications, such that the linear motion assumption is strongly violated. To counteract this, several approaches for CMC have been applied in the MPT literature. Most works [Ber19, Du23, Han22, He21] use the enhanced correlation coefficient (ECC) maximization technique from [Eva08]. Since this image registration method is computationally expensive and hinders a real-time capability of the whole MPT system, a fast CMC alternative based on the matching of efficient ORB descriptors [Rub11] is proposed in this thesis [Sta23c]. Another fast solution to consider potential camera motion can be found in [Nas23]. The positional differences of assigned track–detection pairs are averaged in both spatial directions and then subtracted from the track boxes, before the association is repeated.

Both the target motion from a Kalman filter and the estimated camera motion are leveraged in MAT [Han22] to dynamically set the time interval a lost track can be re-activated. Similarly, the severity of camera motion is utilized to determine whether a lost track should be terminated in a previous work from the author of this thesis [Sta23c]. In contrast to the aforementioned approaches that model motion in the image plane, UCMCTrack [Yi24] employs a Kalman filter on the ground plane, transforming detections in the image plane with a projection matrix derived from the (estimated) camera parameters. Camera motion is then explicitly modeled as process noise in their Kalman filter formulation, removing the need for computing a transformation for each frame of the video.

Other related MPT works with a focus on motion modeling can be found in [Zho20], where the target displacements in two consecutive frames are learned from image data, and in [Qin23], where a transformer-based module for modeling the interaction of different targets is leveraged in the motion prediction. Relying on an accurate motion model, many works confine themselves to a simple association using IoU as similarity measure and thus achieve high inference speeds [Bew16, Boc17, Cao23]. In this thesis, a strong motion model, based on the combination of NSA Kalman filter [Du21] and the HP module [Sta22b], is also employed. However, appearance information is additionally leveraged following an SDE approach. This type of MPT category is presented in the following.

Separate Detection and Embedding

Besides motion cues, the appearance of targets plays an important role in MPT. SDE methods comprise—next to an object detector and a motion module—an additional model for extracting appearance information. A deep neural network from the person REID community is often adopted, which takes the cropped image region of a person detection as input and computes a high-dimensional *embedding* vector as output. One of the most popular SDE frameworks is DeepSORT [Woj17], which applies a wide residual network [Zag16] for appearance feature extraction. The cosine distance between such feature vectors is then used as similarity measure, while the Mahalanobis distance between Kalman filter-predicted tracks and detections is used for motion-based gating, i.e., matching restrictions are applied based on spatial constraints. Another contribution of DeepSORT is a matching cascade that gives priority to tracks that have been observed more recently when associating the current detections. A few years later, the further development StrongSORT [Du23] made various modifications to the DeepSORT framework. Most importantly, a more recent and stronger appearance model termed BoT (*bag of tricks*) [Luo19] is employed, and instead of storing the appearance embeddings of a track in a large feature bank (100 in DeepSORT), only one embedding is maintained that is updated in an exponential moving average (EMA) manner. This not only improves performance but also reduces processing time significantly. Moreover, instead of using the motion-based Mahalanobis distance only for gating, it is combined with the appearance-based cosine distance by a weighted sum. However, it is found in this thesis that IoU-based measures perform better than the Mahalanobis distance, and improved measures for a combined motion- and appearance-based association are proposed [Sta23b, Sta23d]. Further improvements of StrongSORT include the application of the NSA Kalman filter [Du21] and the incorporation of the ECC maximization technique [Eva08] for CMC. Additionally, the matching cascade from DeepSORT is discarded as it limits the tracking accuracy due to unnecessary prior constraints [Du23].

Apart from that, performing the association in multiple stages can also be beneficial for the tracking performance. A popular two-stage matching scheme has been proposed in ByteTrack [Zha22c]. This two-stage approach splits the set of detections in such with low confidence and high confidence by a simple threshold. The low-confidence detections—that have not been used in previous MPT approaches—are matched with the remaining unassigned tracks from the first association stage, which significantly enhances recall. One important finding from [Zha22c] is that appearance features extracted from low-score detection boxes are unreliable as they often contain severe occlusion or motion blur. Thus, appearance information is only used in the first association stage. The two-step matching strategy has been adopted by following works [Aha22, Ren23], and other trackers also introduce multi-stage matching schemes [Jun24, Liu23, Men23, Yan23]. This thesis proposes a two-stage association strategy termed BYTEv2, a further development of the BYTE association [Zha22c], that enables the utilization of previously discarded heavily-occluded detections in the tracking process [Sta23c].

Several methods [Aha22, Du23, Jun24, Mag23] use the already mentioned BoT REID network [Luo19] showing the high importance of strong methods for appearance feature extraction. BoT-SORT [Aha22] combines appearance cosine distance with IoU distance of track–detection pairs by taking the minimum and introduces additional constraints for both cues to enhance the association accuracy. Deep OC-SORT [Mag23] integrates appearance information into the OC-SORT framework [Cao23] and additionally proposes an adaptive weighting of appearance features based on the diversity of embeddings. Another recent SDE approach termed FineTrack [Ren23] enhances the discrimination ability of appearance embeddings by learning part-based features, which is especially beneficial under occlusion.

Joint Detection and Embedding

Due to the typically high computational complexity of SDE approaches, several works have been proposed that perform detection and appearance embedding extraction in a single JDE network. The first of this kind is TrackR-CNN [Voi19], which extends the detection and segmentation network Mask R-CNN [He17] by an embedding head for identity association. To integrate more temporal context, image features from multiple input frames are aggregated with 3D convolutions, before the region proposal network and subsequent heads for detection, segmentation, and appearance embedding are applied. Another JDE work based on a two-stage detector is proposed in [Shu20]. Faster R-CNN [Ren17] is—next to an embedding branch—enlarged by a Siamese tracking branch to predict the target positions in consecutive frames, inspired by the single object tracking (SOT) literature. These two-stage approaches still suffer from a large runtime when a high number of targets is present, because the appearance features have to be computed for each person detection separately.

To counteract that, a single-stage detection approach, based on the Feature Pyramid Network architecture [Lin17a], is equipped with an embedding branch in [Wan20]. Such an efficient single-stage JDE architecture has been taken over by many following works [Lia22, Lu20, Ren24, Wu21, You23, Zha21]. For instance, RetinaTrack [Lu20] uses an additional embedding head on top of the popular RetinaNet detector [Lin17b].

Following works [Lia22, Yu23, Zha21] focus on the problem that learning detection and appearance features are two competing tasks: While features from different targets (but the same object category, namely person) shall be similar for the detection task, they should be dissimilar for the REID task. FairMOT [Zha21] is the first work that focuses on the fairness of learning both tasks simultaneously. The authors find that anchor boxes from common detectors lead to many ambiguities during the training of appearance features as multiple anchors can correspond to the same identity and also, several identities can correspond to a single anchor. To solve this issue, FairMOT builds upon the anchor-free detector CenterNet [Zho19b] and extracts appearance features at the object centers. On top of this, RelationTrack [Yu23] introduces a module that decouples the learned representations into detection-specific and REID-specific features. Task-dependent representations are also the focus in [Lia22], where a cross-correlation network is suggested for learning particularities and commonalities of features for detection and REID. Further improvements are achieved with a scale-aware attention network, which makes the features more robust to changes in object size, and the upgrade of the underlying detection architecture from YOLOv3 [Red18] to YOLOv5 [Joc20]. Other works with feature enhancement modules include [Wan21, Wu21]. In [Wu21], a tracking offset is learned to propagate features from previous frames, which allows for an improved temporal aggregation, and in [Wan21], a GNN is trained to model the relation of targets, which supports the learning of discriminative features for detection and REID.

One of the best-performing JDE approaches in MPT benchmarks is termed UTM [You23]. Its main component is an identity-aware feature enhancement module that uses embeddings from already tracked targets to boost current detection and appearance features by various attention mechanisms. Further components include a learnable memory aggregation module for identity embeddings and an association branch for identity matching. Despite its high complexity, UTM falls behind the best MPT frameworks in the current SOTA, which is dominated by SDE methods or approaches with focus on motion modeling. This indicates that existing JDE solutions cannot (yet) completely solve the problem of the competing tasks of detection and REID within a single network, which is why this thesis follows a SDE approach.

MPT with Graph Neural Networks

GNNs are special types of artificial neural networks for processing data that is structured as a graph. As mentioned earlier, the MPT task can be formulated as a graph problem, where detections correspond to the nodes of a graph and the edges between detections from different time steps represent possible links. One of the best-known earlier methods that applies GNNs for MPT is called MPNTrack [Bra20]. The authors introduce a so-called message passing network (MPN) that is able to propagate information encoded in the nodes and edges throughout the graph via learnable message passing steps. Initially, appearance features extracted by a convolutional neural network (CNN) and geometric cues are used for the node and edge embeddings. After several message passing steps that iteratively aggregate neighboring embeddings, a trained multi-layer perceptron is applied to classify whether the edges are *active*, i.e., the adjacent nodes (detections) belong to the same target. The offline method MPNTrack allows to leverage higher-order information and reason globally over the set of detections and thus shows a good performance in identity preservation over longer time periods.

Another usage of GNNs can be found in [Dai21]. First, an iterative graph clustering method is used to generate an over-complete and diverse set of trajectory proposals. Then, a GNN is trained to predict scores that measure the correctness of the trajectory proposals.

A major problem of graph-based approaches including GNN-based MPT methods is the increasing computational complexity and memory requirements when considering large time spans that lead to very large graphs (even when some pruning strategies are applied). This issue is addressed with a novel hierarchical approach termed SUSHI [Cet23]. The input video is divided into short clips, in which the GNN from [Bra20] is used to generate short trajectories. Then, the same GNN is applied again, whereby the nodes correspond to tracks instead of detections and the appearance features of a track are computed by averaging its detection embeddings, which increases robustness. Furthermore, the motion information encoded in the tracks' embeddings is beneficial for the following association tasks. Repeating the process multiple times leads to tracks with increasing length until the timespan covers the whole input video.

All aforementioned works are offline approaches but there exist also online trackers that make use of GNNs. For example, the JDE method GSDT [Wan21] enhances the features for appearance and detection by leveraging object relations modeled with a GNN. In [You23], an identity association branch is proposed that builds a graph between so-far tracked targets and new detections and uses cross-graph message passing for feature aggregation. Another approach can be found in [Qin23], where a GNN is used within a motion prediction module to fuse interaction cues of targets. Due to their strong capability of modeling relations, an increasing usage of GNNs in the recent MPT literature can be observed.

MPT with Transformers

Next to GNNs, transformer [Vas17] is another popular deep learning architecture leveraged in some MPT methods. Several works build upon the detection transformer (DETR) [Car20], an encoder-decoder structure in that the encoder extracts image information and the decoder finds the best correlations of the encoded image features and so-called *object queries* to perform the detection task. To extend DETR (or its further development Deformable DETR [Zhu21]) to a tracking framework, the concept of *track queries* has been introduced in [Mei22, Sun21a]. Different from object queries, which are responsible for initializing new tracks in transformer-based tracking approaches, track queries encode the appearance and location of already tracked targets and are transferred to consecutive frames and updated continuously. TransTrack [Sun21a] learns two parallel decoders for object and track queries that yield a set of detection and track boxes, respectively. Those boxes are then associated with the Hungarian algorithm [Kuh55] using IoU as similarity measure. In contrast, TrackFormer [Mei22] trains only a single decoder that processes object and track queries jointly and learns the association task implicitly in an end-to-end manner. This also holds true for MOTR [Zen22], which extends the framework by a temporal aggregation network that is responsible for fusing information of previous track queries saved in a query memory bank. Another approach that focuses on temporal information encoded in track queries can be found in [Zhu23], where a temporal attention module for improving the propagation of features to consecutive frames is proposed. Furthermore, two spatial attention modules are introduced for object-to-object and object-to-input attention that allow to reason globally about the relation of the tracked targets.

While queries are decoded into bounding boxes in the aforementioned methods, TransCenter [Xu23] decodes them into center points motivated by point-based detectors like CenterNet [Zho19b]. This change in object representation can be advantageous in crowded scenes, where bounding boxes of targets strongly overlap. A follow-up work finds that transformer-based trackers struggle to bridge longer occlusions, since the spatial information encoded in the track queries can prevent a REID if the target location has changed too much [Gal22]. Thus, the authors integrate a separate REID model and further standard MPT modules, e.g., a Kalman filter, into the TransCenter framework.

Apart from end-to-end-learnable trackers, the transformer architecture is also employed within other tracking paradigms to make use of the global reasoning capabilities of the attention mechanism. For instance, RelationTrack [Yu23] proposes a guided transformer encoder to enhance the discrimination power of learned appearance features within a JDE network. Another example is FineTrack [Ren23], which uses a parallel structure inspired by the multi-head self-attention of the transformer architecture to focus on different parts of the target and learn fine-grained representations.

Further MPT Approaches

This section shortly covers MPT approaches that are of less relevance for this thesis, but were frequently used in earlier works. This includes many variants of probabilistic trackers that are often based on probability hypothesis density filters [Gra12, Lin06] or multiple hypothesis tracking approaches [Kim15, Rei79]. More recently, it was found in [Lar24] that target motion and false detection characteristics in visual tracking differ significantly from radar or sonar tracking, where many probabilistic approaches come from. They propose a novel probabilistic framework that accounts for these differences and achieves competitive results compared to the SOTA.

Being omnipresent in the SOT literature, Siamese networks [Bro93] have also been used for MPT [Jin20, Shu20, Shu21]. These networks comprise two sibling branches with shared weights generating output vectors from two different inputs and are trained with a similarity measure. In the context of MPT, input patches from the same target shall yield a high similarity, whereas images from different targets should have a small similarity. In this sense, they are closely related to REID networks that are frequently used in SDE approaches.

Another type of architecture that has been employed frequently in MPT are recurrent neural networks (RNNs). Building connections between some network outputs to inputs of the next time step, they are suitable to process sequential data and encode temporal relations. RNNs have been used to model the change in target motion or appearance over time [Fan18, Tok21, Wan18a] or even to perform data association and track management [Mil17]. Recently, RNNs have been largely replaced by the transformer architecture [Vas17], which can better capture long-range dependencies that are hard to model in recurrent structures.

2.1.3 Summary and Discussion

This thesis aims at improving the utilization of available information in the tracking process. The focus lies on leveraging the generated detections in the association most effectively and enhancing the use of object cues such as appearance and motion when assigning detections to tracks. For this, the TBD paradigm is followed as it provides a high flexibility allowing to tune individual tracking components separately, which is not possible with end-to-end-learnable MPT approaches based on transformers.

Most tracking frameworks using GNNs are offline methods that process the input video as a whole and are thus not suitable for real-time applications such as autonomous driving or surveillance tasks. Differently, the goal of this thesis is to design an efficient online framework that can be applied for various MPT applications.

Trackers that fuse motion and appearance information obviously have a higher potential than trackers that restrict themselves to one information source. Therefore, this thesis exploits both cues and follows a SDE approach as the related literature indicates that JDE approaches fall behind in terms of performance due to the competition between detection and REID when learned in a single network.

In this thesis, a base framework is built that uses well-established TBD components like the NSA Kalman filter [Du21] for advanced motion modeling and the BoT REID model [Luo19] for appearance feature extraction. Due to these design choices, the base framework is representative for many popular MPT approaches and is a good starting point for further developments. Since efficiency is an often overlooked issue in the MPT literature—many SOTA approaches have a high computational complexity or do not report their runtime [Lar24, Mag23, Men23, Qin23]—this thesis also focuses on the real-time capability of the whole MPT framework. Next to the limited availability of 3D annotations in MPT datasets, the real-time constraints are a reason why tracking is performed in the 2D image plane. To summarize, the goal of this thesis is to develop a real-time-capable TBD-based SDE framework for 2D MPT with a focus on an improved utilization of detections and target information.

2.2 Related Research Areas

The previous section gave a thorough overview of the MPT literature and has placed this thesis in context with existing approaches. In the following, further related research areas are briefly described, emphasizing the connection to the MPT task or the main differences. The presented research areas are multiple object tracking (MOT), SOT, person REID, multi-camera MPT, person detection, and pedestrian trajectory forecasting.

Multiple Object Tracking

A more general task compared to MPT in the sense that various object categories can be treated simultaneously is known as MOT. For example, different vehicles such as cars, buses, or bicycles have to be tracked next to pedestrians in autonomous driving applications. While considering particularities of the several object classes like different motion behaviors can be helpful, the basic ideas do not differ from MPT approaches. From a research perspective, persons are the most interesting objects for MOT with many challenges such as complex motion patterns and strong occlusions in crowded scenes. This is one of the reasons why more than 70 % of the MOT research deals with persons according to the survey in [Luo21].

Single Object Tracking

SOT, also referred to as visual object tracking, is the task of following a predefined but arbitrary object throughout a video sequence. The main focus in the SOT literature lies in designing powerful motion or appearance models to account for the occurring challenges such as illumination variations, scale changes, or out-of-plane rotations [Luo21]. Despite that, SOT is an overall simpler task compared to MOT, which requires to cope with objects appearing or leaving the scene, maintaining multiple identities, and dealing with interactions among targets. Note that a MOT framework can also be used to track single objects of interest, provided that the categories lie within the set of known objects of the detector. On the other hand, some works exist that use SOT methods within a MOT framework [Chu17, Fen22, Zhe21].

Person Re-Identification

The goal of person REID is to find occurrences of a query person of interest in a multi-camera network. REID models therefore have to learn discriminative appearance features to distinguish between different people, which is also helpful in the MPT task. Consequently, REID models are frequently used in MPT frameworks [Aha22, Du23, Jun24, Mag23, Woj17] forming the group of SDE approaches, which already have been discussed in Section 2.1.2.

Aside from that, the general conditions differ between the MPT and REID task. For instance, the appearance of the same person can change dramatically in the REID context due to various viewing angles, illumination conditions, or characteristics of the separate cameras. On the other hand, the appearance information is mostly used for a shorter time period in MPT such that only slight changes occur. Therefore, it is common to train the REID models utilized in MPT on single-camera sequences from MPT datasets [Aha22, Jun24, Mag23]. Note that the query person in the REID context is not restricted to image or video data but can also be represented as a textual description. Other differences can be found in [Ye22], which gives a comprehensive survey of the person REID literature.

Multi-Camera Multi-Person Tracking

When combining the tasks of single-camera MPT and person REID, one basically gets to the multi-camera MPT problem, which is usually solved in two steps. First, single camera tracking is performed in each camera of the network. Second, tracks from different cameras are merged on the basis of appearance features extracted by a REID model. In case of overlapping field of views, positional information is also of high relevance. An important strategy to enhance the inter-camera association accuracy is to enforce constraints derived from the camera network topology. For example, temporal restrictions can prevent wrong matches across cameras with far distance if the time delta between two tracklets is infeasibly small. Such matching constraints play also an important role in single-camera MPT. Several strategies exist to reduce the feasible amount of combinations when associating detections to tracks including motion-based gating mechanisms [Woj17] and prohibiting matches with too different appearance [Aha22] or distance to the camera [Liu23]. For a thorough overview of multi-camera multi-object tracking methods, the interested reader is referred to [Amo23].

Person Detection

Person detection—a special case of general object detection, which aims at recognizing various types of objects—is an indispensable subtask of MPT. Several person-specific detectors that focus on crowded scenes, where most detection (and tracking) errors emerge, have been proposed in the literature [Chu20, Ruk21, Zha18, Zha23, Zhe22]. In a previous work, the author of this thesis has shown that the usage of such specialized person detectors in MPT can lead to a higher tracking accuracy compared to standard detectors [Sta21d]. Consequently, newer approaches [Zha23, Zhe22] that achieve SOTA results on crowded person detection datasets like CrowdHuman (CH) [Sha18a] have the potential to improve the MPT performance.

However, the predominant amount of recent MPT approaches adopts the general purpose detector YOLOX [Ge21] that has been trained on recognizing the person category only. While this may limit the achievable MPT performance as detection and tracking accuracy are closely related, using the same detector enhances the comparability of MPT methods.

Recently, more focus is put on how the generated detections are used in the tracking process. For a long time, it has been the standard practice to discard detections with low confidence scores. Since the introduction of ByteTrack [Zha22c], which incorporates low-confidence detections in a second matching stage, several works treat detections with various characteristics in terms of confidence or localization quality differently [Gao24, Men23]. This thesis also strives for an improved utilization of the available detections, especially in crowded scenes.

Pedestrian Trajectory Forecasting

The goal of pedestrian trajectory forecasting is to predict the future trajectories of multiple pedestrians given their past trajectories. While MPT methods also predict the position of tracked targets to the next time step, or even for a short time interval, the forecasting literature focuses on longer predictions. Another difference is that datasets for trajectory forecasting often comprise videos captured from a bird's eye view. An overview of existing approaches can be found in [Rud20], for instance. Compared to the research areas covered before, the field of trajectory forecasting is less relevant to the MPT task. To date, only a few MPT works have integrated forecasting methods in order to enhance the long-term tracking performance [Den22, Kes22].

3 Concept

This thesis aims at the development of a real-time-capable MPT system with focus on an improved utilization of *available information* that is either not taken into account by existing MPT works or used in a suboptimal way. The available information comprises, for example, the set of generated detections, extracted motion and appearance cues of targets, and context information derived from track or detection statistics.

To solve the MPT task, most methods from the literature follow the TBD paradigm [Aha22, Bew16, Cao23, Woj17, Zha22c] separating the problem mainly into two subtasks: detection and association. This makes a TBD-based framework very flexible such that further components like a REID model for appearance feature extraction or a module to compensate potential camera motion can be easily incorporated to further increase the tracking accuracy. For these reasons, this thesis also employs a TBD-based approach.

In Figure 3.1 the general pipeline of a common TBD-based framework is illustrated and briefly summarized as follows. A detector is applied on the current image yielding a set of raw detections that are filtered with an NMS to remove duplicate detections. While detections under heavy occlusion are discarded, a REID model extracts appearance embeddings for the *normal* detections. After that, those are matched with the tracks from the previous time step, which have been propagated by a motion model. This association is typically based on appearance and/or motion information. Finally, the matched tracks are updated with the corresponding detections, and the unassigned detections start new tracks in the initialization. Note that parts where this thesis focuses on are highlighted green in Figure 3.1.

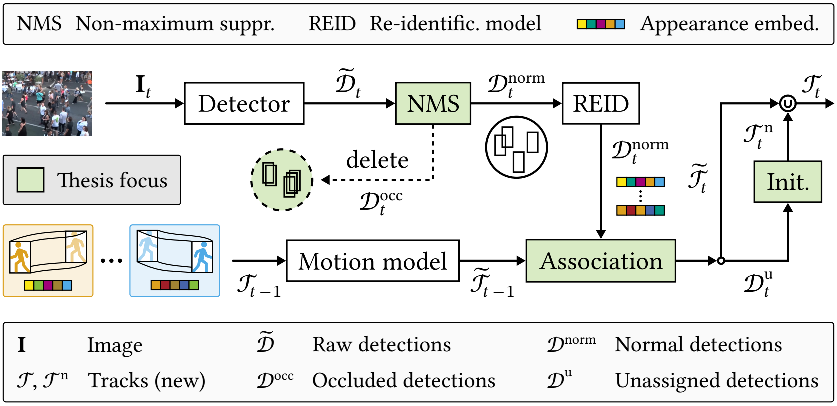

NMS Non-maximum suppr.
REID Re-identific. model
Appearance embed.
$\mathbf{I}_t$
Detector
$\tilde{\mathcal{D}}_t$
NMS
$\mathcal{D}_t^{\text{norm}}$
REID
$\mathcal{T}_t$
Thesis focus
delete
$\mathcal{D}_t^{\text{occ}}$
$\mathcal{D}_t^{\text{norm}}$
$\tilde{\mathcal{T}}_t$
$\mathcal{T}_t^{\text{n}}$
Init.
…
$\mathcal{T}_{t-1}$
Motion model
$\tilde{\mathcal{T}}_{t-1}$
Association
$\mathcal{D}_t^{\text{u}}$
$\mathbf{I}$ Image
$\tilde{\mathcal{D}}$ Raw detections
$\mathcal{D}^{\text{norm}}$ Normal detections
$\mathcal{T}, \mathcal{T}^{\text{n}}$ Tracks (new)
$\mathcal{D}^{\text{occ}}$ Occluded detections
$\mathcal{D}^{\text{u}}$ Unassigned detections

Figure 3.1: Common TBD framework. This thesis focuses on leveraging occluded detections that are typically discarded, enhancing the fusion of motion and appearance information in the association, and utilizing context knowledge in the track initialization.

While MPT works usually try to enhance the overall performance by improving single tracking components like detection, REID, or motion model, how to leverage the available information from standard components most effectively is an underexplored research question. Based on a comprehensive analysis of common weaknesses of current TBD-based MPT methods, several new strategies to improve the utilization of available information in the tracking process are introduced. The developed approaches mainly pursue the following three goals:

- better usage of generated detections in the association,
- enhanced fusion of motion and appearance information, and
- utilization of context knowledge.

Before giving an overview of the proposed tracking framework, the various contributions to reach the aforementioned goals are briefly described.

Better Usage of Generated Detections in the Association

The predictions from typical detectors applied in MPT provide a set of bounding boxes with corresponding confidence scores. For a long time until the introduction of the BYTE association [Zha22c], it has been common to discard all detections with a confidence below a threshold and consider only high-confidence detections in the tracking process. Leveraging the true positives (TPs) from the set of low-confidence detections while introducing no or little FPs in the BYTE association, both recall and precision are increased.

This idea of utilizing so-far discarded detections is expended further in this thesis, and a novel technique to enlarge the set of used detections in the association is proposed. Besides low-confidence detections, detections under strong occlusion are typically filtered by the standard NMS to remove duplicate predictions of the detector. In order to incorporate these occluded detections into the tracking process, an adapted NMS is introduced that outputs a set of occluded detections next to the normal detection set [Sta21c]. A previous work of the author has found that many tracking errors under strong occlusion occur because the task of assigning available detections to so-far tracked targets becomes ambiguous due to missing detections [Sta22b]. Enabling the use of additional detections under heavy occlusion resolves such ambiguities and thus simplifies the association task, besides increasing the detection recall. This has also been shown by another work of the author [Sta21d], where the application of crowd-specific detectors in MPT has improved the tracking performance by increasing detection recall in crowded scenes. To the best of the author's knowledge, the proposed adapted NMS is the only approach that enables the utilization of heavily-occluded detections in the association for standard detectors that rely on an NMS as post-processing.

Two novel association strategies that utilize the occluded detections provided by the adapted NMS are suggested in this thesis: BYTEv2 [Sta23c] and TWC [Sta21c]. BYTEv2, as a further development of BYTE, includes the occluded detections next to the low-confidence detections in a second association stage for matching to the unassigned tracks from the first stage. As mentioned before, this not only enhances the detection recall but also simplifies the association task under heavy occlusion and thus improves the overall performance. Besides its application within the tracking framework of this thesis, the effectiveness of the proposed BYTEv2 association has been demonstrated for various trackers [Sta23c].

While the goal of the TWC approach is the same as for BYTEv2—leveraging the set of occluding detections from the adapted NMS in the association—the mode of operation differs. In contrast to implicitly using the occluded detections in a second association stage, the track information is utilized to identify *clusters* with missing detections. In such clusters, the corresponding occluded detections are explicitly introduced to compensate for the missing detections in the normal detection set. An improved performance in combination with a different detector than the one applied in this thesis indicates a good robustness of the TWC approach w.r.t. the detection model used [Sta21c].

Enhanced Fusion of Motion and Appearance Information

The matching of detections to tracks in each frame of a video sequence builds mostly upon motion or appearance information in TBD-based MPT. Although various fusion strategies exist in the literature, a fair comparison of the approaches is often not possible due to the use of different components, e.g., detection, REID, or motion models, within the respective tracking framework. For the first time, a profound comparison of existing fusion strategies for motion- and appearance-based association is made [Sta23a], using a shared base framework to guarantee meaningful results. Weaknesses of the prevailing methods are identified, and it is shown that they do not fully leverage the available information about the motion and appearance of the tracked targets.

Based on the findings, novel distance measures are proposed [Sta23b, Sta23d] that fuse the available information in a better way and thus outperform previous approaches significantly. The combined distance measures for motion and appearance information not only perform well in the base framework but also yield strong results in combination with the BYTEv2 association strategy. Improvements on three different datasets with varying detection quality further demonstrate the generalization ability of the proposed fusion approaches.

Utilization of Context Knowledge

Next to an optimal usage of available information that is clearly evident, like generated detections or appearance features coming from the detector and REID model, respectively, some information is more concealed. Deriving context knowledge about the density of targets, the ambiguity of track–detection assignments, or physical constraints, such additional information can be leveraged in the tracking process and consequently lead to improved performance. For instance, several works have restricted the change of target bounding box size in the prediction step of the motion model because the person size on the image cannot alter much between two consecutive images when the camera frame rate is high. One of the first methods of this kind—the HP module—has been proposed by the author in [Sta22b] and is also utilized in the MPT framework of this thesis. Another example of using context information in the motion modeling can be found in a further work of the author, where the severity of camera motion is utilized to adaptively change the time an inactive track is propagated in order to react to the increasing uncertainty of the motion state [Sta23c].

Especially in crowded scenes, where most tracking errors occur, the introduction of additional information is desirable. It is shown that explicitly modeling the relation between occluding and occluded tracks can improve the tracking performance in crowds [Sta21b]. Moreover, the number of detections and tracks in combination with the computed distances for the association can be leveraged to identify ambiguous situations that are treated specifically [Sta22a, Sta22b]. As mentioned earlier, the number of detections and tracks also plays a role in the proposed TWC approach that recognizes missing detections in crowded areas and incorporates the set of occluded detections from the adapted NMS in the association.

Another module that exploits context knowledge in crowds introduced in this thesis is the OAI [Sta23b]. Confident detections that remain unassigned after the association are usually used to start new tracks. This easily leads to ghost tracks in crowded scenes, where duplicate detections occur as the detector has difficulties to reason about the boundaries of the persons. The OAI utilizes track information to identify unassigned detections with high overlaps to already tracked targets as duplicates and discards them such that the start of ghost tracks is prevented, which in turn avoids further tracking errors.

Overview of the Proposed Tracking Framework

Following the TBD paradigm, this thesis initially develops a base framework comprising frequently used approaches for detection, REID and motion modeling, which allows a fair comparison with the current SOTA. For instance, YOLOX [Ge21] is adopted as detector which has become the standard detection model on the MPT benchmarks MOT17 [Mil16] and MOT20 [Den20] in recent years. The base framework will be explored in detail in Chapter 5. Being representative for many current MPT methods, it is also a good foundation for the developed tracking modules treated in Chapter 6.

Combining the introduced modules for improving the utilization of available information leads to the proposed tracking framework visualized in Figure 3.2 and described in the following. Note that components comprising the main contributions of this thesis are highlighted in green. Moreover, modules that are runtime-optimized in order to make the whole tracking system real-time capable are colored blue.

In each frame of a video stream, the detector generates a redundant set of detections, which is post-processed with the proposed adapted NMS that yields additional occluded detections next to the normal detection set. The latter is further split into low-confidence and high-confidence detections that are treated differently in the association, which also holds true for the occluded detections. For the high-confidence detections, appearance embeddings are extracted with a REID model that are compared with the appearance features of the so-far tracked targets in the association. Since appearance features from occluded and low-confidence detections are unreliable [Sta23c, Zha22c], the REID model is only applied for the high-confidence detections. The other detections are matched to tracks based on motion information modeled by an improved Kalman filter [Sta22b].

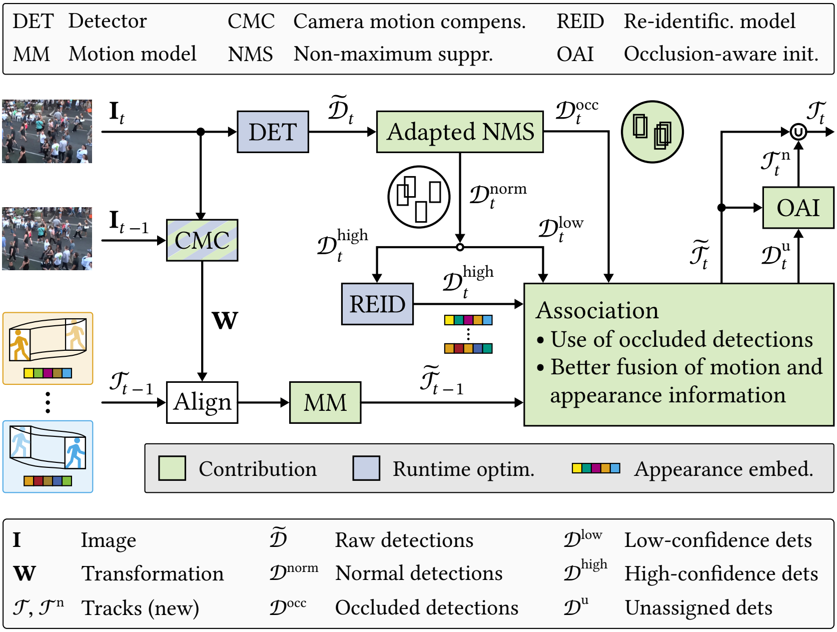

DET Detector
CMC Camera motion compens.
REID Re-identific. model
MM Motion model
NMS Non-maximum suppr.
OAI Occlusion-aware init.
$\mathbf{I}_t$
$\mathbf{I}_{t-1}$
DET
$\tilde{\mathcal{D}}_t$
Adapted NMS
$\mathcal{D}_t^{\text{occ}}$
$\mathcal{T}_t$
$\mathcal{T}_t^{\text{n}}$
OAI
$\mathcal{D}_t^{\text{norm}}$
$\mathcal{D}_t^{\text{low}}$
$\mathcal{D}_t^{\text{high}}$
CMC
$\tilde{\mathcal{T}}_t$
$\mathcal{D}_t^{\text{u}}$
REID
$\mathcal{D}_t^{\text{high}}$
W
Association
• Use of occluded detections
• Better fusion of motion and appearance information
$\mathcal{T}_{t-1}$
Align
MM
$\tilde{\mathcal{T}}_{t-1}$
Contribution
Runtime optim.
Appearance embed.
I Image
W Transformation
$\mathcal{T}, \mathcal{T}^{\text{n}}$ Tracks (new)
$\tilde{\mathcal{D}}$ Raw detections
$\mathcal{D}^{\text{norm}}$ Normal detections
$\mathcal{D}^{\text{occ}}$ Occluded detections
$\mathcal{D}^{\text{low}}$ Low-confidence dets
$\mathcal{D}^{\text{high}}$ High-confidence dets
$\mathcal{D}^{\text{u}}$ Unassigned dets

Figure 3.2: Concept of the proposed tracking framework. The main contributions are the utilization of occluded detections from an adapted NMS, an enhanced fusion of motion and appearance information, and an occlusion-aware track initialization. Moreover, an enhanced motion model is introduced, and an efficient CMC technique is suggested. Together with runtime optimizations of detector and REID model, the real-time capability of the whole system is enabled.

Highlights of the association module of the proposed tracking framework are the usage of occluded detections (BYTEv2, TWC) and the enhanced fusion of motion and appearance information. After the association, the proposed OAI technique is employed to remove unassigned duplicate detections and thus to prevent the start of ghost tracks.

As an additional tracking component, a real-time-capable CMC is introduced [Sta23c], which is an essential module whenever video streams of non-static cameras have to be processed. Given the current and the last image as input, it computes a transformation matrix that is used to align the tracks from the previous time step with the current frame, before the motion model for target movement is applied.

Figure 3.2 indicates that an MPT system has a large computational complexity due to its multiple components. To the best of the author's knowledge, none of the top-performing methods on the common MPT benchmarks comprises all of the computationally expensive but important modules, i.e., detection, REID, and CMC model, and is able to run in real-time. To close this gap, a runtime-optimized variant of the proposed system is developed in Chapter 7. The TBD paradigm allows to exchange the REID model with a lightweight alternative that has a much better runtime–accuracy trade-off as the frequently applied variant from the base framework. With further optimizations including the acceleration of detector and REID model using a library specialized for neural network inference as well as the parallel computation of detection and CMC, a significant speed-up of the framework is achieved without sacrificing its SOTA performance. The final optimized tracking system is capable of tracking 500 targets at a rate of 19 frames per second (FPS) on standard hardware.

4 Experimental Setup

This chapter presents the setup for experiments and evaluating the proposed tracking framework. Datasets used are introduced in Section 4.1, followed by evaluation measures in Section 4.2. Finally, Section 4.3 describes different protocols for evaluating the tracking performance under various conditions.

4.1 Datasets

First, datasets for the actual task of MPT are described in Section 4.1.1. After that, Section 4.1.2 introduces further datasets that are leveraged in the training process of the applied person detection model.

4.1.1 Multi-Person Tracking Datasets

Due to the wide range of applications, there exist numerous datasets for MPT in the literature. An overview of some popular MPT datasets is given in Table 4.1. Large differences can be observed in the dataset size w.r.t. the number of videos, total length, resolution of the images, and number of identities (IDs) to be tracked. As a side note, most of the videos are sampled at 20–30 FPS and some datasets comprise various frame rates.

Next to general datasets that include videos under different scenarios, there exist datasets with special focus on some application domains. These include DanceTrack [Sun22] concentrating on dance videos and SportsMOT [Cui23] comprising sequences of soccer, basketball, and volleyball games. Other popular datasets are HiEve (Human in Events) [Lin23], with focus on complex events such as earthquake escape, fighting, or subway getting on/off, and

Table 4.1: Overview of some popular MPT datasets. The four datasets at the bottom of the table, i.e., MOT17, MOT20, PersonPath22, and SOMPT22, are used in this thesis.

Dataset	Videos	Length	Resolution	IDs	Focus
MOT15 [Lea15]	22	17 min	640×480–1920×1080	1,221	general
MOTSynth [Fab21]	768	1,152 min	1920×1080	45,273	synthetic
P-DESTRE [Kum21]	75	59 min	3840×2160	1,894	drone-based
DanceTrack [Sun22]	100	88 min	1280×720–1920×1080	990	dancing
HiEve [Lin23]	32	33 min	352×258–1920×1080	2,687	events
SportsMOT [Cui23]	240	100 min	1280×720	3,401	sports
MOT17 [Mil16]	14	8 min	640×480–1920×1080	1,331	general
MOT20 [Den20]	8	9 min	1173×880–1920×1080	3,833	crowds
PersonPath22 [Shu22]	236	139 min	720×480–3840×2160	11,970	general
SOMPT22 [Sim23]	14	12 min	1280×720–1920×1080	997	surveillance

P-DESTRE [Kum21], which contains aerial videos captured by a small drone. The largest publicly available dataset is the synthetic MOTSynth [Fab21], which was created using a rendering game engine and is especially of interest when investigating the synthetic-to-real domain gap in the context of MPT.

As of 2024, the two most used datasets for evaluating MPT algorithms are still MOT17 [Mil16] and MOT20 [Den20], two representatives of the MOTChallenge[1]. Being the gold standard for comparing MPT methods with the SOTA, those are also used in this thesis. In addition, PersonPath22 (PP22) [Shu22] as largest real-world MPT dataset is leveraged, which allows to assess the generalization capabilities of the proposed tracking components. To evaluate the MPT performance in a specific application domain, the surveillance-based dataset SOMPT22 [Sim23] is utilized. These four datasets are described next in more detail.

[1] The MOTChallenge is a popular collection of MOT datasets found at https://motchallenge.net (accessed on July 16, 2024).

MOT17

Comprising the same videos as its predecessor MOT16 [Mil16], MOT17 can be regarded as an updated version with improved annotations and an additional provision of public detection sets. Despite the motivation of enabling a fair comparison of tracking methods using the same detections, the public detections are outdated and evaluation is more frequently performed under the *private* protocol that allows to apply an arbitrary person detector.

The 14 videos, 7 in the *train* and *test* split each, comprise a great variety containing day and night scenes with different viewing angles and levels of target density. Moreover, they are captured indoor and outdoor and include both static scenes and scenes with camera motion. The mean (maximum) number of persons per image in the train split is 23 (56), and the average time a person is visible is 7.5 s. A total of 300,373 person ground truth (GT) boxes are annotated in the whole dataset and the frame rate ranges between 14 and 30 FPS. Figure 4.1 shows exemplary frames of the MOT17 dataset.

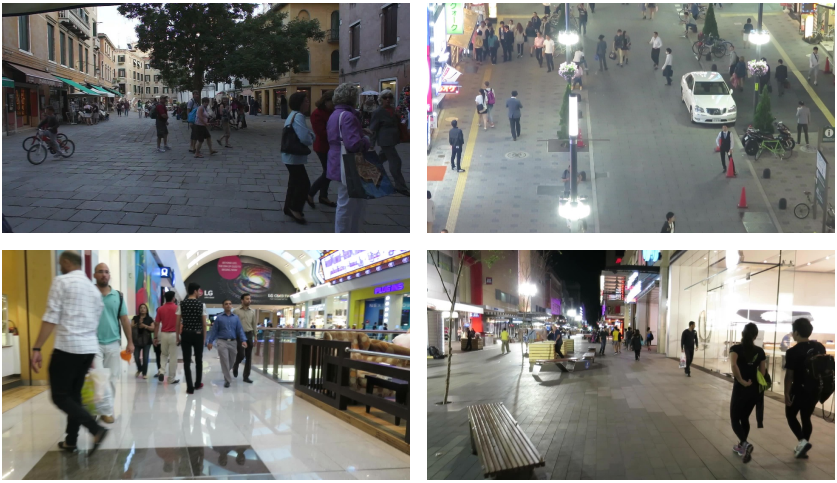


Figure 4.1: Example images of the MOT17 dataset. It comprises day and night scenes, different views as well as sequences with static and moving cameras.

Since the GT of the test set is not publicly available, a validation (*val*) split is created by splitting each train video into two halves and taking the second half following [Aha22, Du23, Wu21, Zha22c, Zho20]. The val split is used for evaluating the proposed tracking components, whereas the test split is taken to compare the whole tracking framework with the SOTA, for which one has to submit the results to the official MOTChallenge evaluation server.

MOT20

The focus of MOT20 [Den20] lies on very crowded scenarios, as this is where naturally most tracking errors occur due to severe occlusion. Per image, the mean (maximum) number of targets is 141 (248), which indicates the largest density among publicly available MPT datasets. MOT20 contains in total 2,102,385 GT boxes, and the average duration of a person being in the camera's field of view in the train split is 20.5 s. The train and test split contain 4 static videos at 25 FPS each, with indoor sequences as well as outdoor day and night scenes. Like for MOT17, the test split of MOT20 is leveraged for comparison with the SOTA by submitting results to the MOTChallenge evaluation server because the annotations are not publicly available. Example images of MOT20 can be found in Figure 4.2.

PersonPath22

To create a large-scale dataset for general MPT, the authors of PP22 [Shu22] combined several datasets for human activity understanding (MEVA [Cor21], VIRAT [Oh11], PathTrack [Man17]), added missing bounding box annotations, and additionally sourced sequences from stock video services. This resulted in a total of 236 videos, 138 and 98 in the train and test split, respectively, and a mean (maximum) number of 21 (139) persons per image. Overall, 741,475 GT boxes are annotated, and a person is visible for 12.4 s on average. A vast variation w.r.t. backgrounds, camera positions, and environment conditions like weather and lighting has been ensured by a team of experts that manually selected the videos from a set of over 8,000 candidates [Shu22]. This large variety makes PP22 an ideal dataset for assessing the

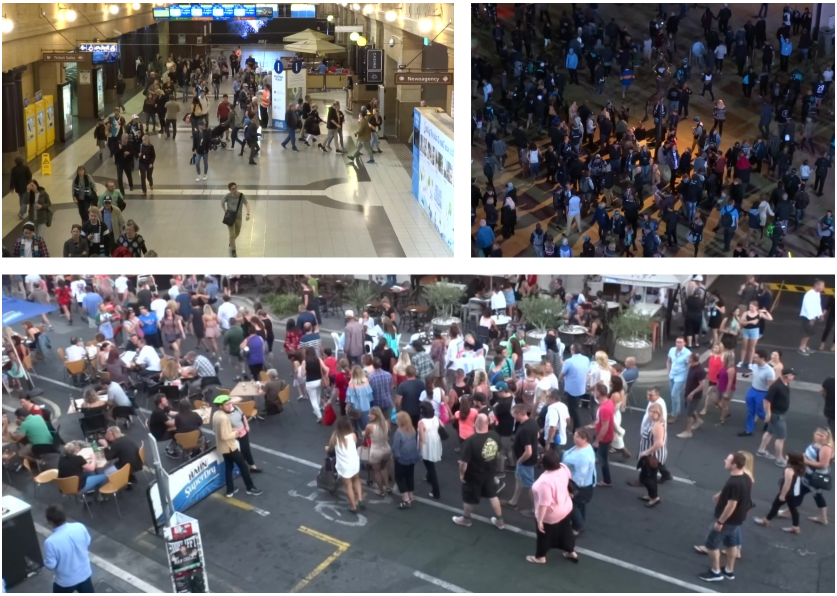


Figure 4.2: Example images of the MOT20 dataset. Its main characteristic is the very high person density in all of the videos.

generalization capabilities of MPT algorithms and thus, it is used as further evaluation dataset in this thesis. Figure 4.3 displays example frames of PP22.

Although the videos are sampled at 15–60 FPS, annotations are only available at a rate of 5 FPS as of July 2024[1]. Accordingly, tracking and evaluation are performed at this low frame rate on PP22 in this thesis, which poses an additional challenge to the tracker. The lack of full annotations might be a reason, why no results on PP22 have been reported in the literature so-far, except in the original paper of the dataset.

[1] https://github.com/amazon-science/tracking-dataset (accessed on July 16, 2024)

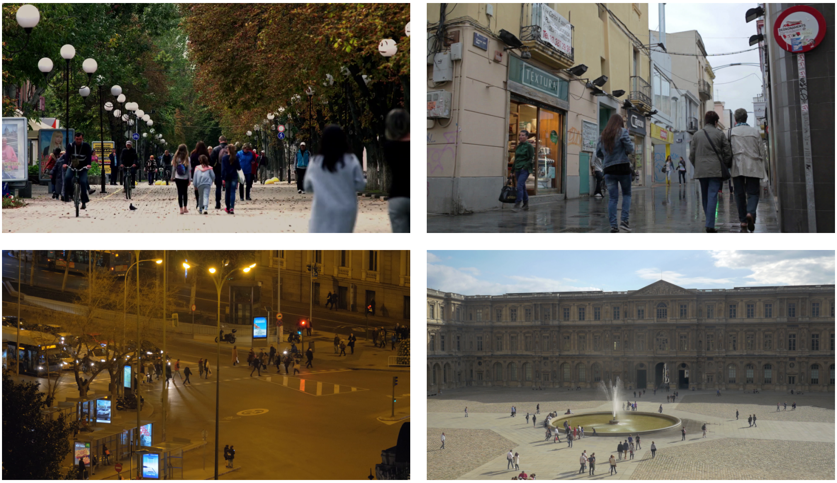


Figure 4.3: Example images of the PP22 dataset. As largest real-world MPT dataset to date, a great diversity in tracking scenarios is covered.

SOMPT22

The *Surveillance Oriented Multi-Pedestrian Tracking 2022* (SOMPT22) [Sim23] dataset comprises 15 videos, 10 in the train and 5 in the test split. Since the GT of the test split is not publicly available and there is no evaluation sever as for the MOTChallenge, only the train split is used for evaluation. A total of 535,904 GT boxes are available, and the average time a person appears in the scene is 26.3 s. All sequences are captured at daytime with 30 FPS from static cameras mounted on poles at a height of 6–8 m for city surveillance. This results in a large field of view and many small persons in the image background that are hard to detect and track as can be seen in Figure 4.4. As a supplement to the general MPT datasets MOT17 and PP22, the SOMPT22 dataset is utilized to evaluate the performance of tracking components in the surveillance context.

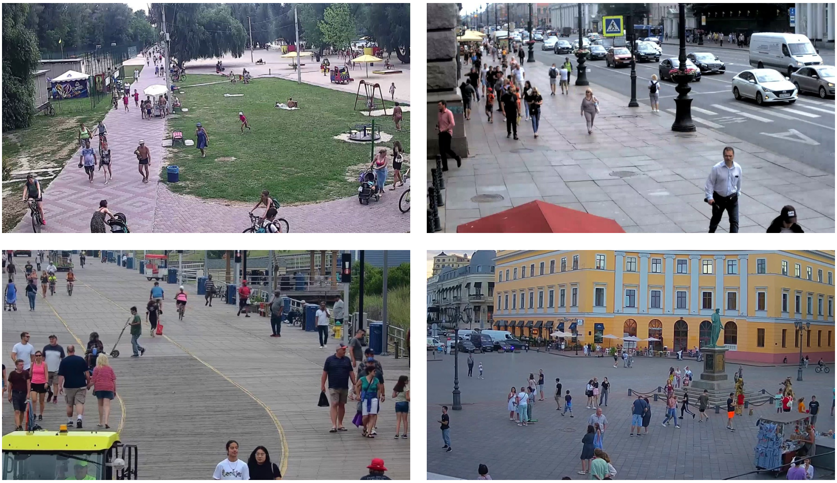


Figure 4.4: Example images of the SOMPT22 dataset. The surveillance-oriented recordings contain an oblique view leading to a high variation in person size and appearance.

4.1.2 Additional Datasets for Person Detection

Besides the MPT datasets from the previous section, three additional datasets are utilized for training the applied person detector as will be described in Section 4.3. These datasets are briefly introduced in the following.

CrowdHuman

CH [Sha18a] is a large dataset specialized for person detection in crowded scenes with about 470,000 human instances and high degrees of occlusion. It consists of 24,370 web-crawled images that are split into a train (15,000), val (4,370), and test (5,000) set. Images with only a small number of persons or small overlaps were discarded in the selection process. This lead to a density of 23 persons per image on average in the train and validation splits, which is much larger than for common person detection datasets. For instance, the COCOPersons subset of the famous COCO dataset [Lin14] contains only four persons per image on average. The CH images vary strongly both in resolution (250×374–10800×7200) and content. The large diversity of the dataset is indicated with example images in Figure 4.5.

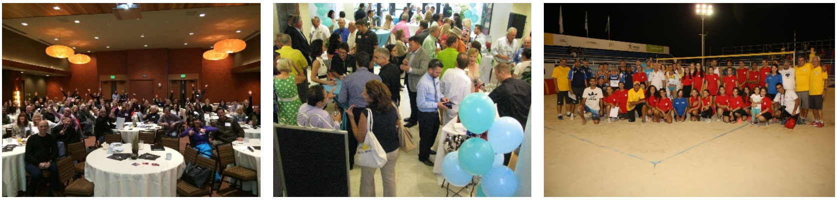


Figure 4.5: Example images of the CH dataset. As the name suggests, the focus of this person detection dataset lies on humans appearing in crowds.

CityPersons

Another dataset for person detection is CityPersons [Zha17]. It is derived from Cityscapes [Cor16], a large dataset for semantic segmentation of urban scenes in the context of autonomous driving. The 5,000 images with fine pixel-wise annotations for semantic labeling of various categories have been adopted and annotated for the task of person detection. This resulted in a total of 35,016 bounding boxes showing 19,654 unique persons in the 2,975 train, 500 val, and 1,525 test images. For benchmarking purposes, the annotations of the test set are withheld. All images have a resolution of 2048×1024 pixels. Figure 4.6 shows examples of the CityPersons dataset.

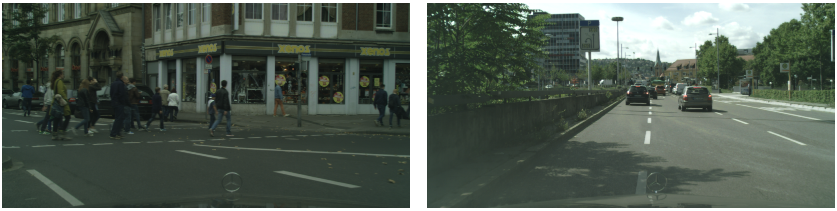


Figure 4.6: Example images of the CityPersons dataset. These are captured from the front of a vehicle in road traffic showing both persons and vehicles.

ETH

Captured by authors of the eponymous university in Zurich, the ETH dataset [Ess07] contains five videos acquired from a moving platform in busy shopping streets. The sequences contain a total of 2,056 frames sampled at 15 FPS with 16,720 annotated person bounding boxes. Besides debayering artifacts, slight motion blur, and missing contrast, the small resolution of 640 × 480 pixels poses a severe challenge for person detection algorithms. Exemplary images of the ETH dataset are depicted in Figure 4.7.

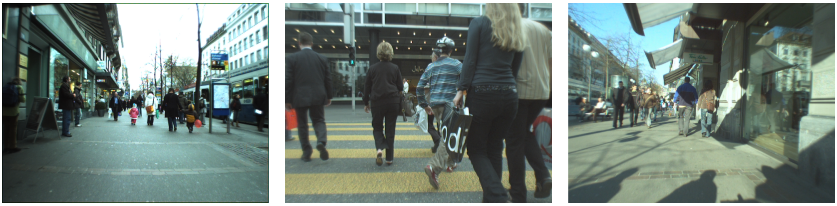


Figure 4.7: Example images of the ETH dataset. These are captured by a moving platform in busy shopping streets.

4.2 Evaluation Measures

To compare the MPT performance of different methods, a commonly accepted evaluation measure is indispensable. However, considering all aspects of MPT is a non-trivial task, since detection, association, and localization accuracy have to be taken into account simultaneously. The first measure that has become a standard in the MPT community is the Multiple Object Tracking Accuracy (MOTA) [Ber08]. Despite its major drawback of significantly preferring detection over association performance, it has been serving as main evaluation measure in the MPT literature and in popular benchmarks as MOTChallenge and KITTI [Gei12] for more than a decade. In 2016, identity F1 (IDF1) [Ris16] was proposed as special measure for multi-target multi-camera tracking. It has also been widely adopted for evaluating MPT in the single-camera context, as it focuses more on the association than detection accuracy and thus is a valuable supplement to MOTA.

Apart from overestimating detection or association accuracy, both MOTA and IDF1 possess other shortcomings that have been addressed with a new metric for MOT termed Higher Order Tracking Accuracy (HOTA) [Lui21]. It is the only popular unified measure for a balanced evaluation of all MOT aspects, i.e., detection, association, and localization. In contrast to MOTA and IDF1, HOTA has the two important characteristics of monotonicity as well as error type differentiability and is the only measure that is a *metric* in the sense of the mathematical definition. For an in-depth comparison of the three evaluation measures, the interested reader is referred to the HOTA paper [Lui21].

Due to its useful properties, HOTA has been adopted as the main evaluation measure by the MOTChallenge and KITTI benchmark and is also leveraged in this thesis to assess the performance of MPT methods. As will be seen in Section 4.2.1, HOTA can be split into several submeasures that are also used in the evaluation. MOTA and IDF1, which are described in Section 4.2.2 and Section 4.2.3, respectively, are reported as additional measures in the comparison with the SOTA. For the computation of HOTA, MOTA, and IDF1 as well as its submeasures, the TrackEval library [Jon20] is utilized. There exist other measures for evaluating MOT performance in the literature, however, these are less relevant and are therefore not considered further. Note that in this thesis, all quantitative evaluation results are given in percent unless otherwise stated.

4.2.1 Higher Order Tracking Accuracy

Like each MPT performance measure, HOTA evaluates how well a predicted set of tracks aligns with the actual GT set of tracks within a video. Before the HOTA computation is described step by step, some notations are introduced as follows. Let $\mathcal{T}_{\mathrm{pred}} = \{\mathrm{T}^1_{\mathrm{pred}}, \mathrm{T}^2_{\mathrm{pred}}, \dots\}$ denote the predicted track set and $\mathcal{T}_{\mathrm{GT}} = \{\mathrm{T}^1_{\mathrm{GT}}, \mathrm{T}^2_{\mathrm{GT}}, \dots\}$ the GT track set, both comprising multiple tracks T, i.e., lists of detections belonging to the same identity (ID). For example, a predicted track is written as $\mathrm{T}^{i}_{\mathrm{pred}} = [\mathrm{D}^{i,f_1}_{\mathrm{pred}}, \mathrm{D}^{i,f_2}_{\mathrm{pred}}, \dots]$ with $\mathrm{D}^{i,f_1}_{\mathrm{pred}}$ being the predicted detection in frame f_1 of the video belonging to track $\mathrm{T}^{i}_{\mathrm{pred}}$. The same notation is used for a GT track, i.e., $\mathrm{T}^{i}_{\mathrm{GT}} = [\mathrm{D}^{i,f_1}_{\mathrm{GT}}, \mathrm{D}^{i,f_2}_{\mathrm{GT}}, \dots]$. The ID of a predicted and GT detection is given by $\mathrm{ID}(\mathrm{D}_{\mathrm{pred}})$ and $\mathrm{ID}(\mathrm{D}_{\mathrm{GT}})$, respectively. Note that the IDs of detections belonging to the same track are equal: $\mathrm{ID}(\mathrm{D}^{i,f_1}) = \mathrm{ID}(\mathrm{D}^{i,f_2}) = \dots$.

Given a localization similarity measure for comparing predicted and GT detections and a minimum localization threshold η, the matching of predicted and GT tracks is performed on the detection level for each frame of the video. In this thesis, tracking is performed with 2D bounding boxes, so the IoU between such boxes is used as similarity measure. Let $\mathcal{D}_{\mathrm{pred}} = \{\mathrm{D}^{i}_{\mathrm{pred}}\}_i$ be the predicted detections and $\mathcal{D}_{\mathrm{GT}} = \{\mathrm{D}^{i}_{\mathrm{GT}}\}_i$ the GT detections of a frame, whereby the frame index is omitted for clarity. A bijective matching of the two detection sets is performed in a way that the resulting HOTA value is maximized. Details of this matching process are not necessary for the general understanding of the HOTA calculation and thus are not considered here but can be found in the original paper [Lui21].

After matching, the predicted detections can be split into three sets that are later used for assessing the detection quality:

- True Positives (TP)—predicted detections with matched GT detection:

$$\mathrm{TP} = \left\{\mathrm{D}^{\mathrm{matched}}_{\mathrm{pred}}\right\}. \tag{4.1}$$

- False Positives (FP)—predicted detections without matched GT detection:

$$\mathrm{FP} = \left\{\mathrm{D}^{\mathrm{unmatched}}_{\mathrm{pred}}\right\}. \tag{4.2}$$

- False Negatives (FN)—GT detections without matched predicted detection:

$$\mathrm{FN} = \left\{\mathrm{D}^{\mathrm{unmatched}}_{\mathrm{GT}}\right\}. \tag{4.3}$$

To evaluate the association accuracy, three measures are defined in the following, which are graphically illustrated in Figure 4.8.

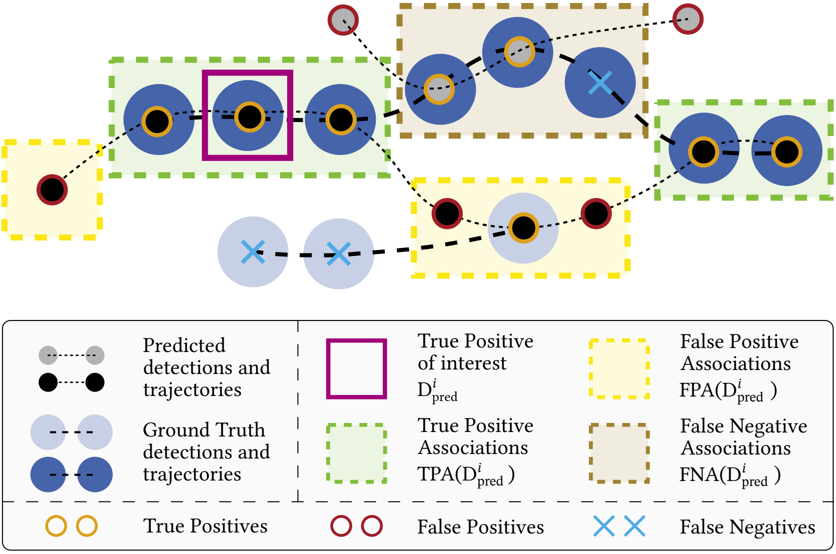

Predicted detections and trajectories
Ground Truth detections and trajectories
True Positive of interest D^i_{pred}
True Positive Associations $\text{TPA}(D^i_{\text{pred}})$
False Positive Associations $\text{FPA}(D^i_{\text{pred}})$
False Negative Associations $\text{FNA}(D^i_{\text{pred}})$
True Positives
False Positives
False Negatives

Figure 4.8: Concepts in the evaluation of HOTA. Given a TP of interest (purple), its TPAs (green), FPAs (yellow), and FNAs (brown) can be determined based on the predicted and GT trajectories. The figure builds upon [Lui21] and is extended by TPs, FPs, and FNs.

These measures are calculated for each true positive detection $D^i_{\text{pred}} \in \text{TP}$ with D^i_{GT} denoting the matched GT detection to D^i_{pred}:

- True Positive Associations (TPA)—set of TPs that have the same predicted ID and same GT ID:

$$\begin{aligned}\text{TPA}(D^i_{\text{pred}}) = \Big\{ D^k_{\text{pred}} \,\Big|\, & D^k_{\text{pred}} \in \text{TP} \\ & \wedge \text{ID}(D^i_{\text{pred}}) = \text{ID}(D^k_{\text{pred}}) \wedge \text{ID}(D^i_{\text{GT}}) = \text{ID}(D^k_{\text{GT}}) \Big\}.\end{aligned} \tag{4.4}$$

- False Positive Associations (FPA)—set of TPs with the same predicted ID but different GT ID or FPs with the same predicted ID:

$$\begin{aligned}\mathrm{FPA}\big(\mathrm{D}^{i}_{\mathrm{pred}}\big) = \Big\{\mathrm{D}^{k}_{\mathrm{pred}} \,\Big|\, \mathrm{D}^{k}_{\mathrm{pred}} &\in \mathrm{TP} \\ &\wedge \mathrm{ID}\big(\mathrm{D}^{i}_{\mathrm{pred}}\big) = \mathrm{ID}\big(\mathrm{D}^{k}_{\mathrm{pred}}\big) \wedge \mathrm{ID}\big(\mathrm{D}^{i}_{\mathrm{GT}}\big) \neq \mathrm{ID}\big(\mathrm{D}^{k}_{\mathrm{GT}}\big)\Big\} \\ \cup \Big\{\mathrm{D}^{k}_{\mathrm{pred}} \,\Big|\, \mathrm{D}^{k}_{\mathrm{pred}} &\in \mathrm{FP} \wedge \mathrm{ID}\big(\mathrm{D}^{i}_{\mathrm{pred}}\big) = \mathrm{ID}\big(\mathrm{D}^{k}_{\mathrm{pred}}\big)\Big\}.\end{aligned} \tag{4.5}$$

- False Negative Associations (FNA)—set of TPs with the same GT ID but a different predicted ID and FNs with the same GT ID:

$$\begin{aligned}\mathrm{FNA}\big(\mathrm{D}^{i}_{\mathrm{pred}}\big) = \Big\{\mathrm{D}^{k}_{\mathrm{pred}} \,\Big|\, \mathrm{D}^{k}_{\mathrm{pred}} &\in \mathrm{TP} \\ &\wedge \mathrm{ID}\big(\mathrm{D}^{i}_{\mathrm{pred}}\big) \neq \mathrm{ID}\big(\mathrm{D}^{k}_{\mathrm{pred}}\big) \wedge \mathrm{ID}\big(\mathrm{D}^{i}_{\mathrm{GT}}\big) = \mathrm{ID}\big(\mathrm{D}^{k}_{\mathrm{GT}}\big)\Big\} \\ \cup \Big\{\mathrm{D}^{k}_{\mathrm{GT}} \,\Big|\, \mathrm{D}^{k}_{\mathrm{GT}} &\in \mathrm{FN} \wedge \mathrm{ID}\big(\mathrm{D}^{i}_{\mathrm{GT}}\big) = \mathrm{ID}\big(\mathrm{D}^{k}_{\mathrm{GT}}\big)\Big\}.\end{aligned} \tag{4.6}$$

In Figure 4.8, an arbitrary TP detection of interest $\mathrm{D}^{i}_{\mathrm{pred}}$ is highlighted in purple, for which the TPAs, FPAs, and FNAs are labeled with green, yellow, and brown boxes, respectively. Remember that these association entities are computed for each TP detection individually. For instance, to determine the set of TPAs for the detection of interest, it is checked for each detection of its predicted trajectory (black filled circles) whether it belongs to the same GT trajectory (dark blue circles), according to Equation (4.4). The set of FPAs and FNAs for the TP of interest are determined with Equations (4.5) and (4.6). Next to the association measures, Figure 4.8 highlights TP and FP detections with orange and red circles, respectively, and indicates FNs with blue crosses.

Remember that the introduced measures are defined for a specific localization threshold η, i.e., the minimum IoU for matching predicted and GT boxes, which has been omitted for clarity. Given a concrete value of η, HOTA_{η} can be calculated as

$$\mathrm{HOTA}_\eta = \sqrt{\frac{\sum_{\mathrm{D}^i_{\mathrm{pred}} \in \mathrm{TP}} \mathrm{AssS}(\mathrm{D}^i_{\mathrm{pred}})}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}} \quad \text{with} \tag{4.7}$$

$$\mathrm{AssS}(\mathrm{D}^i_{\mathrm{pred}}) = \frac{\left|\mathrm{TPA}(\mathrm{D}^i_{\mathrm{pred}})\right|}{\left|\mathrm{TPA}(\mathrm{D}^i_{\mathrm{pred}})\right| + \left|\mathrm{FNA}(\mathrm{D}^i_{\mathrm{pred}})\right| + \left|\mathrm{FPA}(\mathrm{D}^i_{\mathrm{pred}})\right|}, \tag{4.8}$$

where $\mathrm{AssS}(\mathrm{D}^i_{\mathrm{pred}})$ denotes an association score for the true positive detection $\mathrm{D}^i_{\mathrm{pred}} \in \mathrm{TP}$. Consequently, HOTA_η measures the detection and association accuracy at a minimum localization requirement given by $\eta \in (0, 1)$. The final HOTA score is the integration over all possible HOTA_η values, which practically is approximated by an average over 19 values $\mathrm{H} = \{0.05, 0.1, \dots, 0.95\}$:

$$\mathrm{HOTA} = \int_0^1 \mathrm{HOTA}_\eta \, d\eta \approx \frac{1}{19} \sum_{\eta \in \mathrm{H}} \mathrm{HOTA}_\eta. \tag{4.9}$$

Besides being a unified metric for detection, association, and localization accuracy in MOT, HOTA can be decomposed into several submeasures. At each specific localization value η, HOTA_η can be split into detection accuracy (DetA_η) and association accuracy (AssA_η) as follows:

$$\mathrm{HOTA}_\eta = \sqrt{\mathrm{DetA}_\eta \cdot \mathrm{AssA}_\eta} \quad \text{with} \tag{4.10}$$

$$\mathrm{DetA}_\eta = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|} \quad \text{and} \tag{4.11}$$

$$\mathrm{AssA}_\eta = \frac{1}{|\mathrm{TP}|} \sum_{\mathrm{D}^i_{\mathrm{pred}} \in \mathrm{TP}} \mathrm{AssS}(\mathrm{D}^i_{\mathrm{pred}}). \tag{4.12}$$

According to Equation (4.10), HOTA_η is the geometric mean of DetA_η and AssA_η, so detection and association are equally weighted by the HOTA metric. Similar to HOTA, the summarized DetA and AssA can be computed by integrating over DetA_η and AssA_η for all η values as in Equation (4.9). On top of the separation into DetA and AssA, those can be further split into recall and precision measures. The common detection recall (DetRe) and detection precision (DetPr) are calculated and combined to DetA using the sets of TP, FN, and FP from Equations (4.1) to (4.3) as follows (for a specific localization threshold η):

$$\mathrm{DetRe}_\eta = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|}, \tag{4.13}$$

$$\mathrm{DetPr}_\eta = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}|}, \tag{4.14}$$

$$\mathrm{DetA}_\eta = \frac{\mathrm{DetRe}_\eta \cdot \mathrm{DetPr}_\eta}{\mathrm{DetRe}_\eta + \mathrm{DetPr}_\eta - \mathrm{DetRe}_\eta \cdot \mathrm{DetPr}_\eta}. \tag{4.15}$$

The accumulated versions DetRe and DetPr are also generated via integration as in Equation (4.9). DetRe gives the proportion of GT detections that are predicted, while DetPr specifies the percentage of predicted detections that are correct.

Similar to the detection measures, the novel concepts of association recall (AssRe) and association precision (AssPr) can be computed with TP, TPA, FNA, and FPA from Equations (4.1) and (4.4) to (4.6) and combined to AssA for a certain value η:

$$\mathrm{AssRe}_\eta = \frac{1}{|\mathrm{TP}|} \sum_{\mathrm{D}^i_{\mathrm{pred}} \in \mathrm{TP}} \frac{\left|\mathrm{TPA}(\mathrm{D}^i_{\mathrm{pred}})\right|}{\left|\mathrm{TPA}(\mathrm{D}^i_{\mathrm{pred}})\right| + \left|\mathrm{FNA}(\mathrm{D}^i_{\mathrm{pred}})\right|}, \tag{4.16}$$

$$\mathrm{AssPr}_\eta = \frac{1}{|\mathrm{TP}|} \sum_{\mathrm{D}^i_{\mathrm{pred}} \in \mathrm{TP}} \frac{\left|\mathrm{TPA}(\mathrm{D}^i_{\mathrm{pred}})\right|}{\left|\mathrm{TPA}(\mathrm{D}^i_{\mathrm{pred}})\right| + \left|\mathrm{FPA}(\mathrm{D}^i_{\mathrm{pred}})\right|}, \tag{4.17}$$

$$\mathrm{AssA}_\eta = \frac{\mathrm{AssRe}_\eta \cdot \mathrm{AssPr}_\eta}{\mathrm{AssRe}_\eta + \mathrm{AssPr}_\eta - \mathrm{AssRe}_\eta \cdot \mathrm{AssPr}_\eta}. \tag{4.18}$$

Again, the summarized versions AssRe and AssPr follow by integration over η as in Equation (4.9). AssRe measures how well the predicted tracks match the actual GT tracks. For instance, a low AssRe means that an object is represented with multiple predicted tracks. AssPr states how well the predicted tracks keep to tracking the same GT tracks. For example, a low AssPr occurs if a predicted track covers several objects.

The various submeasures of HOTA allow a detailed analysis of different characteristics of a tracking method. In the evaluations of this thesis, the submeasures DetA, AssA, DetRe, DetPr, AssRe, and AssPr are leveraged besides the general HOTA metric.

4.2.2 Multiple Object Tracking Accuracy

As for HOTA, a bijective matching of predicted and GT tracks is performed on the detection level in the MOTA computation. A minimum localization requirement is also enforced with a threshold η. However, MOTA is only calculated for one specific value of η, which is typically set to 0.5. So in the case of tracking with 2D bounding boxes, a minimum IoU of 0.5 between predicted and GT detection is required for matching. Next to TP, FN, and FP (Equations (4.1) to (4.3)), the computation of MOTA requires another concept for measuring association accuracy termed IDSW. An IDSW happens whenever an object ID is mistakenly switched by the tracker or when a track is re-initialized with another ID after it was lost. Formally, the set of IDSW at frame f is given by the true predicted detections that have a different ID as the true predicted detections from the previous frame $f-1$ but the same GT ID:

$$\begin{aligned}\mathrm{IDSW}^{f} = \Big\{\mathrm{D}_{\mathrm{pred}}^{i,f} \,\Big|\, &\mathrm{D}_{\mathrm{pred}}^{i,f} \in \mathrm{TP} \\ &\wedge \mathrm{ID}\big(\mathrm{D}_{\mathrm{pred}}^{i,f}\big) \neq \mathrm{ID}\big(\mathrm{D}_{\mathrm{pred}}^{i,f-1}\big) \wedge \mathrm{ID}\big(\mathrm{D}_{\mathrm{GT}}^{i,f}\big) = \mathrm{ID}\big(\mathrm{D}_{\mathrm{GT}}^{i,f-1}\big)\Big\}.\end{aligned} \tag{4.19}$$

Note that IDSW counts association errors only w.r.t. the previous true detection and does not take longer contexts into account like HOTA. Moreover, the IDSW measure does not include *ID transfers*, i.e., tracking errors where the same predicted ID switches to a different GT ID.

When TP, FN, FP, and IDSW have been determined after the matching process, the MOTA score can be calculated as

$$\mathrm{MOTA} = 1 - \frac{|\mathrm{FN}| + |\mathrm{FP}| + |\mathrm{IDSW}|}{|\mathrm{TP}| + |\mathrm{FN}|}. \tag{4.20}$$

The main problem of the MOTA measure is its strong bias towards detection performance. An evaluation of trackers on the MOT17 benchmark in [Lui21] revealed that the number of detection errors |FN| + |FP| is typically about 100 times higher than the number of association errors |IDSW| in the MOTA formulation. Further shortcomings of the measure are discussed in [Lui21]. Because of its weaknesses, MOTA is only reported as secondary performance measure in the SOTA comparison of this thesis.

4.2.3 Identity F1

In contrast to HOTA and MOTA, IDF1 performs the matching of predicted tracks to GT tracks on the track level, i.e., the full tracks are compared and not their single detections on a frame level. As in MOTA, a minimum localization (IoU) requirement of $\eta = 0.5$ is applied and the bijective matching is carried out such that the final score is maximized. After the matching process, the following types of detections can be specified:

- Identity True Positives (IDTP)—predicted detections in the overlapping parts of two matched tracks.
- Identity False Negatives (IDFN)—GT detections in the non-overlapping parts of two matched tracks and detections of unmatched GT tracks.
- Identity False Positives (IDFP)—predicted detections in the non-overlapping parts of two matched tracks and detections of unmatched predicted tracks.

Based on these definitions, the identity recall (IDRe), identity precision (IDPr) and IDF1 are computed as follows:

$$\text{IDRe} = \frac{|\text{IDTP}|}{|\text{IDTP}| + |\text{IDFN}|}, \tag{4.21}$$

$$\text{IDPr} = \frac{|\text{IDTP}|}{|\text{IDTP}| + |\text{IDFP}|}, \tag{4.22}$$

$$\text{IDF1} = \frac{2|\text{IDTP}|}{2|\text{IDTP}| + |\text{IDFN}| + |\text{IDFP}|}. \tag{4.23}$$

Next to its bias towards association, IDF1 has several drawbacks including a counter-intuitive and non-monotonic behavior when it comes to measuring detection accuracy as analyzed in [Lui21]. Therefore, IDF1 is not used for evaluating single tracking components in this thesis, but is given as another supplementary performance measure in the SOTA comparison.

4.3 Evaluation Protocols

The proposed tracking framework comprises two modules whose performance on an applied dataset strongly depends on the data they were trained on: detector and REID model. In practice, large amounts of training data are not always available in the application domain, especially for MPT because annotating videos with dozens or even hundreds of persons is a time consuming process. In such cases, alternative data has to be used for training which leads to a *domain gap* between training and inference of the used models. The larger this discrepancy between training and testing data, the greater the requirements for the generalization capabilities of the tracking framework.

To assess the performance of proposed tracking components under various generalization difficulties, three evaluation protocols are followed that differ in the utilized datasets for training and testing. Moreover, two further protocols are defined for comparing the performance of the final tracking framework with the SOTA. In the following, the training datasets of detector and REID model for five different evaluation datasets, i.e., MOT17 val, PP22 test, SOMPT22 train, MOT17 test, and MOT20 test, are briefly described and the main purpose of the respective evaluation protocol is stated. Finally, a summary of the key aspects of these protocols is given.

MOT17 Val

Remember that MOT17 val contains the second halves of the MOT17 train sequences. The first halves of the sequences, denoted as MOT17 *train half*, are used for training. As this subset is quite small for training a detection model from scratch, the CH train/val dataset is additionally utilized, which has become a standard procedure in the literature [Aha22, Cao23, Cet23, Du23, Jun24, Sun21a, Wu21, Zen22, Zha22c, Zho20]. The REID model is also trained on MOT17 train half (for details, see Section 5.6.2). Since the domain gap between MOT17 val and MOT17 train half is small, this protocol focuses on the specialization capabilities of the evaluated methods.

PP22 Test

In the evaluation on PP22 test, the detector is trained on a combination of CH train/val and PP22 train. For REID, the same model as for MOT17, which is trained on MOT17 train half, is applied. Thus, good generalization capabilities of the REID model are required. Furthermore, a powerful REID model is beneficial in this evaluation because the low frame rate of five FPS generally makes the predictions of the motion model less accurate.

SOMPT22 Train

The annotations of the SOMPT22 test set are not publicly available. This is why evaluation is performed on the train set. As for PP22 test, the detector trained on CH train/val and PP22 train is applied, as well as the REID model trained on MOT17 train half. Since SOMPT22 is a special dataset for surveillance with different characteristics compared to the more general MOT17 and

PP22 dataset, there is a quite large domain gap between the training and testing scenarios. Thus, this protocol poses the greatest challenges regarding the generalization capabilities of the evaluated tracking approaches.

MOT17 Test

MOT17 test is next to MOT20 test the standard benchmark for comparing an MPT method with the current SOTA. The common protocol of training the detector on MOT17 train, CH train/val, CityPersons train/val, and ETH is adopted [Aha22, Cao23, Du23, Jun24, Zha22c]. Following [Aha22, Du23, Gao24, Jun24, Mag23], the REID model trained on MOT17 train half is applied. On MOT17 test, the performance of trackers on general MPT sequences with a high variety is evaluated.

MOT20 Test

As another well-established dataset in the MPT community, MOT20 test is the second dataset used in the SOTA comparison of this thesis. Again, the typical evaluation protocol from the literature is followed meaning that the detector is trained on a combination of MOT20 train and CH train/val [Aha22, Cao23, Du23, Jun24, Zha22c] and the REID model is trained on MOT20 train half [Aha22, Du23, Gao24, Jun24, Mag23]. The focus of the evaluation on MOT20 test lies on the performance of MPT methods in very crowded scenes, where naturally many tracking errors can occur.

Summary

The key aspects of the just described evaluation protocols are summarized in Table 4.2. As mentioned before, the evaluation protocols differ in terms of the required generalization capability of the tested tracking approaches and their ability to handle scenes with various crowd density levels.

Table 4.2: Overview of evaluation protocols of this thesis. Different combinations of training and evaluation dataset splits mainly lead to various levels of generalization difficulty and crowdedness. The first three rows depict configurations used for analyzing proposed tracking components, whereas the last two rows show protocols for comparing the final tracking framework with the SOTA.

	Evaluation	Detector training	REID training	
Analysis of components	MOT17 val	CH train/val, MOT17 train half	MOT17 train half	↓ Generalizat. difficulty
	PP22 test	CH train/val, PP22 train	MOT17 train half	
	SOMPT22 train	CH train/val, PP22 train	MOT17 train half	
Comparison with SOTA	MOT17 test	CH train/val, MOT17 train, CityPersons train/val, ETH	MOT17 train half	↓ Crowd density
	MOT20 test	CH train/val, MOT20 train	MOT20 train half	

5 Base Framework

For various reasons, most MPT methods found in the literature follow the TBD paradigm [Aha22, Bew16, Cao23, Woj17, Zha22c]. Since the MPT task is divided into the two subtasks detection and association, single tracking modules can be exchanged independently, making the overall system design very flexible. This is beneficial both for the development and evaluation of individual tracking components. In this chapter, a generic TBD framework is built, which serves as baseline for comparison with the proposed methods in Chapter 6. While additional modules are possible, a TBD approach for MPT comprises at least the following three parts:

- **Detection**: Localization of persons on the images of the input video.
- **Association**: Matching of detections belonging to the same target to tracks, i.e., sequences of detections.
- **Track management**: Strategies to determine when tracks are initialized, change their state (e.g., *active*, *inactive*), and are terminated.

MPT is mostly applied on videos with a high frame rate. Therefore, the positions of the persons on the images do not change much from frame to frame and are of high value for the association task. For this reason, MPT systems usually contain a motion model that uses the motion states of the targets to predict their positions in the current frame, before performing a motion-based association. Besides positional information, the appearance of persons is a good indicator for determining whether two detections from different video frames belong to the same person. Therefore, it is common practice in MPT to leverage a model from the person REID community for computing appearance features that are utilized in the association. Consequently, the following two modules are also widely used in MPT systems:

- **Motion model**: Prediction of target positions in consecutive frames.
- **REID model**: Extraction of appearance cues from persons.

The base framework of this thesis comprises all of the five aforementioned modules, which are described next in more detail. Finally, an experimental evaluation of the base framework concludes this chapter.

5.1 Detection

The main task of the detector in the context of MPT is to provide a set of bounding boxes and corresponding confidence scores for each image of the input video. More precisely, let $\mathrm{V} = (\mathbf{I}_1, \ldots, \mathbf{I}_l)$ be the input video to be processed, i.e., a sequence of images with length l. Given a detection model $\mathrm{DET}(\cdot)$ and an image $\mathbf{I}$, a set of k person detections $\widetilde{\mathcal{D}} = \{\widetilde{\mathrm{D}}_1, \ldots, \widetilde{\mathrm{D}}_k\}$ is generated, applying the detector on the image: $\widetilde{\mathcal{D}} = \mathrm{DET}(\mathbf{I})$. The tilde indicates that the detection set is an intermediate result as explained in the next paragraph. A single detection comprises at least a four-dimensional vector $\mathbf{b}$ and a scalar s representing the bounding box and confidence of the detection, respectively. Hence, it is modeled as a tuple $\mathrm{D} = (\mathbf{b}, s)$. The bounding box $\mathbf{b} = (x, y, w, h)^\mathsf{T}$ is fully described by its center coordinates on the image (x, y) as well as its width w and height h. The confidence score s, normalized between zero and one, is a measure of how sure the detector is about the presence of the detected object.

In a multitude of common detector architectures, the obtained detection set $\widetilde{\mathcal{D}}$ is highly redundant, meaning that it contains a high number of duplicate detections of the same person. The reason for this redundancy will be explained in Section 5.1.1, where the detection model used in this thesis is described. To remove duplicate detections, a common technique termed NMS is applied that yields the final set of detections $\mathcal{D} \subseteq \widetilde{\mathcal{D}}$ with $|\mathcal{D}| \leq |\widetilde{\mathcal{D}}|$, which will be presented in Section 5.1.2.

5.1.1 Detection Model

As already mentioned, the modules of the built MPT framework are exchangeable such that an arbitrary detection model can be used. The choice falls on the single-stage detector YOLOX[1] [Ge21] for mainly two reasons:

- The overall system should be real-time capable. The YOLO series [Boc20, Ge21, Joc20, Joc23, Li22, Red16, Red17, Red18, Wan23a] is a popular family of object detectors with a good trade-off between runtime and accuracy.
- YOLOX is the most utilized detector of recent MPT methods allowing a good comparability with the current SOTA.

The basic functionality of YOLOX is explained using its architecture depicted in Figure 5.1.

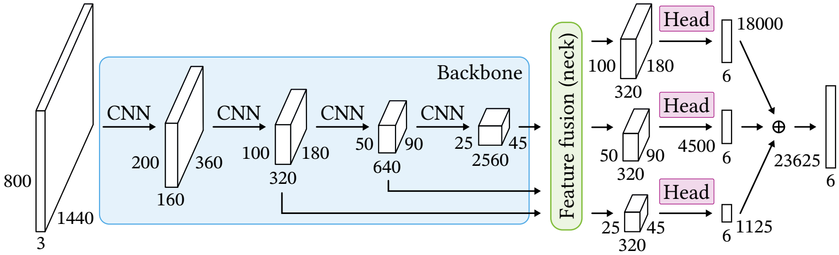


Figure 5.1: Scheme of the YOLOX-X architecture with an exemplary input size of 1440×800 pixels. The backbone extracts feature maps of different sizes, which are then fused in the neck. Given the fused features, the head produces three sets of predictions at various scales. Finally, the predictions are combined yielding the network output.

The input to the network is an image of arbitrary size with three channels (red, green, blue). All shapes of the network can be derived from the input width

[1] "YOLO" stands for *you only look once* and was one of the first one-stage detection approaches, meaning that the detections are generated by looking only once at the image. In contrast, two-stage detectors typically predict region proposals containing object candidates first, before classifying and adjusting those proposals in the second stage. The "X" denotes a variant of the YOLO family with the paper title *YOLOX: Exceeding YOLO Series in 2021.*

and height that are set to 1440 and 800 pixels, respectively, in this thesis. That input resolution is also utilized by many SOTA approaches, enabling a fair comparison in the evaluation. The overall structure of YOLOX is made of a *backbone* for feature extraction, a *neck* for feature enhancement, and a *head* for the actual detection task.

In the backbone, multiple feature maps are extracted with a CNN. With growing depth, the spatial size of the feature maps decreases, while the number of channels and semantic value increases. To improve the semantic information encoded in feature maps of earlier layers, while at the same time enhancing the spatial accuracy in feature maps of deeper layers, a feature fusion module is applied in the network neck. In total, three feature maps with a downsampling factor of 8, 16, and 32 w.r.t. the input size are revised in the feature fusion module. Finally, the network head is applied on the revised feature maps generating detections at three different scales, which is beneficial for detecting objects of various sizes.

Before taking a closer look at the detection head, more details about the backbone shapes are given because these determine the number of detections generated by the model. YOLOv5 [Joc20] introduced a scaling scheme for the network size in order to provide models with different computational complexity, which has been adopted by YOLOX. Starting with the Darknet53 architecture from YOLOv3 [Red18], various variants N (nano), S (small), M (medium), L (large), and X (extra large) are derived by changing the number of layers and filters in the partial CNNs that are indicated in Figure 5.1. In this thesis, the extra large version X is adopted. Consequently, the overall architecture is named *YOLOX-X*. Note that, for example, the number of channels of the feature maps used for detection in YOLOX-X (320, 640, 2560) is 1.25 times larger than in YOLOX-L (256, 512, 2048), which is the original Darknet53 [Red18] architecture from YOLOv3, and 5 times larger than in YOLOX-N (64, 128, 512).

As a consequence of the downsampling in the backbone, it has to be ensured that the input width and height are divisible by 32, which is done by simple padding operations if necessary. The three feature maps used in the network head have a resolution of 180×100, 90×50, and 45×25 pixels, respectively (for input size 1440×800). This yields a total of $180 \cdot 100 + 90 \cdot 50 + 45 \cdot 25 =$

23,625 predictions, since the network head makes a prediction for each spatial position of the feature map input. As will be seen in the next paragraph, one prediction corresponds to a six-dimensional vector if only one object class, i.e., person, has to be detected.

While previous YOLO versions had a coupled detection head using the same features for classification and localization, YOLOX implements a *decoupled head* to resolve the conflict between the two tasks. The decoupled head with its classification and localization branch is illustrated in Figure 5.2.

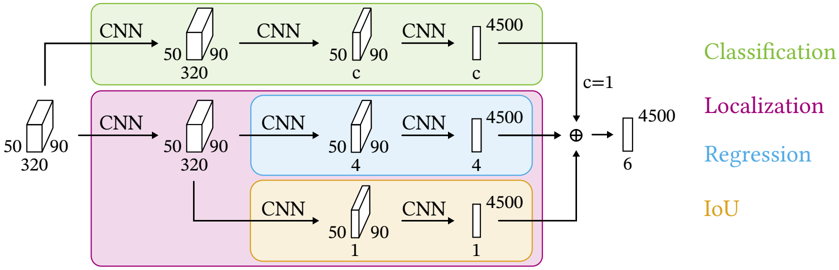


Figure 5.2: Structure of the YOLOX network head with exemplary shapes. Given a feature map as input, the classification branch generates a class score for each spatial position and category. In parallel, the localization branch simultaneously performs bounding box regression and IoU computation for assessing the localization accuracy of the predicted boxes, also for each spatial position of the feature map. Classification, regression, and IoU scores are concatenated to create the network output.

As mentioned before, the network head is applied on three feature maps from the backbone with different shapes to improve the detection performance for various object sizes. The classification branch is a small CNN that yields c confidence scores, one for each considered object category. Since only persons should be detected, $c = 1$ holds. The single output is termed s_{person} and is normalized between zero and one. The localization branch also consists of small CNNs. It is further split into two subbranches for bounding box regression and a newly introduced IoU score. The standard regression branch computes the four bounding box parameters x, y, w, and h. Like the first YOLO version, YOLOX additionally employs an *IoU branch* that predicts the IoU between the GT box and the generated box. The predicted IoU score s_{IoU}, which also lies between zero and one, is multiplied with the score of the classification branch, giving the overall confidence score

$$s = s_{\mathrm{person}} \cdot s_{\mathrm{IoU}}. \tag{5.1}$$

As the IoU is a measure for localization accuracy, this reduces the confidence of poorly localized bounding boxes. To summarize, the YOLOX head predicts six values for each spatial position of the input feature map: one confidence score for detecting a person, four for the bounding box position and size, and one for the IoU as indicator of the localization accuracy.

Besides the decoupled head and the IoU branch, YOLOX introduces further architectural developments as well as improvements in the training process of the single-stage detector, which are out of the scope of this thesis. The interested reader is referred to the original paper of YOLOX [Ge21] and a comprehensive review of the YOLO family found in [Ter23].

5.1.2 Non-Maximum Suppression

With an input size of 1440×800 pixels, a total of 23,625 predictions is made by the YOLOX detector (Figure 5.1). Obviously, not all of these predictions correspond to actual persons. Predictions from feature map positions that contain only information from the background, i.e., image parts without persons, will mostly have a very low confidence score s. Therefore, a minimum confidence threshold s_{min} is applied to filter these background detections. Moreover, adjacent feature map positions are likely to detect the same person multiple times, especially if the person covers a large area on the input image such that large parts of the feature maps contain information from that person. For this reason, an NMS is leveraged to filter duplicate detections as post-processing step.

Given a set of detections $\widetilde{\mathcal{D}} = \{\widetilde{\mathrm{D}}_1, \dots, \widetilde{\mathrm{D}}_k\}$, where each detection $\mathrm{D} = (\mathbf{b}, s)$ consists of a bounding box $\mathbf{b}$ and a confidence score s, optionally, a confidence threshold s_{min} is applied to remove low-confidence detections with $s < s_{\mathrm{min}}$. As first step of the NMS, the detections are sorted with descending confidence. Then, the IoU, denoted by o (*overlap*), between the most confident detection and each other detection is calculated. If it exceeds an overlap threshold o_{NMS}, the less-confident detection is removed from the set of detections $\widetilde{\mathcal{D}}$. When all overlaps with the most confident detection have been computed, the process is repeated with the second most confident detection, while the most confident one is excluded in the following runs. The algorithm terminates when all detections have either been removed or have been kept as the most confident detection, yielding the output set of detections $\mathcal{D} = \{\mathrm{D}_1, \dots, \mathrm{D}_m\}$, $m \leq k$. In practice, it is much smaller than the input detection set $\widetilde{\mathcal{D}}$, so $m < k$ holds. Algorithm 1 summarizes the NMS procedure.

Algorithm 1: Non-Maximum Suppression.

```
Input: Set of detections $\widetilde{\mathcal{D}} = \{\widetilde{\mathrm{D}}_1, \dots, \widetilde{\mathrm{D}}_k\}$ with $\mathrm{D} = (\mathbf{b}, s)$ and
       $\mathbf{b} = (x,y,w,h)$,
       IoU threshold $o_{\text{NMS}}$
Output: Filtered set of detections $\mathcal{D} = \{\mathrm{D}_1, \dots, \mathrm{D}_m\}$, $m \leq k$
// sort detections with descending confidence $s$ and save in list
$\mathrm{L} \leftarrow [(\mathbf{b}_1, s_1), (\mathbf{b}_2, s_2), \dots, (\mathbf{b}_k, s_k) \,|\, s_1 \geq s_2 \geq \cdots \geq s_k]$
$\mathcal{D} \leftarrow \emptyset$                                   // initialize output set
while length(L) $\neq 0$ do
    $\mathrm{D} = (\mathbf{b}, s) \leftarrow$ L.pop(0)                // select most confident detection
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathrm{D}\}$                      // and save it in the output set
    for $\widetilde{\mathrm{D}} = (\tilde{\mathbf{b}}, \tilde{s}) \in \mathrm{L}$ do              // iterate over all other detections
        $o \leftarrow \mathrm{IoU}(\mathbf{b}, \tilde{\mathbf{b}})$                  // calculate intersection over union
        if $o > o_{\text{NMS}}$ then
            $\mathrm{L} \leftarrow \mathrm{L} \setminus \widetilde{\mathrm{D}}$                  // remove less-confident detection
```

It is guaranteed that there are no two detections left in $\mathcal{D}$ with an IoU exceeding o_{NMS}. In sparse scenes without persons occluding other persons, this is desired as only duplicate detections are filtered. However, in dense scenes with many person–person occlusions, also TP detections can be removed, especially if the maximum overlap threshold o_{NMS} is set too small. In conclusion, o_{NMS} should be tuned to achieve a good trade-off between removing most duplicate detections, while eliminating only few TPs.

5.2 Motion Model

In order to achieve a good motion-based association, which means that detections are assigned to tracks based on the estimated track positions derived from the motion of targets, the estimated track positions should be as accurate as possible. The prediction of track positions is the task of the motion model. Formally, a track can be modeled as tuple $\mathrm{T} = (\mathbf{x})$ comprising a state $\mathbf{x}$ that contains the positions of the track on the image. As will be seen in Section 5.2.1, the state $\mathbf{x}$ may include further information about dimensions and velocities. Besides $\mathbf{x}$, a track $\mathrm{T} = (\mathbf{x}, \mathbf{P})$ could also incorporate uncertainties $\mathbf{P}$ about the state quantities or other information, for instance, about the appearance of the tracked target (Section 5.4.2). Given a motion model $\mathrm{MM}(\cdot)$, a track state $\widehat{\mathrm{T}}_t = (\hat{\mathbf{x}}_t, \widehat{\mathbf{P}}_t)$ at time t can be predicted from the state of the previous iteration:

$$\widehat{\mathrm{T}}_t = (\hat{\mathbf{x}}_t, \widehat{\mathbf{P}}_t) = \mathrm{MM}(\mathrm{T}_{t-1}) = \mathrm{MM}(\mathbf{x}_{t-1}, \mathbf{P}_{t-1}). \tag{5.2}$$

While motion paths of persons in natural environments are in general non-linear, the high frame rate available in videos captured for MPT applications allows to approximate the frame-wise motions with a linear model. The omnipresent motion model utilized in MPT is the Kalman filter [Kal60]. Its basic version is described in Section 5.2.1, and two adaptations are treated in Section 5.2.2.

5.2.1 Kalman Filter

The Kalman filter is used to estimate the internal state of a linear dynamic system, which has been discretized in the time domain, on the basis of noisy measurements. Before examining the Kalman filter for the MPT task, its basic formulation and working mechanism are explained. Two steps are performed alternately: the *prediction step* and the *update step*. Those two phases are based on the *system equation* and the *observation equation* of the linear dynamic system, which are described in detail as follows.

Given a multi-dimensional state $\mathbf{x}$, a transition matrix $\mathbf{F}_{t-1}$, a control matrix $\mathbf{B}_{t-1}$ with control vector $\mathbf{u}_{t-1}$, and process noise $\mathbf{w}_{t-1}$, the system equation for the transition from state $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is given by

$$\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{B}_{t-1}\mathbf{u}_{t-1} + \mathbf{w}_{t-1}. \tag{5.3}$$

Note that the process noise $\mathbf{w}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{t-1})$ is assumed to be drawn from a multivariate normal distribution $\mathcal{N}$ with zero mean and covariance matrix $\mathbf{Q}_{t-1}$. The system equation models how the components of the state vector influence each other, how the state is controlled by external inputs, and additionally considers the process noise of the system.

The observation equation describes the relation between an observed measurement vector $\mathbf{z}_t$, the observation matrix $\mathbf{H}_t$, the state $\mathbf{x}_t$, and the measurement noise $\mathbf{v}_t$ at time t:

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t. \tag{5.4}$$

The measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ is also assumed to be drawn from a normal distribution with zero mean and its covariance matrix is denoted by $\mathbf{R}_t$. The observation equation models how the observed measurements are generated from the internal state and also considers the measurement noise.

Based on Equation (5.3), the formulas of the prediction step can be given:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1} + \mathbf{B}_{t-1}\mathbf{u}_{t-1}, \tag{5.5}$$

$$\widehat{\mathbf{P}}_{t|t-1} = \mathbf{F}_{t-1}\widehat{\mathbf{P}}_{t-1}\mathbf{F}_{t-1}^{\mathsf{T}} + \widehat{\mathbf{Q}}_{t-1}. \tag{5.6}$$

The estimated a priori mean of the state $\mathbf{x}$ at time t, given observations up to and including time $t-1$, is denoted by $\hat{\mathbf{x}}_{t|t-1}$. Likewise, the estimated a priori covariance matrix of the state $\mathbf{x}$ is labeled with $\widehat{\mathbf{P}}_{t|t-1}$. It has to be noticed that the covariance matrix of the process noise $\mathbf{Q}_{t-1}$ is not known and must be estimated with a priori information of the system yielding $\widehat{\mathbf{Q}}_{t-1}$.

One can see from Equation (5.6), that the uncertainty of the state estimation increases in the prediction step if the process noise is large. Therefore, it is desired to use measurements to reduce the uncertainty of the estimation in the update step. The formulas of the update step are listed below:

$$\mathbf{K}_t = \widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\intercal(\mathbf{H}_t\widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\intercal + \widehat{\mathbf{R}}_t)^{-1}, \tag{5.7}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}), \tag{5.8}$$

$$\widehat{\mathbf{P}}_t = (\mathbf{I}_n - \mathbf{K}_t\mathbf{H}_t)\widehat{\mathbf{P}}_{t|t-1}. \tag{5.9}$$

Notice that the covariance matrix $\mathbf{R}_t$ of the measurement noise is unknown and has to be estimated yielding $\widehat{\mathbf{R}}_t$ and that $\mathbf{I}_n$ is the identity matrix. $\mathbf{K}_t$ is termed *Kalman gain* and the auxiliary variable $\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}$ is often referred to as *innovation.* The innovation indicates how accurately the estimated a priori mean of the state $\hat{\mathbf{x}}_{t|t-1}$ from the prediction step can explain the measurement $\mathbf{z}_t$ using the observation equation (Equation (5.4)). For a bad prediction, the absolute value of the innovation gets large, while for a good prediction, it gets small. Therefore, the innovation is a measure for the extent of the correction that has to be made for the estimated state from the prediction $\hat{\mathbf{x}}_{t|t-1}$.

The influence of the innovation on the estimated a posteriori mean, i.e., after incorporating the measurement, of the state $\hat{\mathbf{x}}_t$ is controlled by the Kalman gain $\mathbf{K}_t$ as can be seen in Equation (5.8). If the Kalman gain is small, the Kalman filter puts more weight on the prediction than on the measurement and vice versa. Having a closer look at Equation (5.7), this is reasonable because the Kalman gain gets large if the (estimated) uncertainty of the measurement $\widehat{\mathbf{R}}_t$ is low and the Kalman gain gets small if the uncertainty of the measurement $\widehat{\mathbf{R}}_t$ is high. This means that, depending on the uncertainty of the prediction $\widehat{\mathbf{P}}_{t|t-1}$ and the uncertainty of the measurement $\widehat{\mathbf{R}}_t$, the Kalman filter puts more confidence in the one or the other in the update step. That finding can be underlined regarding the following two extreme cases:

- No uncertainty in the measurement: $\widehat{\mathbf{R}}_t = \mathbf{0}$. With Equations (5.7) to (5.9) follows:

$$\begin{aligned}\mathbf{K}_t &= \widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\mathrm{T}(\mathbf{H}_t\widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\mathrm{T} + \mathbf{0})^{-1} \\ &= \widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\mathrm{T}(\mathbf{H}_t^\mathrm{T})^{-1}\widehat{\mathbf{P}}_{t|t-1}^{-1}\mathbf{H}_t^{-1} = \mathbf{H}_t^{-1}, \qquad (5.10)\\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{H}_t^{-1}(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}) = \mathbf{H}_t^{-1}\mathbf{z}_t, \qquad (5.11)\\ \widehat{\mathbf{P}}_t &= (\mathbf{I}_n - \mathbf{H}_t^{-1}\mathbf{H}_t)\widehat{\mathbf{P}}_{t|t-1} = \mathbf{0}. \qquad (5.12)\end{aligned}$$

The Kalman filter relies fully on the measurement and the estimated uncertainty becomes zero.

- Maximum uncertainty in the measurement: $\widehat{\mathbf{R}}_t \to \boldsymbol{\infty}$. With Equations (5.7) to (5.9) follows:

$$\begin{aligned}\mathbf{K}_t &= \lim_{\widehat{\mathbf{R}}_t \to \boldsymbol{\infty}} \widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\mathrm{T}(\mathbf{H}_t\widehat{\mathbf{P}}_{t|t-1}\mathbf{H}_t^\mathrm{T} + \widehat{\mathbf{R}}_t)^{-1} = \mathbf{0}, \qquad (5.13)\\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{0}(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}) = \hat{\mathbf{x}}_{t|t-1}, \qquad (5.14)\\ \widehat{\mathbf{P}}_t &= (\mathbf{I}_n - \mathbf{0}\mathbf{H}_t)\widehat{\mathbf{P}}_{t|t-1} = \widehat{\mathbf{P}}_{t|t-1}. \qquad (5.15)\end{aligned}$$

The Kalman filter relies fully on the prediction and the estimated uncertainty stays the same, i.e., the measurement is not used at all.

Note that Equation (5.9) and the two examples above indicate that, in practical cases, the uncertainty after the update step $\widehat{\mathbf{P}}_t$ is always smaller than the uncertainty after the prediction step $\widehat{\mathbf{P}}_{t|t-1}$.

In the initialization, $\hat{\mathbf{x}}_0$ and $\widehat{\mathbf{P}}_0$ are set, and then, prediction step and update step are alternately repeated if a measurement is available at each discrete time t. If no measurement is available, the update step can be skipped, however, with the consequence that the uncertainty of the estimated state will increase in consecutive prediction steps without any update step.

In the following, the details of the Kalman filter implementation used in this thesis are provided. It is based on the DeepSORT [Woj17] implementation, which has been taken over by many MPT methods [Wan20, Wan21, Zha21, Zha22c]. The Kalman filter is utilized for modeling the motion of the tracked persons with the motion state

$$\mathbf{x} = (x, y, a, h, \dot{x}, \dot{y}, \dot{a}, \dot{h})^{\mathsf{T}}, \tag{5.16}$$

whereby (x,y) is the position on the image, $a = w/h$ and h are the aspect ratio and height of the track bounding box, respectively, and $\cdot$ denotes the temporal derivative.

The detector from Section 5.1 is leveraged for generating a measurement

$$\mathbf{z} = (x, y, a, h)^{\mathsf{T}}. \tag{5.17}$$

Thus, only the first four entries of the state are observable, while the remaining four have to be estimated with the system equation. Since the system contains no control inputs $\mathbf{u}_{t-1}$, the system equation (Equation (5.3)) and the prediction step for the state mean (Equation (5.5)) simplify with $\mathbf{B}_{t-1} = \mathbf{0}$ to

$$\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{w}_{t-1}, \tag{5.18}$$

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1}. \tag{5.19}$$

All other aforementioned equations of the Kalman filter remain identical.

As mentioned earlier, the high frame rate in MPT ensures that only small motion changes occur between two time steps. Therefore, a constant velocity assumption is made. Moreover, the system and observation equations do not change over time, i.e., the time index t can be omitted in the transition matrix $\mathbf{F}_{t-1} = \mathbf{F}$ and observation matrix $\mathbf{H}_t = \mathbf{H}$, which are given below:

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.20}$$

Since the covariance matrices of the process and measurement noise are not known, they have to be estimated in each iteration. The covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$ depend on the depth of the tracked target in the camera frame, since the depth is a good indicator for the extent of movement on the image plane between two time steps. However, there is usually no depth information available in the 2D images. Therefore, the height h of the target's bounding box is leveraged because there is a correlation between height and depth information. Namely, persons close to the camera, i.e., with a small depth, appear larger than persons far away from the camera, i.e., with a large depth. Consequently, the estimated height $\hat{h}_t$ of the state mean $\hat{\mathbf{x}}_t$ and the height h_t of the measurement $\mathbf{z}_t$ are used to compute the estimated process noise covariance $\widehat{\mathbf{Q}}_t$ and estimated measurement noise covariance $\widehat{\mathbf{R}}_t$, respectively:

$$\widehat{\mathbf{Q}}_t = \begin{pmatrix} \alpha\hat{h}_t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha\hat{h}_t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha\hat{h}_t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta\hat{h}_t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta\hat{h}_t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta\hat{h}_t \end{pmatrix}^2 , \tag{5.21}$$

$$\widehat{\mathbf{R}}_t = \begin{pmatrix} \alpha h_t & 0 & 0 & 0 \\ 0 & \alpha h_t & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & \alpha h_t \end{pmatrix}^2 . \tag{5.22}$$

The parameter α is introduced to tune the estimated variances related to the positional variables of the state x, y, and h, while the parameter β is responsible for the velocity variables $\dot{x}$, $\dot{y}$ and $\dot{h}$. In this thesis, $\alpha = 0.05$ and $\beta = 6.25 \cdot 10^{-3}$ are adopted from the DeepSORT [Woj17] implementation, as well as the constant variances related to the aspect ratio a.

In the initialization step, the first measurement $\mathbf{z}_0 = (x_0, y_0, a_0, h_0)$ is used to originally set the estimated state mean $\hat{\mathbf{x}}_0$ and covariance $\widehat{\mathbf{P}}_0$:

$$\hat{\mathbf{x}}_0 = (x_0, y_0, a_0, h_0, 0, 0, 0, 0)^\mathsf{T}, \tag{5.23}$$

$$\widehat{\mathbf{P}}_0 = \begin{pmatrix} 2\alpha h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\alpha h_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\alpha h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10\beta h_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10\beta h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10\beta h_0 \end{pmatrix}^2 . \tag{5.24}$$

While the measurement $\mathbf{z}_0$ is directly copied to the positional entries of the state mean $\mathbf{x}_0$, the velocities are initialized with zero, as no velocity information can be derived from a single measurement. Note that the increased uncertainty of the initial state $\mathbf{x}_0$ is reflected in the higher values of the variances in $\widehat{\mathbf{P}}_0$, especially in those related to the velocities.

5.2.2 Kalman Filter Adaptations

In this section, two small adaptations of the basic Kalman filter in the context of MPT are presented: the NSA Kalman filter from [Du21] and the HP module proposed in a previous work from the author of this thesis [Sta22b].

The idea of the NSA Kalman filter is to leverage the confidence score from the detector as indicator for the measurement noise. A high detection confidence implies a small measurement noise, whereas a low confidence implies a large noise. This should be reflected in the estimation of the measurement covariance matrix. Given a confidence score s_t at time t next to the measurement $\mathbf{z}_t$ and the standard estimated noise covariance $\widehat{\mathbf{R}}_t$ from Equation (5.22), the

NSA measurement covariance $\widetilde{\mathbf{R}}_t$ is calculated as[1]

$$\widetilde{\mathbf{R}}_t = (1 - s_t)^2 \, \widehat{\mathbf{R}}_t. \tag{5.25}$$

If the detector is completely sure about its detection, i.e., $s_t = 1$, the estimated measurement uncertainty will become zero, whereas the estimated uncertainty is not altered for $s_t = 0$.

While the NSA Kalman filter makes changes to the update step, the HP adapts the prediction step. It is found that tracks updated by a measurement with inaccurate height before getting occluded, i.e., with no following measurement for multiple time steps, suffer from a shrinking or growing bounding box during prediction [Sta22b]. This is because the inaccurate measurement distorts the velocity state of the height $\hat{\dot{h}}$ in the update step and since no further measurements are available during occlusion, the bounding box height gets smaller or larger in each prediction step. As a consequence, an association of the track after the occlusion is not possible, if the distortion of the state $\hat{\dot{h}}$ is too severe, because the predicted height will differ significantly from the more accurate height of the newly arriving measurement. To prevent such a behavior, the velocity state of the height is set to zero in the prediction step:

$$\hat{\dot{h}} = 0. \tag{5.26}$$

This ensures that the height of the track is preserved during an occlusion, which simplifies a motion-based association after the occlusion has dissolved.

The performance of the two Kalman filter adaptations will be compared to the standard formulation in Section 5.6.4.

[1] The square in Equation (5.25) does not appear in the paper of [Du21] but in the implementation available at https://github.com/dyhBUPT/StrongSORT (accessed on July 16, 2024).

5.3 Re-Identification Model

Many MPT approaches found in the literature leverage a REID model to extract appearance features from the image regions where persons have been detected [Aha22, Ber19, Du23, Tan17, Woj17]. The features extracted from various person detections are then compared with a similarity measure to perform the association task. This will be elaborated in more detail in Section 5.4.2, while the basic functionality of a REID model is explained next. After that, the model used in the base framework of this thesis is presented.

Given a set of person detections $\mathcal{D} = \{(\mathbf{b}_1, s_1), \dots, (\mathbf{b}_m, s_m)\}$, the corresponding image regions $\mathbf{I}_{\text{reg},1}, \dots, \mathbf{I}_{\text{reg},m}$ are cropped from the image $\mathbf{I}$ using the bounding box information $\mathbf{b}_1, \dots, \mathbf{b}_m$. The image regions are then resized to a fixed size and put into the REID model. It is a CNN that computes a feature vector $\mathbf{f}$ for each input region $\mathbf{I}_{\text{reg}}$. While some sophisticated networks combine multiple local features from different parts of the input region, a typical approach is to extract one global feature vector from the last feature map of the network backbone. This is done by global average pooling (GAP), which takes a three-dimensional feature volume of size $w_\text{f} \times h_\text{f} \times c_\text{f}$ as input, with w_f, h_f, and c_f being the width, height, and channel dimension of the feature volume, respectively. Output of the GAP is a feature vector $\mathbf{f}$ of length c_f, since the average is calculated across the spatial dimensions of the feature volume.

The goal is to train the REID network in a way such that input images of the same person result in *similar* feature vectors with a small distance in the embedding space. Two common loss functions to achieve this goal are presented: the *triplet loss* [Her17] and the *identity loss*. Provided three input images, whereof two depict the identical person and the third shows a different person, the REID network extracts the feature vectors $\mathbf{f}_\text{a}$, $\mathbf{f}_\text{p}$, and $\mathbf{f}_\text{n}$ denoting the anchor, positive, and negative feature, respectively. The anchor feature and positive feature originate from the same person, whereas the negative feature is extracted from the image depicting the other person. Then, the triplet loss $\mathcal{L}_\text{triplet}$ can be calculated as

$$\mathcal{L}_\text{triplet} = \max\left(||\mathbf{f}_\text{a} - \mathbf{f}_\text{p}||_2 - ||\mathbf{f}_\text{a} - \mathbf{f}_\text{n}||_2 + \gamma, 0\right) \tag{5.27}$$

with $\gamma \in [0, \infty)$ being a hyper-parameter termed *margin* and $||\cdot||_2$ representing the Euclidean distance. The triplet loss aims at bringing features from the same person closely together in the embedding space, while pushing features from different persons away from each other. The maximum function prevents the loss becoming negative, and the margin parameter ensures that the loss gets zero only if the negative feature is at least by the value of γ farther away from the anchor feature than the positive feature. Obviously, a training batch has to contain at least two images depicting the same person to enable the use of the triplet loss.

The triplet loss is often combined with the identity loss in the training process of a REID model [Gon22, Her21, Wan18b]. To enable the utilization of the identity loss, the network is extended by a fully-connected (FC) layer and a softmax layer, which are put after the GAP layer that yields the feature vector $\mathbf{f}$ with length c_f. Thus, the FC layer has an input size of c_f. The output size is set to the number of different person identities N_ID available in the training dataset, as the identity loss treats the REID task as a classification problem with N_ID classes. Before calculating the loss, the outputs of the FC layer $p_i, i \in \{1, \dots, N_\mathrm{ID}\}$ are normalized using the softmax function

$$\tilde{p}_i = \frac{\exp(p_i)}{\sum_{j=1}^{N_\mathrm{ID}} \exp(p_j)} \tag{5.28}$$

so that $\tilde{p}_i \in (0,1)$ holds. Given an input image with GT label $g \in \{1, \dots, N_\mathrm{ID}\}$ and class-wise predictions $\tilde{p}_i, i \in \{1, \dots, N_\mathrm{ID}\}$, the identity loss $\mathcal{L}_\mathrm{ID}$ can be computed:

$$\mathcal{L}_\mathrm{ID} = -\sum_{i=1}^{N_\mathrm{ID}} \delta_{ig} \log(\tilde{p}_i). \tag{5.29}$$

Here, δ_{ig} stands for the Kronecker delta, which becomes one if $i = g$ and zero if $i \neq g$. Thus, the identity loss gets low if the correct identity class has been predicted with large probability $\tilde{p}_i$ and vice versa. The identity loss also aims at generating similar features $\mathbf{f}$ from images depicting the same person. Note that the FC layer and softmax layer are only applied in the training phase, while during inference, the network outputs the feature vector **f**. In contrast to the triplet loss, a training batch can contain images of exclusively different persons when using the identity loss.

To improve the performance of the network, REID works [Gon22, Her21, Wan18b] leverage both the triplet loss and the identity loss in the training process. That is also true for the model adopted for this thesis from [He23]. It is termed stronger baseline (SBS) and is a further development of the bag of tricks (BoT) model presented in [Luo19]. Remember that the modularization of the TBD paradigm allows to use any REID model within the base framework. The SBS model has been chosen for two reasons:

- It is designed to achieve a good speed–accuracy trade-off keeping a lightweight network architecture, while using a bag of *training tricks* to improve the performance without additional computational costs.
- It is also utilized in other MPT methods, which enables a fair comparison with those works.

An overview of the SBS network structure is illustrated in Figure 5.3.

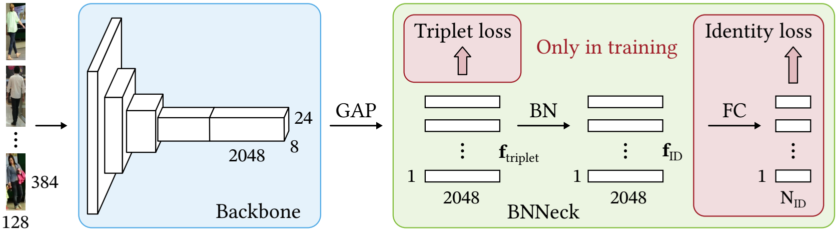


Figure 5.3: Overview of the SBS REID network. Given image crops of persons as input, the backbone extracts feature maps, which are then transformed to feature vectors by GAP. The following BNNeck applies batch normalization (BN) on these features. During training, the triplet loss is leveraged for the non-normalized features, while the identity loss is used together with a FC layer for the normalized features.

A training batch contains $B = PK$ images of size 128×384 pixels with P denoting the number of different persons and K the number of different images per person. The images are put into a modified ResNeSt-50 (S50) [Zha22a] backbone, where the *stride* of the last stage has been changed from 2 to 1 to increase the spatial resolution of the feature map. This comes with very small computational costs, while the performance is significantly increased. With this modification, the overall downsampling factor of the backbone becomes 16 and the size of the last feature volume is 8×24×2048. Consequently, the GAP results in a feature vector $\mathbf{f}$ of length $c_\mathrm{f} = 2048$.

Usually, triplet loss and identity loss are applied on the same feature vector $\mathbf{f}$. In [Luo19], however, it is argued that the targets of the two losses are inconsistent in the embedding space. Therefore, a batch normalization (BN) layer is leveraged within a *BNNeck* to differentiate the two feature vectors $\mathbf{f}_\mathrm{triplet}$ and $\mathbf{f}_\mathrm{ID}$ as can be seen in Figure 5.3. The triplet loss is applied on $\mathbf{f}_\mathrm{triplet}$, which is the output of the GAP layer. Then, BN is performed on $\mathbf{f}_\mathrm{triplet}$ yielding the feature vector $\mathbf{f}_\mathrm{ID}$. Finally, the identity loss is applied on $\mathbf{f}_\mathrm{ID}$. As mentioned before, the FC layer and softmax layer are only used in the training process. During inference, the network outputs the feature vector $\mathbf{f}_\mathrm{ID}$, which embeds valuable information about the appearance of the person depicted on the input image. Consequently, such features are used in the appearance-based matching of the MPT framework.

5.4 Association

In MPT, the task of matching detections of the same persons together to form tracks is termed *association*. One can categorize available approaches into *offline* and *online* methods. In an offline setting, the detections from all images of a video are available, whereas in an online setting, only the detections up to the current time can be used in the association and typically, the tracks are updated in each time step. Offline methods theoretically can achieve better results, since for a time step t, not only information from the past but also from the future can be leveraged. However, they usually suffer from a very high computational complexity, especially for long sequences as the length l of a video enters exponentially into the number of possible assignments. For instance, given a toy example of a video with length $l = 5$ and $n = 4$ person detections in each frame, there are a total of $(n!)^{(l-1)} = 331{,}776$ possible combinations. In contrast, performing the association in each time step in an online manner, there are only $n! \cdot (l-1) = 96$ possibilities. The high complexity of offline methods is one reason why the current SOTA in MPT is dominated by online methods. Furthermore, offline methods cannot be applied in real time and thus have a lower relevance for practical applications. Therefore, this thesis focuses on online association methods following the TBD paradigm, i.e., detections are assigned to tracks in each time step.

After the initialization of tracks, which will be treated in Section 5.5, the current set of detections $\mathcal{D} = \{\mathrm{D}_1, \dots, \mathrm{D}_m\}$ has to be associated with the tracks from the previous time step $\mathcal{T} = \{\mathrm{T}_1, \dots, \mathrm{T}_n\}$. For each detection D_i and each track T_j, a distance $d(\mathrm{D}_i, \mathrm{T}_j)$ is computed and saved in a distance matrix

$$\mathbf{D} = \big(d(\mathrm{D}_i, \mathrm{T}_j)\big)_{i=1,\dots,m;\, j=1,\dots,n} \tag{5.30}$$

of size $m \times n$. Note that the number of detections m and the number of tracks n can differ, as the detector might miss some targets or generate some FP detections. To solve the assignment problem of associating each track with at most one detection, while not utilizing a detection twice, two approaches are common in MPT: *greedy* matching and using the Hungarian algorithm [Kuh55]. In the greedy approach, the detection and track with the smallest distance are matched. Then, the pair with the second smallest distance among the remaining tracks and detections is matched and so on. In contrast, the Hungarian algorithm associates tracks and detections such that the summed distances of matched detections and tracks are minimized. In both strategies, a maximum distance $d_{\max}$ is enforced to prevent unlikely assignments. The two approaches will be evaluated in Section 5.6.5.

Various information about the targets can be represented in the association distance d, whereby mostly motion or appearance cues are leveraged. In the following, common variants for either motion-based or appearance-based association in the context of MPT are presented.

5.4.1 Motion-Based Association

In motion-based association, a motion model is used to predict the track positions (and dimensions) in the current time step t with the information from the previous time step $t-1$ saved in the track states $\{\mathrm{T}_{t-1}\}$. Denoting the prediction function of the motion model with $\mathrm{MM}(\cdot)$, the prediction step can be written as

$$\widehat{\mathcal{T}}_{t|t-1} = \left\{\widehat{\mathrm{T}}^{1}_{t|t-1}, \dots, \widehat{\mathrm{T}}^{n}_{t|t-1}\right\} = \left\{\mathrm{MM}(\mathrm{T}_{t-1}) \,|\, \mathrm{T}_{t-1} \in \mathcal{T}_{t-1}\right\} \tag{5.31}$$

with the predicted tracks $\widehat{\mathcal{T}}_{t|t-1}$ and the tracks from the previous time step $\mathcal{T}_{t-1}$. The predicted tracks $\widehat{\mathcal{T}}_{t|t-1}$ are then compared with the current detections $\mathcal{D}_t$ leveraging a spatial distance function. When using the Kalman filter from Section 5.2.1 as motion model, the predicted track state $\widehat{\mathrm{T}}_{t|t-1}$ contains the eight-dimensional mean vector $\hat{\mathbf{x}}_{t|t-1} = (\hat{x}, \hat{y}, \hat{a}, \hat{h}, \hat{\dot{x}}, \hat{\dot{y}}, \hat{\dot{a}}, \hat{\dot{h}})^{\mathsf{T}}$ and its covariance matrix $\widehat{\mathbf{P}}_{t|t-1}$ from Equations (5.5) and (5.6):

$$\widehat{\mathrm{T}}_{t|t-1} = (\hat{\mathbf{x}}_{t|t-1}, \widehat{\mathbf{P}}_{t|t-1}). \tag{5.32}$$

A simple approach to get a motion-based distance d_{mot} between such a predicted track $\widehat{\mathrm{T}}$ and a detection $\mathrm{D} = (\mathbf{b}, s)$ is to calculate the Euclidean distance between its center positions

$$d_{\mathrm{L2}}(\widehat{\mathrm{T}}, \mathrm{D}) = \|(x_{\mathrm{T}}, y_{\mathrm{T}})^{\mathsf{T}} - (x_{\mathrm{D}}, y_{\mathrm{D}})^{\mathsf{T}}\|_2. \tag{5.33}$$

Here, $(x_{\mathrm{T}}, y_{\mathrm{T}})^{\mathsf{T}}$ is taken from $\hat{\mathbf{x}}$ of $\widehat{\mathrm{T}}$ and $(x_{\mathrm{D}}, y_{\mathrm{D}})^{\mathsf{T}}$ comes from the bounding box $\mathbf{b}$ of D. Notice that the time indices have been omitted for clarity.

Another possibility is to use the (squared) Mahalanobis distance between the track's state (mean and covariance) and the detection in the measurement space. Next to the positional information, the bounding box dimensions are considered as well as the uncertainties of the estimated quantities. To calculate the Mahalanobis distance, mean $\hat{\mathbf{x}}_{t|t-1}$ and covariance $\widehat{\mathbf{P}}_{t|t-1}$ of the track are projected into the measurement space yielding $\mathbf{y}$ and $\mathbf{S}$, respectively:

$$\mathbf{y} = \mathbf{H}\hat{\mathbf{x}}_{t|t-1}, \tag{5.34}$$

$$\mathbf{S} = \mathbf{H}\widehat{\mathbf{P}}_{t|t-1}\mathbf{H}^\top + \widehat{\mathbf{R}}_t. \tag{5.35}$$

Remember that $\mathbf{H}$ denotes the observation matrix from Equation (5.20) and $\widehat{\mathbf{R}}_t$ is the estimated measurement noise at time t from Equation (5.22). Given also the detection $\mathbf{d} = (x_\mathrm{D}, y_\mathrm{D}, a_\mathrm{D}, h_\mathrm{D})^\top$ in the measurement space (confidence score s is omitted), the squared Mahalanobis distance d_Mah is calculated as follows:

$$d_\mathrm{Mah}(\widehat{\mathrm{T}}, \mathrm{D}) = (\mathbf{d} - \mathbf{y})^\top \mathbf{S}^{-1}(\mathbf{d} - \mathbf{y}). \tag{5.36}$$

Note that $\mathbf{S}$ is positive definite in practice, so the existence of $\mathbf{S}^{-1}$ is ensured.

The most used distance measure in motion-based MPT is the IoU distance d_IoU. Let $\mathbf{b}_\mathrm{T} = (x_\mathrm{T}, y_\mathrm{T}, w_\mathrm{T}, h_\mathrm{T})^\top$ be the bounding box derived from the track state $\widehat{\mathrm{T}}$ and $\mathbf{b}_\mathrm{D}$ the box taken from detection D. Further, let A_T and A_D denote the bounding box area of $\mathbf{b}_\mathrm{T}$ and $\mathbf{b}_\mathrm{D}$, respectively. Then, the IoU distance is computed using the ratio of intersection ($\cap$) over union ($\cup$):

$$d_\mathrm{IoU}(\widehat{\mathrm{T}}, \mathrm{D}) = 1 - \mathrm{IoU}(\mathbf{b}_\mathrm{T}, \mathbf{b}_\mathrm{D}) = 1 - \frac{|A_\mathrm{T} \cap A_\mathrm{D}|}{|A_\mathrm{T} \cup A_\mathrm{D}|}. \tag{5.37}$$

Besides positional information, the IoU also takes the bounding box dimensions and the aspect ratio implicitly into account. The performance of the three presented motion-based distances d_L2, d_Mah, and d_IoU used in the MPT task is compared in Section 5.6.5.

Independent from the distance function utilized, whenever a detection $\mathbf{d}_t = (x_t, y_t, a_t, h_t)^\top$ is assigned to a track, the track's state $\widehat{\mathrm{T}}_{t|t-1} = (\hat{\mathbf{x}}_{t|t-1}, \widehat{\mathbf{P}}_{t|t-1})$ is updated using the measurement $\mathbf{z}_t = \mathbf{d}_t$ yielding $\widehat{\mathrm{T}}_t = (\hat{\mathbf{x}}_t, \widehat{\mathbf{P}}_t)$ according to the Kalman filter update Equations (5.8) and (5.9). If the NSA Kalman filter adaptation is applied, the confidence score s_t of the detection is also leveraged as per Equation (5.25). The Kalman filter prediction in the next iteration $t+1$ is then based on the updated track state $\widehat{\mathrm{T}}_t$.

5.4.2 Appearance-Based Association

In appearance-based association, a REID model is used to extract appearance features from the detected image regions. Features from different detections are then compared in order to perform the association. Formally, let $\mathrm{D} = (\mathbf{b}, s, \mathbf{f})$ be a detection comprising an appearance feature vector $\mathbf{f}$ extracted by the REID network in addition to the bounding box $\mathbf{b}$ and confidence score s. Moreover, the state of a track (Equation (5.32)) is extended by a list of associated feature vectors $\mathrm{F}_t = [\mathbf{f}_{t-k}, \dots, \mathbf{f}_t]$:

$$\widehat{\mathrm{T}}^t = (\hat{\mathbf{x}}_t, \widehat{\mathbf{P}}_t, \mathrm{F}_t). \tag{5.38}$$

While the motion states $\hat{\mathbf{x}}_t$ and $\widehat{\mathbf{P}}_t$ as well as the detection bounding box $\mathbf{b}$ and score s still are used in the prediction and update step of the Kalman filter, the appearance-based association uses only the features F_t and $\mathbf{f}$. More precisely, let F_T be the list of feature vectors of a track T (time index t omitted for clarity) and $\mathbf{f}_\mathrm{D}$ the feature vector of a detection D. Then, the appearance-based distance d_app between track T and detection D

$$d_\mathrm{app}(\mathrm{T}, \mathrm{D}) = d_\mathrm{app}(\mathrm{F}_\mathrm{T}, \mathbf{f}_\mathrm{D}) \tag{5.39}$$

can be calculated using only the feature information.

Various approaches to compute the appearance distance exist. Before presenting methods for determining the distance between a list of feature vectors and a single feature vector, the two basic distance measures for comparing two single feature vectors in MPT are described.

First, the Euclidean distance d_L2 between two feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ can be leveraged:

$$d_\mathrm{L2}(\mathbf{f}_1, \mathbf{f}_2) = ||\mathbf{f}_1 - \mathbf{f}_2||_2. \tag{5.40}$$

Second, the cosine distance d_{cos} is another possibility for comparing two feature vectors:

$$d_{\text{cos}}(\mathbf{f}_1, \mathbf{f}_2) = 1 - \frac{\mathbf{f}_1^{\top}\mathbf{f}_2}{||\mathbf{f}_1||_2 \cdot ||\mathbf{f}_2||_2}. \tag{5.41}$$

While the Euclidean distance is not bounded, i.e., $d_{\text{L2}} \in [0, \infty)$, for the cosine distance, $d_{\text{cos}} \in [0, 2]$ holds.

Next, several approaches for computing the appearance distance d_{app} are introduced. In [Woj17], a feature bank of size N_{F} is saved with each track. Specifically, the length of the list with associated feature vectors F is limited to N_{F}. When the list is full and another feature vector $\mathbf{f}$ is assigned to the track, the oldest feature vector of the list is removed. Given a track T with feature bank $\text{F}_{\text{T}} = [\mathbf{f}_1, \dots, \mathbf{f}_N], N \leq N_{\text{F}}$ and a detection D with feature $\mathbf{f}_{\text{D}}$, the minimum cosine distance between $\mathbf{f}_{\text{D}}$ and all track features $\mathbf{f}_j \in \text{F}_{\text{T}}$ is utilized as appearance distance:

$$d_{\text{min}}(\text{T}, \text{D}) = d_{\text{min}}(\text{F}_{\text{T}}, \mathbf{f}_{\text{D}}) = \min_{j=1,\dots,N} \left\{ d_{\text{cos}}(\mathbf{f}_j, \mathbf{f}_{\text{D}}) \right\}. \tag{5.42}$$

Alternatively, one can also leverage the mean distance instead, as in a previous work of this thesis' author [Sta23b]:

$$d_{\text{mean}}(\text{T}, \text{D}) = \frac{1}{N} \sum_{j=1}^{N} d_{\text{cos}}(\mathbf{f}_j, \mathbf{f}_{\text{D}}). \tag{5.43}$$

Naturally, it is beneficial to use features from multiple time steps rather than using only the last associated feature vector of a track, which would correspond to $N_{\text{F}} = 1$. However, this comes with an increased computational burden because the distance function must be evaluated N_{F} times. While some computation time can be saved with the cosine distance, as the normalization in the denominator of Equation (5.41) has only to be computed once for each feature vector $\mathbf{f}_j$, all calculations of the Euclidean distance (Equation (5.40)) have to be performed N_{F} times. Still, the nominator of Equation (5.41) has to be calculated N_{F} times for the cosine distance. As a consequence, the association is slowed down with a growing size of the feature bank.

A different approach of incorporating appearance information of multiple time steps is proposed in [Wan20] and adopted in subsequent works [Aha22, Du23, Jun24, Ren23]. In contrast of maintaining multiple features for each track, a single feature vector $\mathbf{f}_t^{\mathrm{T}}$ is updated iteratively by an EMA according to

$$\mathbf{f}_t^{\mathrm{T}} = \phi \mathbf{f}_{t-1}^{\mathrm{T}} + (1 - \phi)\mathbf{f}_t^{\mathrm{D}}, \tag{5.44}$$

where $\mathbf{f}_t^{\mathrm{D}}$ denotes the detection feature assigned to the track at time t and $\phi \in [0, 1]$ is the weight of the previous feature vector. Consequently, the track state from Equation (5.38) turns to $\widehat{\mathrm{T}}^t = (\hat{\mathbf{x}}_t, \widehat{\mathbf{P}}_t, \mathbf{f}_t^{\mathrm{T}})$. Following [Wan20], the appearance distance between a track T with feature vector before the update step $\mathbf{f}_{t-1}^{\mathrm{T}}$ and a detection D with feature vector $\mathbf{f}_t^{\mathrm{D}}$ can be calculated as

$$d_{\mathrm{EMA}}(\mathrm{T}, \mathrm{D}) = d_{\cos}(\mathbf{f}_{t-1}^{\mathrm{T}}, \mathbf{f}_t^{\mathrm{D}}), \tag{5.45}$$

i.e., the cosine distance is used. The performance of the various appearance-based matching methods will be evaluated in Section 5.6.5.

5.5 Track Management

So far, it has already been covered how detections are generated, how they are associated with tracks, and how tracks are propagated and updated with a motion model. What is left is to answer the following questions:

- How are tracks initialized?
- What happens with detections and tracks that are not matched in the association?
- When are tracks terminated?

These questions are treated by the *track management*.

Because of the imperfection of the detector, the set of detections $\mathcal{D} = \{\mathrm{D}_1, \dots, \mathrm{D}_m\} = \{(\mathbf{b}_1, s_1), \dots, (\mathbf{b}_m, s_m)\}$ contains, next to TPs that should be leveraged in the tracking process, also FPs that should be removed. While a detection with high confidence s is likely a TP, a detection with low confidence is probably a FP. Therefore, a TBD method usually filters the set of detections with a minimum confidence threshold s_{track} before the association, i.e., only detections with $s \geq s_{\text{track}}$ can be assigned to so-far tracked targets, while the others are deleted.

When persons appear in the video for the first time, the corresponding detections will not fit to the tracks and remain unassigned in the association. These detections are then used for initializing new tracks. More precisely, let $\mathbf{d}_t$ be the bounding box of an unassigned detection $\mathrm{D}_t^{\mathrm{u}}$ in the Kalman filter measurement space. Then, a new track $\widehat{\mathrm{T}}_t = (\hat{\mathbf{x}}_t, \widehat{\mathbf{P}}_t)$ is initialized according to Equations (5.23) and (5.24) using the unassigned detection as first measurement $\mathbf{z}_0 = \mathbf{d}_t$.

Besides the minimum confidence threshold, many TBD approaches apply a *continuity* requirement such that a target has to be detected in n_{init} consecutive frames for a successful track initialization [Aha22, Bew16, Du23, Woj17, Zha22c]. This aims at preventing ghost tracks being started from FP detections that occur only in single frames. To achieve this, a *tentative* track state is introduced: When a new track T^{n} is initialized by an unassigned detection, it is deemed tentative until it has been associated with another $n_{\text{init}} - 1$ detections in the next time steps. If during this period, no detection can be assigned to the new track, this tentative track is deleted. Otherwise, it turns *active*, which can be indicated by the superscript: $\mathrm{T}^{\mathrm{n}} \rightarrow \mathrm{T}^{\mathrm{a}}$. Notice that in the evaluation of an online MPT method, only active tracks are typically considered.

To enable the continuation of tracks during failures of the detector or occlusions, tracks without an assigned detection after the association are not immediately terminated but turn *inactive*: $\mathrm{T}^{\mathrm{a}} \rightarrow \mathrm{T}^{\mathrm{i}}$. An inactive track T^{i} is kept for a maximum time period $i_{\max}$ without assigned detection before termination. While deemed inactive, a track is consistently propagated in the Kalman filter prediction step (Equations (5.5) and (5.6)). If a detection is matched to the inactive track in the association during this time period, the track is re-activated: $\mathrm{T}^{\mathrm{i}} \rightarrow \mathrm{T}^{\mathrm{a}}$. The influence of the parameters related to the track management s_{track}, n_{init}, and $i_{\max}$ will be evaluated in the next section.

5.6 Evaluation

As all modules of the base tracking framework have been introduced, an overview of the whole pipeline is given in the following. After that, details about the implementation are provided, before the single modules are evaluated and the results are summarized.

5.6.1 Pipeline Overview

The interplay of the base tracking components is illustrated in Figure 5.4.

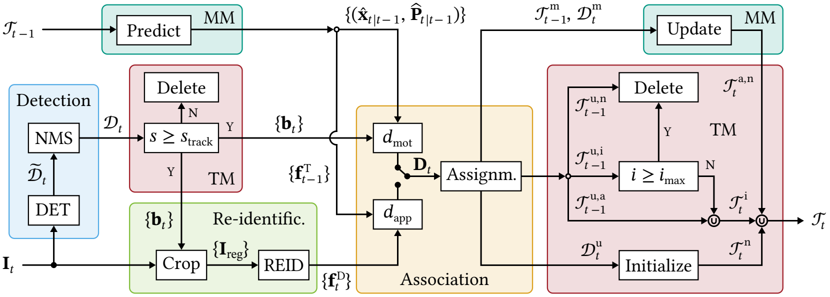

$\mathcal{T}_{t-1}$
Predict
MM
$\{(\hat{\mathbf{x}}_{t|t-1}, \hat{\mathbf{P}}_{t|t-1})\}$
$\mathcal{T}^{\mathrm{m}}_{t-1}, \mathcal{D}^{\mathrm{m}}_{t}$
Update
MM
Detection
Delete
N
$\mathcal{D}_t$
NMS
$s \geq s_{\mathrm{track}}$
Y
$\{\mathbf{b}_t\}$
d_{mot}
$\widetilde{\mathcal{D}}_t$
Y
TM
$\{\mathbf{f}^{\mathrm{T}}_{t-1}\}$
$\mathbf{D}_t$
Assignm.
DET
$\{\mathbf{b}_t\}$
Re-identific.
d_{app}
$\mathbf{I}_t$
$\{\mathbf{I}_{\mathrm{reg}}\}$
Crop
REID
$\{\mathbf{f}^{\mathrm{D}}_{t}\}$
Association
Delete
$\mathcal{T}^{\mathrm{a,n}}_{t}$
$\mathcal{T}^{\mathrm{u,n}}_{t-1}$
Y
TM
$\mathcal{T}^{\mathrm{u,i}}_{t-1}$
N
$i \geq i_{\mathrm{max}}$
$\mathcal{T}^{\mathrm{u,a}}_{t-1}$
$\mathcal{T}^{\mathrm{i}}_{t}$
$\mathcal{T}_{t}$
$\mathcal{D}^{\mathrm{u}}_{t}$
Initialize
$\mathcal{T}^{\mathrm{n}}_{t}$

Figure 5.4: Overview of the base framework with its five modules detection, motion model (MM), re-identification, association, and track management (TM). The switch symbol indicates that either motion or appearance information is used in the association.

In each iteration of the TBD pipeline, an updated set of tracks $\mathcal{T}_t$ is computed given the image of the current time step $\mathbf{I}_t$ and the set of tracks from the previous iteration $\mathcal{T}_{t-1}$. First, the detector (DET) generates a set of preliminary detections $\widetilde{\mathcal{D}}_t$ that is filtered by NMS yielding $\mathcal{D}_t$. As part of the track management, low-confidence detections with $s < s_{\mathrm{track}}$ are deleted, while the bounding boxes $\{\mathbf{b}_t\}$ of the high-confidence detections are leveraged in the association or the REID module. In the latter, they are used to crop the image patches $\{\mathbf{I}_{\mathrm{reg}}\}$ from the input image $\mathbf{I}_t$. These image regions $\{\mathbf{I}_{\mathrm{reg}}\}$ go into the REID model that computes a set of appearance features $\{\mathbf{f}^{\mathrm{D}}_t\}$ for the detections. Before comparing the motion or appearance cues of the current detections to the tracks from the last time step, the tracks $\mathcal{T}_{t-1}$ are predicted with the motion model to get their estimated positions in the current frame.

For the association, the base framework supports two variants: motion-based and appearance-based matching. This is indicated by the switch symbol in Figure 5.4. In the motion-based association, the predicted track positions (and uncertainties) $\{(\hat{\mathbf{x}}_{t|t-1}, \widehat{\mathbf{P}}_{t|t-1})\}$ are compared against the detection boxes $\{\mathbf{b}_t\}$ with a motion distance d_{mot}. Note that in this case, the REID module is not used. In the appearance-based association, however, the detection features $\{\mathbf{f}_t^{\text{D}}\}$, which come from the REID model, are compared against the track features $\{\mathbf{f}_{t-1}^{\text{T}}\}$ with an appearance distance d_{app}. All distances are summarized in the distance matrix $\mathbf{D}_t$, and the assignment problem is solved with the Hungarian algorithm, or a greedy matching is performed.

Depending on whether a track or detection has been assigned with its counterpart, different actions are carried out. The set of matched detections $\mathcal{D}_t^{\text{m}}$ is leveraged to update the matched tracks $\mathcal{T}_{t-1}^{\text{m}}$ with the Kalman filter motion model. This yields a set $\mathcal{T}_t^{\text{a,n}}$ with updated active and tentative tracks. The unmatched detections $\mathcal{D}_t^{\text{u}}$ and tracks $\mathcal{T}_{t-1}^{\text{u}}$ are handled according to the track management as follows. Each unmatched detection of $\mathcal{D}_t^{\text{u}}$ initializes a track yielding a set of new tracks $\mathcal{T}_t^{\text{n}}$. Depending on the state of the unmatched tracks $\mathcal{T}_{t-1}^{\text{u}}$ (active, inactive, tentative), they are treated differently. Tentative tracks that are unmatched $\mathcal{T}_{t-1}^{\text{u,n}}$ are deleted. Unmatched inactive tracks $\mathcal{T}_{t-1}^{\text{u,i}}$ are kept if their inactive time i does not exceed the *inactive patience* i_{max} and deleted otherwise. Together with the unmatched active tracks $\mathcal{T}_{t-1}^{\text{u,a}}$, the kept inactive ones build the set of inactive tracks $\mathcal{T}_t^{\text{i}}$. Finally, the updated set of tracks $\mathcal{T}_t = \mathcal{T}_t^{\text{a,n}} \cup \mathcal{T}_t^{\text{i}} \cup \mathcal{T}_t^{\text{n}}$ is the union of the aforementioned track sets.

5.6.2 Implementation Details

This section briefly describes some implementation details of the tracking components contained in the base framework. As described in Section 4.3, the YOLOX-X detector has been trained on the combination of MOT17 train half and CH train/val following [Aha22, Cao23, Cet23, Du23, Jun24, Sun21a, Wu21, Zen22, Zha22c, Zho20], for application on MOT17 val. Its model weights are taken over from [Zha22c], where the training details can be found. Unless otherwise stated, the overlap threshold of the NMS is set to $o_{\text{NMS}} = 0.7$.

For the SBS REID model, the GT from the first half of the videos from the MOT17 training set is leveraged to generate a dataset for person REID. More precisely, image regions of the persons are cropped in each video frame using the bounding box information, while the identity label is saved next to every image patch. This results in a total of 74,455 images of $N_{\text{ID}} = 487$ different persons. Consequently, the number of output neurons in the last FC layer of the REID model is 487 during training. P and K are set to 4 and 16, respectively, leading to a batch size of $B = 64$. Due to a comparison with BoT-SORT [Aha22] later in this thesis, the model weights are adopted from their work. Further training details can be looked up in the corresponding paper.

When using the EMA update strategy for the track features according to Equation (5.44), ϕ is set to 0.9 following [Aha22, Du23, Wan20]. The impact of all other tracking parameters and different design choices within the base framework is evaluated on MOT17 val as described in the next sections.

5.6.3 Track Management

To evaluate the three parameters of the track management s_{track}, n_{init}, and i_{max}, the motion-based IoU distance d_{IoU} is leveraged for association and the Kalman filter with both adaptations from Section 5.2.2 is applied as motion model. If not otherwise stated, initialized tracks immediately turn active ($n_{\text{init}} = 1$) and the inactive patience i_{max} is set to 1.5 s. Note that a time period is given instead of a number of frames because the MOT17 dataset contains videos of varying frame rates (Section 4.1.1).

The confidence threshold s_{track} is applied to filter the set of detections for the tracking process, since the confidence s of a detection is an indicator whether it is a TP or a FP and only correct detections should be used. To ablate the influence of this track threshold, multiple runs with varying choices of s_{track} have been performed. The results are visualized in Figure 5.5.

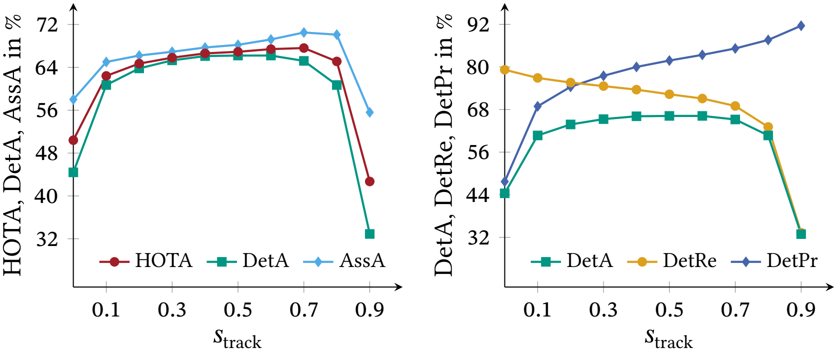


Figure 5.5: Influence of s_{track} in the IoU base framework on MOT17 val. Smaller values lead to a higher DetRe but lower DetPr and vice versa (right). A compromise yields the best DetA, which holds also true for AssA and HOTA (left). Note that the vertical axes do not start at zero and have a different scale and that DetA is shown in both plots.

Remember that HOTA is the geometric mean of DetA and AssA (Equation (4.10)) and that DetA can be decomposed into its submeasures DetRe and DetPr (Equation (4.15)). The track threshold has the largest impact on DetA. Setting a low threshold leads to high DetRe but low DetPr, while a high threshold leads to low recall and high precision. Naturally, DetA has a large influence on AssA since the difficulty of the association task heavily depends on the quality of the provided detections. Thus, AssA also goes up for an increasing s_{track} up to an optimal value and then falls again. The best performance measured in HOTA is achieved for $s_{\text{track}} = 0.7$, which is kept for all following experiments of the base framework.

The idea behind the track initialization utilizing a tentative track state is to suppress FP detections that only occur in single frames. Results with a varying number of n_{init} are given in Table 5.1.

Table 5.1: Influence of n_{init} in the IoU base framework on MOT17 val. Use of tentative tracks enhances DetPr and AssPr but reduces DetRe, which overall leads to a lower HOTA.

n_{init}	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
1	**67.6**	**65.2**	**70.5**	**69.0**	85.2	**75.0**	84.7
2	67.5	65.0	**70.5**	68.6	85.5	**75.0**	84.8
3	67.3	64.7	**70.5**	68.1	**85.7**	**75.0**	**84.9**

For $n_{\text{init}} = 2$ and $n_{\text{init}} = 3$, DetPr and AssPr are improved w.r.t. the baseline ($n_{\text{init}} = 1$), which indicates that indeed FP detections are removed. However, a decreasing DetRe shows that also TP detections are removed from the resulting tracks leading in total to a lower HOTA. It is hypothesized that the strong YOLOX-X detector generates only few high-confidence FPs such that the introduced initialization technique yields no improvements, in contrast to other tracking frameworks with worse detectors, where thresholds $n_{\text{init}} > 1$ are applied [Bew16, Wan20, Woj17]. Consequently, $n_{\text{init}} = 1$ is set for the base framework.

The inactive patience i_{max} determines for how long a track is kept inactive without assigned detection. Its impact on the tracking performance of the IoU base framework can be studied in Figure 5.6.

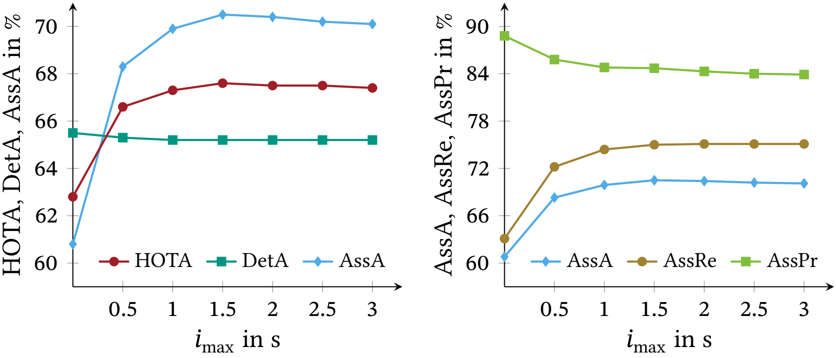


Figure 5.6: Influence of i_{max} in the IoU base framework on MOT17 val. HOTA is mainly affected by a changing AssA, as DetA is nearly constant (left). Keeping tracks longer inactive is beneficial for AssRe, at the cost of AssPr (right). Note that the vertical axes do not start at zero and have a different scale and that AssA is shown in both plots.

Recall that AssA can be decomposed into AssRe and AssPr according to Equation (4.18). One can see that AssRe largely improves up to $i_{\text{max}} = 1.5\,\text{s}$ and then starts to saturate, as the uncertainty of the predicted motion states increases for longer time periods making a successful association less likely. This is also expressed in AssPr, which decreases with a higher inactive patience. Furthermore, the small decrease in DetA can also be explained with the decreasing accuracy of predicted motion states: If a predicted track with a low accuracy is updated with a reasonable detection, the resulting bounding box will be not as accurate as using the box of the detection directly, what would be done in the initialization of a new track if the corresponding track would have already been terminated. The best compromise between AssRe and AssPr is achieved with $i_{\text{max}} = 1.5\,\text{s}$, which is kept for following experiments. With this setting, HOTA is enhanced by 5.8 points w.r.t. $i_{\text{max}} = 0$ demonstrating the high importance of utilizing inactive tracks in MPT.

5.6.4 Motion Model

The two Kalman filter adaptations presented in Section 5.2.2 are evaluated exemplarily with the IoU base framework presented in the previous section. As a reminder, the NSA Kalman filter leverages the detection confidence to improve the estimated measurement noise and thus the update step of the Kalman filter, while the HP module aims at a better prediction step. Experimental results with the two adaptations are summarized in Table 5.2.

Table 5.2: Influence of Kalman filter adaptations in the IoU base framework on MOT17 val. Both the NSA Kalman filter and the proposed HP module improve all tracking measures. Moreover, their combination leads to further gains in HOTA and its submeasures.

NSA	HP	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
✗	✗	66.9	64.7	69.6	68.6	84.7	74.2	84.2
✓	✗	67.3	**65.2**	69.8	**69.0**	**85.2**	74.3	84.4
✗	✓	67.2	64.8	70.2	68.6	84.8	74.5	**84.7**
✓	✓	**67.6**	**65.2**	**70.5**	**69.0**	**85.2**	**75.0**	**84.7**

Adaptively changing the measurement noise according to the detection confidence, the NSA Kalman filter notably enhances DetA, which improves also AssA slightly. On the other hand, HP mainly increases AssA because of the enhanced accuracy of the predicted track boxes that simplify the association task. A further plus in HOTA when combining NSA Kalman filter with HP shows that the two modules yield complementary improvements. W.r.t. the standard Kalman filter formulation, HOTA is increased by 0.7 points, without noticeable computational overhead.

Next to the height, one could also keep the aspect ratio fixed during Kalman filter prediction. Experimental results are given in Table 5.3.

Table 5.3: Keeping different Kalman filter motion states fixed during prediction. Only the HP results in an increased HOTA compared to the baseline.

Fix state	None	Height (HP): $\dot{h} = 0$	Aspect ratio: $\dot{a} = 0$	Both: $\dot{h} = \dot{a} = 0$
HOTA	66.9	**67.2**	66.9	66.9

Keeping the aspect ratio fixed yields the same HOTA as the standard Kalman filter. The same is true when $\dot{h} = \dot{a} = 0$ is set, which means that both height and width of the bounding box are fixed during prediction. However, the width should be variable since in MPT, persons moving in x-direction of the image plane lead to a varying width on the image due to the motion of legs (and arms). In conclusion, the experimental results indicate that it is only reasonable to fix the height of bounding boxes during prediction.

5.6.5 Association

Before different motion- and appearance-based distance functions are evaluated, the two algorithms for solving the assignment problem (Hungarian and greedy assignment) are compared. Moreover, the influence of the maximum association distance on the tracking performance is analyzed.

Table 5.4 gives a comparison of the IoU base framework results when using either the Hungarian algorithm or a greedy assignment.

Table 5.4: Hungarian vs. greedy assignment in the IoU base framework on MOT17 val. Minimizing the overall costs for all assignments, the Hungarian method performs better than the greedy approach. As expected, the same DetA is obtained.

Assignment	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
Hungarian	**67.6**	**65.2**	**70.5**	**69.0**	**85.2**	**75.0**	**84.7**
Greedy	67.4	**65.2**	70.2	**69.0**	**85.2**	74.8	84.4

As expected, the measures related to the detection accuracy are equal, since the same set of detections is used in the association and also all other tracking parameters are identical. Both AssRe and AssPr are higher for the Hungarian algorithm leading to an overall better tracking performance measured in HOTA. An advantage of the greedy assignment is that it is faster. However, the computational burden of the whole tracking framework is dominated by the detection and REID model (Section 7.1), so a small decrease in processing time, when using the greedy assignment compared to the Hungarian algorithm, is mostly negligible. For these reasons, the Hungarian algorithm is leveraged to solve the assignment problem in the remainder of this work.

The Hungarian method assigns detections to tracks such that the overall costs, i.e., the sum of distances, are minimized. To prevent the association of detections and tracks with a too large distance, the maximum distance threshold d_{max} is applied. Its influence on the tracking performance of the IoU base framework is depicted in Figure 5.7.

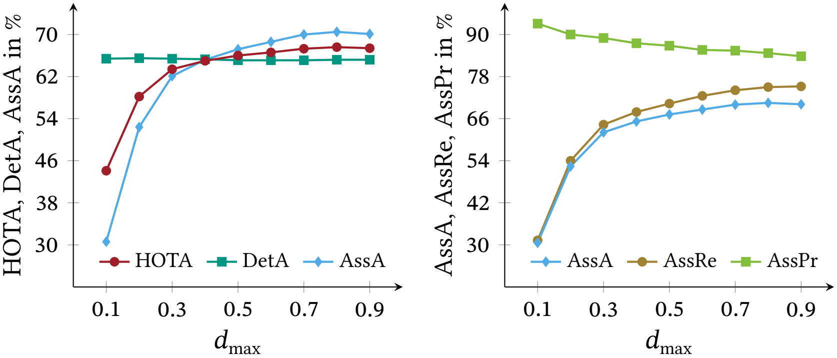


Figure 5.7: Influence of d_{max} in the IoU base framework on MOT17 val. A larger threshold allows more associations such that AssRe increases but AssPr decreases (right). The best overall performance is achieved when allowing associations with small IoU, i.e., with a large distance threshold (left). Notice that the vertical axes do not start at zero and have a different scale and that AssA is shown in both plots.

Allowing a larger distance for matching, AssRe goes up while AssPr decreases. The best compromise is achieved with $d_{\text{max}} = 0.8$, which means that a minimum IoU of $1 - d_{\text{max}} = 0.2$ between a detection and a track is required for association. The slight drop in DetA when enlarging d_{max} can again be explained with the uncertainty of predicted motion states: A small IoU between predicted track and detection box that belong to the same target will mostly occur for inactive tracks that have been predicted for a while without state update. Looking at the HOTA values, one observes that d_{max} has a large influence on the tracking performance and thus should be tuned carefully.

Motion-Based Association

So far, only the IoU distance has been used for determining the similarity between detections and tracks. Table 5.5 compares results generated with IoU distance d_{IoU}, Mahalanobis distance d_{Mah}, and Euclidean distance d_{L2} as introduced in Section 5.4.1.

Table 5.5: Comparison of motion-based distances in the base framework on MOT17 val. IoU works much better than L2 and Mahalanobis distance.

d	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
d_{L2}	64.3	65.1	64.0	68.9	85.1	67.7	83.7
d_{Mah}	63.3	65.0	62.1	68.9	85.1	65.1	75.0
d_{IoU}	**67.6**	**65.2**	**70.5**	**69.0**	**85.2**	**75.0**	**84.7**

Using the Euclidean distance between the center points of tracks and detections yields worse results as the IoU, which additionally takes size and aspect ratio of objects into account. Although the Mahalanobis distance leverages such information, the results are much worse than using the IoU and even worse in comparison to the Euclidean distance. As already stated by the authors of DeepSORT [Woj17], one of the most popular MPT methods that uses the Mahalanobis distance, the Kalman filter provides only a rough estimate of the object location if the state uncertainty is high. Furthermore, it has been shown by previous works of this thesis' author [Sta23a, Sta23b] that IoU-based distance measures perform better than the Mahalanobis distance in MPT, where tracking is performed in the image space. Utilized by many SOTA approaches [Aha22, Du23, Jun24, Men23, Zha22c], the IoU distance can be regarded as the standard measure for motion-based association in MPT.

Appearance-Based Association

Next to the motion of persons, their appearance is a valuable information for the association task. Typically, a REID model is leveraged to extract appearance features from person detections that are then compared against each other with a distance function d_{app}. Note that this comes with additional computational expenses that usually cannot be neglected, especially in real-world applications. The runtime will be analyzed comprehensively in Chapter 7, but the focus is put on the performance of the appearance-based association in the following.

Two measures are common in MPT for computing the distance between the appearance feature vectors: the Euclidean distance d_{L2} and the cosine distance d_{cos}. To compare the two measures, experiments are conducted on MOT17 val with a feature bank size of $N_{\text{F}} = 30$, which corresponds to a time period of one second in most of the sequences. For these experiments, the minimum distance d_{min} among all pairs between a track's feature vectors and the detection feature vector is used as association distance. Table 5.6 depicts the results.

Table 5.6: L2 vs. cosine distance in the appearance base framework on MOT17 val. Similar results are obtained with both distance measures.

d	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
d_{L2}	67.4	**65.4**	70.0	**69.1**	**85.4**	74.9	**84.6**
d_{cos}	**67.5**	65.3	**70.1**	**69.1**	85.3	**75.0**	84.4

The evaluation shows that both measures perform similarly. However, when computing multiple cosine distances, some computation time can be saved since the normalization term has only to be computed once as explained in Section 5.4.2. Thus, the calculation of the cosine distances is in total faster, so d_{cos} is used in the following experiments.

Next, the influence of the feature bank size N_F on the performance is analyzed. In addition, the mean distance d_mean is applied besides the minimum distance d_min. The resulting HOTA values are given in Table 5.7.

Table 5.7: Influence of feature bank size N_F and strategies for appearance distance calculation in the base framework on MOT17 val. d_cos is used as distance between single feature vectors, and HOTA is reported as main evaluation measure. Taking multiple past features into account significantly improves the performance in both strategies.

N_F	1	10	20	30	60	100
d_min	66.3	67.4	**67.9**	67.5	67.7	67.3
d_mean	66.3	**67.9**	67.2	67.2	65.8	65.2

Three observations can be made: First, taking multiple features from different time steps into account significantly improves the performance compared to just using the last associated feature as track feature, i.e., $N_\text{F} = 1$. Second, incorporating features from too far in the past, i.e., setting N_F too large, degrades the accuracy because the appearance of targets changes over time. Third, d_min and d_mean can achieve similarly good results, while taking the minimum distance is more robust w.r.t. the feature bank size N_F.

As an alternative approach to incorporate appearance information from different time steps into the track features, the EMA update rule has been introduced in Section 5.4.2. The evaluation results for different calculation strategies of the appearance distance are summarized in Table 5.8.

Table 5.8: Comparison of different calculation strategies for appearance distance. While the same HOTA is obtained, the EMA strategy is most efficient as only one cosine distance has to be computed per track–detection pair.

d	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
d_min	**67.9**	65.3	71.0	69.1	85.3	75.3	**85.9**
d_mean	**67.9**	**65.4**	70.8	**69.2**	**85.4**	75.8	84.7
d_EMA	**67.9**	65.3	**71.1**	69.1	85.3	**76.6**	83.4

Some notable differences are found in AssRe and AssPr, but the overall tracking performance is similar. However, the computation of d_EMA is significantly faster since only one feature, which is iteratively updated in each time step, has to be compared with the detection feature for each track. In other words, only one cosine distance has to be calculated per track–detection pair. For d_{min} and d_{mean}, the cosine distance has to be calculated $N_{\text{F}} = 20$ and $N_{\text{F}} = 10$ times, respectively, to achieve the same performance (Table 5.7). Thus, the EMA distance d_{EMA} is much more efficient. For this reason, it is used as appearance distance in the rest of this thesis.

Motion-Based vs. Appearance-Based Association

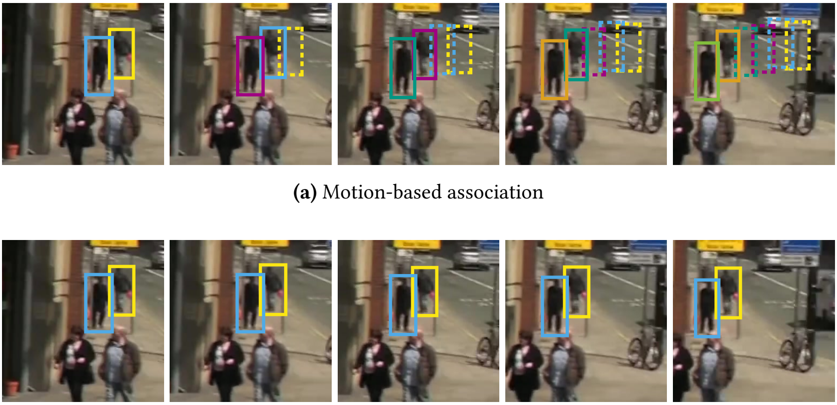
In this section, a detailed comparison of the motion- and appearance-based association in the base framework is presented. First, a quantitative examination shows that comparable results are obtained on MOT17 val. After that, various qualitative examples are provided, which demonstrate that in different situations, the one or the other association method performs better. Furthermore, the advantages and disadvantages of the two approaches are elaborated.

Table 5.9 opposes the evaluation results for the best motion-based and best appearance-based method from the previous sections.

Table 5.9: Best motion-based vs. best appearance-based association method in the base framework on MOT17 val. The latter achieves slightly better results due to a higher AssRe.

d	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
$d_{\text{mot}} = d_{\text{IoU}}$	67.6	65.2	70.5	69.0	85.2	75.0	**84.7**
$d_{\text{app}} = d_{\text{EMA}}$	**67.9**	**65.3**	**71.1**	**69.1**	**85.3**	**76.6**	83.4

The appearance-based association overall achieves a better accuracy, however, with the expense of a much higher computational complexity due to the additional REID model (Section 7.1.2). While d_{app} yields a higher AssRe, using d_{mot} results in a higher AssPr. This is because the appearance distance does not make any spatial restrictions and considers only the extracted appearance features. In situations, where the predicted motion states of the Kalman filter are poor, e.g., due to unaccounted camera motion, the appearance-based association can still achieve good results, while the motion-based association can fail. Figure 5.8 shows qualitative tracking results for motion- and appearance-based association on an example sequence with notable camera motion.



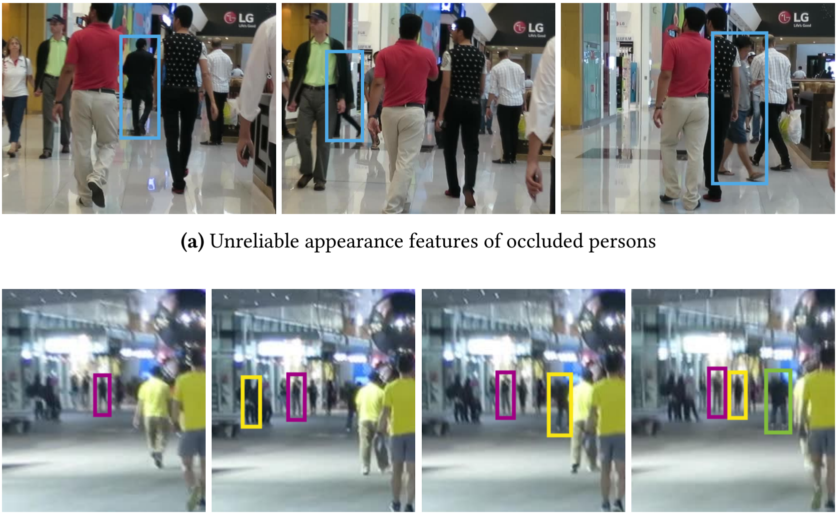
(a) Motion-based association

(b) Appearance-based association

Figure 5.8: Effect of camera motion on motion- and appearance-based association. (a) If not compensated, the camera motion deteriorates the target's motion states and tracking fails in motion-based association. (b) As long as no severe motion blur is introduced, camera motion does not affect appearance-based association.

Note that here and in the following, active and inactive tracks are visualized with solid and dashed boxes, respectively, while each track has its own color and some tracks are omitted for clarity. In the motion-based association (Figure 5.8a), the distances d_{IoU} between the predicted tracks and the real positions of the targets are larger than the threshold $d_{\max}$ due to the camera motion. This leads to a lot of inactive ghost tracks and also many IDSWs, since some of the predicted track positions fall together with the positions of real targets. In contrast, the appearance-based association has no problems to deal with the camera motion as can be seen in Figure 5.8b.

While applying no spatial restrictions can be beneficial if the available motion information is poor, the appearance-based association has issues if the extracted features are unreliable. For instance, the REID model might extract unreliable features from severely-occluded persons, blurry images, or small targets as shown in Figure 5.9.



(a) Unreliable appearance features of occluded persons

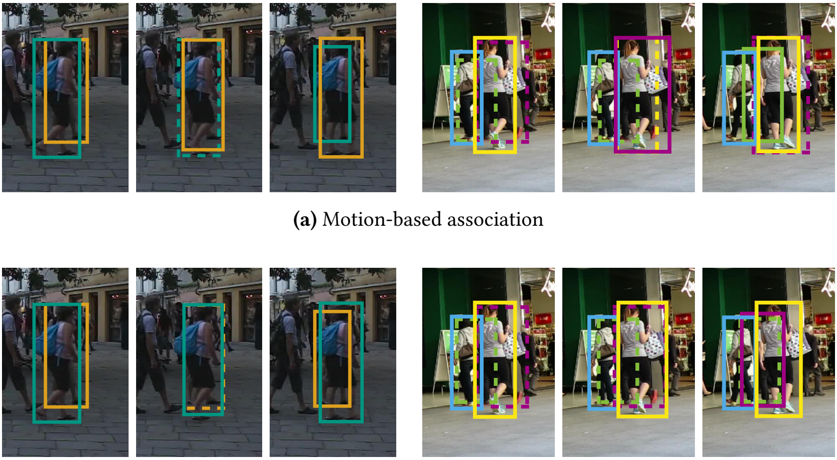
(b) Unreliable appearance features for small and blurred persons

Figure 5.9: Failures of appearance-based association due to missing spatial restrictions. Appearance features can be unreliable under strong occlusion (a) or for small and blurred persons (b) and thus lead to wrong associations.

In the middle and right image of Figure 5.9a, the bounding box contains not only image regions belonging to the occluded target but also body parts from other persons. Thus, the extracted features can be misleading which results in incorrect associations. The same risk is prevalent when facing small or blurred detection boxes that can also result in misleading features and thus in wrong associations as visible in Figure 5.9b.

Analyzing more qualitative results, one observes that the appearance-based association can also achieve good results under quite large degrees of occlusion. Since the training data contains occluded persons, too, the REID model can learn to focus on the actual target while ignoring image parts of nearby persons, at least to some extent. However, it is difficult to determine whether the extent of occlusion is, or is not, too severe for the REID model to extract reasonable appearance features. On the other hand, the motion-based association has also problems under severe occlusion, mainly for two reasons: First, multiple targets share similar positions on the image. Second, the detected bounding boxes might be inaccurate, because it is hard for the detector to reason about the exact boundaries of occluded persons, which can lead to distorted motion states. To conclude, there are situations with occlusion where the appearance-based association works better and other situations where the motion-based association has an advantage. The following qualitative examples underline this statement.

Two sequences are shown in Figure 5.10, for which the appearance-based association is superior to the motion-based association.



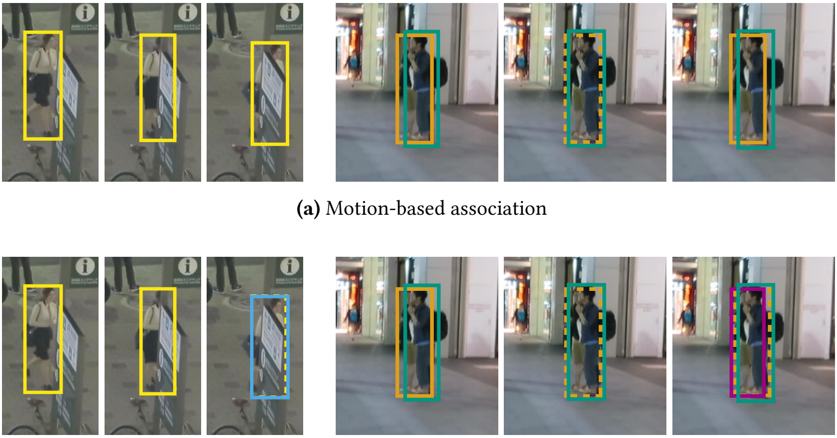
(a) Motion-based association

(b) Appearance-based association

Figure 5.10: Examples for better performance of appearance-based association under occlusion.

In the middle frame of the left sequence, two predicted track boxes are nearly identical such that an IDSW occurs in the motion-based association, while the appearance-based association makes no error. In the right sequence, the situation is as follows for the motion-based association (Figure 5.10a). Next to the blue and the yellow active track in the first frame, two occluded targets are modeled by the inactive green and purple track. In the middle frame, the large detection is erroneously assigned to the purple track instead of the yellow track, and in the right frame, the middle detection is falsely assigned to the green track instead of the purple track. The reason for the wrong assignments lies again in the similar spatial positions of the targets. In other words, the accuracy of the motion model is not high enough to correctly solve the assignment problem in the motion-based association. Surprisingly, the appearance-based association in Figure 5.10b assigns all detections correctly despite the fairly strong occlusions.

Whereas the previous examples have shown superior performance of the appearance-based association under occlusion, the two scenes in Figure 5.11 depict situations, where the motion-based association is in favor.



(a) Motion-based association

(b) Appearance-based association

Figure 5.11: Examples for better performance of motion-based association under occlusion.

The left sequence displays a person walking behind a static obstacle and thereby becoming more and more occluded. While the motion-based association works fine, the detection in the last frame is not assigned to the track but a new track is initialized (blue) in the appearance-based association. This is because the cosine distance between the feature vector extracted from the occluded detection box and the track's feature vector exceeds the maximum distance threshold. The same conclusion can be drawn from the right sequence with the exception that the occluding entity is not a static obstacle but another person.

5.6.6 Summary and Analysis

In this section, findings from the evaluation are summarized and opportunities for further developments are worked out. To improve the tracking performance, one can try to enhance the accuracy of the detector, motion model, or REID network, for instance. However, the use of available information in the base framework is not optimal as pointed out shortly. The aim of this thesis is to develop a framework that improves the utilization of available information in the tracking process, especially by improving the track management and association.

In the base framework, the following shortcomings are identified:

- Only confident detections are kept for the tracking task and heavily-overlapping detections are removed by NMS.
- No information about already tracked targets is leveraged in the initialization of tracks.
- Either motion or appearance information is used in the association but not both.

Next, these weaknesses are explained in more detail, and approaches developed in this thesis to eliminate them are briefly outlined.

Detections with a lower confidence score than s_{track} are not used in the tracking process. However, Figure 5.5 clearly shows that this simple filtering strategy removes a lot of correct detections since DetRe in the optimal setting (DetRe $= 69.0$ for $s_{\text{track}} = 0.7$) is about 10 points lower than if all detections are used (DetRe $= 79.2$ for $s_{\text{track}} = 0.0$). If one manages to leverage the correct low-confidence detections while filtering the incorrect ones, the overall tracking accuracy can be largely increased, which has been demonstrated by the BYTE association in [Zha22c]. The same holds true for strongly-overlapping detections: Instead of suppressing all detections with overlaps higher than

o_{NMS} in the NMS, only the FPs should be filtered. For the first time, two different approaches are proposed in this thesis to leverage such heavily-occluded detections in the tracking process. The first method builds upon the BYTE association and utilizes both low-confidence and heavily-occluded detections in a second matching stage, while preventing incorrect detections from starting FP tracks [Sta23c]. The second method leverages the positional information of so-far tracked targets to identify regions with missing detections and adaptively incorporates detections with high overlaps into the association [Sta21c]. Both strategies will be introduced in Section 6.1.

For the track initialization, the continuity requirement presented in Section 5.5 has not led to any improvements (Table 5.1). Instead of determining whether a unmatched detection is confirmed in consecutive frames, the surroundings of the detection can indicate whether it is a TP or a FP. If an unmatched detection arises at a dense region, where already multiple targets have been tracked, it is likely that the detection is a duplicate. The OAI technique presented in Section 6.1.3 utilizes the track information to identify and remove such duplicate detections, which prevents the start of ghost tracks [Sta23b]. To the best of the author's knowledge, the OAI is the first method in MPT that explicitly takes the surroundings of an unmatched detection for track initialization into account.

The third shortcoming of the base framework, that either motion *or* appearance information is used in the association, has already been analyzed in detail in the previous section. It was demonstrated that in different situations, motion- or appearance-based association is favorable, which indicates that the combination of both information sources has a large potential for improving the association accuracy. While several fusion strategies exist in the literature, it will be shown through a detailed analysis in Section 6.2 that the prevailing approaches do not utilize the available information effectively. Based on the findings, novel combined motion- and appearance-based distance functions are proposed that do not have the weaknesses of previous fusion methods [Sta23b, Sta23d].

Note that the Kalman filter adaptations from Section 5.2.2 can also be regarded as methods that improve the usage of available information: The NSA Kalman filter enhances the update step leveraging the confidence score of the detection, which is not utilized in the standard formulation. The proposed HP module [Sta22b] takes advantage of the fact that in MPT with typically high frame rates and persons moving with limited speed, the size of bounding boxes cannot change notably during a short time period. Exploiting such context knowledge is also a form of improving the use of available information. As the Kalman filter adaptations have significantly enhanced the tracking performance (Table 5.2), it is reasonable to further improve the utilization of available information in the tracking process.

6 Utilization of Occluded Detections and Target Information

It has been found in the previous section that the base framework, which follows the TBD paradigm like many methods from the literature [Aha22, Bew16, Cao23, Woj17, Zha22c], does not fully utilize the available information in the tracking process. This *available information* can contain, for instance, the set of detections, extracted appearance features, or motion states of the tracked targets. The main goal of this thesis is to use such information in the best possible way.

At first, the focus lies on improving the utilization of detections and tracks under occlusion, as this is where naturally most errors occur in the MPT task. Section 6.1 introduces two different approaches for enlarging the set of used detections in the association to enhance the matching accuracy. Furthermore, the track information is leveraged in an OAI technique to suppress the start of ghost tracks from duplicate detections in crowded regions.

After that, various strategies to fuse motion and appearance information are examined in Section 6.2 and combined distance functions for a motion- *and* appearance-based association are introduced, which clearly outperform previous fusion approaches from the literature.

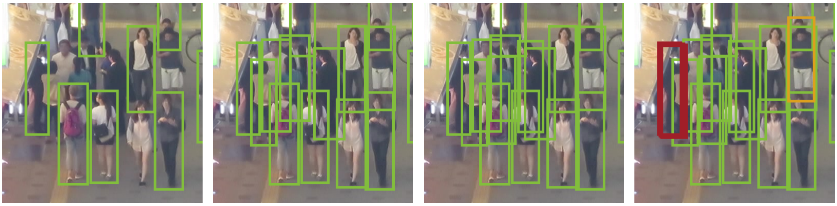
By combination of the approaches in Section 6.3, the overall tracking performance is further improved showing that the proposed modules work well together and complement each other.

In Section 6.4, an efficient model for CMC is introduced, which is an important component for MPT when dealing with non-static cameras and makes an additional module of the proposed tracking framework.

Finally, a comparison of the tracking framework with the SOTA on two MPT benchmarks is made in Section 6.5, and the findings of this chapter are summarized in Section 6.6.

6.1 Improved Use of Detections and Tracks under Occlusion

To remove duplicate detections, the NMS is applied, which ensures that the filtered detections have a maximum overlap (IoU) of o_{NMS} on the image (Section 5.1.2). This is illustrated exemplarily in Figure 6.1, where the filtered detections for different values of o_{NMS} are shown.

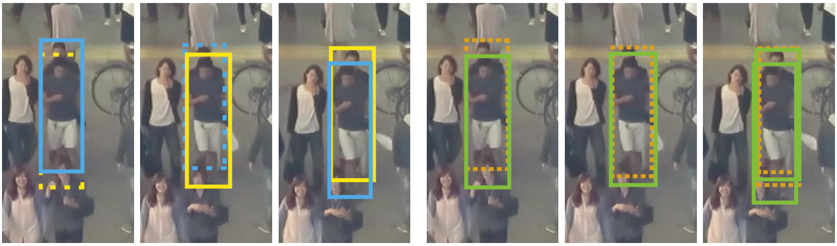


(a) $o_{\mathrm{NMS}} = 0.0$ **(b)** $o_{\mathrm{NMS}} = 0.3$ **(c)** $o_{\mathrm{NMS}} = 0.6$ **(d)** $o_{\mathrm{NMS}} = 0.9$

Figure 6.1: Filtered detections with different NMS thresholds o_{NMS}. Duplicate detections are depicted in red, and the detection of the person that is only kept with a very high NMS threshold, i.e., $o_{\mathrm{NMS}} = 0.9$, is highlighted in orange.

With a growing NMS threshold o_{NMS}, the detection recall increases, however, at the cost of a decreasing precision. Note that for $o_{\mathrm{NMS}} = 0.9$ in Figure 6.1d, the left-most person is detected multiple times, which is depicted with red overlapping boxes. Using such a high NMS threshold would lead to a lot of ghost tracks initialized by the FP duplicate detections and thus to a significant decrease in tracking performance. However, the severely-occluded person on the right top of the image (orange box) is only detected with a very high NMS threshold, see also Figure 6.1d. Leveraging such a TP detection cannot only enhance the detection recall but also the association accuracy, since the assignment task becomes simpler if no detections are missing.

With less detections than tracks under severe occlusion, the association easily fails as can be seen in Figure 6.2a, where tracking results of the base framework are depicted.



(a) Active (solid) and inactive (dashed) tracks **(b)** Used (green) and removed (orange) detect.

Figure 6.2: Failure of standard association (a) and available detections (b) under person–person occlusion. The removal of detections in the NMS is caused by too large overlaps.

The standard association uses only one detection in the first an second frame and two detections in the third frame of the sequence. Notice that the optimal setting for the standard association is $o_{\text{NMS}} = 0.7$ (Table 6.4). Because of the missing detections and imperfections of the motion model, an IDSW occurs. Figure 6.2b shows the used detections in green and the removed detections, which one would obtain when setting $o_{\text{NMS}} = 0.9$, in orange. Leveraging such additional detections can simplify the association task and prevent IDSWs under severe occlusion, as will be seen later in this section.

Two different approaches are suggested in this thesis that both have the goal to utilize as many occluded TP detections as possible, while not introducing duplicate detections into the tracking results. Both methods build upon an adapted version of the NMS proposed in this thesis that aims at an increased detection recall under occlusion [Sta21d]. Several works exist in the literature that adjust the NMS process to enhance the detection performance in crowded scenes [Chu20, Hos17, Hua20, Liu19, Xie20]. For instance, the detector is enlarged by a density subnetwork and the NMS threshold is increased for detections with large estimated densities in [Liu19]. Similarly, [Xie20] introduces a count-and-similarity branch in the Faster R-CNN detector [Ren17] to identify distinct proposals. Another approach is found in [Chu20], where one region proposal makes multiple predictions for distinct targets and a set NMS is applied on the generated detection sets. In contrast to the aforementioned methods, the proposed adapted NMS can be applied without the need for changes to the network architecture or training process of the detector. Thus, it is a generic method that can be used together with a multitude of different detection models. Its working mechanism is explained in the following.

Adapted NMS

The adapted NMS aims at providing an additional set of *occluded* detections next to the *normal* set of detections. To achieve this, two standard NMS are performed with various overlap thresholds and then, the resulting detection sets are subtracted. More formally, let $\widetilde{\mathcal{D}}$ be the unfiltered detection set coming from the detection model and o_{NMS1} and o_{NMS2} the maximum overlap thresholds measured in IoU of the first and second NMS, respectively. The two filtered detection sets $\mathcal{D}_{\text{NMS1}}$ and $\mathcal{D}_{\text{NMS2}}$ can be computed as

$$\mathcal{D}_{\text{NMS1}} = \text{NMS}(\widetilde{\mathcal{D}}, o_{\text{NMS1}}) \qquad \text{and} \qquad \mathcal{D}_{\text{NMS2}} = \text{NMS}(\widetilde{\mathcal{D}}, o_{\text{NMS2}}), \tag{6.1}$$

where NMS(·) stands for the procedure in Algorithm 1. The second NMS shall apply a larger threshold: $o_{\text{NMS1}} < o_{\text{NMS2}}$. Then, $\mathcal{D}_{\text{NMS1}}$ corresponds to the standard detection set, and the additional set of occluded detections $\mathcal{D}_{\text{occ}}$ is obtained by subtraction of the two sets:

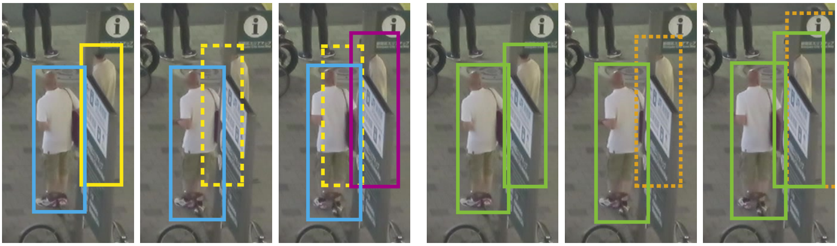
$$\mathcal{D}_{\text{occ}} = \mathcal{D}_{\text{NMS2}} \setminus \mathcal{D}_{\text{NMS1}}. \tag{6.2}$$

This additional detection set can be leveraged in several ways in the association. Two different approaches are suggested. The first method incorporates the detections from $\mathcal{D}_{\text{occ}}$ in a second association stage, where they are matched with unassigned tracks from the first stage [Sta23c]. While this allows the TPs of $\mathcal{D}_{\text{occ}}$ being used in the association, the FPs are not utilized for track initialization such that the start of duplicate tracks is prevented. The second approach makes use of the track information to identify track clusters, where the number of corresponding detections within the standard set

$\mathcal{D}_{\text{NMS1}}$ is less than the number of tracks, i.e., missing detections are recognized [Sta21c]. Then, the additional detection set $\mathcal{D}_{\text{occ}}$ is used within these clusters, while also ensuring that no duplicate tracks are started. The two different methods are thoroughly presented next.

6.1.1 Two-Stage Association

The set of occluded detections $\mathcal{D}_{\text{occ}}$ usually contains much more duplicate FP detections than TPs. Directly using them in the association and in the initialization would lead to many tracking errors. Therefore, a second association stage is introduced similar as in the BYTE[1] tracking framework [Zha22c]. In BYTE, low-confidence detections—which are typically discarded just like the occluded detections—are matched with the unassigned tracks from the first association stage. In addition to discarding TP detections with severe person–person occlusions by NMS as in Figure 6.2, TP detections of persons with severe obstacle occlusion can be removed from the association due to a low confidence score ($s < s_{\text{track}}$). An example is given in Figure 6.3a, where the tracking results of the standard association are depicted for frames 892, 914, and 937 of the MOT17-04 sequence.



(a) Active (solid) and inactive (dashed) tracks **(b)** Used (green) and removed detect. (orange)

Figure 6.3: Failure of standard association (a) and available detections (b) under obstacle–person occlusion. The removal of detections is caused by too low confidence scores.

[1] Authors being affiliated with the Chinese company ByteDance explains the name of the tracker.

Between frames 893 and 936 inclusively, the confidence score of the person detection occluded by the static obstacle is below s_{track}, so no update of the yellow track is performed. It turns inactive and is predicted with the motion model in the consecutive frames. When in frame 937, a confident detection for the occluded person is available again, the predicted track position is quite inaccurate such that the association fails and a new track (purple) is erroneously started. Figure 6.3b shows the used detections in green and the removed detections due to low confidence in orange. The goal of the BYTE association is to leverage these low-confidence detections to improve the detection recall and the association accuracy. Note that there can be other reasons besides occlusion for low-confidence TP detections as motion blur, small object size, or unusual appearance, to name a few.

While BYTE uses detections with low-confidence, detections with high overlaps are still discarded, so the risk for tracking errors under strong occlusion due to missing detections as in Figure 6.2 remains high. Therefore, a two-stage association technique—termed BYTEv2 as extension to BYTE—that additionally leverages heavily-occluded detections in the tracking process is proposed in this thesis [Sta23c]. Before the further development BYTEv2 is presented, the functionality of the basic BYTE association is introduced in the following.

Given the set of detections filtered by standard NMS $\mathcal{D}$, the track threshold s_{track}, and a minimum confidence threshold s_{min}, the detection set is split into low-confidence detections $\mathcal{D}_{\text{low}}$ and high-confidence detections $\mathcal{D}_{\text{high}}$:

$$\mathcal{D}_{\text{low}} = \{\mathrm{D}_i | \mathrm{D}_i \in \mathcal{D}, s_{\text{min}} \leq s_i < s_{\text{track}}\}, \tag{6.3}$$

$$\mathcal{D}_{\text{high}} = \{\mathrm{D}_i | \mathrm{D}_i \in \mathcal{D}, s_i \geq s_{\text{track}}\} \tag{6.4}$$

with s_i denoting the confidence score of the i-th detection D_i. Notice that detections with very low confidence below s_{min} are removed. As in the base framework, the high-confidence detections $\mathcal{D}_{1,\text{BYTE}} = \mathcal{D}_{\text{high}}$ are matched with the set of tracks $\mathcal{T}$ based on a distance function d_1 in the *first* association stage. After that, the active unassigned tracks $\mathcal{T}^{\text{u,a}} \subseteq \mathcal{T}$ are matched to the low-confidence detections $\mathcal{D}_{2,\text{BYTE}} = \mathcal{D}_{\text{low}}$ based on the distance function d_2 in the *second* association stage. Note that various distance measures can be applied for d_1 and d_2 and that the maximum allowed matching distance generally differs among the two stages: $d_{\max,1} \neq d_{\max,2}$. Since a lot of low-confidence detections are FPs, only unassigned detections with a confidence score $s \geq s_{\text{init}} \geq s_{\text{track}}$ are allowed to initialize new tracks.

Incorporating the usually discarded low-confidence detections into the association, BYTE improves the usage of available information in the tracking process. However, it does not account for the heavily-occluded detections filtered by NMS. To solve this problem, BYTEv2 additionally utilizes the set of occluded detections $\mathcal{D}_{\text{occ}}$ in the second association stage. Since detections under heavy occlusion tend to be more inaccurate—as the detector has difficulties to reason about the boundaries of objects—only detections $\widetilde{\mathcal{D}}_{\text{occ}}$ with a confidence score larger than s_{occ} are leveraged:

$$\widetilde{\mathcal{D}}_{\text{occ}} = \{\mathrm{D}_i | \mathrm{D}_i \in \mathcal{D}_{\text{occ}}, s_i \geq s_{\text{occ}}\}. \tag{6.5}$$

This detection set is added to the set of low-confidence detections from Equation (6.3) yielding the detection set for the second association stage of BYTEv2

$$\mathcal{D}_{2,\text{BYTEv2}} = \mathcal{D}_{\text{low}} \cup \widetilde{\mathcal{D}}_{\text{occ}}. \tag{6.6}$$

Like for BYTE, only unassigned detections from the first association stage, i.e., unassigned high-confidence detections, with score $s \geq s_{\text{init}}$ can start new tracks. As a summary, the pipeline of BYTEv2 is illustrated in Figure 6.4, where the adapted NMS and the two association stages are highlighted. Note that time indices, some components (detection, REID, and motion model) as well as the final part of the track management for matched (m), unmatched (u), and new (n) tracks are omitted for clarity (compare with Figure 5.4).

As in the base framework, high-confidence detections $\mathcal{D}_{\text{high}}$ are matched with the tracks $\mathcal{T}$ in the first association stage. Then, low-confidence detections $\mathcal{D}_{\text{low}}$ and confident heavily-occluded detections $\widetilde{\mathcal{D}}_{\text{occ}}$ coming from the adapted NMS are matched with the unassigned active tracks $\mathcal{T}_1^{\text{u,a}}$ in the second association stage. In contrast to unassigned detections from the first stage $\mathcal{D}_1^{\text{u}}$ that are considered for track initialization, unassigned ones from the second stage $\mathcal{D}_2^{\text{u}}$ are deleted to prevent the start of duplicate tracks.

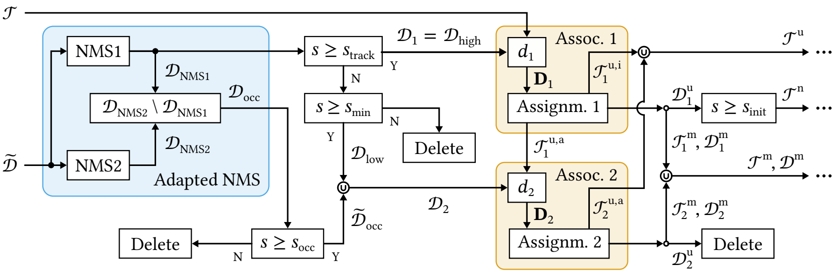

$\mathcal{T}$
$\tilde{\mathcal{D}}$
NMS1
NMS2
$\mathcal{D}_{\text{NMS1}}$
$\mathcal{D}_{\text{NMS2}} \setminus \mathcal{D}_{\text{NMS1}}$
$\mathcal{D}_{\text{NMS2}}$
$\mathcal{D}_{\text{occ}}$
Adapted NMS
$s \geq s_{\text{track}}$
Y
N
$s \geq s_{\min}$
N
Y
Delete
$\mathcal{D}_{\text{low}}$
$\tilde{\mathcal{D}}_{\text{occ}}$
$s \geq s_{\text{occ}}$
N
Y
Delete
$\mathcal{D}_1 = \mathcal{D}_{\text{high}}$
$\mathcal{D}_2$
Assoc. 1
d_1
$\mathbf{D}_1$
Assignm. 1
$\mathcal{T}_1^{\text{u,i}}$
$\mathcal{T}_1^{\text{u,a}}$
Assoc. 2
d_2
$\mathbf{D}_2$
Assignm. 2
$\mathcal{T}_2^{\text{u,a}}$
$\mathcal{T}^{\text{u}}$ …
$\mathcal{D}_1^{\text{u}}$
$s \geq s_{\text{init}}$
$\mathcal{T}^{\text{n}}$ …
$\mathcal{T}_1^{\text{m}}, \mathcal{D}_1^{\text{m}}$
$\mathcal{T}^{\text{m}}, \mathcal{D}^{\text{m}}$ …
$\mathcal{T}_2^{\text{m}}, \mathcal{D}_2^{\text{m}}$
$\mathcal{D}_2^{\text{u}}$
Delete

Figure 6.4: Overview of the proposed BYTEv2 pipeline. Main components that differ from the base framework are the adapted NMS that enables the use of occluded detections in the tracking process and the two-stage association.

Evaluation

In this section, the BYTE and BYTEv2 association are evaluated and compared with each other. For all experiments, the IoU distance is leveraged as association distance in both stages: $d_1 = d_2 = d_{\text{IoU}}$. In addition to such a motion-based association, it will be shown in Section 6.3 that BYTEv2 can also be successfully applied with an appearance-based distance measure.

BYTE mainly makes two adaptations w.r.t. the base framework: the incorporation of low-confidence detections in a second association stage and the introduction of a confidence threshold $s_{\text{init}} \geq s_{\text{track}}$ for track initialization. Evaluation results of these adaptations can be found in Table 6.1.

Table 6.1: Ablation of the BYTE two-stage association and initialization on MOT17 val. Both strategies improve the overall tracking performance and lead to further gains when applied together.

Two-stage association	Initialization	s_{track}	s_{init}	HOTA	DetA	AssA
✗	✗	0.7	–	67.6	65.2	70.5
✓	✗	0.7	–	68.4	66.8	70.6
✗	✓	0.6	0.7	67.8	66.1	70.0
✓	✓	0.6	0.7	**68.8**	**67.1**	**71.0**

The first row corresponds to the base framework with standard association (single stage) and no separate initialization threshold, i.e., the unmatched detections with $s \geq s_{\text{track}}$ start new tracks. If one uses the low-confidence detections in a second matching stage, HOTA is increased by 0.8 points showing the high potential of incorporating typically discarded detections in the association. Using a higher confidence threshold for track initialization than for association (third row), HOTA is slightly increased by 0.2 points. Both improvements stem mainly from an increase of DetA. However, if both adaptations are applied together, AssA is also significantly enhanced (+1.0 w.r.t. the base framework). This synergy effect is further indicated by a total increase of 1.2 HOTA when combining the two components that individually yield only a plus of 0.8 and 0.2 HOTA. Note that Table 6.1 shows the optimal values of s_{track} and s_{init} for the respective configuration.

The influence of the maximum association distance in the second association stage of BYTE $d_{2,\text{max}}$ on the tracking performance can be seen in Table 6.2.

Table 6.2: Influence of $d_{2,\text{max}}$ in the BYTE association on MOT17 val. A too small distance threshold prevents correct matches in the second association stage, while a too high threshold introduces wrong matches.

$d_{2,\text{max}}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
HOTA	68.1	68.3	68.3	**68.8**	68.2	67.9	67.3	67.2	66.6

The same conclusion as for the distance threshold of the base framework is drawn: Whereas a too small threshold misses TP associations, a too large threshold introduces FP associations. The best tracking performance of BYTE on MOT17 val is achieved with $d_{2,\text{max}} = 0.4$. Note that this is smaller than the distance in the first association $d_{1,\text{max}} = 0.8$ that has been taken over from the base framework, like other basic parameters (i_{max}, n_{init}, etc.). Thus, a stricter matching criterion is applied for the on average more inaccurate low-confidence detections.

The minimum confidence of detections in the second association s_{min} is set to 0.0 on MOT17 val. On the PP22 and SOMPT22 dataset, a slightly higher value of 0.1 yields the best results. The key message here is that with the BYTE association, detections with very low confidence can improve the tracking performance, since they are only used for assigning to already tracked targets and not for track initialization.

This also holds true for the occluded detections that are utilized in the second association stage of the further development BYTEv2. Table 6.3 lists the evaluation results of BYTEv2, BYTE, and the standard association on the three datasets MOT17 val, PP22 test, and SOMPT22 train.

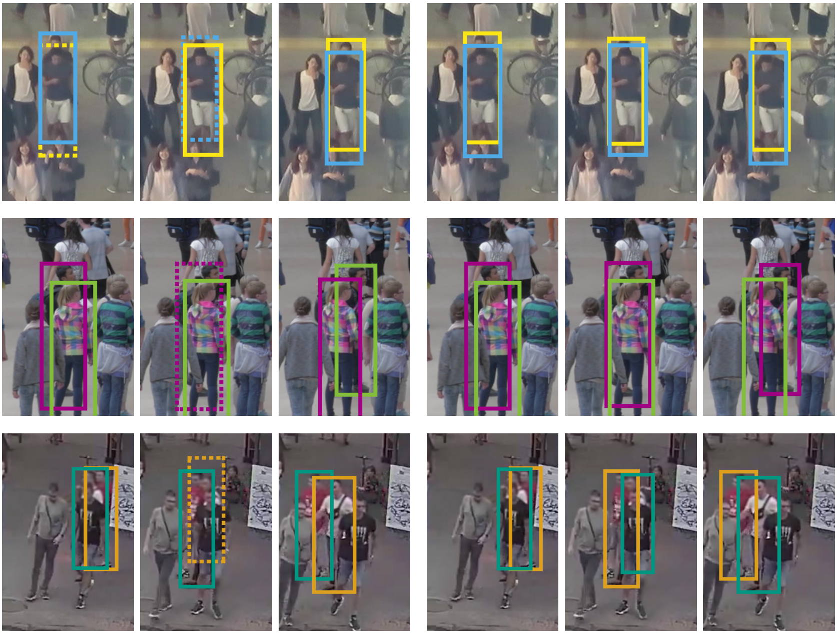
Table 6.3: Comparison of association methods on three different datasets. The proposed BYTEv2 further improves upon BYTE and clearly outperforms the standard association in all evaluation measures.

	MOT17 val			PP22 test			SOMPT22 train		
Association	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
Standard	67.6	65.2	70.5	62.1	67.7	57.5	55.4	56.0	55.0
BYTE	68.8	67.1	71.0	63.0	**68.2**	58.7	56.0	57.1	55.1
BYTEv2	**69.3**	**67.4**	**71.7**	**63.4**	**68.2**	**59.5**	**56.3**	**57.4**	**55.5**

Leveraging the occluded detections $\mathcal{D}_{\text{occ}}$ that are discarded in BYTE and the standard association, BYTEv2 further increases the tracking performance on all datasets. This indicates a good generalization ability w.r.t. different qualities of the available detections, since the domain gap varies among the three evaluation protocols (Section 4.3). Compared to the standard association, HOTA is enhanced by 1.7, 1.3, and 0.9 points on MOT17 val, PP22 test, and SOMPT22 train, respectively.

Note that BYTEv2 is a generic association strategy that can be applied within any TBD-based method. It has been shown in a previous work of the author [Sta23c] that consistent performance improvements w.r.t. the BYTE baseline are achieved when using BYTEv2 in various tracking frameworks.

The superior tracking performance of BYTEv2 especially shows up in crowded scenes, where severely-occluded detections are not leveraged in BYTE and the standard association, which easily leads to IDSWs. Figure 6.5 depicts qualitative tracking results of BYTE and BYTEv2 on three sequences of the evaluation datasets. Remember that inactive tracks are drawn in dashed lines and that only the interesting tracks are visualized for clarity.



(a) BYTE

(b) BYTEv2

Figure 6.5: Qualitative comparison of BYTE and BYTEv2 on example sequences from the MOT17, PP22, and SOMPT22 dataset (top to bottom). The utilization of heavily-occluded detections simplifies the association task (b) and prevents IDSWs that occur in the BYTE baseline (a).

In all three sequences, missing detections lead to an IDSW in BYTE, while no such error occurs in BYTEv2. Next to missing detections making the association task more difficult, propagated inactive tracks become increasingly inaccurate without assigned detections. This can be seen in the middle frame of the last sequence in Figure 6.5a, where the inaccurate inactive orange track contributes to the association failure in BYTE. In contrast, the motion state is successfully updated in this frame with the additional occluded detection in BYTEv2 such that the tracking error is prevented.

Building upon the advanced NMS presented earlier in this section, BYTEv2 introduces an additional parameter into the tracking framework: the overlap threshold o_{NMS2} of the second NMS. Its influence on the tracking performance, together with the first NMS threshold o_{NMS1}, has been exemplarily investigated on MOT17 val. The results are summarized in Table 6.4.

Table 6.4: Influence of the NMS thresholds o_{NMS1} and o_{NMS2} in the BYTE and BYTEv2 association on MOT17 val. The introduction of heavily-occluded detections in BYTEv2 with $o_{\text{NMS2}} \leq 0.9$ enhances the tracking performance for all evaluated values of o_{NMS1}.

o_{NMS1}	o_{NMS2}	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
0.6	–	68.6	66.6	71.1	72.9	**81.6**	76.1	**83.3**
0.6	0.7	68.7	66.9	71.1	73.3	81.5	76.3	83.1
0.6	0.8	68.8	67.2	70.9	73.7	81.5	76.1	82.7
0.6	0.9	69.0	67.1	71.5	73.7	81.3	76.6	83.1
0.6	1.0	68.2	66.7	70.2	74.2	80.1	76.3	81.3
0.7	–	68.8	67.1	71.0	73.9	81.0	76.3	82.8
0.7	0.9	**69.3**	**67.4**	**71.7**	74.3	81.1	**76.9**	83.0
0.8	–	67.4	67.1	68.4	**74.5**	80.2	73.6	81.7
0.8	0.9	68.2	66.9	70.0	**74.5**	80.0	75.0	82.8

The rows where no value for o_{NMS2} is given (–) correspond to the BYTE baseline. As mentioned earlier for the standard association, the optimal setting when applying a single NMS is $o_{\text{NMS1}} = 0.7$. Higher values, e.g., $o_{\text{NMS1}} = 0.8$, lead to a greater DetRe, but with the cost of a reduced DetPr and AssA. In contrast, smaller values like $o_{\text{NMS1}} = 0.6$ increase the precision of detection and association at the expense of a reduced recall. For the parameter o_{NMS2} of BYTEv2, two findings can be derived from Table 6.4. First, the overall tracking accuracy measured in HOTA consistently improves for a growing overlap threshold until $o_{\text{NMS2}} = 0.9$. Setting it too high, e.g., $o_{\text{NMS2}} = 1.0$, leads to a drop in DetPr, since a lot of duplicate detections are introduced. The reason for this is as follows. Whenever a ghost track has been started by a duplicate detection, the chance is high that the ghost track is consistently matched with a duplicate occluded detection if o_{NMS2} is set to a very high value. The second finding is that for various values of o_{NMS1}, BYTEv2 achieves notable improvements. Setting $o_{\text{NMS2}} = 0.9$, a gain of 0.4, 0.5, and 0.6 HOTA is obtained for

$o_{\text{NMS1}} \in \{0.6, 0.7, 0.8\}$, which indicates a good robustness of BYTEv2 w.r.t. the choice of its parameter o_{NMS2}.

Finally, the influence of the second additional parameter, the confidence threshold for occluded detections s_{occ}, is ablated in Table 6.5.

Table 6.5: Influence of s_{occ} in the BYTEv2 association on MOT17 val. It balances the number of FPs and FNs in the set of used occluded detections.

s_{occ}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
HOTA	68.7	68.8	69.1	69.2	69.2	69.2	**69.3**	69.1	68.8

A too small confidence threshold introduces many FP detections, whereas a too large threshold removes TPs. Choosing $s_{\text{occ}} = 0.7$ yields the best results. In the range of $s_{\text{occ}} \in [0.3, 0.8]$, HOTA values above the BYTE baseline are achieved, which demonstrates that the performance of BYTEv2 is robust to the choice of its parameter s_{occ}.

In this section, the two-stage association method BYTEv2 has been introduced. It enhances the tracking performance in crowded scenes by incorporating occluded detections in the association and thus improving the utilization of available detections. Another method with the same goal but a quite different approach is presented in the following.

6.1.2 Tracking with Clusters

In situations where missing detections occur when only the normal set of detections (termed $\mathcal{D}_{\text{NMS1}}$ in the previous section) is leveraged, BYTEv2 *implicitly* integrates the additional occluded detections $\mathcal{D}_{\text{occ}}$ with the second association stage. This section introduces an *explicit* approach to treat such situations with missing detections [Sta21d]. The basic idea is to utilize positional information of already tracked targets to find track *clusters* in the image where the number of tracks is larger than the number of normal detections. In the local image regions of these clusters, additional occluded detections $\mathcal{D}_{\text{occ}}$ from the adapted NMS are incorporated. Then, the assignment problem is solved in each cluster separately. To the best of the author's knowledge, no comparable approach exists in the MPT literature. The procedure of the TWC method is summarized for a single time step below, before the individual steps are presented in detail.

1. Build track clusters based on the overlaps of all tracks in the image.
2. Put each of the normal detections into the cluster containing the track with the highest overlap.
3. For clusters with a greater number of tracks than detections, put the suitable occluded detections into the cluster.
4. Solve the assignment problem in each cluster separately.
5. Treat the unassigned detections and tracks.

To build clusters of tracks having high overlaps, which correlates with a high chance for missing detections, the IoU between all tracks $\mathrm{T} \in \mathcal{T}$ is computed. Here, T denotes a predicted track with bounding box $\mathbf{b}$ in the current frame, on which the motion model has been applied, and the time index is omitted for clarity. Moreover, the IoU between two tracks shall be the IoU between its predicted track boxes: $\mathrm{IoU}(\mathrm{T}_1, \mathrm{T}_2) = \mathrm{IoU}(\mathbf{b}_1, \mathbf{b}_2)$. Then, a track cluster $\widetilde{\mathcal{C}} = \{\mathrm{T}_1, \dots, \mathrm{T}_k\}$ contains tracks that are *connected* in the sense that two neighboring tracks have an IoU of at least o_{cluster}. Formally, for all tracks T_i of a cluster $\widetilde{\mathcal{C}}$, there exists a sequence of tracks $[\mathrm{T}_i, \mathrm{T}_1, \dots, \mathrm{T}_n, \mathrm{T}_j]$ within the cluster, where the IoU of subsequent tracks is greater or equal o_{cluster}:

$$\begin{aligned}\widetilde{\mathcal{C}} = \{\mathrm{T}_i \,|\, \forall\, \mathrm{T}_i, \mathrm{T}_j \in \widetilde{\mathcal{C}}\ \exists\, [\mathrm{T}_i, \mathrm{T}_1, \dots, \mathrm{T}_n, \mathrm{T}_j]&: \\ \mathrm{IoU}(\mathrm{T}_i, \mathrm{T}_1), \dots, \mathrm{IoU}(\mathrm{T}_n, \mathrm{T}_j) \geq o_{\text{cluster}}&\}.\end{aligned} \tag{6.7}$$

Note that not all track pairs of a cluster must have an IoU above o_{cluster}. For instance, the tracks T_a, T_b, and T_c with $\mathrm{IoU}(\mathrm{T}_a, \mathrm{T}_b), \mathrm{IoU}(\mathrm{T}_b, \mathrm{T}_c) \geq o_{\text{cluster}}$ build a cluster, even if $\mathrm{IoU}(\mathrm{T}_a, \mathrm{T}_c) < o_{\text{cluster}}$ holds, as T_a is *connected* to T_c over T_b. Since o_{cluster} is set quite high in practice, a cluster contains only few tracks and many clusters comprise just a single track, i.e., if the track has no minimum overlap of o_{cluster} with other tracks. To compute the track clusters according to Equation (6.7), a graph is built, where each track is a node and two nodes are connected with an edge if the IoU between the tracks of the corresponding nodes exceeds o_{cluster}. Then, the set of clusters $\widetilde{\mathfrak{C}} = \{\widetilde{\mathcal{C}}_1, \dots, \widetilde{\mathcal{C}}_l\}$ emerges from the set of connected components of the graph. Thus, $\bigcup_{k=1}^{l} \widetilde{\mathcal{C}}_k = \mathcal{T}$ holds. So far, the clusters contain only tracks, which is denoted by the tilde. In the next step, the detections are integrated into the clusters.

For each *normal* detection with high confidence $\mathrm{D}_{\mathrm{norm}} \in \mathcal{D}_1 = \{\mathrm{D}_i \in \mathcal{D}_{\mathrm{NMS1}} \,|\, s_i > s_{\mathrm{track}}\}$, the IoU with all tracks $\mathrm{T} \in \mathcal{T}$ is computed. Then, a detection is put into the cluster containing the track with the highest overlap, if the IoU between that track and the detection exceeds the minimum matching threshold $1 - d_{\max}$. Notice that $d_{\max}$ denotes the maximum allowed IoU distance for association. If there is no track with a sufficiently large IoU, the detection is put into a preliminary set of unassigned detections $\widetilde{\mathcal{D}^{\mathrm{u}}}$ that is later used to initialize new tracks. After all normal detections are treated, a cluster can contain both tracks and detections. For example, a cluster $\mathcal{C}$ containing the tracks T_a and T_b as well as the detections D_a and D_b is denoted by $\mathcal{C} = \{\mathrm{T}_a, \mathrm{T}_b, \mathrm{D}_a, \mathrm{D}_b\}$ without tilde.

In the third step, the number of tracks $n_{\mathrm{T}} = |\mathcal{C} \cap \mathcal{T}|$ and the number of detections $n_{\mathrm{D}} = |\mathcal{C} \cap \mathcal{D}_1|$ in each cluster $\mathcal{C}$ are compared. If $n_{\mathrm{T}} > n_{\mathrm{D}}$ holds, which means that missing detections are identified, the additional set of occluded detections $\mathcal{D}_{\mathrm{occ}}$ from the adapted NMS is involved. As in BYTEv2, only confident occluded detections $\widetilde{\mathcal{D}}_{\mathrm{occ}}$ with score $s > s_{\mathrm{occ}}$ are considered (Equation (6.5)). For each occluded detection $\mathrm{D}_{\mathrm{occ}} \in \widetilde{\mathcal{D}}_{\mathrm{occ}}$, the track with highest IoU is searched. If the IoU exceeds the matching threshold $1 - d_{\max}$ and the track is in a cluster with missing detections ($n_{\mathrm{T}} > n_{\mathrm{D}}$), the occluded detection is put into this cluster. The process of assigning detections to track clusters is summarized in Algorithm 2.

After both normal and occluded detections have been put into the clusters, the association task is solved in each cluster separately. Clusters with $n_{\mathrm{T}} = n_{\mathrm{D}} = 1$ are straightforward: The track and detection can be associated, since the maximum association distance $d_{\max}$ has already been enforced when putting the detection into the cluster containing the single track. For $n_{\mathrm{T}} = 1, n_{\mathrm{D}} = 0$, the track turns inactive and for all other cases, the Hungarian algorithm is applied to solve the assignment problem.

Algorithm 2: Assigning Detections to Clusters (D2C).

```
Input: Set of track clusters $\widetilde{\mathfrak{C}} = \{\widetilde{\mathcal{C}}_1, \dots, \widetilde{\mathcal{C}}_l\}$ with $\widetilde{\mathcal{C}} = \{\mathrm{T}_1, \dots, \mathrm{T}_k\}$ and
       $\bigcup_{j=1}^{l} \widetilde{\mathcal{C}}_j = \mathcal{T}$,
       set of confident normal detections $\mathcal{D}_1$,
       set of confident occluded detections $\widetilde{\mathcal{D}}_{\mathrm{occ}}$,
       maximum IoU distance for matching $d_{\max}$
Output: Set of clusters with tracks and detections $\mathfrak{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_l\}$ with
        $\mathcal{C} = \{\mathrm{T}_1, \dots, \mathrm{T}_k, \mathrm{D}_1, \dots, \mathrm{D}_n\}$,
        preliminary set of unassigned detections $\widetilde{\mathcal{D}^{\mathrm{u}}}$

$\mathfrak{C} \leftarrow \widetilde{\mathfrak{C}}$, $\widetilde{\mathcal{D}^{\mathrm{u}}} \leftarrow \varnothing$                    // initialize output sets
for $\mathrm{D}_{\mathrm{norm}} \in \mathcal{D}_1$ do                              // iterate over all normal detections
    $o_{\max} \leftarrow \max_k\{\mathrm{IoU}(\mathrm{D}_{\mathrm{norm}}, \mathrm{T}_k)\}$              // find track with maximum overlap
    if $o_{\max} \geq 1 - d_{\max}$ then
        $\widetilde{\mathcal{C}}_k \leftarrow \widetilde{\mathcal{C}} \mid \mathrm{T}_k \in \widetilde{\mathcal{C}}$                    // find respective track cluster
        $\mathcal{C}_k \leftarrow \widetilde{\mathcal{C}}_k \cup \{\mathrm{D}_{\mathrm{norm}}\}$         // put detection into cluster in output set
    else
        $\widetilde{\mathcal{D}^{\mathrm{u}}} \leftarrow \widetilde{\mathcal{D}^{\mathrm{u}}} \cup \{\mathrm{D}_{\mathrm{norm}}\}$                    // save unassigned detection
for $\mathcal{C} \in \mathfrak{C}$ do                                        // iterate over all clusters
    $n_{\mathrm{T}} \leftarrow |\mathcal{C} \cap \mathcal{T}|$                         // count number of tracks in cluster
    $n_{\mathrm{D}} \leftarrow |\mathcal{C} \cap \mathcal{D}_1|$                    // count number of detections in cluster
    if $n_{\mathrm{T}} > n_{\mathrm{D}}$ then
        for $\mathrm{D}_{\mathrm{occ}} \in \widetilde{\mathcal{D}}_{\mathrm{occ}}$ do            // iterate over all occluded detections
            // find maximum overlap with tracks in cluster
            $o_{\max} \leftarrow \max_k\{\mathrm{IoU}(\mathrm{D}_{\mathrm{occ}}, \mathrm{T}_k) \mid \mathrm{T}_k \in \mathcal{C}\}$
            if $o_{\max} \geq 1 - d_{\max}$ then
                $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathrm{D}_{\mathrm{occ}}\}$                 // put detection into cluster
                $\widetilde{\mathcal{D}}_{\mathrm{occ}} \leftarrow \widetilde{\mathcal{D}}_{\mathrm{occ}} \setminus \{\mathrm{D}_{\mathrm{occ}}\}$  // remove assigned detection from set
```

As a final association step, the unassigned normal detections from the clusters are compared with the unassigned tracks without the cluster limitation. This can lead to additional correct matches in cases where a detection has been assigned to the *wrong* cluster due to inaccuracies of the motion model or the detected bounding box. The remaining unassigned detections $\{\mathrm{D}^{\mathrm{u}}_{\mathrm{norm}}\}$

are then combined with the preliminary set of single detections $\widetilde{\mathcal{D}}^{\mathrm{u}}$, i.e., detections that have not been put into any cluster. This yields the total set of unassigned normal detections $\mathcal{D}^{\mathrm{u}} = \widetilde{\mathcal{D}}^{\mathrm{u}} \cup \{\mathrm{D}^{\mathrm{u}}_{\mathrm{norm}}\}$. Note that the unassigned occluded detections $\{\mathrm{D}^{\mathrm{u}}_{\mathrm{occ}}\}$ are removed to prevent the start of ghost tracks, as also done in BYTEv2. Lastly, the total set of unassigned normal detections $\mathcal{D}^{\mathrm{u}}$ is leveraged for track initialization.

Figure 6.6 gives an overview of the TWC approach. The final association step, update of matched tracks, and the track initialization are left out for clarity (compare with Figure 5.4).

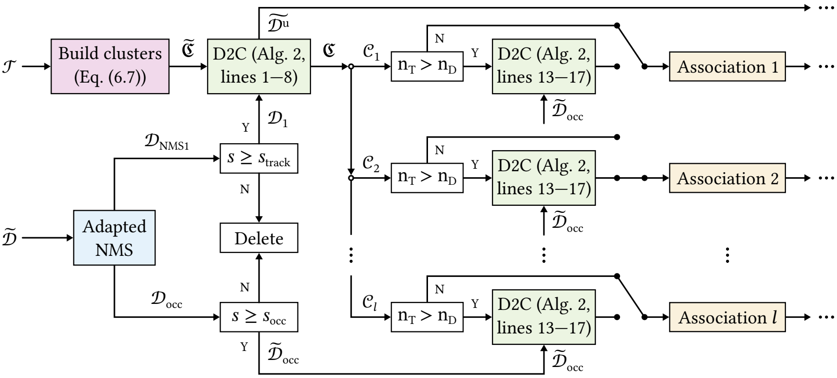

$\mathcal{T}$
Build clusters (Eq. (6.7))
$\mathfrak{C}$
D2C (Alg. 2, lines 1–8)
$\widetilde{\mathcal{D}}^{\mathrm{u}}$
$\mathfrak{C}$
$\mathcal{C}_1$
$\mathcal{C}_2$
$\mathcal{C}_l$
N
Y
$\mathrm{n_T > n_D}$
D2C (Alg. 2, lines 13–17)
$\widetilde{\mathcal{D}}_{\mathrm{occ}}$
Association 1
Association 2
Association l
…
$\mathcal{D}_1$
$\mathcal{D}_{\mathrm{NMS1}}$
$s \geq s_{\mathrm{track}}$
Delete
$s \geq s_{\mathrm{occ}}$
$\widetilde{\mathcal{D}}$
Adapted NMS
$\mathcal{D}_{\mathrm{occ}}$

Figure 6.6: Pipeline of the proposed TWC. The adapted NMS and the association is highlighted in blue and orange, respectively. The assignment of detections to clusters (D2C) of Algorithm 2 is depicted green, and the building of clusters according to Equation (6.7) is colored red. The association task is performed in each cluster separately. The occluded detections are only used if $n_{\mathrm{T}} > n_{\mathrm{D}}$ holds, which is indicated by the switch symbols, e.g., $n_{\mathrm{T}} > n_{\mathrm{D}}$ is true in cluster $\mathcal{C}_2$.

First, track clusters are built according to Equation (6.7). Then, the confident normal detections $\mathcal{D}_1$ and occluded detections $\widetilde{\mathcal{D}}_{\mathrm{occ}}$ from the adapted NMS are assigned to the clusters (D2C, Algorithm 2). Different from previously presented approaches, the association task is divided: It is performed in each cluster $\mathcal{C}_k \in \mathfrak{C}$ separately. Note that, according to line 12 in Algorithm 2, the occluded detections $\widetilde{\mathcal{D}}_{\mathrm{occ}}$ are only used if $n_{\mathrm{T}} > n_{\mathrm{D}}$ holds, i.e., if missing detections are identified.

Evaluation

The TWC approach is compared with the basic tracking framework in the following. Again, the IoU distance is taken as association distance $d = d_{\text{IoU}}$. Table 6.6 lists the main performance measures for both methods on the three evaluation datasets.

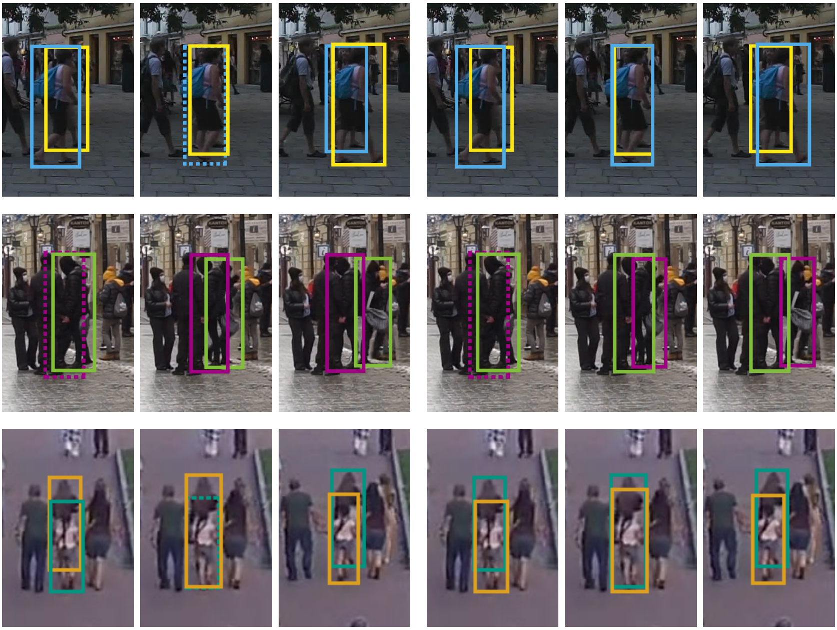
Table 6.6: Comparison of TWC and the base framework on three different datasets. Notable improvements are only obtained on MOT17 val.

	MOT17 val			PP22 test			SOMPT22 train		
Method	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
Baseline	67.6	65.2	70.5	**62.1**	**67.7**	57.5	55.4	56.0	55.0
TWC	**68.1**	**65.5**	**71.2**	**62.1**	**67.7**	**57.6**	**55.5**	**56.1**	**55.1**

With a plus of 0.5 HOTA and an increase in AssA of 0.7 points, the TWC approach outperforms the baseline on the MOT17 dataset. However, on the PP22 and SOMPT22 dataset, only marginal gains can be observed. Before reasons of this worse performance on these datasets are given, positive examples of the TWC are shown in Figure 6.7.

In all of the three depicted sequences, the results of the TWC approach are superior to the results of the base framework, since no IDSWs occur in contrast to the baseline. Looking at the top and bottom sequence, respectively, the improved tracking performance of TWC can be attributed to an additional occluded detection that is leveraged in the middle frame. No detections are missing which simplifies the association task and the IDSW of the base framework is prevented.

In the sequence displayed in the second row of Figure 6.7, no additional detections are incorporated by the TWC approach. The correct assignments in the middle frame stem from the separate association in each cluster. In the example, each cluster contains only a single track, as no two tracks overlap



(a) Base framework

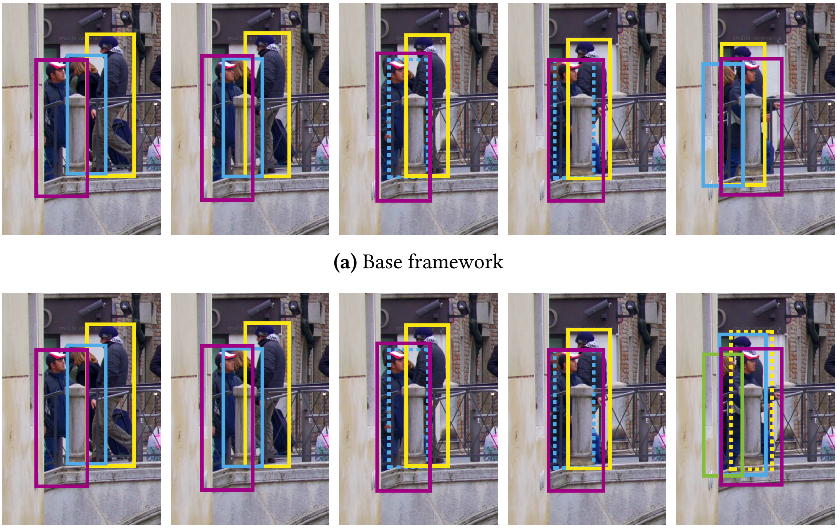
(b) TWC

Figure 6.7: Qualitative comparison of the base framework and TWC on example sequences from the MOT17, PP22, and SOMPT22 dataset (top to bottom). Leveraging heavily-occluded detections, the TWC can resolve ambiguities in the association (b) that lead to IDSWs in the base framework (a).

by more than $o_{\text{cluster}} = 0.7$. The two shown detections[1] have both been put into the cluster containing the green track. Since the green detection fits better, it is assigned to the track, while the other unassigned detection is matched to the purple track in the final association step.

This kind of *greedy* matching introduced in some cases of the TWC approach can be beneficial but may also harm the performance as illustrated in the example of Figure 6.8.

[1] To be precise, the updated track boxes after the Kalman filter update step are depicted and not the assigned detections. This detail is ignored in favor of a more comprehensible explanation.



(a) Base framework

(b) TWC

Figure 6.8: Failure case of TWC (b) compared to the base framework that makes no error in this sequence (a). The failure is caused by the divided association in the TWC approach that does not take all possible track–detection assignments into account.

Due to an inaccurate motion prediction of the inactive blue track, the green and the blue detection are assigned to the cluster of the blue track in the last frame of Figure 6.8b. The other tracks again build their own clusters because the overlaps are lower than o_{cluster}. As a consequence of the inaccurate motion prediction, the blue detection is put in the *wrong* cluster and assigned to the blue track. However, it should be assigned to the yellow track. The remaining green detection does not fit to any unassigned track (the yellow one) in the final association step, so it starts a new track. Without the separation of the association in the clusters but rather considering all possible assignments with the Hungarian algorithm, the base framework makes no error on this example sequence as can be seen in Figure 6.8a. Since the Hungarian method is generally preferable over a greedy matching (Table 5.4), the separation of the association task can be regarded as a drawback of the TWC approach.

Moreover, the example in Figure 6.8 shows that TWC depends on a very accurate motion prediction to prevent detections being assigned to the wrong cluster. Thus, it also requires detections with high accuracy, especially under occlusion. The quality of detections in crowded scenes might be worse on the PP22 and SOMPT22 dataset compared to the MOT17 dataset, as MOT17 is the dataset on which the applied detector has been trained. This can be an explanation why the TWC performs not as good on these datasets (Table 6.6). Furthermore, the frame rate of the sequences in the PP22 dataset is much lower compared to the ones in MOT17 (5 Hz vs. 30 Hz). This negatively influences the accuracy of the motion prediction and thus the performance of the TWC approach.

Before discussing differences compared to the previously presented BYTEv2 association, the parameters of TWC are ablated. Table 6.7 summarizes the influence of the overlap thresholds of the adapted NMS o_{NMS1} and o_{NMS2} on MOT17 val. Note that $o_{\text{cluster}} = o_{\text{NMS1}}$ is set for the TWC approach and that the rows without o_{NMS2} (–) are results of the base framework.

Table 6.7: Influence of the NMS thresholds o_{NMS1} and o_{NMS2} in TWC on MOT17 val. For $o_{\text{NMS2}} \leq 0.9$, leveraging heavily-occluded detections improves the tracking performance for all evaluated values of o_{NMS1}.

o_{NMS1}	o_{NMS2}	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
0.6	–	67.1	64.7	70.1	68.3	**85.5**	74.3	84.9
0.6	0.7	67.4	65.0	70.5	68.7	85.2	75.0	84.8
0.6	0.8	67.7	65.3	70.6	69.0	85.4	75.1	84.9
0.6	0.9	67.9	65.3	71.0	69.4	84.9	75.2	85.0
0.6	1.0	66.8	64.7	69.5	69.0	84.3	74.1	83.7
0.7	–	67.6	65.2	70.5	69.0	85.2	75.0	84.7
0.7	0.9	**68.1**	**65.5**	**71.2**	**69.5**	85.1	**75.7**	**85.3**
0.8	–	66.5	65.2	68.2	69.3	84.8	73.0	83.5
0.8	0.9	66.6	65.3	68.4	69.3	84.9	73.2	83.7

Similar findings are made as for BYTEv2: Up to $o_{\text{NMS2}} = 0.9$, the performance improves for all evaluated values of o_{NMS1}, but setting o_{NMS2} too high ($o_{\text{NMS2}} = 1.0$), the performance decreases because duplicate detections are introduced.

For $o_{\mathrm{NMS2}} = 0.9$, HOTA is enhanced by 0.8, 0.5, and 0.1 points for $o_{\mathrm{NMS1}} = 0.6$, $o_{\mathrm{NMS1}} = 0.7$, and $o_{\mathrm{NMS1}} = 0.8$, respectively, w.r.t. the baseline.

In Table 6.8, the HOTA values when applying different confidence thresholds for the occluded detections s_{occ} in TWC are given.

Table 6.8: Influence of s_{occ} in TWC on MOT17 val. The best results are achieved with a small score threshold for the occluded detections.

s_{occ}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
HOTA	67.9	**68.1**	67.8	67.3	67.4	67.1	67.4	67.9	67.4

The results show that the TWC approach is not as robust to the choice of s_{occ} as BYTEv2. One reason for this could be that in TWC, occluded detections can be preferred over the *normal* detections in the association, as all detections within a cluster are treated equally. Contrarily, in BYTEv2, the occluded detections do not alter the assignment of normal detections to tracks, as those are matched in the first stage. The occluded detections are only considered for matching to the remaining unassigned tracks in the second association stage. Another advantage of the BYTEv2 association is that the maximum distance threshold for the second stage, i.e., for the occluded detections, can be set different as for the first stage, i.e., for the normal detections. A more strict requirement for matching the on average more inaccurate occluded detections is beneficial for the overall association accuracy.

Besides leveraging the occluded detections in the tracking process, the main idea behind the TWC approach is to utilize the track information to identify areas with missing detections and only incorporate occluded detections from these areas. BYTEv2 achieves exactly that by considering all occluded detections in the second association stage but not incorporating the occluded detections that do not fit to any of the unassigned tracks. Overall, BYTEv2 is superior to TWC because it does not suffer from the elaborated shortcomings of the TWC approach.

6.1.3 Occlusion-Aware Initialization

In the previous section, the spatial proximity of tracks has been leveraged to identify areas in the image with missing detections in order to incorporate additional detections and improve the association performance. Next, another method is introduced that makes use of the available track information. Its goal is to enhance the accuracy of the track initialization process.

Before the remaining unassigned detections from the association are taken to initialize new tracks, they are filtered with a minimum confidence threshold s_{init} to remove FP detections that would introduce ghost tracks, i.e., FP tracks. Still, some FP detections with high confidence can remain in the filtered detection set. To further remove such FPs, an often applied continuity requirement has already been investigated in the base framework, which allows the initialization of a track only if multiple detections of the same target are available in consecutive frames. Since the strategy not only filters FP detections but also many TPs, the tracking performance could not be improved (Table 5.1).

While looking at the continuity of detections, the surroundings of a detection have not been considered by previous literature in the track initialization process. For an isolated detection, it is difficult to determine whether it is a TP or a FP as no information about the surroundings is available. However, in crowded scenes, FP duplicate detections can be revealed with the help of the current track information. The idea of the OAI is to compute overlaps of unassigned detections with the current set of tracks and prohibit a track initialization if overlaps are too large [Sta23b]. It is argued that an unassigned detection with a severe overlap to an already tracked target is likely a duplicate detection and shall be removed to prevent the start of ghost tracks. Formally, let $\mathcal{D}^{\text{u}}$ denote the set of unassigned detections remaining from the association and $\mathcal{T}$ the updated tracks after the Kalman filter update step. Given an unassigned detection $\text{D}^{\text{u}} \in \mathcal{D}^{\text{u}}$, the maximum overlap o_{max} measured in IoU with all tracks $\text{T} \in \mathcal{T}$ is computed:

$$o_{\text{max}}(\text{D}^{\text{u}}, \mathcal{T}) = \max_{\text{T} \in \mathcal{T}} \{\text{IoU}(\text{D}^{\text{u}}, \text{T})\}. \tag{6.8}$$

If the maximum overlap exceeds a predefined threshold o_{init}, the detection is deemed a duplicate and deleted. Consequently, the final set of detections that is used in the OAI $\mathcal{D}_{\mathrm{init}}$ follows as

$$\mathcal{D}_{\mathrm{init}} = \{\mathrm{D}^{\mathrm{u}} \,|\, \mathrm{D}^{\mathrm{u}} \in \mathcal{D}^{\mathrm{u}} \wedge o_{\max}(\mathrm{D}^{\mathrm{u}}, \mathcal{T}) \leq o_{\mathrm{init}}\}. \tag{6.9}$$

The procedure of the OAI is illustrated for two updated tracks and an unassigned detection in Figure 6.9.

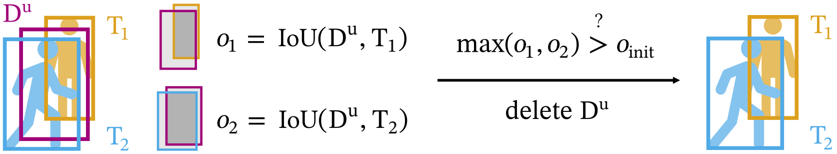

D^{u}
T_1
T_2
$o_1 = \mathrm{IoU}(\mathrm{D}^{\mathrm{u}}, \mathrm{T}_1)$
$o_2 = \mathrm{IoU}(\mathrm{D}^{\mathrm{u}}, \mathrm{T}_2)$
$\max(o_1, o_2) \overset{?}{>} o_{\mathrm{init}}$
delete D^{u}
T_1
T_2

Figure 6.9: Scheme of the proposed OAI. The unassigned duplicate detection D^{u} is deleted because its maximum IoU to already tracked targets, namely o_2, exceeds the threshold o_{init}. Thus, the start of a FP track is prevented.

In the toy example, an additional FP detection under severe occlusion D^{u} remains unassigned. The two overlaps o_1 and o_2 to the tracks T_1 and T_2 of the detection's surroundings are computed. Since the maximum overlap $o_{\max} = o_2$ exceeds the threshold o_{init} of the OAI, the detection is deemed a FP and deleted. Thus, no ghost track is started from this duplicate detection.

Evaluation

Looking at the surroundings of a detection, the OAI follows an approach different from BYTE and BYTEv2 that apply two confidence thresholds s_{track} and s_{init} for detections being used in the association and for initialization, respectively. The OAI also differs from the continuity strategy of starting tracks only from detections that are confirmed in n_{init} consecutive frames (Section 5.5). However, it is possible to combine the OAI with the two other approaches. Table 6.9 summarizes results with various initialization settings, strategies, and combinations.

Table 6.9: Comparison of different initialization strategies and settings on MOT17 val. Using a slightly higher score threshold for initialization than for association ($s_{\text{init}} > s_{\text{track}}$) increases HOTA, while leveraging a tentative track state ($n_{\text{init}} > 1$) does not. Additionally employing the OAI leads to further improvements.

s_{track}	s_{init}	n_{init}	OAI	HOTA	DetA	AssA	DetRe	DetPr
0.6	0.6	1	✗	67.4	**66.2**	69.2	**71.1**	83.4
0.7	0.7	1	✗	67.6	65.2	70.5	69.0	**85.2**
0.6	0.7	1	✗	67.8	66.1	70.0	70.8	83.9
0.6	0.7	2	✗	67.7	65.8	70.0	70.3	84.1
0.6	0.7	1	✓	**68.2**	65.9	71.0	70.0	84.7
0.6	0.7	2	✓	68.1	65.6	**71.2**	69.6	84.9

The first two rows correspond to the standard initialization of the base framework, where only a single confidence threshold $s_{\text{track}} = s_{\text{init}}$ is used to filter the detections for association and track initialization. Setting $s_{\text{track}} = s_{\text{init}} = 0.7$ yields the best results (see also Figure 5.5). Utilizing a slightly higher threshold for initialization as in BYTE(v2), HOTA is increased by 0.2 points. The fourth row shows that using a continuity requirement in the track initialization ($n_{\text{init}} = 2$) does not enhance the overall performance, also not in combination with the BYTE(v2) initialization (last row). The second last row depicts the results of the OAI when combined with the BYTE(v2) initialization. One observes that another gain of 0.4 HOTA is achieved compared to only using the two confidence thresholds. DetA remains nearly constant with an increase in DetPr but a decrease in DetRe. This is because some TPs are also removed from the OAI, besides filtering FP detections. However, AssA is increased considerably by 1.0 points, which indicates that the removal of unassigned detections with high overlaps to already tracked targets prevents many incorrect assignments. Next to the superior tracking performance in comparison with the standard initialization, another advantage is that the OAI can be combined with other strategies like using the continuity requirement. If AssA is more important to the application than DetA, it is beneficial to combine the OAI with $n_{\text{init}} = 2$, which results in a slightly higher AssA (+0.2).

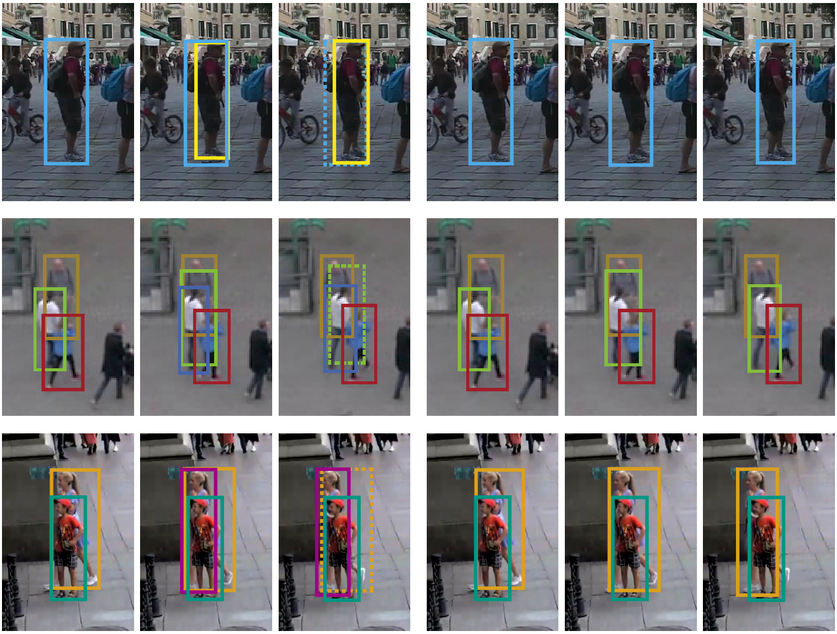
Next, the generalization ability of the OAI is investigated, conducting experiments on the three evaluation datasets. Table 6.10 lists the quantitative tracking results for the OAI in combination with the BYTE(v2) initialization compared to the standard method.

Table 6.10: Comparison of initialization strategies on three different datasets. The OAI yields, on top of the BYTE(v2) strategy, notable enhancements of the overall performance measured in HOTA compared to the standard initialization.

	MOT17 val			PP22 test			SOMPT22 train		
Initialization	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
Standard	67.6	65.2	70.5	62.1	**67.7**	57.5	55.4	56.0	55.0
+ BYTE(v2)	67.8	**66.1**	70.0	62.2	67.2	58.1	56.1	**56.7**	55.7
+ OAI	**68.2**	65.9	**71.0**	**62.4**	67.1	**58.5**	**56.3**	**56.7**	**56.0**

In combination with the BYTE(v2) initialization, the OAI improves the overall tracking performance measured in HOTA with a plus of 0.6, 0.3, and 0.9 points w.r.t. the standard initialization on the MOT17, PP22, and SOMPT22 dataset, respectively. Furthermore, better results in comparison to using only the BYTE(v2) initialization are achieved.

The benefits of the OAI can also be observed qualitatively. In Figure 6.10, tracking results when applying only the BYTE(v2) initialization and additionally using the OAI are depicted. In all three example sequences, an IDSW occurs in the baseline as a consequence of an unassigned duplicate detection that starts a ghost track. The OAI, however, identifies the duplicate detections as FPs with the help of the track information from the surrounding targets. Removing the FPs, it successfully prevents the wrong track initializations and thus also the IDSWs. Note that next to the initial IDSWs in the baseline, the propagated ghost tracks could lead to further errors in consecutive frames, which is also prevented by the OAI.



(a) Baseline

(b) OAI

Figure 6.10: Qualitative comparison of OAI (b) and baseline initialization (a) on example sequences from the MOT17, PP22, and SOMPT22 dataset (top to bottom). The OAI prevents both the start of a ghost track and an IDSW in each depicted sequence.

Only one additional parameter is introduced with the OAI: the maximum allowed overlap between an unassigned detection and the already tracked targets o_{init} to initialize a new track. The influence of this parameter on the tracking performance is evaluated in Table 6.11.

Table 6.11: Influence of o_{init} in the OAI on MOT17 val. A too small overlap threshold delays the initialization under vanishing occlusion, whereas less ghost tracks are prevented with a too high threshold.

o_{init}	0.05	0.2	0.35	0.5	0.65	0.8	0.95
HOTA	64.3	67.0	**68.2**	68.0	67.9	67.8	67.8

Obviously, o_{init} must not be set too small as this delays the initialization of targets until they are nearly completely visible without any remaining occlusion. In crowded scenes, some targets are partially occluded the whole time such that they are not tracked at all when setting a too low overlap threshold. On the other hand, if o_{init} is set too large, the influence of the OAI vanishes, since $o_{\text{init}} = 1$ corresponds to not applying the method at all. For the applied values of $o_{\text{init}} \in \{0.35, 0.5, 0.65\}$, higher HOTA values compared to the baseline (67.8) are achieved on MOT17 val, which indicates a good robustness of the OAI to the choice of o_{init}, when not set too small.

To conclude, the OAI improves the accuracy of the track initialization and thus enhances the overall tracking performance. This is achieved by leveraging the current set of tracks to derive context information of unassigned detections. The information is used to identify whether such detections are duplicates, and if so, they are deleted to prevent the start of ghost tracks.

6.1.4 Combinations

Three methods to improve the use of available detections and tracks have been introduced: BYTEv2, TWC, and the OAI. The first two mainly aim at increasing the association performance incorporating the additional occluded detections from the adapted NMS, while the OAI enhances the initialization accuracy. With these different objectives, it is promising to combine the approaches. Results on the three evaluation datasets are found in Table 6.12.

Table 6.12: Combination of approaches to improve the use of available detections and tracks. The largest improvements w.r.t. the baseline are achieved with BYTEv2. Further small gains are obtained when additionally employing the OAI and TWC.

			MOT17 val			PP22 test			SOMPT22 train		
BYTEv2	OAI	TWC	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
✗	✗	✗	67.6	65.2	70.5	62.1	67.7	57.5	55.4	56.0	55.0
✓	✗	✗	69.3	67.4	71.7	63.4	**68.2**	59.5	56.3	57.4	55.5
✓	✓	✗	**69.4**	**67.5**	71.9	63.5	68.0	59.9	56.4	57.4	**55.6**
✓	✓	✓	**69.4**	67.4	**72.0**	**63.6**	68.0	**60.0**	**56.5**	**57.6**	**55.6**

The first two rows depict the performance measures of the base framework and of BYTEv2 alone, respectively. Besides the large improvements of BYTEv2 w.r.t. the base framework, combining it with the OAI yields further improvements on all of the three datasets.

The last line of Table 6.12 shows the results when additionally leveraging the TWC approach. In this combination, the TWC is performed first, with the adaptation that the unassigned occluded detections are saved. Then, the second association stage of BYTEv2 is executed with the unassigned occluded detections and the low-confidence normal detections. Finally, the OAI is performed using the unassigned high-confidence normal detections. The overall tracking performance measured in HOTA is only slightly increased on PP22 and SOMPT22 but keeps equal on MOT17 compared to BYTEv2+OAI. So only marginal gains can be achieved when additionally applying TWC in combination with BYTEv2. This is expected as both methods have the same goal, while it has already been elaborated in Section 6.1.2 that BYTEv2 yields better results than TWC. Moreover, the TWC approach has another shortcoming: It is designed to work with the IoU distance d_{IoU} as association measure. As will be seen in the next section, the utilization of combined motion- and appearance-based association distances can significantly improve the tracking accuracy. Application of such a sophisticated association distance is not directly possible following the TWC approach, which makes the combination BYTEv2+OAI+TWC inferior to BYTEv2+OAI. Consequently, BYTEv2+OAI is the best tracking framework within this thesis to improve the utilization of detections and tracks. It will be used in Section 6.3 together with more advanced distance measures that are presented in the following.

6.2 Fusion of Motion and Appearance Information

In the base framework, either motion or appearance information has been used. To achieve a high processing speed of the overall system, several trackers rely fully on motion information [Bew16, Cao23, Yan23, Zha22c, Zho20] and refrain from using a REID model to extract appearance information, since this comes with an additional computational overhead. However, it will be shown in Chapter 7 that even a sophisticated tracking framework as presented in this thesis can achieve real-time speed on standard hardware despite using both motion and appearance cues. With this finding in mind, there is hardly any reason why one should not leverage the valuable appearance information in the association task.

The fusion of available motion and appearance information is an underexplored research field of MPT. Although there are various approaches to combine the two different cues [Aha22, Du23, Mag23, Wan20, Woj17], there is a lack of in-depth analyses of the combined distance measures found in the literature. For instance, some formulas for fusing motion and appearance distances are introduced without thorough explanation, and often, no ablation studies, which would give more insights into the working mechanisms, are performed [Aha22, Wan20, Woj17]. Furthermore, a detailed comparison of the existing fusion approaches is missing. As the tracking frameworks generally comprise various detectors, motion models, etc., one cannot simply compare the tracking results from two publications that propose different fusion approaches for motion and appearance information. For these reasons, this thesis conducts a detailed analysis of fusion strategies from the literature, and shortcomings in existing methods are identified. Based on that, distance measures for an improved utilization of available motion and appearance information are proposed [Sta23b, Sta23d].

In Section 6.2.1, existing association distances for motion and appearance information are combined with the base framework, making sure that all other tracking components are identical, thus enabling a fair comparison. The different distance measures are analyzed in detail and their weaknesses are elaborated. Then, improvements are proposed in Section 6.2.2 on the basis of the findings. Finally, a comprehensive evaluation is given in Section 6.2.3.

6.2.1 Existing Fusion Approaches

One of the most popular tracking frameworks and one of the first approaches that leveraged appearance features extracted by a deep CNN for person REID in MPT is DeepSORT [Woj17]. For motion information, the squared Mahalanobis distance d_{Mah} (Equation (5.36)) is calculated, and for appearance information, the cosine distance of the extracted features is taken. Note that here and in the following, the EMA strategy is used to compute the appearance distance: $d_{\text{app}} = d_{\text{EMA}}$ (Equations (5.44) and (5.45)). DeepSORT introduces a *gating* mechanism that forbids the association of detections and tracks with a squared Mahalanobis distance above $d_{\text{max,Mah}} = 9.4877$. This value is drawn from the inverse chi-square distribution with 4 dimensions of freedom for a confidence level of 95 %. Consequently, the distance measure of DeepSORT d_{DS} can be stated as

$$d_{\text{DS}} = \begin{cases} d_{\text{app}} & \text{if } d_{\text{Mah}} \leq d_{\text{max,Mah}} \\ \varkappa & \text{otherwise} \end{cases} \tag{6.10}$$

with $\varkappa \gg d_{\text{max}}$ denoting a large constant, e.g., $\varkappa = 10{,}000$ in the implementation (note: $d_{\text{app}} = d_{\cos} \in [0, 2]$). Next to the motion constraint $d_{\text{max,Mah}}$, the maximum allowed association distance d_{max} prevents matches of detections and tracks with large appearance distance d_{app}. Although both motion and appearance information is considered, the DeepSORT fusion strategy has two major weaknesses. First, the squared Mahalanobis distance is used for calculating motion similarity. As already stated in the evaluation of the base framework (Section 5.6.5), the squared Mahalanobis distance is not very accurate if the uncertainties of the tracks' motion state estimates from the Kalman filter are high. Second, apart from utilizing the motion information for gating, within the gating area, i.e., $d_{\text{Mah}} \leq d_{\text{max,Mah}}$, the exact values of the motion distance d_{Mah} do not matter and only the appearance distance determines the outcome of the association. In other words, the DeepSORT distance d_{DS} uses motion information only for preventing unlikely assignments but relies fully on appearance information for the remaining association candidates. This strategy of utilizing the available information is not very effective.

The way of combining the squared Mahalanobis distance and appearance distance is improved in the JDE framework from [Wan20]. Instead of using the motion information only for gating, the two distances are fused with a weighted sum to

$$d_{\mathrm{JDE}} = \lambda d_{\mathrm{app}} + (1 - \lambda) d_{\mathrm{Mah}}, \tag{6.11}$$

where $\lambda \in [0, 1]$ is the weighting factor that determines the influence of the two information sources. The distance d_{JDE} is adopted from other well-known MPT works, for example, FairMOT [Zha21] and StrongSORT [Du23]. In all of the three works, λ is set to 0.98 but an analysis of the choice of this value is missing. In the JDE paper, this value is not even given and it has to be looked up in the official implementation[1]. The reason for such a high λ lies in the different scales of the combined distance measures. While the appearance cosine distance is bound in $[0, 2]$, the Mahalanobis distance can become indefinitely large, i.e., $d_{\mathrm{Mah}} \in [0, \infty)$. Besides the undesirable scale difference of the involved single distance measures, which makes d_{JDE} very sensitive to the choice of λ, the imprecise Mahalanobis distance is still used for motion information in the JDE distance.

Another approach for combining motion and appearance cues is found in BoT-SORT[2] [Aha22]. Different from the previous fusion methods, the IoU distance d_{IoU} is applied for motion information instead of the squared Mahalanobis distance. Two gating thresholds $d_{\mathrm{max,IoU}}$ and $d_{\mathrm{max,app}}$ are introduced that are intended to prevent unlikely assignments with the help of IoU distance and appearance distance, respectively. A further difference to the DeepSORT and JDE distance lies in the fusion strategy of the BoT-SORT distance measure d_{BoT}. In contrast of using a weighted sum, the minimum of motion and appearance distance is leveraged, whereby the latter is scaled by 0.5 such that both IoU distance and appearance distance are in the range $[0, 1]$. Putting all

[1] https://github.com/Zhongdao/Towards-Realtime-MOT (accessed on July 16, 2024)

[2] The REID model BoT from [Luo19] is the namesake.

together, the BoT-SORT distance d_{BoT} can be calculated as follows:

$$d_{\mathrm{BoT}} = \min\{\tilde{d}_{\mathrm{app}}, d_{\mathrm{IoU}}\} \quad \text{with} \tag{6.12}$$

$$\tilde{d}_{\mathrm{app}} = \begin{cases} 0.5\, d_{\mathrm{app}} & \text{if } d_{\mathrm{app}} \leq d_{\mathrm{max,app}} \wedge d_{\mathrm{IoU}} \leq d_{\mathrm{max,IoU}} \\ 1 & \text{otherwise} \end{cases} . \tag{6.13}$$

Unfortunately, the authors of BoT-SORT do not give a motivation of why using the minimum of motion and appearance distance should be beneficial compared to using a weighted sum as in d_{JDE}. Moreover, only a negligible gain is achieved with d_{BoT} compared to simply applying the IoU distance d_{IoU} alone in their experiments on MOT17 val [Aha22]. This indicates that fusing the two distance measures for motion and appearance information by taking the minimum is not a powerful strategy. That claim can be supported by the following two facts. First, the BoT-SORT distance represents either the appearance distance or the motion distance between a detection and a track—depending on which distance is smaller—but not both. Thus, one of the two information sources is discarded. Second, when comparing the distances between a track and two candidate detections for association, the one distance can be the appearance distance while the other one is the motion distance. The method provides no sound basis for directly comparing distances of two different types, i.e., motion and appearance distance, which can lead to unpredictable results. Based on the findings of this section, improved combinations of motion and appearance distances are proposed in the following.

6.2.2 Proposed Distance Measures

The analysis of existent combined distance functions for motion and appearance information has revealed several shortcomings:

- Use of the imprecise Mahalanobis distance for motion information (DeepSORT, JDE).
- Suboptimal fusion of the two information sources in the sense that motion cues are only used for gating (DeepSORT) or that either one or the other information is decisive but not both (BoT-SORT).

- Missing detailed understanding of the working mechanism and the influence of involved parameters due to a lack of ablation experiments.

To address the first problem, various IoU-based distance measures shall be investigated for motion information. Remember that the evaluation in the base framework has demonstrated that the IoU distance achieves better results than the squared Mahalanobis distance when only motion information is considered (Table 5.5). Moreover, further developments of the IoU, e.g., generalized IoU (GIoU) [Rez19] or distance IoU (DIoU) [Zhe20], have shown promising results when combined with appearance information in a previous work of this thesis' author [Sta23a]. As will be seen shortly, the GIoU and DIoU have the advantage that they yield different values for non-overlapping boxes with various spatial distances, for which the IoU is always zero. These measures are briefly introduced as follows.

To calculate the GIoU of two boxes A and B, the smallest box C enclosing both A and B has to be computed as intermediate step. Then, the $\mathrm{GIoU}(A, B)$ between A and B is given by

$$\mathrm{GIoU}(A,B) = \mathrm{IoU}(A,B) - \frac{|C \setminus (A \cup B)|}{|C|} \tag{6.14}$$

with $|C \setminus (A \cup B)|$ denoting the area of C minus the union of A and B. The subtrahend in Equation (6.14) gets zero if $A = B$ and tends to one if the spatial distance of the boxes A and B tends to infinity. As the IoU is bound in the range $[0, 1]$, the GIoU lies in $(-1, 1]$.

While the distance of two boxes A and B is to a certain extent encoded in the enclosing box C, it is not modeled explicitly. This is done by the DIoU that takes the Euclidean distance $d_{\mathrm{L2}}(A, B)$ of the box centers into account. The smallest enclosing box C still has to be computed, since its diagonal d_C is necessary for calculating the DIoU:

$$\mathrm{DIoU}(A,B) = \mathrm{IoU}(A,B) - \frac{d_{\mathrm{L2}}^2(A,B)}{d_C^2}. \tag{6.15}$$

With the same argumentation as for the GIoU, it can be derived that the DIoU lies also in the range $(-1, 1]$.

The three IoU-based distance measures are illustrated and compared with two examples in Figure 6.11.

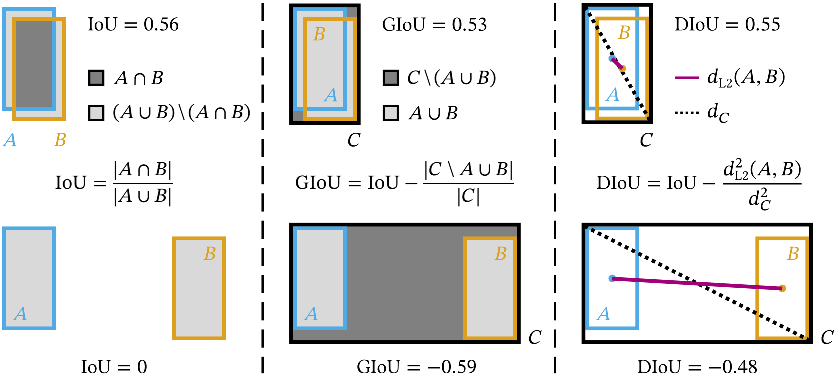

IoU = 0.56
$A \cap B$
$(A \cup B) \setminus (A \cap B)$
A
B
$\mathrm{IoU} = \frac{|A \cap B|}{|A \cup B|}$
GIoU = 0.53
$C \setminus (A \cup B)$
$A \cup B$
C
$\mathrm{GIoU} = \mathrm{IoU} - \frac{|C \setminus A \cup B|}{|C|}$
DIoU = 0.55
$d_{\mathrm{L2}}(A, B)$
d_C
$\mathrm{DIoU} = \mathrm{IoU} - \frac{d^2_{\mathrm{L2}}(A, B)}{d^2_C}$
IoU = 0
GIoU = −0.59
DIoU = −0.48

Figure 6.11: Illustration and comparison of IoU, GIoU, and DIoU with two examples. While similar values for two overlapping boxes are obtained with all measures, GIoU and DIoU yield negative values for non-overlapping boxes, for which IoU is always zero.

IoU, GIoU, and DIoU all assess the similarity of bounding boxes and give nearly identical results when comparing boxes with high overlaps. However, regarding non-overlapping boxes, the IoU is always zero no matter how far the boxes are away from each other. In contrast, GIoU and DIoU take on negative values if the examined boxes do not overlap, and become the smaller the larger the distance between the box centers.

Just like with the IoU, one gets a distance measure for GIoU and DIoU, respectively, when subtracting them from one:

$$d_{\mathrm{GioU}}(A, B) = 1 - \mathrm{GIoU}(A, B), \tag{6.16}$$

$$d_{\mathrm{DioU}}(A, B) = 1 - \mathrm{DIoU}(A, B). \tag{6.17}$$

Note that $d_{\mathrm{GioU}}, d_{\mathrm{DioU}} \in [0, 2)$ holds.

Combining the presented IoU-based distances with the appearance distance, neither a motion-based gating nor a minimum function should be used because such strategies cannot exploit the full potential of the two information sources. Instead, utilizing a weighted sum as in d_{JDE} leverages both components and furthermore, the influence of each component can be controlled. Thus, the following combined distance measures are proposed:

$$\begin{aligned} d_{\mathrm{comb,IoU}} &= \lambda d_{\mathrm{app}} + (1-\lambda) d_{\mathrm{IoU}}, && (6.18)\\ d_{\mathrm{comb,GIoU}} &= \lambda d_{\mathrm{app}} + (1-\lambda) d_{\mathrm{GIoU}}, && (6.19)\\ d_{\mathrm{comb,DIoU}} &= \lambda d_{\mathrm{app}} + (1-\lambda) d_{\mathrm{DIoU}}. && (6.20) \end{aligned}$$

In the next section, those distance measures are compared with the previous fusion approaches from the literature and a detailed analysis is conducted.

6.2.3 Evaluation

To evaluate the performance of the presented distance measures, the base framework from Chapter 5 is leveraged. Except replacing the association distance, all tracking components (detection, REID, motion model, and track management) are kept unchanged to enable a fair comparison. Experiments are conducted on the datasets MOT17 val, PP22 test, and SOMPT22 train.

Because of different scales and composition of the applied distance measures, the maximum allowed distance for association $d_{\max}$ is adapted for every measure separately. Moreover, parameters of the distance functions, for instance, the weighting factor λ for motion and appearance distance, are also tuned such that the shown evaluation measures represent the best achievable results. This is also important for a meaningful comparison, since parameter configurations of the distance functions from the literature are given without validation, and taking over these parameter values in a different tracking framework can lead to suboptimal results. Indeed, by tuning the parameters of the distance measures, results can be significantly improved. For example, setting $d_{\mathrm{max,app}} = 0.4$ and $d_{\mathrm{max,IoU}} = 0.3$ in BoT-SORT instead of using the values given in the paper [Aha22], i.e., $d_{\mathrm{max,app}} = d_{\mathrm{max,IoU}} = 0.5$, HOTA is enhanced by 0.5 points on MOT17 val. This underlines the importance of a detailed analysis of existing fusion approaches in order to identify weaknesses and improve the utilization of the available information.

Table 6.13 gives a fair quantitative comparison of the existent fusion approaches for motion and appearance distances d_{DS}, d_{JDE}, and d_{BoT}, which has been missing so far in the MPT literature. Moreover, results of the proposed combined distances d_{comb} from the previous section are listed, as well as the baseline results, where only either motion information d_{IoU} or appearance information d_{app} was used.

Table 6.13: Comparison of association distance measures on three different datasets. The proposed combined distances for motion and appearance information (last rows) clearly outperform previous fusion approaches from the literature (middle rows). The baseline results using either motion or appearance cues are also given (first rows).

	MOT17 val			PP22 test			SOMPT22 train		
Distance	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
d_{IoU}	67.6	65.2	70.5	62.1	67.7	57.5	55.4	56.0	55.0
d_{app}	67.9	65.3	71.1	61.5	66.7	57.4	56.0	**56.3**	56.0
d_{DS}	69.0	66.4	72.2	63.9	68.0	60.7	56.4	**56.3**	56.7
d_{JDE}	68.9	66.4	72.1	65.7	**68.1**	63.8	56.6	**56.3**	57.0
d_{BoT}	68.7	66.4	71.6	65.5	67.8	63.8	57.2	56.2	58.3
$d_{\text{comb,IoU}}$	69.7	**66.5**	73.6	66.0	**68.1**	64.6	57.7	**56.3**	59.4
$d_{\text{comb,GIoU}}$	69.7	66.4	73.5	**66.6**	**68.1**	**65.6**	**58.2**	**56.3**	**60.3**
$d_{\text{comb,DIoU}}$	**69.8**	66.4	**73.8**	66.5	**68.1**	65.5	**58.2**	**56.3**	**60.3**

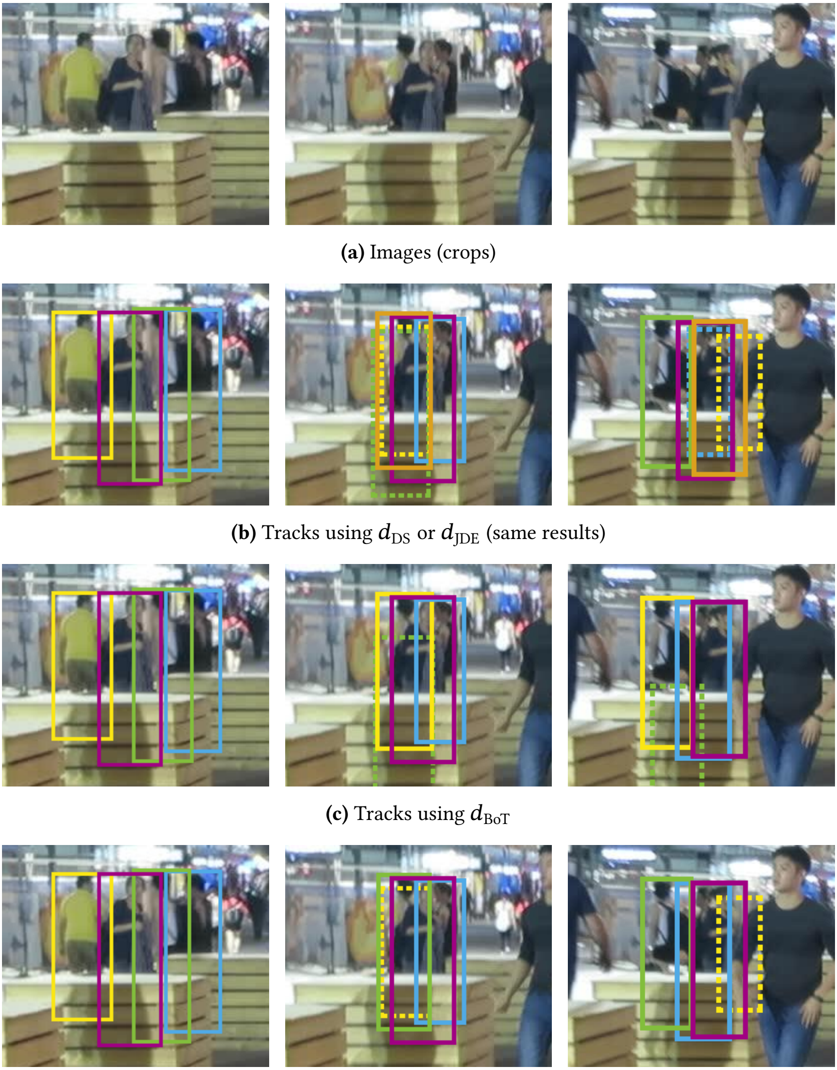
The first essential observation from Table 6.13 is that all distance functions incorporating both motion and appearance information (rows 3–8) outperform the baselines in that only one information source is leveraged (rows 1–2). Gains in HOTA up to 1.1 (MOT17 val), 3.6 (PP22 test), and 1.2 (SOMPT22 train) points w.r.t. the best baseline are obtained. Especially in the PP22 dataset, a good fusion strategy is important since the accuracy of the motion information is lower due to the small frame rate of 5 Hz (Section 4.1.1).

Comparing the three distance measures from the literature, one cannot deem one to be superior to the others: d_{DS}, d_{JDE}, and d_{BoT} achieve the best results among themselves on MOT17 val, PP22 test, and SOMPT22 train, respectively. All those methods fuse the available information in a suboptimal manner, as they are clearly outperformed by the proposed combined distance measures $d_{\text{comb,IoU}}$, $d_{\text{comb,GIoU}}$, and $d_{\text{comb,DIoU}}$. The overall best results are achieved with $d_{\text{comb,DIoU}}$ yielding a plus of 0.8, 0.8, and 1.0 HOTA on MOT17 val, PP22 test, and SOMPT22 train, respectively, compared to the best previous fusion approach. The enhanced tracking performance is attributable to the improved AssA because DetA is similar among the methods as the same set of detections is used. Compared to the best baseline, $d_{\text{comb,DIoU}}$ increases AssA by 2.7 points on MOT17 val, 8.1 points on PP22 test, and 4.3 points on SOMPT22 train.

A qualitative example sequence, where only the proposed combined distance measures solve the association task without error, is shown in Figure 6.12. Due to the high complexity of the example sequence involving four persons, the images without any bounding boxes are depicted in Figure 6.12a. The woman in front is barely moving in the sequence, while two men are walking from right to left behind her. Additionally, a man with a yellow shirt is walking further behind from left to right and becomes fully occluded in the middle and right frame. All of the four persons are heavily occluded, either by each other or by static obstacles.

Leveraging d_{DS} and d_{JDE} yields the same results, which are shown in Figure 6.12b. The orange detection in the middle frame belongs to the green track but is not assigned to it, as the distance exceeds the maximum threshold $d_{\max}$. Instead, it starts a new track that leads to further IDSWs in the right frame.

In Figure 6.12c, the tracking results of using d_{BoT} as association distance are depicted. Although the man with the yellow shirt is fully occluded, the yellow detection is assigned to his track in the middle frame, since the minimum of IoU distance and appearance distance is taken as association measure. Concretely, the IoU distance of the yellow detection to the yellow track is smaller than the IoU distance to the green track due to a bad motion prediction as a consequence of camera motion. Note that the camera motion cannot be seen, since different image parts are cropped to improve the visualization. The takeaway message is that d_{BoT} fully ignores the appearance distance in this situation which causes an IDSW.



(a) Images (crops)

(b) Tracks using d_{DS} or d_{JDE} (same results)

(c) Tracks using d_{BoT}

(d) Tracks using $d_{\text{comb,IoU}}$ or $d_{\text{comb,GIoU}}$, or $d_{\text{comb,DIoU}}$ (same results)

Figure 6.12: Qualitative comparison of fusion methods for motion and appearance information on an example sequence from MOT17 val (a). The proposed distance measures produce no IDSWs on this sequence (d), in contrast to previous approaches (b-c).

Finally, Figure 6.12d shows the tracks generated applying the proposed combined distance measures $d_{\text{comb,IoU}}$, $d_{\text{comb,GIoU}}$, and $d_{\text{comb,DIoU}}$. Fusing an IoU-based distance for motion information with the appearance distance by a weighted sum, all targets are tracked correctly.

So far, it has been demonstrated both quantitatively and qualitatively, that the proposed distance measures achieve a better tracking performance than previous fusion approaches from the literature. What is left is to compare the proposed measures with each other. Recalling Equations (6.18) to (6.20), one notes that the three measures differ only in the variant of the IoU-based motion distance that they are employing. The results from Table 6.13 show that the three combined measures perform on par on MOT17 val, but $d_{\text{comb,GIoU}}$ and $d_{\text{comb,DIoU}}$ yield better results than $d_{\text{comb,IoU}}$ on PP22 test and SOMPT22 train. Before analyzing the reason for this superior performance, it is pointed out that the improvements of GIoU and DIoU are only achievable in combination with the appearance distance, which is demonstrated by the following results.

Table 6.14 summarizes the tracking measures on the three evaluation datasets when applying only the motion distances d_{IoU}, d_{GIoU}, and d_{DIoU} for association. Note that the maximum distance threshold $d_{\max}$ again has been tuned independently for each distance to enable a fair comparison.

Table 6.14: Comparison of IoU-based distance measures on three different datasets. Similar results are obtained, whereby the standard IoU overall performs slightly better than GIoU and DIoU.

	MOT17 val			PP22 test			SOMPT22 train		
Distance	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
d_{IoU}	**67.6**	**65.2**	**70.5**	**62.1**	**67.7**	**57.5**	**55.4**	56.0	**55.0**
d_{GIoU}	67.4	**65.2**	70.0	61.9	**67.7**	57.2	**55.4**	**56.1**	54.9
d_{DIoU}	**67.6**	**65.2**	**70.5**	61.9	**67.7**	57.1	**55.4**	56.0	**55.0**

A similar performance is obtained with all three measures, while the standard IoU distance overall performs slightly better than the GIoU and DIoU distance. In an IoU-based association, $d_{\max} < 1$ holds, which means that not a single assignment is made where the corresponding detection and predicted track box are not overlapping. However, this is where the GIoU and DIoU have an advantage over the basic IoU—the GIoU and DIoU differ for non-overlapping boxes with varying distance, whereas the IoU is always zero (Figure 6.11).

Notice that using the combined distance

$$d_{\text{comb}} = \lambda d_{\text{app}} + (1-\lambda) d_{\text{mot}} \quad \text{with} \quad d_{\text{mot}} \in \{d_{\text{IoU}}, d_{\text{GIoU}}, d_{\text{DIoU}}\}, \tag{6.21}$$

a choice of $\lambda = 0.7$, and a distance threshold of $d_{\max} = 0.55$ (optimal setting), two boxes can be associated even if they do not overlap, i.e., IoU $= 0$. For instance, an assignment is possible in the case of $d_{\text{mot}} = d_{\text{IoU}} = 1 - \text{IoU}$, if the following holds:

$$d_{\text{app}} < \frac{d_{\max} - (1-\lambda) d_{\text{IoU}}}{\lambda} \tag{6.22}$$

$$d_{\text{app}} < \frac{0.55 - (1-0.7)\cdot 1}{0.7} \qquad (d_{\max} = 0.55, \lambda = 0.7, d_{\text{IoU}} = 1) \tag{6.23}$$

$$d_{\text{app}} \lesssim 0.36. \tag{6.24}$$

Thus, if the appearance distance is low enough, i.e., $d_{\text{app}} \lesssim 0.36$ in the example configuration, two boxes can be matched no matter how far away from each other they are when using the IoU for measuring motion distance. In contrast, consider two boxes with large spatial distance and leveraging the GIoU or DIoU instead: $\text{GIoU}, \text{DIoU} \to -1$. One gets

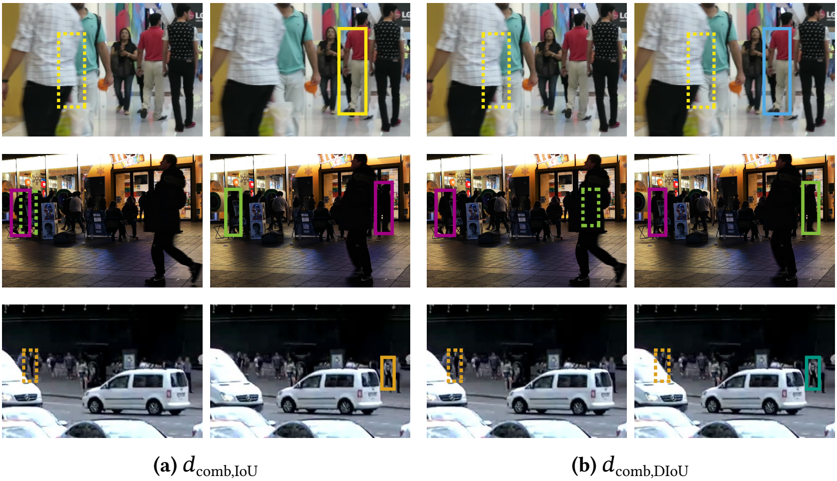
$$d_{\text{app}} < \frac{d_{\max} - (1-\lambda) d_{\text{GIoU}}}{\lambda} \tag{6.25}$$

$$d_{\text{app}} < \frac{0.55 - (1-0.7)\cdot 2}{0.7} \qquad (d_{\max} = 0.55, \lambda = 0.7, d_{\text{GIoU}} = 2) \tag{6.26}$$

$$d_{\text{app}} \lesssim -0.07 < 0, \tag{6.27}$$

so an assignment is prevented no matter how small the appearance distance is. The two calculations show that the advanced IoU distances d_{GIoU} and d_{DIoU} give the combined distance d_{comb} the ability to prevent unlikely assignments based on motion distances of non-overlapping boxes. This is the main reason why $d_{\text{comb,GIoU}}$ and $d_{\text{comb,DIoU}}$ yield an overall higher tracking accuracy than

$d_{\text{comb,IoU}}$ (Table 6.13). Figure 6.13b illustrates examples, where the distance $d_{\text{comb,IoU}}$ makes incorrect assignments of boxes that are far away from each other, which can be prevented with $d_{\text{comb,DIoU}}$.



(a) $d_{\text{comb,IoU}}$

(b) $d_{\text{comb,DIoU}}$

Figure 6.13: Qualitative comparison of using $d_{\text{comb,IoU}}$ and $d_{\text{comb,DIoU}}$ as association distance on example sequences from the MOT17, PP22, and SOMPT22 dataset (top to bottom). The DIoU prevents infeasible assignments with large spatial distances on the image (b), which cannot be done by the IoU within the combined distance measure, as the IoU is always zero for non-overlapping boxes, independent from their distance (a).

Note that the small appearance distance between the associated boxes, which is required for matching, in Figure 6.13a is caused by heavy occlusion, bad lighting conditions, and small object sizes (top to bottom).

In contrast to applying the advanced IoU-based distances $d_{\text{comb,GIoU}}$ and $d_{\text{comb,DIoU}}$, one can prevent assignments of far-apart boxes by fusing the combined distance $d_{\text{comb,IoU}}$ with a minimum IoU requirement $d_{\text{max,IoU}} = 1 - \text{IoU}_{\text{min}}$ similar as in BoT-SORT (Equation (6.13)):

$$\tilde{d}_{\text{comb,IoU}} = \begin{cases} d_{\text{comb,IoU}} & \text{if } d_{\text{IoU}} \leq d_{\text{max,IoU}} \\ d_{\text{max}} + \epsilon & \text{otherwise} \end{cases} \tag{6.28}$$

with ϵ being a very small value, e.g., 10^{-5}. However, such a strategy will also prevent correct assignments of non-overlapping boxes in cases where the predicted track boxes are inaccurate, e.g., when facing camera motion or a low frame rate. This also has been validated experimentally: Exchanging $d_{\text{comb,IoU}}$ with $\tilde{d}_{\text{comb,IoU}}$, i.e., enforcing the minimum IoU requirement, has lead to a decrease in tracking performance on all three evaluation datasets.

Besides preventing unlikely assignments due to a large distance between boxes, $d_{\text{comb,GIoU}}$ and $d_{\text{comb,DIoU}}$ allow the association of non-overlapping boxes if the appearance distance is *sufficiently high*. The larger the motion distance, the smaller the appearance distance must be to allow an assignment, which intuitively is a good property.

The last investigation of this section deals with the weighting factor λ that determines the influence of motion and appearance information in the combined distance d_{comb}. Table 6.15 lists the HOTA values for different choices of λ exemplarily for $d_{\text{comb,DIoU}}$ on MOT17 val.

Table 6.15: Influence of λ in the combined distance measure $d_{\text{comb,DIoU}}$ on MOT17 val. Setting $\lambda = 0.7$, which means that more weight is put on the appearance distance than on the DIoU distance according to Equation (6.20), gives the best results.

λ	0.0	0.1	0.3	0.5	0.6	0.7	0.8	0.9	1.0
HOTA	67.6	67.7	67.9	68.6	69.0	**69.8**	69.7	69.0	68.1

The best results are obtained with $\lambda = 0.7$. This means that more weight is put on the appearance information than on the motion information. In comparison to considering both distances equally, i.e., $\lambda = 0.5$, HOTA is enhanced by 1.2 points, which shows the high importance of fusing the available information effectively. Note that in other configurations, for instance, if a different motion model is used or if the quality of the REID model that extracts the appearance information changes, another balance between motion and appearance distance might be optimal. The proposed combined distance measures allow to adapt to such situations by changing the weighting parameter λ accordingly.

To summarize, it has been shown that existing fusion approaches from the literature for motion and appearance distances have several weaknesses and do not fully utilize the available information. Based on the findings, novel combined distance measures have been proposed that significantly outperform the previous approaches on three different datasets.

6.3 Combination of the Proposed Approaches

Two different directions have been explored to improve the usage of available information in the tracking process. Section 6.1 introduced novel strategies to enhance the utilization of detections and tracks, while Section 6.2 proposed new distance measures for a better combination of motion and appearance information in the association. As various aspects are treated, combining the two concepts is promising.

Recall one of the main ideas of the two-stage association BYTEv2 from Section 6.1.1: matching the high-confidence detections without severe occlusion with the current tracks, before associating the low-confidence and heavily-occluded detections to the remaining unassigned tracks. So far, the IoU distance d_{IoU} has been used as association distance in both stages: $d_1 = d_2 = d_{\text{IoU}}$. However, it is possible to use arbitrary distance functions, so the simple IoU distance shall be exchanged with one of the combined distances d_{comb} from Section 6.2.2. Since the fusion of DIoU and appearance distance $d_{\text{comb,DIoU}}$ has achieved the overall best results in the basic one-stage matching (Table 6.13), it is leveraged as distance measure in the first association stage of BYTEv2: $d_1 = d_{\text{comb,DIoU}}$. With a similar argumentation as in [Zha22c], the IoU is kept as association distance in the second stage. Namely, for the heavily-occluded detections leveraged in the second stage, often only unreliable appearance features can be extracted. Experimental results have shown that such unreliable features can be misleading and harm the association accuracy.

The OAI is not affected by the change of the association distance d_1 in BYTEv2. Since it has lead to a further increase in tracking performance when combined with BYTEv2 (Table 6.12), it is also used in the following experiments. Table 6.16 shows the quantitative tracking results of the combination of BYTEv2+OAI and the advanced distance measure $d_{\text{comb,DIoU}}$ on the three evaluation datasets. For reference, the obtained results when only using one of the two components are also given, as well as the results of the IoU base framework (first row).

Table 6.16: Combination of proposed approaches for the improved utilization of available detections and tracks (BYTEv2+OAI) and for a better fusion of motion and appearance information ($d_{\text{comb,DIoU}}$) on three different datasets. In addition to increasing all performance measures individually, applying BYTEv2+OAI and $d_{\text{comb,DIoU}}$ together leads to further improvements.

		MOT17 val			PP22 test			SOMPT22 train		
BYTEv2+OAI	$d_{\text{comb,DIoU}}$	HOTA	DetA	AssA	HOTA	DetA	AssA	HOTA	DetA	AssA
✗	✗	67.6	65.2	70.5	62.1	67.7	57.5	55.4	56.0	55.0
✓	✗	69.4	**67.5**	71.9	63.5	68.0	59.9	56.4	57.4	55.6
✗	✓	69.8	66.4	73.8	66.5	68.1	65.5	58.2	56.3	60.3
✓	✓	**71.0**	**67.5**	**75.1**	**67.4**	**68.5**	**66.8**	**59.4**	**57.8**	**61.3**

On all datasets, the combination of the proposed approaches leads to further enhancements of the tracking performance. Compared to only applying one improvement, the combination yields a plus of 1.2, 0.9, and 1.2 HOTA on MOT17 val, PP22 test, and SOMPT22 train, respectively. Especially the gains on SOMPT22 train are noteworthy, as the combination achieves a larger increase in HOTA w.r.t. the baseline (+4.0) than the sum of increases (+3.8) when applying only a single component (+1.0/+2.8).

In comparison to the base framework, the proposed tracking system, i.e., BYTEv2+OAI+$d_{\text{comb,DioU}}$, notably improves DetA (+2.3 on MOT17 val, +0.8 on PP22 test, +1.8 on SOMPT22 train) despite using the same detection model. This is mainly achieved by a better utilization of the available detections and the incorporation of usually discarded low-confidence and heavily-occluded detections into the tracking process. On the other hand, large enhancements of AssA compared to the baseline (+4.6 on MOT17 val, +9.3 on PP22 test, +6.3 on SOMPT22 train) are attributable to a good fusion of motion and appearance information, an improved association strategy, and the consideration of detections' neighborhoods in the track initialization. Putting all together, the proposed tracking system obtains a superior tracking performance indicated by a plus of 3.4, 5.3, and 4.0 HOTA, on MOT17 val, PP22 test, and SOMPT22 train, respectively, compared to the base framework.

These remarkable results are achieved through a sophisticated utilization of available information, while only standard components for detection, appearance feature extraction, and motion modeling are adopted from the MPT literature. This shows the high importance of striving for an optimal interplay of the single tracking modules and ensuring that the available information is leveraged as effectively as possible.

6.4 Camera Motion Compensation

The displacement of image content as a consequence of camera motion can also be regarded as available information that has not been treated so far in this thesis. However, the example in Figure 5.8 has shown that tracking errors can be introduced due to bad predictions of the motion model if present camera motion is not considered. Note that camera motion not only includes spatial translations of a mobile camera carried by an object (car, drone, person, etc.) but can also arise in fixed mounting positions from PTZ cameras to monitor large areas. Thus, building a tracking system that takes camera motion into account is of high importance for many applications.

The majority of MPT approaches found in the literature that incorporate a CMC method [Aha22, Ber19, Du23, Han22, He21, Khu21] utilize the ECC maximization from [Eva08], which is too slow to be applied in real time (Table 7.8). Maximizing the ECC requires to solve a nonlinear optimization problem, which can approximately be solved by an iterative scheme. In each iteration, an $N_\mathrm{I} \times N_\mathrm{p}$ Jacobian matrix of the transformed image pixels w.r.t. the transformation parameters has to be calculated, with N_I denoting the number of image pixels and N_p being the number of parameters of the transformation. This results in a large computational complexity as multiple iterations have to be performed for convergence (up to 50 in the experiments on MOT17 val).

For more details about the ECC algorithm, the interested reader is referred to the original paper [Eva08].

To enable a real-time-capable CMC, a fast alternative strategy based on *keypoint* detection and the matching of local image *descriptors* is introduced for the use in MPT for the first time [Sta23c]. As the detection of keypoints has to be performed only once per image, for which efficient algorithms exist, the proposed method can run significantly faster than the ECC technique (Table 7.8). In this thesis, a thorough empirical evaluation of different algorithms for the detection of image keypoints and extraction of visual descriptors is performed in order to find the combination with the best accuracy–runtime trade-off. In other words, a configuration is searched that is capable of accurately compensating occurring camera motion, while not significantly increasing the computational complexity of the overall tracking system.

Before going into details of the evaluation, the basic functionality of the proposed CMC approach is explained. In each frame of the video sequence, a keypoint detector is applied that identifies distinctive points on the image, e.g., corners of objects. For each point $\mathbf{p} = (p_x, p_y)^\top \in \mathcal{P}$ from the set of keypoints $\mathcal{P}$, the subsequent descriptor extraction algorithm computes a vector $\mathbf{f}_{\mathrm{KP}}$ that stores image information of the keypoint location. This vector can have binary entries $\{0, 1\}$ or contain floating-point values depending on the specific descriptor extractor used. Consequently, different distance functions to compare the computed descriptor vectors have to be applied. Typical choices are the Hamming distance for binary vectors and the Euclidean distance for real-valued vectors.

After keypoint detection and descriptor extraction, the set of descriptors $\mathcal{F}_t$ from the current frame $\mathbf{I}_t$ is matched with the descriptors $\mathcal{F}_{t-1}$ from the previous frame $\mathbf{I}_{t-1}$ based on an appropriate distance function. The matching is performed in a brute force manner meaning that all descriptors from the first image are compared with all descriptors from the second image. Then, the N matched keypoints $(\mathbf{p}_{t-1}^1, \mathbf{p}_t^1), \dots, (\mathbf{p}_{t-1}^N, \mathbf{p}_t^N)$ with smallest descriptor distances are leveraged to estimate a transformation matrix $\mathbf{W}$. Finally, the transformation matrix can be used to align the two consecutive images, thus to compensate potential camera motion. Notice that the random sample consensus (RANSAC) algorithm [Fis81] is applied to increase the robustness of the transformation estimation w.r.t. outliers, i.e., incorrectly matched keypoints. The concept of the proposed CMC approach is illustrated in Figure 6.14.

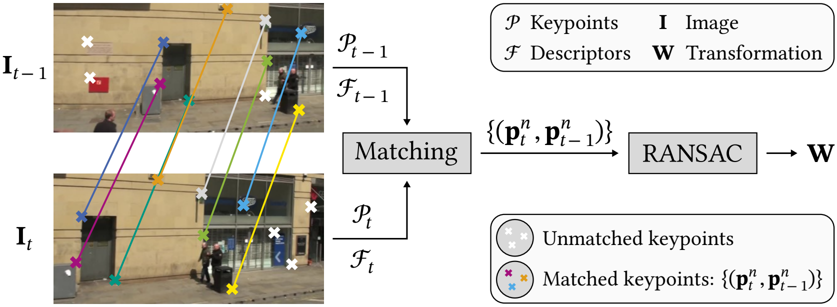


Figure 6.14: Overview of the proposed CMC approach. Keypoints are detected on two consecutive frames, for which visual descriptors are extracted. Based on the similarity of these descriptors, the keypoints are matched. Afterwards, the matched keypoints are used to compute a transformation matrix for image alignment, whereby the RANSAC algorithm is leveraged for an increased robustness.

To evaluate the performance of the CMC in the context of MPT, the proposed tracking framework from the previous section (BYTEv2+OAI+$d_{\text{DIoU,comb}}$) is leveraged. Experiments using different keypoint detectors and descriptor extractors are conducted on MOT17 val. While different types of transformations (translation, affine transformation, homography, etc.) can be estimated from the matched keypoints to compute the matrix $\mathbf{W}$, assuming a partially affine transformation with four degrees of freedom (scale q, rotation θ, translation in x-direction t_x, translation in y-direction t_y)

$$\mathbf{W} = \begin{pmatrix} \cos(\theta) \cdot q & -\sin(\theta) \cdot q & t_x \\ \sin(\theta) \cdot q & \cos(\theta) \cdot q & t_y \end{pmatrix} \tag{6.29}$$

has lead to the best results. Applying the CMC in the tracking system means that the tracks from the previous iteration are aligned with the current frame by the matrix $\mathbf{W}$ before performing the motion prediction step of the Kalman filter. Formally, let $(x, y)^\mathsf{T}$ be the top-left or bottom-right coordinates of a track's bounding box, the transformed coordinates $(x', y')^{\mathsf{T}}$ are calculated, in the case of $\mathbf{W}$ representing a (partially) affine transformation, as follows:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{W} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) \cdot q \cdot x - \sin(\theta) \cdot q \cdot y + t_x \\ \sin(\theta) \cdot q \cdot x + \cos(\theta) \cdot q \cdot y + t_y \end{pmatrix}. \tag{6.30}$$

To conduct experiments with different keypoint detectors and descriptor extractors, the OpenCV library [Bra00] is utilized. The following 12 methods have been tested for keypoint detection: A-KAZE [Alc13], AGAST [Mai10], FAST [Ros06], GFTT [Shi94], Harris-Laplace [Mik04], KAZE [Alc12], MSER [Nis08], MSD [Tom15], ORB [Rub11], SIFT [Low04], SimpleBlobDetector [Bra00], StarDetector [Agr08]. To extract descriptors, the following 12 algorithms have been used: A-KAZE [Alc13], BRIEF [Cal10], BRISK [Leu11], BoostDesc [Trz13], DAISY [Tol10], FREAK [Ala12], KAZE [Alc12], LATCH [Lev16], LUCID [Zie12], ORB [Rub11], SIFT [Low04], VGG [Sim14]. Note that some methods can be applied both for keypoint detection and descriptor extraction (A-KAZE, KAZE, ORB, SIFT) and that not all detector–extractor combinations are possible. In total, experiments with 120 combinations have been conducted, whereby the standard parameter settings of the detectors and extractors are adopted from the OpenCV library.

The highest HOTA is achieved by 12 combinations. ORB+BRIEF, as the fastest one, is compared to the baseline without CMC on MOT17 val in Table 6.17.

Table 6.17: Effect of the ORB+BRIEF CMC on the tracking performance on MOT17 val. An increase in HOTA is mainly caused by an improved AssA.

CMC	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
–	71.0	67.5	75.1	72.9	**83.1**	79.2	**87.2**
ORB+BRIEF	**71.4**	**67.6**	**75.9**	**73.3**	82.8	**80.0**	86.9

Compensating camera motion yields a plus of 0.4 HOTA due to an increased DetRe and AssRe. However, note that not all sequences of the MOT17 dataset contain camera motion. Whereas the performance on scenes with static cameras is similar, large improvements up to 2.5 HOTA are observed on sequences with severe camera motion (MOT17-13). The static scenes are not excluded from the evaluation because estimating an (approximate) identity matrix for consecutive frames without camera motion is as important as precisely estimating the transformation matrix under severe camera motion. If the CMC method estimates a large motion, where none is present, many tracking errors can be introduced. To investigate if such unwanted errors are made by the presented CMC, experiments are performed on PP22 test and SOMPT22 train that hardly contain any camera motion. The results are given in Table 6.18.

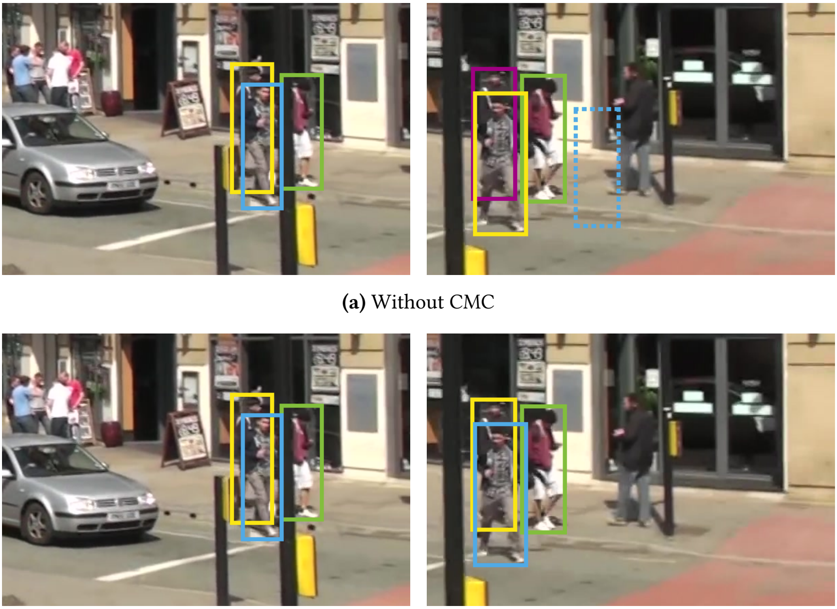
Table 6.18: Proposed CMC on two different datasets without notable camera motion. The similar results show that the CMC introduces hardly any tracking errors in static scenes.

	PP22 test			SOMPT22 train		
CMC	HOTA	DetA	AssA	HOTA	DetA	AssA
–	**67.4**	**68.5**	**66.8**	**59.4**	**57.8**	**61.3**
ORB+BRIEF	**67.4**	**68.5**	**66.8**	59.3	**57.8**	61.1

The same tracking measures are obtained with and without CMC on PP22, while a small decrease of 0.1 HOTA is observed when using the CMC module on SOMPT22. This indicates that not many errors are introduced by the CMC method on static scenes. The advantage of being capable to compensate severe camera motion clearly outweighs a potentially small decrease in tracking performance on static scenes. As a future work, one could use additional information like the frame rate and motion limits of the camera to exclude infeasible transformation matrices originating from wrong keypoint matches. This might be necessary in scenes with very low contrast, where the chance for incorrect matches is increased. Moreover, if too many keypoints are found on moving persons in very crowded scenes, this might also impede a correct estimation of the camera motion. To counter this, keypoints on detected person regions could be excluded from the keypoint matching. However, such a strategy has not been necessary on the evaluation datasets, although they contain scenes with a high number of persons.

Not all of the 12 detector–extractor combinations that have achieved the highest HOTA on MOT17 val can be applied in real time, either since a too slow detector (Harris-Laplace) or extractor (VGG) is involved. The best results according to runtime–accuracy trade-off in the experiments are obtained using the ORB detector. ORB+BoostDesc, ORB+BRIEF, ORB+BRISK, and ORB+LATCH achieve the best tracking performance, while taking BRIEF as descriptor extractor has the lowest runtime. Therefore, ORB and BRIEF are used as keypoint detector and descriptor extractor, respectively, in the CMC module of the final tracking system of this thesis. A detailed runtime analysis of the CMC as well as the full tracking framework is given in Chapter 7.

Figure 6.15 shows a qualitative example from the MOT17 dataset, where the usage of ORB+BRIEF improves the tracking performance of the proposed framework BYTEv2+OAI+$d_{\text{comb,DIoU}}$ under severe camera motion.



(a) Without CMC

(b) With CMC

Figure 6.15: Qualitative comparison of the proposed framework with (a) and without (b) CMC on an example sequence with severe camera motion from the MOT17 dataset. The CMC prevents wrong associations caused by inaccurate motion predictions.

Without CMC, the predicted track positions do not match with the actual target positions, which leads to a wrong association (yellow), an incorrect track start (purple), and a ghost track (blue) as can be seen in Figure 6.15a. In contrast, applying the proposed ORB+BRIEF method, all targets are tracked correctly, which is depicted in Figure 6.15b.

Finally, a comparison of the proposed CMC with the prevailing ECC method is performed. Results of these two approaches applied together with the BYTEv2+OAI+$d_{\text{comb,DIoU}}$ framework on MOT17 val are listed in Table 6.19, and results without CMC are also given.

Table 6.19: Comparison of CMC methods on MOT17 val. The proposed method ORB+BRIEF achieves similar results as the ECC baseline, while being able to run in real time.

CMC	Real-time capability	HOTA	DetA	AssA
–	–	71.0	67.5	75.1
ECC	✗	**71.4**	67.5	**75.9**
ORB+BRIEF	✓	**71.4**	**67.6**	**75.9**

Note that the parameters of the ECC method (type of transformation, number of iterations, etc.) have been tuned carefully for a fair comparison. While the ECC can achieve comparable results to the proposed CMC method, it cannot be applied in real time, which will be shown in Section 7.4.

With its high accuracy and computational efficiency, the presented CMC method makes a good extension to the overall tracking system of this thesis, which is compared with the SOTA in the following section.

6.5 Comparison with the State of the Art

The datasets MOT17 [Mil16] and MOT20 [Den20] have been used for many years to evaluate the accuracy of MPT approaches and are still the standard benchmarks. Consequently, those two are leveraged to compare the proposed tracking system with the SOTA.

For MOT17 and MOT20, two evaluation protocols exist—using *public* and *private* detections. The public detection sets are provided with the datasets while the private protocol allows to use an arbitrary object detector. The idea behind the public protocol is to enable a fair comparison between tracking methods. However, two major drawbacks speak against the use of the public detections: First, they are outdated as they are generated with detection methods that are now up to 20 years old [Fel04, Ren17, Yan16]. Second, trackers that incorporate additional tasks into the detection, for instance, JDE methods [Lu20, Ren24, Wan20, Wan23b, Zha21], or transformer-based approaches that solve detection and tracking end-to-end within one network [Mei22, Sun21a, Zen22, Zhu23] cannot be directly evaluated with the public detection protocol. Moreover, the public detections are provided after a standard NMS has been performed. As a consequence, alternative filter approaches like the introduced adapted NMS cannot be applied, which also prevents the use of the proposed BYTEv2 association strategy.

While the private protocol—which is used for evaluation in this thesis—allows to utilize any detector, YOLOX-X [Ge21] with weights from ByteTrack [Zha22c] has become the default detection model for evaluation on the MOT benchmarks in recent years. This improves the comparability of various trackers under the private protocol. Another established standard is to interpolate fragmented tracks in order to recover missed detections under occlusion. To analyze the impact of this post-processing on the evaluation measures, different types of interpolation are performed on the generated tracks of the proposed tracking framework on MOT17 val. The results are listed in Table 6.20.

Table 6.20: Comparison of interpolation approaches on MOT17 val. Higher DetRe and AssRe lead to a significant increase in HOTA when performing interpolation. Both the minimum length requirement and the Gaussian smoothed interpolation (GSI) lead to improvements compared to the standard linear interpolation (LI).

Interpolation	Minimum length	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
✗	✗	71.4	67.6	75.9	73.3	**82.8**	80.0	**86.9**
LI	✗	73.0	69.5	77.2	**77.1**	80.5	82.2	85.9
LI	✓	73.2	69.7	77.3	76.8	81.1	82.3	86.0
GSI	✓	**73.5**	**69.9**	**77.8**	77.0	81.3	**82.7**	86.3

A simple linear interpolation (LI) leads to a large plus of 1.6 HOTA and boosts both DetA and AssA due to the increase of DetRe and AssRe, respectively. Interpolating only fragmented tracks with a minimum overall length of one second yields another slight enhancement, just like Gaussian smoothing of the interpolated trajectories (GSI) as introduced in [Du23]. An overall improvement of the HOTA measure by 2.1 points shows the importance of interpolation to achieve competitive results on the MOT benchmarks.

Besides interpolation, some parameters of the tracking framework have a large influence on the final performance. In particular, the confidence threshold for track initialization s_{init} has a high impact. This is why a lot of SOTA methods apply various thresholds for different videos of the MOT17 test set [Aha22, Cao23, Liu23, Men23, Zha22c] to take specific characteristics of the scenes into account. For example, if a video contains many low-resolution persons due to large camera distances, the detection confidences are probably lower than for scenes showing only high-resolution persons. Therefore, setting a smaller initialization threshold for such sequences can lead to a higher recall and thus to a better overall tracking performance. The applied values of the proposed tracking framework for s_{init} on the MOT17 test sequences are given for reference in Table 6.21.

Table 6.21: Applied initialization thresholds s_{init} on the MOT17 test sequences.

MOT17-01	MOT17-03	MOT17-06	MOT17-07	MOT17-08	MOT17-12	MOT17-14
0.75	0.8	0.75	0.6	0.65	0.8	0.4

All other tracking parameters are taken over from the evaluation on MOT17 val to show the generalization ability of the proposed tracking framework. Table 6.22 gives a comparison with the 20 best-performing trackers on the MOT17 test set, separated by offline (top) and online (bottom) approaches. In addition to the performance measures, it is indicated for each tracker whether specific modules are used (REID, CMC) and if a different detection model as in the proposed framework is applied.

Table 6.22: Comparison of the best-performing trackers on MOT17 test. Highest values are bold and second highest underlined. REID and CMC specify whether the tracker uses appearance information and a module to compensate camera motion, respectively. Offline methods are listed at the top and online methods at the bottom. One tracker that does not use the YOLOX detector with weights from [Zha22c] is indicated by *.

Method	REID	CMC	HOTA	DetA	AssA	MOTA	IDF1
RTU++ [Wan22a]	✓	✗	63.9	64.5	63.7	79.5	79.1
BASE [Lar24]	✗	✗	64.5	66.3	63.1	81.9	78.6
SUSHI [Cet23]	✓	✗	66.5	65.5	**67.8**	81.1	**83.1**
OC-SORT [Cao23]	✗	✗	63.2	63.2	63.4	78.0	77.5
FOR_Tracking [Nas23]	✗	✓	63.3	64.7	62.2	80.4	77.7
QDTrack [Fis23]	✓	✗	63.5	64.5	62.6	78.7	77.5
Unfctrack [Hua23]	✓	✗	63.5	64.4	62.9	79.8	77.9
BPMTrack [Gao24]	✓	✓	63.6	65.5	62.0	81.3	78.1
UTM [You23] *	✓	✗	64.0	65.9	62.5	81.8	78.7
FineTrack [Ren23]	✓	✗	64.3	64.5	64.5	80.0	79.5
StrongSORT [Du23]	✓	✓	64.4	64.6	64.4	79.6	79.5
SAT [Wan22b]	✓	✗	64.4	64.8	64.4	80.0	79.8
Deep OC-SORT [Mag23]	✓	✓	64.9	64.1	65.9	79.4	80.6
BoT-SORT [Aha22]	✓	✓	65.0	64.9	65.5	80.5	80.2
MotionTrack [Qin23]	✗	✗	65.1	65.4	65.1	81.1	80.1
SparseTrack [Liu23]	✗	✓	65.1	65.3	65.1	81.0	80.1
LG-Track [Men23]	✓	✗	65.4	65.6	65.4	81.4	80.4
ConfTrack [Jun24]	✓	✓	65.4	64.8	66.3	80.0	81.2
UCMCTrack [Yi24]	✗	✓	65.7	65.3	66.4	80.6	81.0
C-BIoU [Yan23]	✗	✗	66.0	66.3	66.1	**82.8**	82.5
Proposed	✓	✓	**66.7**	**66.6**	67.1	82.5	82.7

As a first observation, most of the SOTA trackers are considered online methods that process frame after frame. Note that nearly all listed approaches apply interpolation as post-processing, which strictly speaking violates the online property. However, they are still considered online in this comparison (and by the MPT community) as only a small delay of the tracking results, for instance, $i_{\text{max}} = 1.5\,\text{s}$ for the proposed tracker, is introduced by the interpolation. In contrast, the actual offline methods process the whole sequence at once afterwards.

The second finding drawn from Table 6.22 is that most top-performing online methods either apply a CMC or follow an alternative approach to handle irregular motions (C-BIoU [Yan23]). So, a valid treatment of occurring camera motion is important to obtain a high tracking performance.

Third, some approaches do not use a REID model to achieve a low runtime [Liu23, Yan23, Yi24] at the cost of not utilizing the important appearance information. Unfortunately, a valid runtime comparison of the different trackers is currently impossible because some methods report their runtime excluding modules like detection or CMC [Cao23, Cet23, Gao24, Yan23, Yi24] or do not provide a runtime at all [Lar24, Mag23, Men23, Qin23, Ren23, Wan22b, You23, Zha24]. The two trackers StrongSORT [Du23] and BoT-SORT [Aha22] comprising both REID and CMC modules run only at a small rate of 7.5 (1.5) and 4.5 (2.4) FPS on MOT17 (MOT20), respectively, according to their papers. This indicates that no top-performing method that includes both a REID model and CMC can run in real time on videos with high frame rates on standard hardware. That also holds true for the proposed tracking framework. However, it will be shown in Chapter 7 that a runtime-optimized version can be built that achieves real-time capability while maintaining the high tracking accuracy.

Regarding accuracy, the proposed framework of this thesis achieves the overall best tracking performance among all methods on MOT17 test set, with a plus of 0.7 HOTA w.r.t the second-best online method C-BIoU [Yan23] and a plus of 0.2 HOTA compared to the best offline approach SUSHI [Cet23]. Obtaining the highest DetA (+0.3 w.r.t. second-best entry) indicates a superior utilization of available detections by the proposed adapted NMS in combination with the BYTEv2 association strategy. Moreover, the best AssA among all online methods (+1.0 w.r.t second-best entry) is attributable to the introduced combined association distance, which has been shown to be superior to other approaches of fusing motion and appearance information from the literature. Only the offline method SUSHI [Cet23] achieves a higher AssA (+0.7) due to its better long-term association capabilities (+0.4 IDF1), since all detections of the whole sequence are available at once.

While all except one tracker leverage the identical YOLOX model with the weights provided by [Zha22c]—which in the first place enables a fair comparison—one has to note that different detection confidence thresholds are applied, which are not reported by many methods. Next to the missing or incomparable runtimes, this can be seen as another limitation of the comparison on the MOT17 test set.

The lack of information on used confidence thresholds is also a problem on MOT20. Again, the applied thresholds for track initialization s_{init} of this thesis on the MOT20 test sequences are provided for reference in Table 6.23.

Table 6.23: Applied initialization thresholds s_{init} on the MOT20 test sequences.

MOT20-04	MOT20-06	MOT20-07	MOT20-08
0.65	0.4	0.65	0.4

Besides tuning s_{init} for each sequence independently, it has become a common practice to utilize the original input resolution of sequences on the MOT20 dataset for detection [Aha22, Cao23, Liu23, Men23, Zha22c]. So, instead of keeping the default input size of 1440×800 pixels for the YOLOX detector as on MOT17, the original sizes of the sequences are used (Table 4.1). Another difference to the evaluation on MOT17 test is that the CMC module is turned off since there is hardly any camera motion on MOT20.

The comparison of the proposed framework with the best-performing tracking methods on the MOT20 test set is given in Table 6.24. The proposed tracker obtains the overall best performance, surpassing the second best entry ConfTrack [Jun24] by 0.7 HOTA. Furthermore, the highest DetA (+1.1 w.r.t. second-best entry) and MOTA (+0.4 w.r.t. second-best entry) are achieved.

Again, the reason behind the best DetA lies in the usage of heavily-occluded detections in the BYTEv2 association. The consideration of such detections is especially important for very crowded scenes as in the MOT20 benchmark (on average 141 persons per image, Section 4.1.1). Next to

Table 6.24: Comparison of the best trackers on MOT20 test. Highest values are bold and second highest underlined. REID specifies whether the tracker uses appearance information. Offline methods are listed at the top and online methods at the bottom. Trackers that do not use the YOLOX detector with weights from [Zha22c] are indicated by *.

Method	REID	HOTA	DetA	AssA	MOTA	IDF1
RTU++ [Wan22a]	✓	62.8	63.1	62.6	76.5	76.8
BASE [Lar24]	✗	63.5	64.1	63.2	78.2	77.6
SUSHI [Cet23]	✓	64.3	61.5	**67.5**	74.3	79.8
ByteTrack [Zha22c]	✗	61.3	63.4	59.6	77.8	75.2
FOR_Tracking [Nas23]	✗	61.4	62.3	66.2	76.8	76.4
QuoVadis [Den22]	✓	61.5	63.3	59.9	77.8	75.7
SuppTrack [Zha24] *	✗	61.9	63.8	60.1	78.2	75.5
BPMTrack [Gao24]	✓	62.3	63.9	60.9	78.3	76.7
OC-SORT [Cao23]	✗	62.4	62.4	62.5	75.7	76.3
UTM [You23] *	✓	62.5	63.7	61.4	78.2	76.9
SAT [Wan22b]	✓	62.6	62.1	63.2	75.0	76.6
StrongSORT [Du23]	✓	62.6	61.3	64.0	73.8	77.0
MotionTrack [Qin23]	✗	62.8	64.0	61.8	78.0	76.5
UCMCTrack [Yi24]	✗	62.8	62.4	63.5	75.6	77.4
BoT-SORT [Aha22]	✓	63.3	64.0	62.9	77.8	77.5
LG-Track [Men23]	✓	63.5	64.1	62.9	77.8	77.4
SparseTrack [Liu23]	✗	63.5	64.1	63.1	78.1	77.6
FineTrack [Ren23]	✓	63.6	63.6	63.8	77.9	79.0
Deep OC-SORT [Mag23]	✓	63.9	62.4	65.7	75.6	79.2
ConfTrack [Jun24]	✓	64.8	63.6	66.2	77.2	**80.2**
Proposed	✓	**65.5**	**65.2**	66.0	**78.7**	79.6

SuppTrack [Zha24], which improves the detection recall in crowds by incorporating a detection branch that focuses on the heads of persons, and BPMTrack [Gao24], which also makes use of the introduced adapted NMS, the proposed tracking framework is the only method in Table 6.24 that explicitly accounts for improving the detection recall under severe occlusion. Since most tracking errors occur in such circumstances, further enhancing the detection performance in crowded scenes, e.g., by applying *crowd-specific* person detectors as in a previous work of this thesis' author [Sta21d], is a promising direction for future research.

ConfTrack [Jun24] is the only online approach that achieves a higher AssA (+0.2) than the proposed tracker on MOT20 test. One of the main contributions of ConfTrack is a further development of the NSA Kalman filter, introducing a confidence-weighted Kalman update and using a constant box prediction inspired by the proposed HP module of this thesis. Moreover, the detection confidence is incorporated into the association distance and a new treatment of low-confidence tracks in the association is proposed. All of these approaches are other ways of *utilizing the available information* in the tracking process which underlines the high importance of this thesis' topic.

To conclude, the proposed tracking framework sets a new SOTA on the standard MPT benchmarks MOT17 and MOT20. DetA is notably enhanced with the integration of heavily-occluded detections through the BYTEv2 association strategy. Considering both benchmarks, also the best AssA is achieved, which is attributable to an improved fusion of target information and an effective CMC technique.

6.6 Summary

With an improved use of available information—generated detections, extracted appearance and motion information of targets as well as derived context knowledge—the proposed framework has significantly enhanced the MPT performance, especially under occlusion.

First, an adapted NMS was introduced, which enables the incorporation of severely-occluded detections into the association that are typically discarded. Based on this adapted NMS, two novel association strategies, BYTEv2 and TWC, were proposed. Both approaches are designed to leverage the additional TPs from the set of occluded detections without introducing FPs in the tracking process. This does not only increase detection recall but also simplifies the assignment task in crowded scenes leading to an overall higher tracking accuracy. The two proposed association strategies are among only a few approaches found in the MPT literature that leverage detections under such heavy occlusion in the tracking process [Gao24, Zha24].

Next to the better usage of available detections in the association, the track information is consulted to derive knowledge of the surroundings from unassigned detections in the proposed OAI. The method prevents the start of ghost tracks in crowded areas and thus further enhances the tracking performance when combined with the aforementioned association strategies.

Besides a sophisticated use of detections and tracks, an effective utilization of the extracted appearance and motion information from the targets is also important. To identify shortcomings of existing association measures, a detailed analysis of fusion strategies of motion and appearance distances from the literature was conducted for the first time. Based on that, combined distance measures have been proposed that clearly outperform previous fusion approaches on three MPT datasets.

A combination of BYTEv2, OAI, and the combined distance measure has shown that the various approaches work well together and yield complementary gains. W.r.t the base framework, the overall tracking accuracy measured in HOTA was increased by 3.4, 5.3, and 4.0 points on the MOT17 val, PP22 test, and SOMPT22 train dataset, respectively.

Moreover, an efficient CMC method based on keypoint detection and descriptor extraction, which makes an important addition to the tracking framework whenever moving cameras are involved, was introduced. An extensive amount of keypoint detectors and descriptor extractors was compared to build a CMC method with both high accuracy and real-time capability, which improves over the prevailing CMC approach from the MPT literature.

Putting all together, the proposed framework achieves the best tracking performance on the standard MPT benchmarks MOT17 and MOT20. The high flexibility of the framework allows to integrate modules of other trackers from the literature to further improve its performance, which is outlined in Section 8.2. Some qualitative tracking results are visualized in Figures 6.16 and 6.17 to demonstrate the SOTA performance of the proposed tracker under different scenarios.

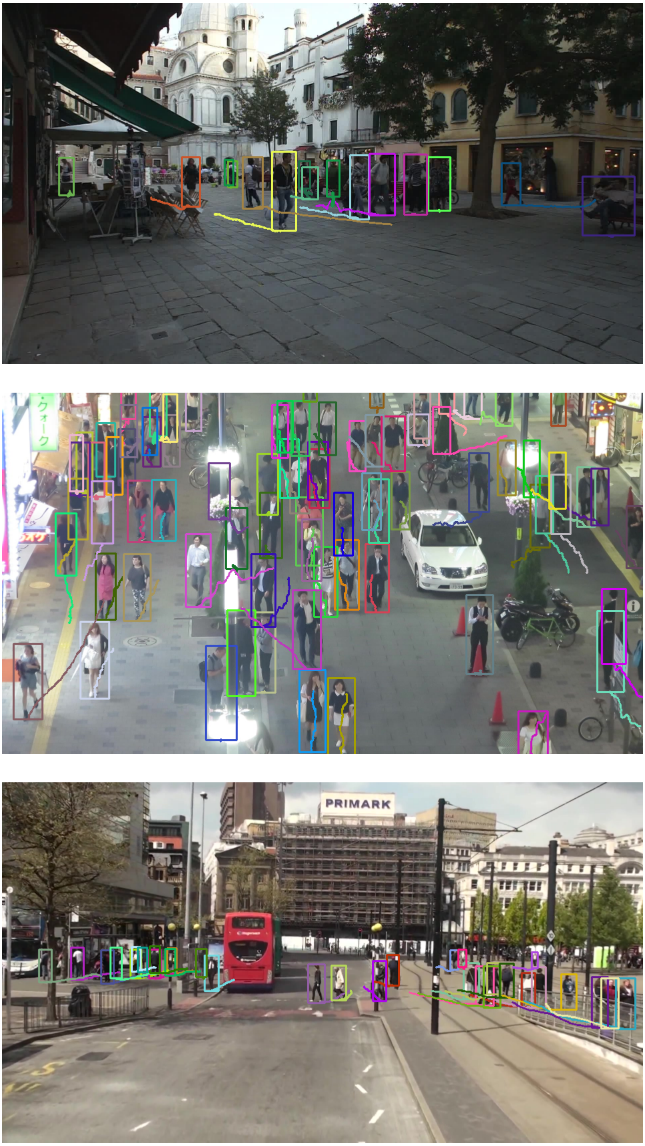


Figure 6.16: Qualitative results of the proposed tracking framework on example sequences of MOT17 test. The trajectories display the targets' positions of the last three seconds.

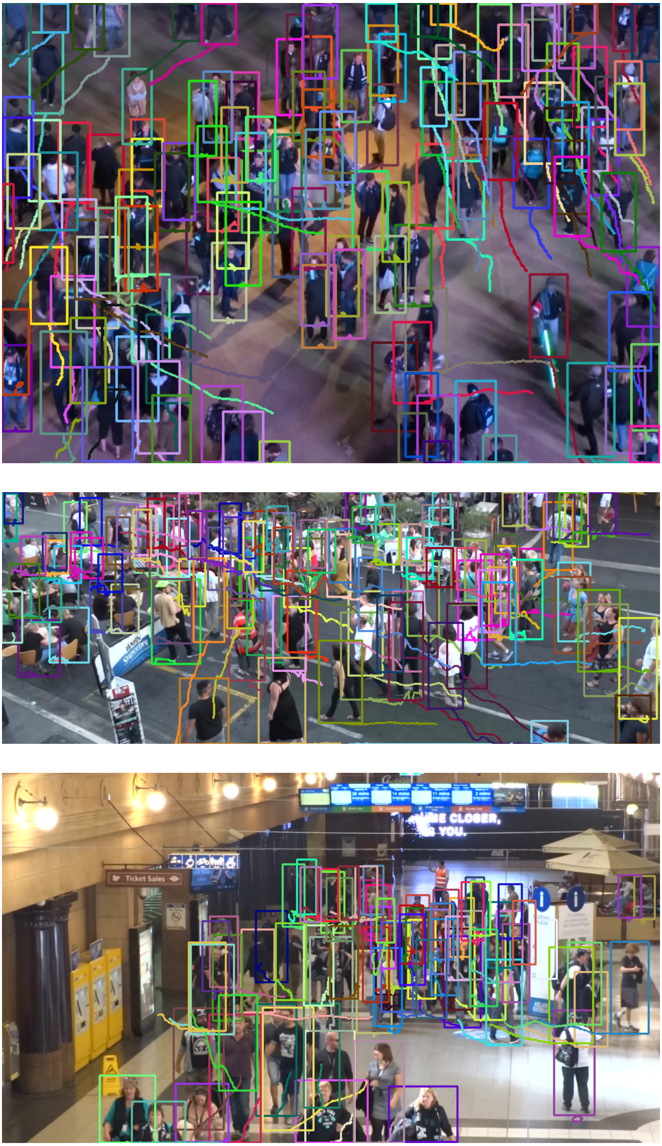


Figure 6.17: Qualitative results of the proposed tracking framework on example sequences of MOT20 test. The trajectories display the targets' positions of the last three seconds.

Looking at the displayed trajectories that show the targets' positions of the last three seconds, one observes that the persons are tracked robustly through frequent occlusions under various camera views and target densities. Also in very crowded scenes from the MOT20 benchmark (Figure 6.17), accurate tracking results are obtained with the proposed framework.

So far, the runtime of the tracking algorithms has not been considered. However, like most of the SOTA trackers that involve multiple modules as detection, REID, and CMC model, the proposed framework is not real-time capable without modifications. For this reason, a runtime-optimized system, which achieves real-time capability without sacrificing the SOTA performance, is developed in the next chapter.

7 Runtime Optimization

For many MPT applications, especially in the security domain, a *real-time* processing is required. While there is no universal understanding of what real-time processing means in the MPT community, this thesis adopts the definition of [Kuo06, Mur17] that states real-time capability of a tracking system as the ability to process one frame in less time than is given between two consecutive frames of a video stream. Thus, the real-time capability of a tracking algorithm depends on the frame rate of the camera, which, for example, is on average about 15 FPS for industry video surveillance cameras as of 2021 [IPV21]. Taking these 15 FPS as reference, the time limit for the overall tracking system to process a frame is 66.7 ms—including detection, motion modeling, feature extraction, association, and track management. This makes clear that building a real-time-capable MPT system with common hardware is a non-trivial task.

In particular, the two optional tracking components REID and CMC are computationally expensive, which is why tracking systems from the literature either do not use these models (Section 6.5) or do not achieve real-time capability. On the contrary, this chapter shows that the proposed tracking system runs in real time without significantly sacrificing the overall performance, when modifications are made, namely, using a more efficient REID model, executing some modules in parallel, and utilizing a high-performance library for neural network inference like NVIDIA TensorRT[1].

The remainder of this chapter is organized as follows. Section 7.1 gives a thorough runtime analysis of the proposed tracking system. Then, the usage of a more efficient REID model is investigated in Section 7.2, since the so-far used

[1] https://developer.nvidia.com/tensorrt (accessed on July 16, 2024)

model prevents a real-time capability of the overall tracking system. In Section 7.3, the inference library TensorRT is utilized to speed up the detection and REID model. Then, the runtime of the proposed CMC module is compared with the prevailing approach from the literature, and its influence on the total runtime is studied in Section 7.4. Finally, the applied changes w.r.t. the proposed framework from the previous chapter are summarized, and the runtime of the optimized system is examined in Section 7.5.

7.1 Runtime Analysis of the Proposed System

To obtain a detailed understanding about the computational complexity of the proposed tracking system under different settings, i.e., number of targets or input resolution, and to identify bottlenecks in the framework, the runtime of each tracking component is analyzed independently. The runtime of a component is measured for 10,000 iterations after performing 100 warmup iterations that do not contribute to the calculation of mean and standard deviation of the runtime. This warmup is especially important when deep neural networks as the detector or REID model are executed on the GPU, since some deferred initializations and optimizations are happening in the first few iterations.

The code of the tracking system is written in Python and executed on an Intel Xeon E5-2698 v4 CPU. Some of the code leverages multi-threading with a maximum number of 16 threads. An NVIDIA Tesla V100-SXM2-32GB is used as GPU. To inference the detection and REID model on the GPU, the PyTorch[1] library is utilized.

In the following subsections, the runtime of detection, REID, and motion model as well as the parts of the association are analyzed. Note that the runtimes of the OAI and the parts of the track management, e.g., initialization of new tracks, changes of the track state, etc., are negligible w.r.t. the other modules, so no analysis of these components is carried out. Moreover, the CMC as optional additional module is treated separately in Section 7.4. Lastly, notice

[1] https://pytorch.org (accessed on July 16, 2024)

that operations that would not be performed in a real application environment, like loading images from the hard disk or saving the tracking results, are excluded from the runtime measurements.

7.1.1 Detection

The detection task consists of two parts: inference of the detection model (YOLOX-X [Ge21]) given an image as input and a subsequent NMS for removing duplicate detections. The runtime of the detector is highly dependent on the resolution of the input image as depicted in Figure 7.1. Note that the input sizes are chosen such that width and height are divisible by 32, which is a prerequisite of the YOLOX-X detector.

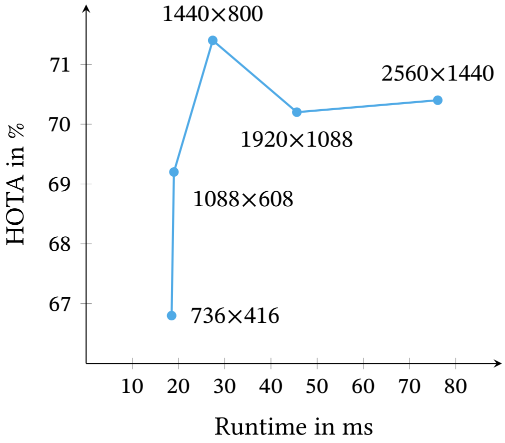


Figure 7.1: Runtime of the YOLOX-X detector for various image sizes and obtained HOTA values on MOT17 val. The best result of 71.4 HOTA is achieved with a resolution of 1440×800 pixels, which leads to a runtime of 27.4 ± 0.3 ms. Notice that the vertical axis does not start at zero.

One observes that the best results are achieved with the standard input resolution of 1440 × 800 pixels, on which the model has been trained. If the model would be trained on a higher input resolution, increased HOTA values might be possible. However, the goal is not to get the most powerful detection model but one with a good accuracy–runtime trade-off that is suitable for being used in a real-time-capable MPT system. Considering that further acceleration can be achieved with an optimized inference library (Section 7.3), the runtime of 27.4 ms for the input resolution of 1440×800 pixels is affordable. Consequently, this input size is used for the YOLOX-X detector in the experiments of the following sections.

Table 7.1 compares the standard NMS with the proposed adapted NMS, which outputs the set of occluded detections next to the normal detection set.

Table 7.1: Runtime in ms of the standard NMS and the proposed adapted NMS. A small overhead of roughly 1 ms is introduced to the tracking framework by the adapted NMS. The runtime of both NMS variants hardly depends on the image size.

Image size	736×416	1088×608	1440×800	1920×1088	2560×1440
Standard NMS	**1.28**±0.10	**1.23**±0.21	**1.19**±0.16	**1.15**±0.10	**1.21**±0.16
Adapted NMS	2.08±0.18	2.33±0.42	2.22±0.37	2.03±0.16	2.13±0.26

Although the number of predictions made by the model is about 12 times higher for the largest input size w.r.t. the smallest one (75,600 for 2560×1440 vs. 6,279 for 736×416), there is no significant runtime difference for various image resolutions. This is mainly due to the fact that the vast majority of predictions has a very low confidence ($s < 0.01$) and thus is removed before the actual NMS algorithm is performed. Moreover, due to strong parallelization on the GPU, no significant runtime differences have been observed for scenes with various person numbers. As the adapted NMS basically involves executing the standard NMS two times, its mean processing time is about twice as large. In other words, the adapted NMS comes with an overhead of about one millisecond. Enabling the usage of occluded detections in the association and thus notably improving the overall tracking performance, this small overhead is worthwhile.

7.1.2 Re-Identification Model

Whereas the runtime of the detection is to a large extent independent from the number of targets on the image, the runtime of the REID model increases with the number of detected persons. For each person detection considered in the association, the REID model extracts features that are used to calculate the appearance distances. As the input size of the network is fixed and the detected image crops are resized accordingly, the runtime of the REID model is independent from the image resolution. For the so-far used REID model (SBS S50 [Luo19]), the runtime measurements are shown in Figure 7.2.

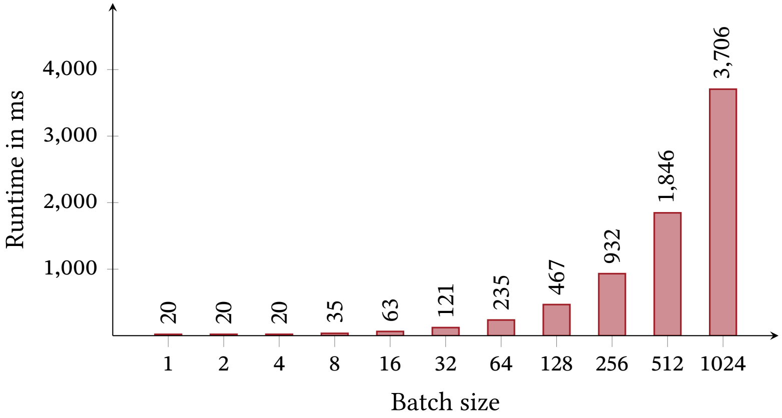


Figure 7.2: Runtime of the SBS S50 REID model with various batch sizes. The results clearly show that this REID model is not able to run in real time when a large number of targets has to be tracked. Note the logarithmic scale of the horizontal axis.

The batch size corresponds to the maximum number of person detections for that appearance features can be extracted in one inference pass of the REID model. The more person detections are processed, the longer the runtime of the REID model. Given the average frame rate of 15 FPS for industry video surveillance cameras as reference, the applied REID model can only run in real time if no more than about 16 persons appear on the image. However, dozens or even hundreds of persons need to be tracked in real-world applications, for instance, when monitoring large public spaces. If 256 persons are present, a mean runtime of 932 ms is needed which roughly corresponds only to 1 FPS. To conclude, the so-far applied REID model prevents the real-time capability of the whole system when a large number of targets has to be tracked. Therefore, a more efficient alternative will be introduced in Section 7.2.

7.1.3 Motion Model

Just like the REID model, the runtime of the applied Kalman filter as motion model is independent from the input size of the image but increases with the number of targets to be tracked. This holds true both for the prediction step as well as the update step. The individual runtimes are listed together with the total runtime in dependence of the batch size (number of targets) in Table 7.2.

Table 7.2: Runtime in ms of the Kalman filter prediction and update step with various batch sizes. The implementation runs fully on the CPU, so an acceleration for large batch sizes through parallelization on the GPU is conceivable.

Batch size	1	2	4	8	16	32
Prediction	0.09±0.01	0.10±0.01	0.12±0.03	0.13±0.02	0.17±0.04	0.24±0.05
Update	0.08±0.01	0.10±0.01	0.13±0.03	0.15±0.03	0.25±0.05	0.38±0.07
Total	0.17±0.01	0.20±0.03	0.24±0.05	0.28±0.05	0.42±0.09	0.62±0.11

Batch size	64	128	256	512	1024
Prediction	0.36±0.07	0.65±0.14	1.13±0.23	2.34±0.47	4.28±0.73
Update	0.72±0.11	1.33±0.23	2.28±0.40	4.76±0.78	9.85±1.09
Total	1.09±0.17	1.97±0.34	3.42±0.55	7.10±1.03	14.13±1.48

With a total runtime of about 14 ms including prediction and update step for 1024 targets, one can state that the Kalman filter is suitable to be applied in a real-time-capable MPT system. Notice that the implementation runs fully on the CPU. If the Kalman filter has to be applied on a large number of targets, extensive parallelization of a GPU implementation could lead to a further speed-up. This, however, is outside the scope of this thesis.

7.1.4 Association

The association task demands to calculate a distance between every track–detection pair, before the linear sum assignment (LSA) problem can be solved, i.e., matching each detection to (at most) one track and vice versa while minimizing the summed matching distances. Thus, the runtime of the association is both related to the number of targets to be tracked and the computational complexity of the applied distance measure d. The introduced combined distance $d_{\text{comb,DIoU}}$ of the proposed tracking system from Equation (6.20) requires to compute both the DIoU distance d_{DIoU} (Equation (6.15)) and the cosine appearance distance d_{app} (Equation (5.41)). Whereas the DIoU distance performs calculations on the fixed four-dimensional bounding boxes, the computational complexity of the cosine distance depends on the length l_{f} of the feature vector of detections and tracks. Runtime measurements have been performed with various batch sizes for the DIoU distance computation and the cosine appearance distance using three different feature dimensions. The results can be found in Table 7.3. Note that here and in the following, it is assumed that the number of detections is identical to the number of tracks (and to the batch size). While in practical cases, the equality generally does not hold, the numbers are typically close together, so the measured runtimes are meaningful approximations.

The runtime of the DIoU distance computation is basically constant among all evaluated batch sizes. This is because a GPU implementation from the *torchvision*[1] library, a part of the PyTorch project, is leveraged that is strongly parallelized. Computing the DIoU distance between two sets of 1024 bounding boxes, thus about one million times, lasts less than one millisecond.

The calculation time of the cosine appearance distance notably depends on the length of the feature vector l_{f}. Taking a target number of 256 as example, which is about the maximum number of persons in one image of the very crowded MOT20 dataset (Section 4.1.1), the mean runtime of the appearance distance for $l_{\text{f}} = 128$ is only 0.59 ms compared to 8.31 ms for $l_{\text{f}} = 2048$. Note

[1] https://pytorch.org/vision (accessed on July 16, 2024)

Table 7.3: Runtime in ms of the distance computation for d_{DIoU} and d_{app} with different feature vector lengths l_{f} and various batch sizes. Due to strong parallelization, the runtime for the d_{DIoU} computation is independent from the batch size. Regarding d_{app}, the runtime is significantly reduced when using smaller feature vectors.

Batch size	1	2	4	8	16	32
d_{DIoU}	0.83±0.18	0.83±0.09	0.83±0.07	0.88±0.07	0.88±0.08	0.85±0.13
d_{app}, $l_{\text{f}} = 128$	**0.02**±0.01	**0.02**±0.00	**0.03**±0.01	**0.02**±0.00	**0.04**±0.01	**0.08**±0.01
d_{app}, $l_{\text{f}} = 512$	**0.02**±0.00	0.03±0.01	0.06±0.01	0.06±0.01	0.09±0.01	0.21±0.02
d_{app}, $l_{\text{f}} = 2048$	0.04±0.01	0.07±0.02	0.09±0.01	0.16±0.03	0.39±0.03	0.64±0.07

Batch size	64	128	256	512	1024
d_{DIoU}	0.94±0.15	0.87±0.10	0.87±0.11	**0.83**±0.08	**0.88**±0.10
d_{app}, $l_{\text{f}} = 128$	**0.16**±0.01	**0.27**±0.02	**0.59**±0.04	1.70±0.12	5.62±0.39
d_{app}, $l_{\text{f}} = 512$	0.39±0.04	0.72±0.06	2.44±0.39	5.71±0.40	12.69±1.79
d_{app}, $l_{\text{f}} = 2048$	2.20±0.13	4.22±0.23	8.31±1.91	18.76±1.31	46.62±3.93

that the length of the extracted feature vectors l_{f} depends on the applied REID network. For the so-far used SBS S50 model, $l_{\text{f}} = 2048$ holds, which comes with a relatively high computational complexity in the distance calculation next to its high runtime (Figure 7.2). However, it will be shown in Section 7.2 that a more efficient REID network can achieve comparable results, while being faster and additionally generating more compact feature representations with $l_{\text{f}} = 128$. This saves a lot of runtime not only for extracting features in the network backbone but also in the distance computation of the association, especially when a large number of targets is present.

After the distance matrix is computed, the LSA problem has to be solved. No matter which algorithm is used, the runtime obviously increases with the size of the distance matrix and thus the number of targets. However, different algorithms or implementations can be favorable depending on the number of targets as will be seen shortly. The most common approach to solve the LSA problem in MPT is to use the Hungarian method [Kuh55] or one of its many variants. Especially the Jonker–Volgenant (JV) algorithm [Jon87] is often applied [Aha22, Ber19, Bew16, Wan20, Zha22c] as efficient variant of the Hungarian method. Moreover, different implementations from several libraries exist. For this thesis, runtime experiments with two implementations of the JV algorithm have been performed using the *SciPy*[1] and *LAPJV*[2] library, respectively. Table 7.4 lists the runtime of the two implementations for various batch sizes.

Table 7.4: Runtime in ms to solve the LSA problem with different implementations of the JV algorithm from the SciPy and LAPJV library for various batch sizes. Up to a batch size of 256, SciPy is faster, while LAPJV has a lower runtime for larger batches.

Batch size	1	2	4	8	16	32
SciPy	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.01**±0.00
LAPJV	0.02±0.01	0.02±0.01	0.02±0.00	0.02±0.01	0.03±0.01	0.03±0.01

Batch size	64	128	256	512	1024
SciPy	**0.12**±0.02	**0.47**±0.01	**2.57**±0.24	12.13±1.00	62.45±3.93
LAPJV	1.33±0.02	2.50±0.03	5.54±0.52	**6.41**±0.59	**25.14**±1.94

One observes that the runtime for both variants is negligible w.r.t. the full tracking pipeline up to a batch size of 32 . Interestingly, the SciPy implementation is notably faster than the one of the LAPJV library for a batch size of $B \leq 256$, whereas the contrary holds for $B \geq 512$. Thus, either the one or the other one may be used depending on the size of the distance matrix. This decision can be made in each iteration of the whole tracking pipeline during operation. For most practical cases, $B < 256$ holds, so the runtime of solving the LSA problem is below 3 ms using the SciPy implementation. If, in special cases, the number of targets to be tracked is in the region of 1,000 or even higher, a GPU implementation, e.g., from [Dat16], needs to be considered in order to achieve real-time capability of the overall tracking system.

[1] https://scipy.org (accessed on July 16, 2024)

[2] https://github.com/src-d/lapjv (accessed on July 16, 2024)

7.1.5 Summary

The total runtime of the proposed tracking system comprises the runtime for the detector (Figure 7.1), adapted NMS (Table 7.1), REID network (Figure 7.2), motion model (Table 7.2), computation of d_{DIoU} as well as d_{app} with $l_{\mathrm{f}} = 2048$ (Table 7.3), and solving the LSA problem (Table 7.4). For the sake of completeness, it is noted that the second stage of the BYTEv2 association requires to solve an additional LSA problem. However, the runtime for this step is negligible, as only the unassigned tracks, which make up a small number, are involved. The total runtime of the tracker is visualized for various batch sizes in Figure 7.3, whereby the runtime is given with and without the use of appearance information.

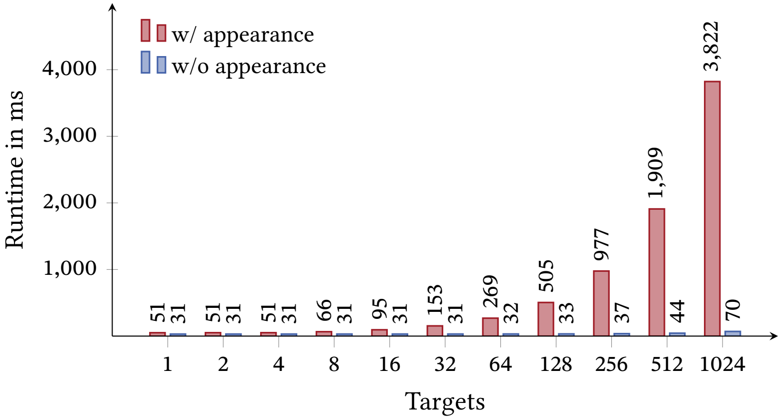


Figure 7.3: Runtime of the proposed tracking system with and without using appearance information for various numbers of targets. Utilizing appearance information drastically increases the overall runtime, especially when a large number of targets has to be tracked. Notice the logarithmic scale of the horizontal axis.

The plot clearly shows that leveraging appearance cues, which involves the application of the REID model and computing the cosine appearance distances between the extracted feature vectors, prevents the real-time capability of the whole tracking system. The reference rate of 15 FPS can only be achieved for a maximum number of 8 targets when the REID model is used. If one would omit the appearance information by not applying the REID model and using only motion information in the association, tracking 1,024 targets could be realized with approximately 14 FPS. However, ignoring the available appearance information would come with a significant decrease in tracking accuracy. Therefore, a more efficient REID model is introduced in the next section as an alternative approach to achieve real-time capability of the tracking system.

7.2 Efficient Re-Identification Model

The runtime analysis of the tracking system has shown that the computationally complex REID model is the component that prevents real-time capability. Before a more efficient REID model is introduced, another possible solution to the problem is briefly discussed.

Over the past years, JDE networks have been proposed in the literature [Lu20, Ren24, Wan20, Wan23b, Zha21], which combine the two tasks of object detection and appearance extraction within one model (Section 2.1.2). Instead of cropping detected image patches and feeding them into a separate REID model, a JDE network outputs next to each predicted detection box a feature vector that represents the appearance of the corresponding detection. While the JDE approach saves the additional computation time of a separate REID model, the quality of detection and appearance feature extraction generally falls behind a SDE approach, due to the competition between the two tasks in a single network. As already discussed in Section 2.1.2, some JDE works mitigate this issue by decoupling REID and detection features within the network architecture [Guo23, Jin23, Lia22, Yu23]. However, an analysis of the best-performing methods on the MOT17 (Table 6.22) and MOT20 (Table 6.24) benchmark reveals that the current SOTA in MPT is dominated by SDE approaches or methods with focus on motion modelling.

Therefore, the SDE paradigm is maintained and a more efficient REID model is introduced to replace the so-far used SBS S50 network, which prevents real-time capability. To be concrete, the extremely lightweight *Omni-Scale Network* (OSNet) [Zho19a] is employed. The main idea of OSNet is to learn so-called *omni-scale* features by combining features at different scales. Multi-scale features are generated from parallel convolutional paths in the network architecture and dynamically fused with input-dependent weights to omni-scale features. This allows to combine information from small local image regions (for instance, shoes or logos on shirts) with more global whole body regions, which is important to distinguish similar looking persons with only little appearance differences. The specific architecture design is tailored to the instance-level recognition task in REID, in contrast to many CNNs that are adopted for the REID task but are originally designed for category-level recognition tasks [Zho19a]. This is an advantage over the so-far used SBS model [Luo19], which uses a ResNeSt-50 [Zha22a] backbone that has been developed mainly for classifying objects of different categories and not distinguishing instances of the person class.

Another merit of OSNet is its low computational complexity that is achieved with the help of depthwise separable convolutions [Cho17, Sif14]. Moreover, the hyper-parameters *width* and *resolution multiplier* enable to tune the size of the model and the runtime–accuracy trade-off. For this thesis, the width multiplier (β in the OSNet paper) and resolution multiplier (γ in the OSNet paper) are set to 0.25 and 1.0, respectively. Next to the number of convolutional filters, the width multiplier determines the output size of the last FC layer that corresponds to the length l_f of the generated appearance feature vectors. The resulting feature dimension, number of network parameters, and input resolution are found in Table 7.5, which gives a comparison of some traits of the two employed REID models SBS and OSNet.

Table 7.5: Comparison of some traits of the SBS and OSNet REID model. Due to a smaller feature vector, a lower input size, and many less parameters, OSNet is notably faster than SBS, while achieving comparable results because of a REID-specific design.

Network	Backbone	REID-specific design	l_f	Input size	Parameters
SBS	ResNeSt-50	✗	2048	128×384	25.4 M
OSNet	$\text{OSNet}_{\beta=0.25}$	✓	128	128×256	0.2 M

The employed OSNet variant has only about 0.2 million network parameters, which is more than 100 times less than the SBS model. The smaller input size of 128×256 pixels does also contribute to the higher efficiency as well as the more compact representation of the generated feature vectors ($l_\mathrm{f} = 128$ vs. $l_\mathrm{f} = 2048$). The latter especially pays out in the computation of the cosine appearance distance as seen previously in Table 7.3.

To investigate whether the OSNet model can achieve comparable results within the proposed tracking system, it is trained on the REID dataset generated based on MOT17 as described in Section 5.6.2. For a fair comparison, the same training settings as for the SBS model have been applied. Then, the overall tracking performance measured in HOTA has been evaluated on MOT17 val, and runtime experiments with various batch sizes have been conducted. The results are depicted in Figure 7.4.

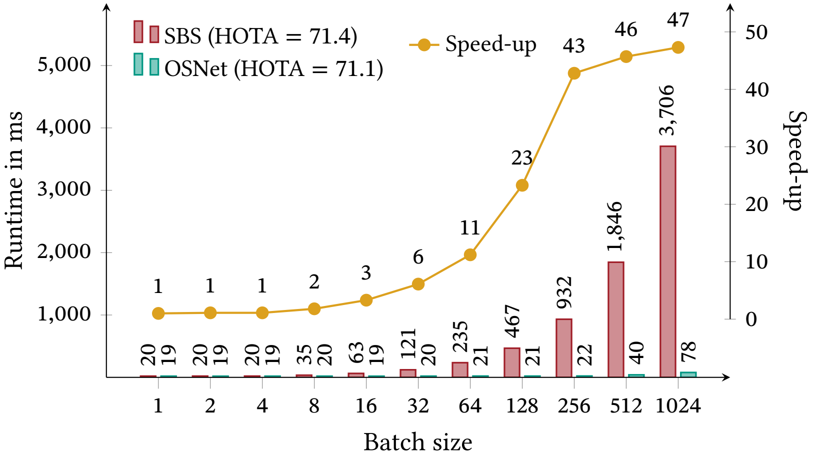


Figure 7.4: Runtime comparison of SBS and OSNet and achieved HOTA values with the tracking system on MOT17 val. Speed-up of OSNet w.r.t. SBS is also given. OSNet is overall much faster than SBS, especially for large batch sizes, while the performance in HOTA is only slightly lower. Note the logarithmic scale of the horizontal axis.

While the runtime advantage for small batch sizes ($B \leq 4$) is minor, OSNet runs significantly faster than SBS for larger batch sizes ($B \geq 8$). For $B = 8$, OSNet runs at about twice the speed as SBS and for $B = 1024$, it is approximately 47 times faster. The processing time of OSNet keeps nearly constant up to a batch size of 256, since the lightweight network can benefit from strong parallelization on the GPU. The reduced model capacity comes only with a small degradation in tracking performance of 0.3 HOTA. This is tolerable as the whole tracking system can run in real time with OSNet, even when a large number of targets has to be tracked. The REID model as well as the detector can be further accelerated using a specialized inference library, which will be discussed in the next section.

7.3 Deployment in TensorRT

TensorRT is a library for deep learning model inference, which is strongly optimized to run on NVIDIA GPUs. With the help of quantization, layer and tensor fusion, kernel tuning, etc., the runtime of a model can be significantly reduced, while (nearly) maintaining the original accuracy. That the conversion to TensorRT does not come with a loss of accuracy has been checked by exchanging the PyTorch models with its TensorRT counterparts and evaluating the tracking performance on MOT17 val, where the same HOTA values have been obtained. The performed optimizations are dependent on the specific hardware platform used, so a TensorRT converted model (also referred to as *engine*) can only run on the type of GPU, where it has been created. To speed up the proposed tracking system, the detection model and REID model have been converted with TensorRT on a Tesla V100-SXM2-32GB GPU. The degree of acceleration and its influence on the runtime of the whole pipeline are described in the following.

7.3.1 Detector

When generating a TensorRT engine, for instance, from a PyTorch model, one has to define the maximum batch size that the resulting model will be able to handle. Since the proposed tracking system works in an online fashion, directly processing frame after frame, the required batch size for the detector is one. However, to examine if a batch processing of multiple input images has the potential of speeding up the system, the applied YOLOX-X detector is converted to TensorRT for various batch sizes. The resulting runtimes are listed together with the PyTorch baseline in Table 7.6.

Table 7.6: Runtime in ms of YOLOX-X with PyTorch and TensorRT for various batch sizes. OOM (out of memory) means that the GPU memory is too small for inference with the respective batch size. Speed-up of TensorRT vs. PyTorch is also given. In an online processing ($B = 1$), the TensorRT engine is 1.5 times faster than the PyTorch baseline.

Batch size	1	2	3	4	10	11
PyTorch	27.4±0.3	53.4±0.3	73.2±0.3	100.4±0.4	228.1±0.7	OOM
TensorRT	**18.6**±0.2	**33.4**±0.2	**43.3**±0.3	OOM	OOM	OOM
Speed-up	1.5	1.6	1.7	–	–	–

If the frames of the incoming video are processed one after another, i.e., $B = 1$ (*online* processing), the TensorRT model is about 1.5 times faster than the PyTorch counterpart, while maintaining the same accuracy. If the specific application allows to output the tracking results with a small delay, one could process multiple frames simultaneously by the detector and the other tracking modules (*batch* processing). For $B = 3$, a mean processing time of 14.4 ms per frame is achieved by the TensorRT model, which is an additional speed-up of about 30 % compared to the online processing.

One has to note that TensorRT engines typically require more memory than PyTorch models for saving optimization data and intermediate results. Therefore, the maximum batch size fitting into the applied GPU with 32GB memory is three, whereas the PyTorch model can inference up to ten images of size 1440×800 pixels simultaneously. However, the runtime with PyTorch is 22.8 ms per image for $B = 10$, which is about 1.6 times the runtime per image of the TensorRT engine at its maximum batch size $B = 3$. To conclude, TensorRT leads to a significant acceleration of the applied YOLOX-X detector both for online and batch processing.

7.3.2 Re-Identification Model

The second module that can be accelerated with the TensorRT library is the $\text{OSNet}_{\beta=0.25}$ REID model. It has been converted for various maximum batch sizes, while allowing the batch size to change *dynamically* during inference. This is an important property, since the batch size can be set to the number of actual targets in each iteration and the input of the network does not have to contain any dummy values as it would be required if the TensorRT engine was generated using a *static* batch size. The runtimes of the generated TensorRT engines for various batch sizes are compared with the PyTorch counterpart in Table 7.7. Note that the given batch size is both the maximum allowed batch size for the respective TensorRT engine as well as the applied batch size for measuring the runtime. In total, 13 engines with different maximum batch sizes have been created that are compared with the same PyTorch model, which is only restricted in its maximum batch size by the memory of the used GPU.

Table 7.7: Runtime in ms of OSNet$_{\beta=0.25}$ with PyTorch and TensorRT for various batch sizes. OOM (out of memory) means that the GPU memory is too small for inference with the respective batch size. Speed-up of TensorRT vs. PyTorch is also given. The TensorRT engine is notably faster than the PyTorch baseline, in particular for small batch sizes.

Batch size	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192
PyTorch	19.4±1.8	18.8±0.8	19.3±1.2	19.5±1.6	19.0±1.4	19.9±1.3	21.0±2.1	20.1±1.1	21.8±0.8	40.4±0.1	78.4±0.1	153.1±0.2	348.7±2.0	703.4±2.6
TensorRT	**1.0**±0.0	**1.4**±0.0	**1.4**±0.1	**1.6**±0.1	**1.9**±0.1	**2.5**±0.1	**3.5**±0.0	**5.5**±0.0	**9.3**±0.0	**16.7**±0.1	**31.3**±0.1	**60.7**±0.2	**118.6**±0.3	OOM
Speed-up	19.4	13.4	13.8	12.2	10.0	8.0	6.0	3.7	2.3	2.4	2.5	2.5	2.9	–

The smaller the maximum batch size of the TensorRT engine, the smaller the range of the input shape that has to be compliant with the engine, the more optimizations can be performed when creating the engine. Thus, one observes consistent runtime decreases when lowering the batch size up to $B = 1$ for TensorRT, whereas the runtime is nearly constant for $B \leq 256$ for the PyTorch model. This is because the PyTorch model has to be capable of handling any batch size and cannot perform optimizations that would be possible if the applicable batch size was limited.

The achieved speed-up of the OSNet$_{\beta=0.25}$ model using TensorRT in comparison to the PyTorch baseline lies in the range of $[2.3, 19.4]$ depending on the applied maximum batch size, which corresponds to the number of targets the TensorRT engine is able to process simultaneously. The advantage of the PyTorch model being capable of handling a large batch size of $B = 8192$ (and larger) in contrast to TensorRT, which runs out of memory (OOM), is more of a theoretical nature, since such a large number of targets hardly occurs in a real MPT application.

In contrast to the detector, processing a batch of thousands of images in parallel is possible as the size of the input images is only 128×256 for the REID model in comparison to 1440×800 for the detector. Therefore, the amount of memory needed to save intermediate network results like feature maps is significantly smaller. The small memory consumption of the REID model allows to perform a batch processing of stacked image patches from consecutive frames, if a further acceleration of the tracking pipeline is needed.

7.4 Efficient Camera Motion Compensation

Whenever the camera position or orientation is not static, compensating introduced camera motion is important to maintain a high accuracy of the tracking system. It has already been stated in Section 6.4 that the commonly applied ECC method [Eva08] for CMC is too slow to run in real time. For this reason, a more efficient alternative based on the ORB keypoint detector [Rub11] and BRIEF descriptor extractor [Cal10] has been proposed in Section 6.4. To further accelerate the processing, a lightweight configuration of the ORB detector is investigated following a previous work from the author of this thesis [Sta23c]. In the lightweight configuration, the number of levels in the involved image pyramid is reduced from 8 to 2, and the scale factor between consecutive pyramid levels is increased from 1.2 to 2. The settings of the BRIEF descriptor extractor as well as the other settings of the ORB detector are taken over from the standard configuration of the OpenCV library. To compare the runtime of the different CMC methods, experiments have been conducted on MOT17 val, whereby the efficient OSNet$_{\beta=0.25}$ has been applied as REID model. As size of the input images, the resolution of the YOLOX-X detector input has been adopted: 1440×800 pixels. The experimental results are listed in Table 7.8. Note that the first row depicts the baseline results when no CMC is used for reference.

Table 7.8: Runtime in ms and accuracy of CMC methods on MOT17 val using OSNet$_{\beta=0.25}$ as REID model. The proposed ORB+BRIEF method is significantly faster than the frequently used ECC and achieves better results. The lightweight configuration roughly halves the runtime while maintaining the same HOTA.

CMC method	Configuration	Runtime	HOTA	DetA	AssA
–	–	–	70.6	67.4	74.5
ECC	–	313.6±223.6	70.9	**67.6**	74.9
ORB+BRIEF	standard	20.8±1.8	**71.1**	67.5	**75.3**
ORB+BRIEF	lightweight	**10.9**±0.9	**71.1**	**67.6**	**75.3**

With a mean runtime of more than 300 ms, it is clear that the ECC method prevents the real-time capability of the whole tracking system. Moreover, a high standard deviation of 224 ms is obtained, as the algorithm requires different numbers of iterations to converge depending on the image content. The proposed CMC approach is about 15 times faster than ECC with the OpenCV standard configuration and even 29 times faster when the introduced lightweight configuration is used. In terms of accuracy on MOT17 val, the lightweight model is on par with the standard configuration and both versions perform better than the ECC baseline in combination with the fast OSNet REID model.

A more detailed runtime analysis including the time for the individual components of the ORB+BRIEF method is found in Table 7.9.

Table 7.9: Runtime in ms of the components of the proposed CMC method ORB+BRIEF for the standard and lightweight configuration. The components comprise keypoint detection, feature extraction, keypoint matching, and computation of transformation. The lightweight configuration substantially decreases the runtime for keypoint detection.

Configuration	Detection	Extraction	Matching	Transformation	Total
Standard	14.37±1.23	2.13±0.31	4.10±0.47	0.15±0.03	20.76±1.84
Lightweight	**4.67**±0.40	**2.02**±0.27	**4.02**±0.38	**0.14**±0.03	**10.85**±0.91

In the standard configuration, the most time is spent for the ORB keypoint detector. Lowering the size of the involved image pyramid in the lightweight configuration, this time is reduced by two thirds without loss of accuracy. The runtime for the following components, i.e., descriptor extraction for all detected keypoints, matching of keypoints, and computation of the transformation matrix, does not differ significantly among the configurations. Making the ORB keypoint detector more lightweight, the overall runtime of ORB+BRIEF is reduced by half leading to a mean processing time of about 11 ms.

As last runtime optimization of the tracking system, the parallel execution of detection and CMC is examined. The detection of persons in the current frame and the compensation of camera motion between the last and the current frame can be performed independently. Therefore, a parallel execution of detection (detector and adapted NMS) and CMC in two separate threads is performed and the total runtime including synchronization is measured. Moreover, a sequential processing of detector, adapted NMS, and CMC is carried out. The results are shown in Table 7.10.

Table 7.10: Runtime in ms of detector, adapted NMS, and CMC as well as total runtime for sequential execution (left) and parallel execution (right) of detection, i.e., detector and adapted NMS, and CMC. The parallel execution is about 1.6 times faster.

Detector	Adapted NMS	CMC	Total	Parallel
18.4±0.2	2.6±0.3	13.2±1.5	34.2±1.7	**21.1**±0.4

With a mean runtime of approximately 21 ms, the parallel execution of detection and CMC is about 1.6 times faster than processing them sequentially. This optimization further contributes to the real-time capability of the proposed tracking framework.

7.5 Runtime of the Accelerated System

The optimizations from the previous sections are combined to an accelerated tracking system. Before a detailed runtime analysis is provided, the applied changes w.r.t. the proposed tracking framework from Chapter 6 are summarized in the following:

- Exchange of the SBS S50 model with $\text{OSNet}_{\beta=0.25}$. Next to a huge speed-up of the REID module, this comes with a reduction of the appearance feature length l_f and thus an acceleration of the cosine distance computation.
- Deployment of the detection model YOLOX-X and the REID model $\text{OSNet}_{\beta=0.25}$ with the fast inference library TensorRT.
- Introduction of a lightweight version of the proposed ORB+BRIEF method for CMC.
- Parallel implementation of detection and CMC.

The resulting runtimes and throughputs of the pipeline for various batch sizes are visualized and compared with the baseline in Figure 7.5. Note that the total runtime of the baseline slightly differs from Figure 7.3 as the runtime for the CMC (20.8 ms, Table 7.9) has been added for a fair comparison.

Recall that the average frame rate of a surveillance industry camera is 15 FPS. Whereas the baseline system cannot achieve this frame rate for any number of targets, the optimized system runs at 18.6 FPS while tracking 512 targets simultaneously. For such a high number of targets, the baseline runs only at 0.5 FPS, which is about 36 times slower than the optimized framework. Remember that this great acceleration comes only with a small decrease in tracking performance (0.3 HOTA on MOT17 val).

While the runtime of the proposed system obviously depends on the utilized hardware, the results from Figure 7.5 show that it is possible to design a SOTA real-time-capable MPT system that contains all relevant modules including the often computationally complex REID and CMC. As future work, the system could be further accelerated by increasing the degree of parallelization, for instance, by implementing more modules on the GPU (motion model, LSA) or executing more modules concurrently.

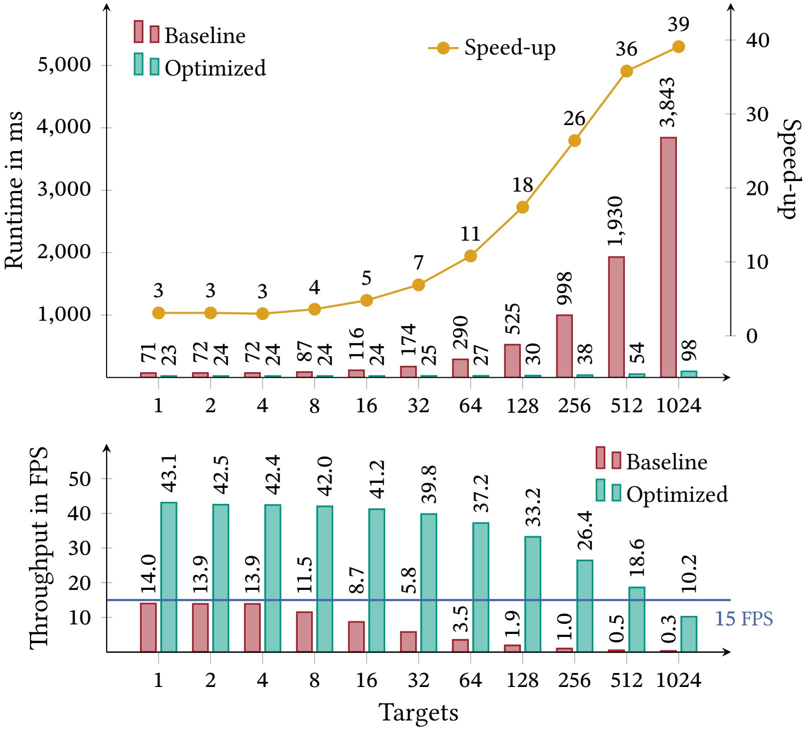


Figure 7.5: Runtime comparison of the optimized and baseline tracking system for various numbers of targets. Achieved speed-ups and throughputs are also given. The optimized system is significantly faster than the baseline and is able to track 512 targets in real time, i.e., with more than 15 FPS. Notice the logarithmic scale of the horizontal axes.

8 Conclusions and Outlook

This chapter summarizes the results and findings of this thesis and draws conclusions in Section 8.1. After that, an outlook on possible future work to further develop the proposed tracking system is given in Section 8.2.

8.1 Conclusions

This thesis proposes a novel MPT framework that focuses on the improved utilization of detections and target information. Following the TBD paradigm, it is very flexible and can be tuned for various applications by exchanging single tracking components. As a starting point of the development, a base framework has been built that is representative for many MPT approaches from the literature. A detailed analysis of this baseline has revealed several weaknesses of existing methods when it comes to using the available information in the tracking process: the rejection of generated detections under heavy occlusion, an insufficient fusion of motion and appearance information in the association, and the ignorance of the neighborhood of unassigned detections when initializing new tracks. Furthermore, two small extensions of the basic motion model have shown the large potential of integrating previously unused information. The NSA Kalman filter leverages the confidence score of a detection to improve the Kalman filter update step by taking the measurement uncertainty into account. Enforcing physical constraints, the HP module prevents infeasible changes of the target size during the Kalman filter prediction step. Both modifications lead to a significant enhancement of the motion model accuracy and thus the overall tracking performance by exploiting additional information.

To enable the use of detections under severe occlusion that have been discarded by previous tracking frameworks, an adapted NMS is proposed that outputs a set of heavily-occluded detections next to the normal detection set. Based on the adapted NMS, two different association strategies are suggested that leverage these additional detections. After assigning high-confidence detections to tracks in the first association stage, BYTEv2 matches heavily-occluded detections to the remaining unassigned tracks in a second association stage. Besides increasing the detection recall in crowded scenes, this comes with a simplification of the association task as less detections are missing and ambiguities of track–detection assignments are eliminated. The same goal is achieved with the TWC association technique. Using the track information, areas with missing detections are identified and the heavily-occluded detections from the adapted NMS are incorporated in these areas to reduce the number of missing detections and improve the association accuracy. Both introduced strategies for including heavily-occluded detections in the association significantly enhance the tracking performance under occlusion, which is where naturally most errors occur.

Several fusion mechanisms for motion and appearance information can be found in the MPT literature. However, due to differences in the tracking frameworks, a reasonable comparison has been missing so far. In this thesis, the developed base framework is used to conduct a thorough analysis of existing fusion approaches and enable a fair comparison for the first time. It is found that prevailing methods fuse the two information sources in a suboptimal way in the sense that motion cues are only used for gating or that either one or the other information is decisive but not both. Based on that, novel distance measures for a combined motion- and appearance-based association are introduced that outperform the previous approaches by a large margin on three different datasets. This demonstrates the high importance of fusing the available information effectively.

Next to the association, other parts of the tracking task are enhanced. To improve the initialization process of tracks in crowded areas, the OAI is proposed, which identifies and removes duplicate detections with the help of track information and thus prevents the start of ghost tracks. Furthermore, an efficient CMC method is suggested that enables the application of the tracking framework for non-static cameras. By an extensive comparison of various algorithms for keypoint detection and descriptor extraction, an efficient combination is found that performs on par with the prevailing CMC method from the MPT literature while being much faster. Putting all developed components together, the proposed tracking system surpasses the SOTA on the well-established MPT benchmarks MOT17 and MOT20. This accomplishment is mainly attributable to a better use of available information in the tracking process, since the same detection, REID, and motion model as in the competing tracking frameworks are applied.

Due to the complexity of an MPT system, most SOTA methods are not real-time capable or refrain from using computationally expensive components as CMC or REID. In contrast, several optimizations are performed in this thesis to accelerate the proposed tracking system without removing important modules. It is shown that utilizing a more efficient REID model, the time for appearance feature extraction can be reduced up to a factor of 47 while nearly maintaining the original accuracy. A further speed-up is achieved by parallelization, a lightweight CMC variant, and the use of TensorRT as special library for neural network inference to accelerate detection and REID model. The final optimized tracking system runs at 19 FPS while tracking 512 persons simultaneously without sacrificing the SOTA performance. With its high accuracy and efficiency, it can be easily integrated in a real-world application.

8.2 Outlook

Although the proposed MPT system achieves impressive results under various scenarios, there is still room for improvement. Some ideas to further enhance the performance are outlined in the following.

As one of the main findings of this work, integrating detections under heavy occlusion in the tracking process helps to simplify the association task in crowded scenes, where most tracking errors occur. Thus, exploiting detectors that aim at enhancing the detection recall under occlusion for the task of MPT is a promising research direction, as already shown in a previous study of this thesis' author [Sta21d]. Furthermore, increasing the detection recall of persons by explicitly focusing on body parts that have typically a higher visibility than the whole body, such as the head, is another promising idea [Zha24]. The interplay of the proposed tracking framework with such specialized detection approaches for crowded scenes can be investigated in the future, since the followed TBD paradigm allows a simple exchange of the applied detector.

It has been shown that a sophisticated fusion of motion and appearance cues significantly enhances the association accuracy. However, the two information sources are combined in the same way at each time step, ignoring potential deficiencies in the extracted features. For instance, the motion state of an inactive track can have a high uncertainty when no measurements, i.e., detections, were available for a long time period or severe camera motion occurred. On the other hand, the extracted appearance features of detections might be misleading under strong occlusion or for small objects. Considering such context knowledge and putting more weight on the motion or appearance information whenever the other one is unreliable could lead to a further boost of the association accuracy. Next to a manual design of measures for the reliability of extracted features, learning the fusion mechanism implicitly from data with GNNs is a possible alternative direction [Bra20, Cet23, You23].

Large potential lies in the use of available information at points of the MPT pipeline where it has not been considered yet. Including the detection confidence in the association distance and using a special association strategy for low-confidence tracks are two examples found in the recent method ConfTrack [Jun24] that could be employed in the proposed framework. Due to the high modularity, adopting an advanced motion model [Cao23, Jun24, Qin23] or appearance model [Ren23, You23] from other trackers is also conceivable.

Besides motion and appearance cues, other information has already been exploited for MPT in the past, which is rarely used by recent SOTA methods. This includes human poses, relations of targets, or 3D information, to name a few. The findings of this work indicate that, on the one hand, integrating such additional knowledge into the association task is still a promising research direction, and, on the other hand, more attention needs to be paid to the correct merging of the available information. Moreover, with additional tracking components and increasing complexity, more emphasis must be placed on the efficiency of MPT systems to allow the deployment in real-world applications.

Bibliography

[Agr08] Agrawal, Motilal; Konolige, Kurt and Blas, Morten Rufus: "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching". In: *Computer Vision – ECCV 2008*. Ed. by Forsyth, David; Torr, Philip and Zisserman, Andrew. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 102–115. doi: 10.1007/978-3-540-88693-8_8 (cit. on p. 163).

[Agr24] Agrawal, Harshit; Halder, Agrya and Chattopadhyay, Pratik: "A systematic survey on recent deep learning-based approaches to multi-object tracking". In: *Multimedia Tools and Applications* 83.12 (2024), pp. 36203–36259. doi: 10.1007/S11042-023-16910-9 (cit. on p. 18).

[Aha22] Aharon, Nir; Orfaig, Roy and Bobrovsky, Ben-Zion: BoT-SORT: Robust Associations Multi-Pedestrian Tracking. 2022. arXiv: 2206.14651 (cit. on pp. 3, 4, 19–23, 27, 35, 36, 39, 50, 65, 66, 69, 84, 93, 94, 96, 97, 104, 115, 144, 146, 147, 150, 160, 168–172, 186).

[Ala12] Alahi, Alexandre; Ortiz, Raphael and Vandergheynst, Pierre: "FREAK: Fast Retina Keypoint". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 510–517. doi: 10.1109/CVPR.2012.6247715 (cit. on p. 163).

[Alc12] Alcantarilla, Pablo Fernández; Bartoli, Adrien and Davison, Andrew J.: "KAZE Features". In: *Computer Vision – ECCV 2012*. Ed. by Fitzgibbon, Andrew; Lazebnik, Svetlana; Perona, Pietro; Sato, Yoichi and Schmid, Cordelia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 214–227. doi: 10.1007/978-3-642-33783-3_16 (cit. on p. 163).

[Alc13] Alcantarilla, Pablo Fernández; Nuevo, Jesús and Bartoli, Adrien: "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces". In: *Proceedings of the British Machine Vision Conference*. Ed. by Burghardt, Tilo; Damen, Dima; Mayol-Cuevas, Walterio W. and Mirmehdi, Majid. Article no. 13. BMVA Press, 2013. url: https://bmva-archive.org.uk/bmvc/2013/Papers/paper0013 (cit. on p. 163).

[Amo23] Amosa, Temitope Ibrahim; Sebastian, Patrick; Izhar, Lila Iznita; Ibrahim, Oladimeji; Ayinla, Lukman Shehu; Bahashwan, Abdulrahman Abdullah; Bala, Abubakar and Samaila, Yau Alhaji: "Multi-camera multi-object tracking: A review of current trends and future advances". In: *Neurocomputing* 552 (2023). Article no. 126558. doi: 10.1016/J.NEUCOM.2023.126558 (cit. on p. 36).

[Bao21] Bao, Qian; Liu, Wu; Cheng, Yuhao; Zhou, Boyan and Mei, Tao: "Pose-Guided Tracking-by-Detection: Robust Multi-Person Pose Tracking". In: *IEEE Transactions on Multimedia* 23 (2021), pp. 161–175. doi: 10.1109/TMM.2020.2980194 (cit. on p. 21).

[Bas22] Bashar, Mk; Islam, Samia; Hussain, Kashifa Kawaakib; Hasan, Md. Bakhtiar; Rahman, A. B. M. Ashikur and Kabir, Md. Hasanul: Multiple Object Tracking in Recent Times: A Literature Review. 2022. arXiv: 2209.04796 (cit. on p. 18).

[Ben11] Benfold, Ben and Reid, Ian: "Stable Multi-Target Tracking in Real-Time Surveillance Video". In: *CVPR 2011*. 2011, pp. 3457–3464. doi: 10.1109/CVPR.2011.5995667 (cit. on p. 21).

[Ber08] Bernardin, Keni and Stiefelhagen, Rainer: "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics". In: *EURASIP Journal on Image and Video Processing* 2008 (2008). Article no. 246309. url: https://jivp-eurasipjournals.springeropen.com/articles/10.1155/2008/246309 (cit. on p. 55).

[Ber11] Berclaz, Jerome; Fleuret, Francois; Turetken, Engin and Fua, Pascal: "Multiple Object Tracking Using K-Shortest Paths Optimization". In: *IEEE Transactions on Pattern Analysis and Machine*

Intelligence 33.9 (2011), pp. 1806–1819. DOI: 10.1109/TPAMI.2011.21 (cit. on p. 19).

[Ber19] BERGMANN, Philipp; MEINHARDT, Tim and LEAL-TAIXÉ, Laura: “Tracking Without Bells and Whistles”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* 2019, pp. 941–951. DOI: 10.1109/ICCV.2019.00103 (cit. on pp. 18, 19, 24, 84, 160, 186).

[Bew16] BEWLEY, Alex; GE, Zongyuan; OTT, Lionel; RAMOS, Fabio and UPCROFT, Ben: “Simple online and realtime tracking”. In: *2016 IEEE International Conference on Image Processing (ICIP).* 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003 (cit. on pp. 3, 18–20, 23, 25, 39, 69, 94, 99, 115, 143, 186).

[BMW24] BMW GROUP: Autonomous, driverless transport vehicle navigates precisely through BMW Group Plant Regensburg press plant. 2024. URL: https://www.press.bmwgroup.com/global/article/detail/T0442979EN/autonomous-driverless-transport-vehicle-navigates-precisely-through-bmw-group-plant-regensburg-press-plant. Accessed on July 16, 2024 (cit. on p. 2).

[Boc17] BOCHINSKI, Erik; EISELEIN, Volker and SIKORA, Thomas: “High-Speed Tracking-by-Detection Without Using Image Information”. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* 2017. DOI: 10.1109/AVSS.2017.8078516 (cit. on p. 25).

[Boc20] BOCHKOVSKIY, Alexey; WANG, Chien-Yao and LIAO, Hong-Yuan Mark: YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. arXiv: 2004.10934 (cit. on p. 71).

[Bra00] BRADSKI, Gray: “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* 25.11 (2000), pp. 120–125. URL: https://www.proquest.com/docview/202684726 (cit. on p. 163).

[Bra20] BRASÓ, Guillem and LEAL-TAIXÉ, Laura: “Learning a Neural Solver for Multiple Object Tracking”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020, pp. 6246–6256. DOI: 10.1109/CVPR42600.2020.00628 (cit. on pp. 19, 21, 29, 30, 204).

[Bre09] BREITENSTEIN, Michael D.; REICHLIN, Fabian; LEIBE, Bastian; KOLLER-MEIER, Esther and VAN GOOL, Luc: “Robust Tracking-by-Detection using a Detector Confidence Particle Filter”. In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 1515–1522. DOI: 10.1109/ICCV.2009.5459278 (cit. on p. 21).

[Bri19] BRIDGEMAN, Lewis; VOLINO, Marco; GUILLEMAUT, Jean-Yves and HILTON, Adrian: “Multi-Person 3D Pose Estimation and Tracking in Sports”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 2487–2496. DOI: 10.1109/CVPRW.2019.00304 (cit. on p. 22).

[Bro93] BROMLEY, Jane; BENTZ, James W.; BOTTOU, Léon; GUYON, Isabelle; LECUN, Yann; MOORE, Cliff; SÄCKINGER, Eduard and SHAH, Roopak: “Signature Verification Using A “Siamese” Time Delay Neural Network”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 7.4 (1993), pp. 669–688. DOI: 10.1142/S0218001493000339 (cit. on p. 32).

[Cal10] CALONDER, Michael; LEPETIT, Vincent; STRECHA, Christoph and FUA, Pascal: “BRIEF: Binary Robust Independent Elementary Features”. In: *Computer Vision – ECCV 2010*. Ed. by DANIILIDIS, Kostas; MARAGOS, Petros and PARAGIOS, Nikos. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792. DOI: 10.1007/978-3-642-15561-1_56 (cit. on pp. 163, 195).

[Cam17] CAMPLANI, Massimo; PAIEMENT, Adeline; MIRMEHDI, Majid; DAMEN, Dima; HANNUNA, Sion; BURGHARDT, Tilo and TAO, Lili: “Multiple human tracking in RGB-depth data: a survey”. In: *IET Computer Vision* 11.4 (2017), pp. 265–285. DOI: 10.1049/IET-CVI.2016.0178 (cit. on p. 22).

[Cao23] CAO, Jinkun; PANG, Jiangmiao; WENG, Xinshuo; KHIRODKAR, Rawal and KITANI, Kris: “Observation-Centric SORT: Rethinking

SORT for Robust Multi-Object Tracking". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 9686–9696. DOI: 10.1109/CVPR52729.2023.00934 (cit. on pp. 3, 19, 22–25, 27, 39, 65, 66, 69, 96, 115, 143, 168–172, 204).

[Car20] Carion, Nicolas; Massa, Francisco; Synnaeve, Gabriel; Usunier, Nicolas; Kirillov, Alexander and Zagoruyko, Sergey: "End-to-End Object Detection with Transformers". In: *Computer Vision – ECCV 2020*. Ed. by Vedaldi, Andrea; Bischof, Horst; Brox, Thomas and Frahm, Jan-Michael. Cham: Springer International Publishing, 2020, pp. 213–229. DOI: 10.1007/978-3-030-58452-8_13 (cit. on p. 30).

[Cet23] Cetintas, Orcun; Brasó, Guillem and Leal-Taixé, Laura: "Unifying Short and Long-Term Tracking with Graph Hierarchies". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 22877–22887. DOI: 10.1109/CVPR52729.2023.02191 (cit. on pp. 18, 30, 65, 96, 169, 170, 172, 204).

[Cho12] Choi, Wongun and Savarese, Silvio: "A Unified Framework for Multi-target Tracking and Collective Activity Recognition". In: *Computer Vision – ECCV 2012*. Ed. by Fitzgibbon, Andrew; Lazebnik, Svetlana; Perona, Pietro; Sato, Yoichi and Schmid, Cordelia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 215–230. DOI: 10.1007/978-3-642-33765-9_16 (cit. on p. 21).

[Cho15] Choi, Wongun: "Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3029–3037. DOI: 10.1109/ICCV.2015.347 (cit. on p. 21).

[Cho17] Chollet, François: "Xception: Deep Learning with Depthwise Separable Convolutions". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195 (cit. on p. 190).

[Cho19] Choudhary, Arti and Gokhale, Sharad: “Evaluation of emission reduction benefits of traffic flow management and technology upgrade in a congested urban traffic corridor”. In: *Clean Technologies and Environmental Policy* 21 (2019), pp. 257–273. doi: 10.1007/S10098-018-1634-Z (cit. on p. 2).

[Chu17] Chu, Qi; Ouyang, Wanli; Li, Hongsheng; Wang, Xiaogang; Liu, Bin and Yu, Nenghai: “Online Multi-Object Tracking Using CNN-based Single Object Tracker with Spatial-Temporal Attention Mechanism”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4846–4855. doi: 10.1109/ICCV.2017.518 (cit. on p. 35).

[Chu19] Chu, Peng and Ling, Haibin: “FAMNet: Joint Learning of Feature, Affinity and Multi-Dimensional Assignment for Online Multiple Object Tracking”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6171–6180. doi: 10.1109/ICCV.2019.00627 (cit. on p. 21).

[Chu20] Chu, Xuangeng; Zheng, Anlin; Zhang, Xiangyu and Sun, Jian: “Detection in Crowded Scenes: One Proposal, Multiple Predictions”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12211–12220. doi: 10.1109/CVPR42600.2020.01223 (cit. on pp. 36, 117).

[Cia20] Ciaparrone, Gioele; Luque Sánchez, Francisco; Tabik, Siham; Troiano, Luigi; Tagliaferri, Roberto and Herrera, Francisco: “Deep learning in video multi-object tracking: A survey”. In: *Neurocomputing* 381 (2020), pp. 61–88. doi: 10.1016/J.NEUCOM.2019.11.023 (cit. on p. 18).

[Cor16] Cordts, Marius; Omran, Mohamed; Ramos, Sebastian; Rehfeld, Timo; Enzweiler, Markus; Benenson, Rodrigo; Franke, Uwe; Roth, Stefan and Schiele, Bernt: “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223. doi: 10.1109/CVPR.2016.350 (cit. on p. 54).

[Cor21] Corona, Kellie; Osterdahl, Katie; Collins, Roderic and Hoogs, Anthony: "MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection". In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1059–1067. doi: 10.1109/WACV48630.2021.00110 (cit. on p. 50).

[Cui23] Cui, Yutao; Zeng, Chenkai; Zhao, Xiaoyu; Yang, Yichun; Wu, Gangshan and Wang, Limin: "SportsMOT: A Large Multi-Object Tracking Dataset in Multiple Sports Scenes". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 9887–9897. doi: 10.1109/ICCV51070.2023.00910 (cit. on pp. 1, 3, 10, 47, 48).

[Cyb23] CyberLink: The Importance of AI People Tracking Technology: Locating Missing Persons. 2023. url: https://www.cyberlink.com/faceme/insights/articles/845/people-tracking-technology-locating-missing-people. Accessed on July 16, 2024 (cit. on p. 2).

[Dai21] Dai, Peng; Weng, Renliang; Choi, Wongun; Zhang, Changshui; He, Zhangping and Ding, Wei: "Learning a Proposal Classifier for Multiple Object Tracking". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2443–2452. doi: 10.1109/CVPR46437.2021.00247 (cit. on p. 29).

[Dat16] Date, Ketan and Nagi, Rakesh: "GPU-accelerated Hungarian algorithms for the Linear Assignment Problem". In: *Parallel Computing* 57 (2016), pp. 52–72. doi: 10.1016/J.PARCO.2016.05.012 (cit. on p. 187).

[Deh15] Dehghan, Afshin; Assari, Shayan Modiri and Shah, Mubarak: "GMMCP tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4091–4099. doi: 10.1109/CVPR.2015.7299036 (cit. on p. 19).

[Den20] Dendorfer, Patrick; Rezatofighi, Hamid; Milan, Anton; Shi, Javen; Cremers, Daniel; Reid, Ian; Roth, Stefan; Schindler, Konrad and Leal-Taixé, Laura: MOT20: A benchmark for multi object tracking in crowded scenes. 2020. arXiv: 2003.09003 (cit. on pp. 11, 44, 48, 50, 166).

[Den22] Dendorfer, Patrick; Yugay, Vladimir; Ošep, Aljoša and Leal-Taixé, Laura: "Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?" In: *Advances in Neural Information Processing Systems.* Ed. by Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K. and Oh, A. Curran Associates, Inc., 2022, pp. 15657–15671. url: https://proceedings.neurips.cc/paper_files/paper/2022/hash/647dc4a76b3efdd676f50f32949299a8-Abstract-Conference.html (cit. on pp. 22, 37, 172).

[Dol14] Dollár, Piotr; Appel, Ron; Belongie, Serge and Perona, Pietro: "Fast Feature Pyramids for Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014), pp. 1532–1545. doi: 10.1109/TPAMI.2014.2300479 (cit. on p. 21).

[Du21] Du, Yunhao; Wan, Junfeng; Zhao, Yanyun; Zhang, Binyu; Tong, Zhihang and Dong, Junhao: "GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW).* 2021, pp. 2809–2819. doi: 10.1109/ICCVW54120.2021.00315 (cit. on pp. 24–26, 33, 82, 83).

[Du23] Du, Yunhao; Zhao, Zhicheng; Song, Yang; Zhao, Yanyun; Su, Fei; Gong, Tao and Meng, Hongying: "StrongSORT: Make DeepSORT Great Again". In: *IEEE Transactions on Multimedia* 25 (2023), pp. 8725–8737. doi: 10.1109/TMM.2023.3240881 (cit. on pp. 19, 21–24, 26, 27, 35, 50, 65, 66, 84, 93, 94, 96, 97, 104, 144, 146, 160, 168–170, 172).

[Du24] Du, Chenjie; Lin, Chenwei; Jin, Ran; Chai, Bencheng; Yao, Yingbiao and Su, Siyu: “Exploring the State-of-the-Art in Multi-Object Tracking: A Comprehensive Survey, Evaluation, Challenges, and Future Directions”. In: *Multimedia Tools and Applications* (2024). doi: 10.1007/S11042-023-17983-2 (cit. on p. 18).

[Elh21] Elharrouss, Omar; Almaadeed, Noor and Al-Máadeed, Somaya: “A review of video surveillance systems”. In: *Journal of Visual Communication and Image Representation* 77 (2021). Article no. 103116. doi: 10.1016/J.JVCIR.2021.103116 (cit. on p. 1).

[Ess07] Ess, Andreas; Leibe, Bastian and Van Gool, Luc: “Depth and Appearance for Mobile Scene Analysis”. In: *2007 IEEE 11th International Conference on Computer Vision*. 2007. doi: 10.1109/ICCV.2007.4409092 (cit. on p. 55).

[Eva08] Evangelidis, Georgios D. and Psarakis, Emmanouil Z.: “Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1858–1865. doi: 10.1109/TPAMI.2008.113 (cit. on pp. 5, 24, 26, 160, 161, 195).

[Fab21] Fabbri, Matteo; Brasó, Guillem; Maugeri, Gianluca; Cetintas, Orcun; Gasparini, Riccardo; Ošep, Aljoša; Calderara, Simone; Leal-Taixé, Laura and Cucchiara, Rita: “MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?” In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 10829–10839. doi: 10.1109/ICCV48922.2021.01067 (cit. on p. 48).

[Fai19] Faiza Gul, Wan Rahiman and Alhady, Syed Sahal Nazli: “A comprehensive study for robot navigation techniques”. In: *Cogent Engineering* 6.1 (2019). Ed. by Chen, Kun. Article no. 1632046. doi: 10.1080/23311916.2019.1632046 (cit. on p. 1).

[Fan18] Fang, Kuan; Xiang, Yu; Li, Xiaocheng and Savarese, Silvio: “Recurrent Autoregressive Networks for Online Multi-Object Tracking”. In: *2018 IEEE Winter Conference on Applications of*

Computer Vision (WACV). 2018, pp. 466–475. DOI: 10.1109/WACV.2018.00057 (cit. on p. 32).

[Fel04] FELZENSZWALB, Pedro F. and HUTTENLOCHER, Daniel P.: “Efficient Belief Propagation for Early Vision”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2004. DOI: 10.1109/CVPR.2004.1315041 (cit. on pp. 21, 167).

[Fen22] FENG, Weitao; LI, Baopu and OUYANG, Wanli: “Multi-Object Tracking with Multiple Cues and Switcher-Aware Classification”. In: *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2022. DOI: 10.1109/DICTA56598.2022.10034575 (cit. on p. 35).

[Fis23] FISCHER, Tobias; HUANG, Thomas E.; PANG, Jiangmiao; QIU, Linlu; CHEN, Haofeng; DARRELL, Trevor and YU, Fisher: “QD-Track: Quasi-Dense Similarity Learning for Appearance-Only Multiple Object Tracking”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.12 (2023), pp. 15380–15393. DOI: 10.1109/TPAMI.2023.3301975 (cit. on p. 169).

[Fis81] FISCHLER, Martin A. and BOLLES, Robert C.: “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692 (cit. on p. 162).

[Fro18] FROSSARD, Davi and URTASUN, Raquel: “End-to-end Learning of Multi-sensor 3D Tracking by Detection”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 635–642. DOI: 10.1109/ICRA.2018.8462884 (cit. on p. 22).

[Gal22] GALOR, Amit; ORFAIG, Roy and BOBROVSKY, Ben-Zion: Strong-TransCenter: Improved Multi-Object Tracking based on Transformers with Dense Representations. 2022. arXiv: 2210.13570 (cit. on p. 31).

[Gao24] Gao, Yan; Xu, Haojun; Li, Jie and Gao, Xinbo: "BPMTrack: Multi-Object Tracking With Detection Box Application Pattern Mining". In: *IEEE Transactions on Image Processing* 33 (2024), pp. 1508–1521. doi: 10.1109/TIP.2024.3364828 (cit. on pp. 23, 24, 37, 66, 169, 170, 172, 173).

[Ge21] Ge, Zheng; Liu, Songtao; Wang, Feng; Li, Zeming and Sun, Jian: YOLOX: Exceeding YOLO Series in 2021. 2021. arXiv: 2107.08430 (cit. on pp. 21, 36, 44, 71, 74, 167, 181).

[Gei12] Geiger, Andreas; Lenz, Philip and Urtasun, Raquel: "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074 (cit. on p. 55).

[Gon22] Gong, Yunpeng; Huang, Liqing and Chen, Lifei: Eliminate Deviation with Deviation for Data Augmentation and a General Multi-modal Data Learning Method. 2022. arXiv: 2101.08533 (cit. on pp. 85, 86).

[Gra12] Granstrom, Karl and Orguner, Umut: "A PHD Filter for Tracking Multiple Extended Targets Using Random Matrices". In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5657–5671. doi: 10.1109/TSP.2012.2212888 (cit. on p. 32).

[Guo23] Guo, Wen; Quan, Wuzhou; Gao, Junyu; Zhang, Tianzhu and Xu, Changsheng: "Feature Disentanglement Network: Multi-Object Tracking Needs More Differentiated Features". In: *ACM Transactions on Multimedia Computing, Communications and Applications* 20.3 (2023). Article no. 83. doi: 10.1145/3626825 (cit. on p. 189).

[Han22] Han, Shoudong; Huang, Piao; Wang, Hongwei; Yu, En; Liu, Donghaisheng and Pan, Xiaofeng: "MAT: Motion-aware multi-object tracking". In: *Neurocomputing* 476 (2022), pp. 75–86. doi: 10.1016/J.NEUCOM.2021.12.104 (cit. on pp. 23–25, 160).

[He17] He, Kaiming; Gkioxari, Georgia; Dollár, Piotr and Girshick, Ross: “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV).* 2017, pp. 2980–2988. doi: 10.1109/ICCV.2017.322 (cit. on p. 27).

[He21] He, Jiawei; Huang, Zehao; Wang, Naiyan and Zhang, Zhaoxiang: “Learnable Graph Matching: Incorporating Graph Partitioning with Deep Feature Learning for Multiple Object Tracking”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2021, pp. 5295–5305. doi: 10.1109/CVPR46437.2021.00526 (cit. on pp. 24, 160).

[He23] He, Lingxiao; Liao, Xingyu; Liu, Wu; Liu, Xinchen; Cheng, Peng and Mei, Tao: “FastReID: A Pytorch Toolbox for General Instance Re-identification”. In: *Proceedings of the 31st ACM International Conference on Multimedia.* New York: Association for Computing Machinery, 2023, pp. 9664–9667. doi: 10.1145/3581783.3613460 (cit. on p. 86).

[Her17] Hermans, Alexander; Beyer, Lucas and Leibe, Bastian: In Defense of the Triplet Loss for Person Re-Identification. 2017. arXiv: 1703.07737 (cit. on p. 84).

[Her21] Herzog, Fabian; Ji, Xunbo; Teepe, Torben; Hörmann, Stefan; Gilg, Johannes and Rigoll, Gerhard: “Lightweight Multi-Branch Network For Person Re-Identification”. In: *2021 IEEE International Conference on Image Processing (ICIP).* 2021, pp. 1129–1133. doi: 10.1109/ICIP42928.2021.9506733 (cit. on pp. 85, 86).

[Hor20] Horňáková, Andrea; Henschel, Roberto; Rosenhahn, Bodo and Swoboda, Paul: “Lifted Disjoint Paths with Application in Multiple Object Tracking”. In: *Proceedings of the 37th International Conference on Machine Learning.* PMLR, 2020, pp. 4364–4375. url: https://proceedings.mlr.press/v119/hornakova20a (cit. on p. 19).

[Hos17] Hosang, Jan; Benenson, Rodrigo and Schiele, Bernt: “Learning non-maximum suppression”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6469–6477. doi: 10.1109/CVPR.2017.685 (cit. on p. 117).

[Hua20] Huang, Xin; Ge, Zheng; Jie, Zequn and Yoshie, Osamu: “NMS by Representative Region: Towards Crowded Pedestrian Detection by Proposal Pairing”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10747–10756. doi: 10.1109/CVPR42600.2020.01076 (cit. on p. 117).

[Hua23] Huang, Junchao; He, Xiaoqi and Zhao, Sheng: The Detection and Rectification for Identity-Switch Based on Unfalsified Control. 2023. arXiv: 2307.14591 (cit. on p. 169).

[IPV21] IPVM: Frame Rate Guide for Video Surveillance. 2021. url: https://ipvm.com/reports/frame-rate-surveillance-guide. Accessed on July 16, 2024 (cit. on pp. 8, 179).

[Iza12] Izadinia, Hamid; Saleemi, Imran; Li, Wenhui and Shah, Mubarak: “(MP)^{2}T: Multiple People Multiple Parts Tracker”. In: *Computer Vision – ECCV 2012*. Ed. by Fitzgibbon, Andrew; Lazebnik, Svetlana; Perona, Pietro; Sato, Yoichi and Schmid, Cordelia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 100–114. doi: 10.1007/978-3-642-33783-3_8 (cit. on p. 21).

[Jin20] Jin, Jiating; Li, Xingwei; Li, Xinlong and Guan, Shaojie: “Online Multi-object Tracking with Siamese Network and Optical Flow”. In: *2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC)*. 2020, pp. 193–198. doi: 10.1109/ICIVC50857.2020.9177480 (cit. on p. 32).

[Jin23] Jin, Yan; Gao, Fang; Yu, Jun; Wang, Jiabao and Shuang, Feng: “Multi-Object Tracking: Decoupling Features to Solve the Contradictory Dilemma of Feature Requirements”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 33.9 (2023), pp. 5117–5132. doi: 10.1109/TCSVT.2023.3249162 (cit. on p. 189).

[Joc20] Jocher, Glenn: YOLOv5 by Ultralytics. 2020. url: https://github.com/ultralytics/yolov5. Accessed on July 16, 2024 (cit. on pp. 28, 71, 72).

[Joc23] Jocher, Glenn; Chaurasia, Ayush and Qiu, Jing: Ultralytics YOLO. 2023. url: https://github.com/ultralytics/ultralytics. Accessed on July 16, 2024 (cit. on p. 71).

[Jon20] Jonathon Luiten, Arne Hoffhues: TrackEval. 2020. url: https://github.com/JonathonLuiten/TrackEval. Accessed on July 16, 2024 (cit. on p. 56).

[Jon87] Jonker, R. and Volgenant, A.: "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems". In: *Computing* 38.4 (1987), pp. 325–340. doi: 10.1007/BF02278710 (cit. on p. 186).

[Jun24] Jung, Hyeonchul; Kang, Seokjun; Kim, Takgen and Kim, HyeongKi: "ConfTrack: Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box". In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 6569–6578. doi: 10.1109/WACV57701.2024.00645 (cit. on pp. 22, 24, 27, 35, 65, 66, 93, 96, 104, 169, 171–173, 204).

[Kal60] Kalman, R. E.: "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45. doi: 10.1115/1.3662552 (cit. on pp. 20, 23, 76).

[Kes22] Kesa, Oluwafunmilola; Styles, Olly and Sanchez, Victor: "Multiple Object Tracking and Forecasting: Jointly Predicting Current and Future Object Locations". In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2022, pp. 560–569. doi: 10.1109/WACVW54805.2022.00062 (cit. on p. 37).

[Khu21] Khurana, Tarasha; Dave, Achal and Ramanan, Deva: "Detecting Invisible People". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 3154–3164. doi: 10.1109/ICCV48922.2021.00316 (cit. on pp. 22, 160).

[Kim15] Kim, Chanho; Li, Fuxin; Ciptadi, Arridhana and Rehg, James M.: “Multiple Hypothesis Tracking Revisited”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4696–4704. doi: 10.1109/ICCV.2015.533 (cit. on p. 32).

[Kok16] Kok, Ven Jyn; Lim, Mei Kuan and Chan, Chee Seng: “Crowd behavior analysis: A review where physics meets biology”. In: *Neurocomputing* 177 (2016), pp. 342–362. doi: 10.1016/J.NEUCOM.2015.11.021 (cit. on p. 2).

[Kuh55] Kuhn, H. W.: “The Hungarian Method for the Assignment Problem”. In: *Naval Research Logistics Quarterly* 2.1–2 (1955), pp. 83–97. doi: 10.1002/NAV.3800020109 (cit. on pp. 31, 88, 186).

[Kum21] Kumar, S. V. Aruna; Yaghoubi, Ehsan; Das, Abhijit; Harish, B. S. and Proença, Hugo: “The P-DESTRE: A Fully Annotated Dataset for Pedestrian Detection, Tracking, and Short/Long-Term Re-Identification From Aerial Devices”. In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 1696–1708. doi: 10.1109/TIFS.2020.3040881 (cit. on p. 48).

[Kuo06] Kuo, Sen M.; Lee, Bob H. and Tian, Wenshun: Real-Time Digital Signal Processing: Implementations and Applications. John Wiley & Sons Ltd, 2006. doi: 10.1002/0470035528 (cit. on p. 179).

[Lar24] Larsen, Martin Vonheim; Rolfsjord, Sigmund; Gusland, Daniel; Ahlberg, Jörgen and Mathiassen, Kim: “BASE: Probably a Better Approach to Visual Multi-Object Tracking”. In: *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*. Ed. by Radeva, Petia; Furnari, Antonino; Bouatouch, Kadi and Sousa, A. Augusto. SciTePress, 2024, pp. 110–121. doi: 10.5220/0012386600003660 (cit. on pp. 18, 32, 33, 169, 170, 172).

[Laz17] Lazar, Jonathan; Feng, Jinjuan Heidi and Hochheiser, Harry: Research Methods in Human-Computer Interaction, 2nd Edition. Cambridge: Morgan Kaufmann, 2017. url: https://www.oreilly.

com/library/view/research-methods-in/9780128093436 (cit. on p. 1).

[Lea15] Leal-Taixé, Laura; Milan, Anton; Reid, Ian; Roth, Stefan and Schindler, Konrad: MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. 2015. arXiv: 1504.01942 (cit. on p. 48).

[Lea17] Leal-Taixé, Laura; Milan, Anton; Schindler, Konrad; Cremers, Daniel; Reid, Ian and Roth, Stefan: Tracking the Trackers: An Analysis of the State of the Art in Multiple Object Tracking. 2017. arXiv: 1704.02781 (cit. on p. 18).

[Lei08] Leibe, Bastian; Schindler, Konrad; Cornelis, Nico and Van Gool, Luc: "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1683–1698. doi: 10.1109/TPAMI.2008.170 (cit. on p. 22).

[Leu11] Leutenegger, Stefan; Chli, Margarita and Siegwart, Roland Y.: "BRISK: Binary Robust Invariant Scalable Keypoints". In: *2011 International Conference on Computer Vision*. 2011, pp. 2548–2555. doi: 10.1109/ICCV.2011.6126542 (cit. on p. 163).

[Lev16] Levi, Gil and Hassner, Tal: "LATCH: Learned Arrangements of Three Patch Codes". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2016. doi: 10.1109/WACV.2016.7477723 (cit. on p. 163).

[Li22] Li, Chuyi; Li, Lulu; Jiang, Hongliang; Weng, Kaiheng; Geng, Yifei; Li, Liang; Ke, Zaidan; Li, Qingyuan; Cheng, Meng; Nie, Weiqiang; Li, Yiduo; Zhang, Bo; Liang, Yufei; Zhou, Linyuan; Xu, Xiaoming; Chu, Xiangxiang; Wei, Xiaoming and Wei, Xiaolin: YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. 2022. arXiv: 2209.02976 (cit. on p. 71).

[Lia22] Liang, Chao; Zhang, Zhipeng; Zhou, Xue; Li, Bing; Zhu, Shuyuan and Hu, Weiming: "Rethinking the Competition Between Detection and ReID in Multiobject Tracking". In: *IEEE Transactions on Image Processing* 31 (2022), pp. 3182–3196. doi: 10.1109/TIP.2022.3165376 (cit. on pp. 28, 189).

[Lin06] Lin, L.; Bar-Shalom, Y. and Kirubarajan, T.: “Track Labeling and PHD Filter for Multitarget tracking”. In: *IEEE Transactions on Aerospace and Electronic Systems* 42.3 (2006), pp. 778–795. doi: 10.1109/TAES.2006.248213 (cit. on p. 32).

[Lin14] Lin, Tsung-Yi; Maire, Michael; Belongie, Serge; Hays, James; Perona, Pietro; Ramanan, Deva; Dollár, Piotr and Zitnick, C. Lawrence: “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Ed. by Fleet, David; Pajdla, Tomas; Schiele, Bernt and Tuytelaars, Tinne. Cham: Springer International Publishing, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48 (cit. on p. 53).

[Lin17a] Lin, Tsung-Yi; Dollár, Piotr; Girshick, Ross; He, Kaiming; Hariharan, Bharath and Belongie, Serge: “Feature Pyramid Networks for Object Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106 (cit. on p. 28).

[Lin17b] Lin, Tsung-Yi; Goyal, Priya; Girshick, Ross; He, Kaiming and Dollár, Piotr: “Focal Loss for Dense Object Detection”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2999–3007. doi: 10.1109/ICCV.2017.324 (cit. on p. 28).

[Lin23] Lin, Weiyao; Liu, Huabin; Liu, Shizhan; Li, Yuxi; Xiong, Hongkai; Qi, Guojun and Sebe, Nicu: “HiEve: A Large-Scale Benchmark for Human-Centric Video Analysis in Complex Events”. In: *International Journal of Computer Vision* 131.11 (2023), pp. 2994–3018. doi: 10.1007/S11263-023-01842-6 (cit. on pp. 47, 48).

[Liu19] Liu, Songtao; Huang, Di and Wang, Yunhong: “Adaptive NMS: Refining Pedestrian Detection in a Crowd”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6452–6461. doi: 10.1109/CVPR.2019.00662 (cit. on p. 117).

[Liu20] Liu, Qiankun; Chu, Qi; Liu, Bin and Yu, Nenghai: "GSM: Graph Similarity Model for Multi-Object Tracking". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Bessiere, Christian. International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 530–536. doi: 10.24963/IJCAI.2020/74 (cit. on pp. 19, 21).

[Liu23] Liu, Zelin; Wang, Xinggang; Wang, Cheng; Liu, Wenyu and Bai, Xiang: SparseTrack: Multi-Object Tracking by Performing Scene Decomposition based on Pseudo-Depth. 2023. arXiv: 2306.05238 (cit. on pp. 22, 27, 36, 168–172).

[Low04] Lowe, David G.: "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. doi: 10.1023/B:VISI.0000029664.99615.94 (cit. on p. 163).

[Lu20] Lu, Zhichao; Rathod, Vivek; Votel, Ronny and Huang, Jonathan: "RetinaTrack: Online Single Stage Joint Detection and Tracking". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 14656–14666. doi: 10.1109/CVPR42600.2020.01468 (cit. on pp. 28, 167, 189).

[Luc21] Luckey, Daniel; Fritz, Henrieke; Legatiuk, Dmitrii; Dragos, Kosmas and Smarsly, Kay: "Artificial Intelligence Techniques for Smart City Applications". In: *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*. Ed. by Toledo Santos, Eduardo and Scheer, Sergio. Cham: Springer International Publishing, 2021, pp. 3–15. doi: 10.1007/978-3-030-51295-8_1 (cit. on p. 1).

[Lui21] Luiten, Jonathon; Ošep, Aljoša; Dendorfer, Patrick; Torr, Philip; Geiger, Andreas; Leal-Taixé, Laura and Leibe, Bastian: "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking". In: *International Journal of Computer Vision* 129 (2021), pp. 548–578. doi: 10.1007/S11263-020-01375-2 (cit. on pp. 56–58, 63, 64).

[Luo19] Luo, Hao; Gu, Youzhi; Liao, Xingyu; Lai, Shenqi and Jiang, Wei: "Bag of Tricks and a Strong Baseline for Deep Person Re-Identification". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 1487–1495. doi: 10.1109/CVPRW.2019.00190 (cit. on pp. 26, 27, 33, 86, 87, 146, 183, 190).

[Luo21] Luo, Wenhan; Xing, Junliang; Milan, Anton; Zhang, Xiaoqin; Liu, Wei and Kim, Tae-Kyun: "Multiple object tracking: A literature review". In: *Artificial Intelligence* 293 (2021). Article no. 103448. doi: 10.1016/J.ARTINT.2020.103448 (cit. on pp. 18, 34, 35).

[Mag23] Maggiolino, Gerard; Ahmad, Adnan; Cao, Jinkun and Kitani, Kris: "Deep OC-Sort: Multi-Pedestrian Tracking by Adaptive Re-Identification". In: *2023 IEEE International Conference on Image Processing (ICIP)*. 2023, pp. 3025–3029. doi: 10.1109/ICIP49359.2023.10222576 (cit. on pp. 23, 27, 33, 35, 66, 144, 169, 170, 172).

[Mai10] Mair, Elmar; Hager, Gregory D.; Burschka, Darius; Suppa, Michael and Hirzinger, Gerhard: "Adaptive and Generic Corner Detection Based on the Accelerated Segment Test". In: *Computer Vision – ECCV 2010*. Ed. by Daniilidis, Kostas; Maragos, Petros and Paragios, Nikos. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 183–196. doi: 10.1007/978-3-642-15552-9_14 (cit. on p. 163).

[Man17] Manen, Santiago; Gygli, Michael; Dai, Dengxin and Van Gool, Luc: "PathTrack: Fast Trajectory Annotation with Path Supervision". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 290–299. doi: 10.1109/ICCV.2017.40 (cit. on p. 50).

[Mei22] Meinhardt, Tim; Kirillov, Alexander; Leal-Taixé, Laura and Feichtenhofer, Christoph: "TrackFormer: Multi-Object Tracking with Transformers". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 8834–

8844. DOI: 10.1109/CVPR52688.2022.00864 (cit. on pp. 20, 22, 31, 167).

[Men23] Meng, Ting; Fu, Chunyun; Huang, Mingguang; Wang, Xiyang; He, Jiawei; Huang, Tao and Shi, Wankai: Localization-Guided Track: A Deep Association Multi-Object Tracking Framework Based on Localization Confidence of Detections. 2023. arXiv: 2309.09765 (cit. on pp. 22, 27, 33, 37, 104, 168–172).

[Mik04] Mikolajczyk, Krystian and Schmid, Cordelia: "Scale & Affine Invariant Interest Point Detectors". In: *International Journal of Computer Vision* 60.1 (2004), pp. 63–86. DOI: 10.1023/B:VISI.0000027790.02288.F2 (cit. on p. 163).

[Mil16] Milan, Anton; Leal-Taixe, Laura; Reid, Ian; Roth, Stefan and Schindler, Konrad: MOT16: A Benchmark for Multi-Object Tracking. 2016. arXiv: 1603.00831 (cit. on pp. 8, 11, 44, 48, 49, 166).

[Mil17] Milan, Anton; Rezatofighi, S. Hamid; Dick, Anthony; Reid, Ian and Schindler, Konrad: "Online Multi-Target Tracking Using Recurrent Neural Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (2017). DOI: 10.1609/AAAI.V31I1.11194 (cit. on p. 32).

[Mur17] Murray, Samuel: Real-Time Multiple Object Tracking - A Study on the Importance of Speed. 2017. arXiv: 1709.03572 (cit. on p. 179).

[Nas23] Nasseri, Mohammad Hossein; Babaee, Mohammadreza; Moradi, Hadi and Hosseini, Reshad: "Online relational tracking with camera motion suppression". In: *Journal of Visual Communication and Image Representation* 90 (2023). Article no. 103750. DOI: 10.1016/J.JVCIR.2022.103750 (cit. on pp. 24, 169, 172).

[Nis08] Nistér, David and Stewénius, Henrik: "Linear Time Maximally Stable Extremal Regions". In: *Computer Vision – ECCV 2008.* Ed. by Forsyth, David; Torr, Philip and Zisserman, Andrew. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 183–196. DOI: 10.1007/978-3-540-88688-4_14 (cit. on p. 163).

[Oh11] Oh, Sangmin; Hoogs, Anthony; Perera, Amitha; Cuntoor, Naresh; Chen, Chia-Chih; Lee, Jong Taek; Mukherjee, Saurajit; Aggarwal, J. K.; Lee, Hyungtae; Davis, Larry; Swears, Eran; Wang, Xioyang; Ji, Qiang; Reddy, Kishore; Shah, Mubarak; Vondrick, Carl; Pirsiavash, Hamed; Ramanan, Deva; Yuen, Jenny; Torralba, Antonio; Song, Bi; Fong, Anesco; Roy-Chowdhury, Amit and Desai, Mita: “A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video”. In: *CVPR 2011*. 2011, pp. 3153–3160. doi: 10.1109/CVPR.2011.5995586 (cit. on p. 50).

[Oku04] Okuma, Kenji; Taleghani, Ali; Freitas, Nando de; Little, James J. and Lowe, David G.: “A Boosted Particle Filter: Multitarget Detection and Tracking”. In: *Computer Vision - ECCV 2004*. Ed. by Pajdla, Tomás and Matas, Jiří. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 28–39. doi: 10.1007/978-3-540-24670-1_3 (cit. on p. 21).

[Pan20] Pang, Bo; Li, Yizhuo; Zhang, Yifan; Li, Muchen and Lu, Cewu: “TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 6307–6317. doi: 10.1109/CVPR42600.2020.00634 (cit. on p. 21).

[Pat22] Pattanashetty, Vishal B.; Mane, Venkatesh; Iyer, Nalini C. and Kore, Shweta: “Traffic Rules Violation Detection System”. In: *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*. Ed. by Joshi, Amit; Mahmud, Mufti; Ragel, Roshan G. and Thakur, Nileshsingh V. Singapore: Springer Singapore, 2022, pp. 77–87. doi: 10.1007/978-981-16-0739-4_8 (cit. on p. 2).

[Pir11] Pirsiavash, Hamed; Ramanan, Deva and Fowlkes, Charless C.: “Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects”. In: *CVPR 2011*. 2011, pp. 1201–1208. doi: 10.1109/CVPR.2011.5995604 (cit. on p. 19).

[Pol18] Polacco, Alex and Backes, Kayla: "The Amazon Go Concept: Implications, Applications, and Sustainability". In: *Journal of Business and Management* 24.1 (2018), pp. 79–92. doi: 10.6347/JBM.201803_24(1).0004 (cit. on p. 3).

[Pun21] Punn, Narinder Singh; Sonbhadra, Sanjay Kumar; Agarwal, Sonali and Rai, Gaurav: Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. 2021. arXiv: 2005.01385 (cit. on p. 2).

[Qin23] Qin, Zheng; Zhou, Sanping; Wang, Le; Duan, Jinghai; Hua, Gang and Tang, Wei: "MotionTrack: Learning Robust Short-Term and Long-Term Motions for Multi-Object Tracking". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 17939–17948. doi: 10.1109/CVPR52729.2023.01720 (cit. on pp. 19, 25, 30, 33, 169, 170, 172, 204).

[Qui16] Quintana, Marcos; Menéndez, José Manuel; Alvarez, Federico and López, Juan Pedro: "Improving retail efficiency through sensing technologies: A survey". In: *Pattern Recognition Letters* 81 (2016), pp. 3–10. doi: 10.1016/J.PATREC.2016.05.027 (cit. on p. 1).

[Red16] Redmon, Joseph; Divvala, Santosh; Girshick, Ross and Farhadi, Ali: "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91 (cit. on p. 71).

[Red17] Redmon, Joseph and Farhadi, Ali: "YOLO9000: Better, Faster, Stronger". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690 (cit. on p. 71).

[Red18] Redmon, Joseph and Farhadi, Ali: YOLOv3: An Incremental Improvement. 2018. arXiv: 1804.02767 (cit. on pp. 28, 71, 72).

[Rei79] Reid, Donald B.: "An Algorithm for Tracking Multiple Targets". In: *IEEE Transactions on Automatic Control* 24.6 (1979), pp. 843–854. doi: 10.1109/TAC.1979.1102177 (cit. on p. 32).

[Ren17] Ren, Shaoqing; He, Kaiming; Girshick, Ross and Sun, Jian: "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149. doi: 10.1109/TPAMI.2016.2577031 (cit. on pp. 19, 21, 27, 117, 167).

[Ren23] Ren, Hao; Han, Shoudong; Ding, Huilin; Zhang, Ziwen; Wang, Hongwei and Wang, Faquan: "Focus On Details: Online Multi-Object Tracking with Diverse Fine-Grained Representation". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 11289–11298. doi: 10.1109/CVPR52729.2023.01086 (cit. on pp. 27, 32, 93, 169, 170, 172, 204).

[Ren24] Ren, Weihong; Wu, Denglu; Cao, Hui; Chen, Xi'ai; Han, Zhi and Liu, Honghai: Joint Counting, Detection and Re-Identification for Multi-Object Tracking. 2024. arXiv: 2212.05861 (cit. on pp. 28, 167, 189).

[Rez19] Rezatofighi, Hamid; Tsoi, Nathan; Gwak, JunYoung; Sadeghian, Amir; Reid, Ian and Savarese, Silvio: "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 658–666. doi: 10.1109/CVPR.2019.00075 (cit. on p. 148).

[Ris16] Ristani, Ergys; Solera, Francesco; Zou, Roger; Cucchiara, Rita and Tomasi, Carlo: "Performance Measures and a Data Set for Multi-target, Multi-camera Tracking". In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Hua, Gang and Jégou, Hervé. Cham: Springer International Publishing, 2016, pp. 17–35. doi: 10.1007/978-3-319-48881-3_2 (cit. on p. 55).

[Rob19] Robillard, Jean-Marc: How Security System Automation Can Help Tackle Operator Fatigue. 2019. url: https://www.securitymagazine.com/articles/89915-how-security-system-automation-can-help-tackle-operator-fatigue. Accessed on July 16, 2024 (cit. on p. 3).

[Rod09] Rodriguez, Mikel; Ali, Saad and Kanade, Takeo: “Tracking in Unstructured Crowded Scenes”. In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 1389–1396. doi: 10.1109/ICCV.2009.5459301 (cit. on p. 21).

[Ros06] Rosten, Edward and Drummond, Tom: “Machine Learning for High-Speed Corner Detection”. In: *Computer Vision – ECCV 2006*. Ed. by Leonardis, Aleš; Bischof, Horst and Pinz, Axel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. doi: 10.1007/11744023_34 (cit. on p. 163).

[Ros23] Rosell, Niklas: Ensuring privacy when undertaking surveillance. 2023. url: https://www.axis.com/blog/secure-insights/privacy-security-industry. Accessed on July 16, 2024 (cit. on p. 3).

[Rub11] Rublee, Ethan; Rabaud, Vincent; Konolige, Kurt and Bradski, Gary: “ORB: an efficient alternative to SIFT or SURF”. In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544 (cit. on pp. 24, 163, 195).

[Rud20] Rudenko, Andrey; Palmieri, Luigi; Herman, Michael; Kitani, Kris M.; Gavrila, Dariu M. and Arras, Kai O.: “Human motion trajectory prediction: a survey”. In: *The International Journal of Robotics Research* 39.8 (2020), pp. 895–935. doi: 10.1177/0278364920917446 (cit. on p. 37).

[Ruk21] Rukhovich, Danila; Sofiiuk, Konstantin; Galeev, Danil; Barinova, Olga and Konushin, Anton: “IterDet: Iterative Scheme for Object Detection in Crowded Environments”. In: *Structural, Syntactic, and Statistical Pattern Recognition*. Ed. by Torsello, Andrea; Rossi, Luca; Pelillo, Marcello; Biggio, Battista and Robles-Kelly, Antonio. Cham: Springer International Publishing, 2021, pp. 344–354. doi: 10.1007/978-3-030-73973-7_33 (cit. on p. 36).

[Sad17] Sadeghian, Amir; Alahi, Alexandre and Savarese, Silvio: “Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies”. In: *2017 IEEE International*

Conference on Computer Vision (ICCV). 2017, pp. 300–311. DOI: 10.1109/ICCV.2017.41 (cit. on p. 21).

[Sha18a] Shao, Shuai; Zhao, Zijian; Li, Boxun; Xiao, Tete; Yu, Gang; Zhang, Xiangyu and Sun, Jian: CrowdHuman: A Benchmark for Detecting Human in a Crowd. 2018. arXiv: 1805.00123 (cit. on pp. 36, 53).

[Sha18b] Sharma, Sarthak; Ansari, Junaid Ahmed; Krishna Murthy, J. and Madhava Krishna, K.: "Beyond Pixels: Leveraging Geometry and Shape Cues for Online Multi-Object Tracking". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 3508–3515. DOI: 10.1109/ICRA.2018.8461018 (cit. on p. 22).

[Shi94] Shi, Jianbo and Tomasi, Carlo: "Good Features to Track". In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794 (cit. on p. 163).

[Shu20] Shuai, Bing; Berneshawi, Andrew G.; Modolo, Davide and Tighe, Joseph: Multi-Object Tracking with Siamese Track-RCNN. 2020. arXiv: 2004.07786 (cit. on pp. 27, 32).

[Shu21] Shuai, Bing; Berneshawi, Andrew; Li, Xinyu; Modolo, Davide and Tighe, Joseph: "SiamMOT: Siamese Multi-Object Tracking". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 12367–12377. DOI: 10.1109/CVPR46437.2021.01219 (cit. on p. 32).

[Shu22] Shuai, Bing; Bergamo, Alessandro; Büchler, Uta; Berneshawi, Andrew; Boden, Alyssa and Tighe, Joseph: "Large Scale Real-World Multi-person Tracking". In: *Computer Vision – ECCV 2022*. Ed. by Avidan, Shai; Brostow, Gabriel; Cissé, Moustapha; Farinella, Giovanni Maria and Hassner, Tal. Cham: Springer Nature Switzerland, 2022, pp. 504–521. DOI: 10.1007/978-3-031-20074-8_29 (cit. on pp. 6–8, 48, 50).

[Sif14] Sifre, Laurent and Mallat, Stéphane: Rigid-Motion Scattering for Texture Classification. 2014. arXiv: 1403.1687 (cit. on p. 190).

[Sim14] Simonyan, Karen; Vedaldi, Andrea and Zisserman, Andrew: "Learning Local Feature Descriptors Using Convex Optimisation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014), pp. 1573–1585. doi: 10.1109/TPAMI.2014.2301163 (cit. on p. 163).

[Sim23] Simsek, Fatih Emre; Cigla, Cevahir and Kayabol, Koray: "SOMPT22: A Surveillance Oriented Multi-pedestrian Tracking Dataset". In: *Computer Vision – ECCV 2022 Workshops*. Ed. by Karlinsky, Leonid; Michaeli, Tomer and Nishino, Ko. Cham: Springer Nature Switzerland, 2023, pp. 659–675. doi: 10.1007/978-3-031-25072-9_44 (cit. on pp. 8, 48, 52).

[Spe21] Specker, Andreas; Stadler, Daniel; Florin, Lucas and Beyerer, Jürgen: "An Occlusion-aware Multi-target Multi-camera Tracking System". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021, pp. 4168–4177. doi: 10.1109/CVPRW53098.2021.00471 (cit. on p. 13).

[Sta20] Stadler, Daniel; Sommer, Lars Wilko and Beyerer, Jürgen: "PAS Tracker: Position-, Appearance- and Size-Aware Multi-object Tracking in Drone Videos". In: *Computer Vision – ECCV 2020 Workshops*. Ed. by Bartoli, Adrien and Fusiello, Andrea. Cham: Springer International Publishing, 2020, pp. 604–620. doi: 10.1007/978-3-030-66823-5_36 (cit. on p. 13).

[Sta21a] Stadler, Daniel: "Multi-Object Tracking in Drone Videos". In: *Proceedings of the 2020 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Ed. by Beyerer, Jürgen and Zander, Tim. KIT Scientific Publishing, 2021, pp. 123–133. doi: 10.5445/IR/1000135221 (cit. on p. 13).

[Sta21b] Stadler, Daniel and Beyerer, Jürgen: "Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 10953–10962. doi: 10.1109/CVPR46437.2021.01081 (cit. on pp. 13, 43).

[Sta21c] Stadler, Daniel and Beyerer, Jürgen: "Multi-Pedestrian Tracking with Clusters". In: *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2021. doi: 10.1109/AVSS52988.2021.9663829 (cit. on pp. 12, 13, 41, 42, 112, 119).

[Sta21d] Stadler, Daniel and Beyerer, Jürgen: "On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking". In: *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2021. doi: 10.1109/AVSS52988.2021.9663836 (cit. on pp. 36, 41, 117, 127, 172, 204).

[Sta22a] Stadler, Daniel: "Multi-Person Tracking with a Multi-Hypothesis Approach for Ambiguous Assignments". In: *Proceedings of the 2021 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Ed. by Beyerer, Jürgen and Zander, Tim. KIT Scientific Publishing, 2022, pp. 153–167. doi: 10.5445/IR/1000148359 (cit. on p. 43).

[Sta22b] Stadler, Daniel and Beyerer, Jürgen: "Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds". In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2022, pp. 133–142. doi: 10.1109/WACVW54805.2022.00019 (cit. on pp. 12, 24, 25, 41, 43, 44, 82, 83, 113).

[Sta23a] Stadler, Daniel: "A Detailed Study of the Association Task in Tracking-by-Detection-based Multi-Person Tracking". In: *Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Ed. by Beyerer, Jürgen and Zander, Tim. KIT Scientific Publishing, 2023, pp. 59–85. doi: 10.5445/IR/1000161972 (cit. on pp. 12, 42, 103, 148).

[Sta23b] Stadler, Daniel and Beyerer, Jürgen: "An Improved Association Pipeline for Multi-Person Tracking". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*

(CVPRW). 2023, pp. 3170–3179. DOI: 10.1109/CVPRW59228.2023.00319 (cit. on pp. 13, 21, 26, 42, 43, 92, 103, 112, 137, 144).

[Sta23c] Stadler, Daniel and Beyerer, Jürgen: "BYTEv2: Associating More Detection Boxes Under Occlusion for Improved Multi-person Tracking". In: *Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges*. Ed. by Rousseau, Jean-Jacques and Kapralos, Bill. Cham: Springer Nature Switzerland, 2023, pp. 79–94. DOI: 10.1007/978-3-031-37660-3_6 (cit. on pp. 12, 13, 24, 25, 27, 41–45, 112, 118, 120, 124, 161, 195).

[Sta23d] Stadler, Daniel and Beyerer, Jürgen: "Past Information Aggregation for Multi-Person Tracking". In: *2023 IEEE International Conference on Image Processing (ICIP)*. 2023, pp. 321–325. DOI: 10.1109/ICIP49359.2023.10223159 (cit. on pp. 13, 21, 26, 42, 112, 144).

[Sun21a] Sun, Peize; Cao, Jinkun; Jiang, Yi; Zhang, Rufeng; Xie, Enze; Yuan, Zehuan; Wang, Changhu and Luo, Ping: TransTrack: Multiple Object Tracking with Transformer. 2021. arXiv: 2012.15460 (cit. on pp. 20, 22, 31, 65, 96, 167).

[Sun21b] Sun, ShiJie; Akhtar, Naveed; Song, HuanSheng; Mian, Ajmal and Shah, Mubarak: "Deep Affinity Network for Multiple Object Tracking". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2021), pp. 104–119. DOI: 10.1109/TPAMI.2019.2929520 (cit. on p. 21).

[Sun22] Sun, Peize; Cao, Jinkun; Jiang, Yi; Yuan, Zehuan; Bai, Song; Kitani, Kris and Luo, Ping: "DanceTrack: Multi-Object Tracking in Uniform Appearance and Diverse Motion". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 20961–20970. DOI: 10.1109/CVPR52688.2022.02032 (cit. on pp. 10, 47, 48).

[Tan15] Tang, Siyu; Andres, Bjoern; Andriluka, Mykhaylo and Schiele, Bernt: "Subgraph Decomposition for Multi-Target Tracking". In: *2015 IEEE Conference on Computer Vision* *and Pattern Recognition (CVPR)*. 2015, pp. 5033–5041. doi: 10.1109/CVPR.2015.7299138 (cit. on p. 19).

[Tan16] Tang, Siyu; Andres, Bjoern; Andriluka, Mykhaylo and Schiele, Bernt: "Multi-person Tracking by Multicut and Deep Matching". In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Hua, Gang and Jégou, Hervé. Cham: Springer International Publishing, 2016, pp. 100–111. doi: 10.1007/978-3-319-48881-3_8 (cit. on p. 19).

[Tan17] Tang, Siyu; Andriluka, Mykhaylo; Andres, Bjoern and Schiele, Bernt: "Multiple People Tracking by Lifted Multicut and Person Re-identification". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3701–3710. doi: 10.1109/CVPR.2017.394 (cit. on pp. 18, 19, 21, 84).

[Tei09] Teizer, Jochen and Vela, Patricio A.: "Personnel tracking on construction sites using video cameras". In: *Advanced Engineering Informatics* 23.4 (2009), pp. 452–462. doi: 10.1016/J.AEI.2009.06.011 (cit. on p. 2).

[Ter23] Terven, Juan; Córdova-Esparza, Diana-Margarita and Romero-González, Julio-Alejandro: "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS". In: *Machine Learning and Knowledge Extraction* 5.4 (2023), pp. 1680–1716. doi: 10.3390/MAKE5040083 (cit. on p. 74).

[Tok21] Tokmakov, Pavel; Li, Jie; Burgard, Wolfram and Gaidon, Adrien: "Learning to Track with Object Permanence". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 10840–10849. doi: 10.1109/ICCV48922.2021.01068 (cit. on p. 32).

[Tol10] Tola, Engin; Lepetit, Vincent and Fua, Pascal: "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.5 (2010), pp. 815–830. doi: 10.1109/TPAMI.2009.77 (cit. on p. 163).

[Tom15] Tombari, Federico and Di Stefano, Luigi: "Interest Points via Maximal Self-Dissimilarities". In: *Computer Vision – ACCV 2014.* Ed. by Cremers, Daniel; Reid, Ian; Saito, Hideo and Yang, Ming-Hsuan. Cham: Springer International Publishing, 2015, pp. 586–600. doi: 10.1007/978-3-319-16808-1_39 (cit. on p. 163).

[Tra22] Tran, Trung: 6 Typical Examples of Robots in Everyday Life. 2022. url: https://www.orientsoftware.com/blog/robots-in-everyday-life. Accessed on July 16, 2024 (cit. on p. 2).

[Trz13] Trzcinski, Tomasz; Christoudias, Mario; Fua, Pascal and Lepetit, Vincent: "Boosting Binary Keypoint Descriptors". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition.* 2013, pp. 2874–2881. doi: 10.1109/CVPR.2013.370 (cit. on p. 163).

[Tya21] Tyagi, Riya: The Landscape of AI & Robotic Guides in Museums & Cultural Places. 2021. url: https://www.aldebaran.com/en/blog/news-trends/landscape-ai-robotic-guides-museums-cultural-places. Accessed on July 16, 2024 (cit. on p. 2).

[Vas17] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz and Polosukhin, Illia: "Attention is All you Need". In: *Advances in Neural Information Processing Systems.* Ed. by Guyon, I.; Luxburg, U. Von; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S. and Garnett, R. Curran Associates, Inc., 2017. url: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (cit. on pp. 18, 20, 30, 32).

[Voi19] Voigtlaender, Paul; Krause, Michael; Ošep, Aljoša; Luiten, Jonathon; Sekar, Berin Balachandar Gnana; Geiger, Andreas and Leibe, Bastian: "MOTS: Multi-Object Tracking and Segmentation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2019, pp. 7934–7943. doi: 10.1109/CVPR.2019.00813 (cit. on p. 27).

[Wan18a] Wan, Xingyu; Wang, Jinjun and Zhou, Sanping: “An Online and Flexible Multi-object Tracking Framework Using Long Short-Term Memory”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 1311–13118. doi: 10.1109/CVPRW.2018.00169 (cit. on p. 32).

[Wan18b] Wang, Guanshuo; Yuan, Yufeng; Chen, Xiong; Li, Jiwei and Zhou, Xi: “Learning Discriminative Features with Multiple Granularities for Person Re-Identification”. In: *Proceedings of the 26th ACM International Conference on Multimedia*. New York: Association for Computing Machinery, 2018, pp. 274–282. doi: 10.1145/3240508.3240552 (cit. on pp. 85, 86).

[Wan20] Wang, Zhongdao; Zheng, Liang; Liu, Yixuan; Li, Yali and Wang, Shengjin: “Towards Real-Time Multi-Object Tracking”. In: *Computer Vision – ECCV 2020*. Ed. by Vedaldi, Andrea; Bischof, Horst; Brox, Thomas and Frahm, Jan-Michael. Cham: Springer International Publishing, 2020, pp. 107–122. doi: 10.1007/978-3-030-58621-8_7 (cit. on pp. 4, 20, 21, 28, 79, 93, 97, 99, 144, 146, 167, 186, 189).

[Wan21] Wang, Yongxin; Kitani, Kris and Weng, Xinshuo: “Joint Object Detection and Multi-Object Tracking with Graph Neural Networks”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 13708–13715. doi: 10.1109/ICRA48506.2021.9561110 (cit. on pp. 21, 28, 30, 79).

[Wan22a] Wang, Shuai; Sheng, Hao; Yang, Da; Zhang, Yang; Wu, Yubin and Wang, Sizhe: “Extendable Multiple Nodes Recurrent Tracking Framework With RTU++”. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 5257–5271. doi: 10.1109/TIP.2022.3192706 (cit. on pp. 18, 169, 172).

[Wan22b] Wang, Shuai; Yang, Da; Wu, Yubin; Liu, Yang and Sheng, Hao: “Tracking Game: Self-adaptative Agent based Multi-object Tracking”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. New York: Association for Computing

Machinery, 2022, pp. 1964–1972. DOI: 10.1145/3503161.3548231 (cit. on pp. 169, 170, 172).

[Wan23a] WANG, Chien-Yao; BOCHKOVSKIY, Alexey and LIAO, Hong-Yuan Mark: "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2023, pp. 7464–7475. DOI: 10.1109/CVPR52729.2023.00721 (cit. on p. 71).

[Wan23b] WANG, Haidong; HE, Xuan; LI, Zhiyong; YUAN, Jin and LI, Shutao: "JDAN: Joint Detection and Association Network for Real-Time Online Multi-Object Tracking". In: *ACM Transactions on Multimedia Computing, Communications and Applications* 19.1s (2023). Article no. 45. DOI: 10.1145/3533253 (cit. on pp. 167, 189).

[Wen16] WEN, Longyin; LEI, Zhen; LYU, Siwei; LI, Stan Z. and YANG, Ming-Hsuan: "Exploiting Hierarchical Dense Structures on Hypergraphs for Multi-Object Tracking". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.10 (2016), pp. 1983–1996. DOI: 10.1109/TPAMI.2015.2509979 (cit. on p. 21).

[Woj17] WOJKE, Nicolai; BEWLEY, Alex and PAULUS, Dietrich: "Simple online and realtime tracking with a deep association metric". In: *2017 IEEE International Conference on Image Processing (ICIP).* 2017, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962 (cit. on pp. 3, 4, 18–21, 23, 25, 35, 36, 39, 69, 79, 81, 84, 92, 94, 99, 103, 115, 144, 145).

[Wu21] WU, Jialian; CAO, Jiale; SONG, Liangchen; WANG, Yu; YANG, Ming and YUAN, Junsong: "Track to Detect and Segment: An Online Multi-Object Tracker". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2021, pp. 12347–12356. DOI: 10.1109/CVPR46437.2021.01217 (cit. on pp. 28, 50, 65, 96).

[Xie20] Xie, Jin; Cholakkal, Hisham; Muhammad Anwer, Rao; Shahbaz Khan, Fahad; Pang, Yanwei; Shao, Ling and Shah, Mubarak: “Count- and Similarity-Aware R-CNN for Pedestrian Detection”. In: *Computer Vision – ECCV 2020.* Ed. by Vedaldi, Andrea; Bischof, Horst; Brox, Thomas and Frahm, Jan-Michael. Cham: Springer International Publishing, 2020, pp. 88–104. doi: 10.1007/978-3-030-58520-4_6 (cit. on p. 117).

[Xu20] Xu, Yihong; Ošep, Aljoša; Ban, Yutong; Horaud, Radu; Leal-Taixé, Laura and Alameda-Pineda, Xavier: “How to Train Your Deep Multi-Object Tracker”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020, pp. 6786–6795. doi: 10.1109/CVPR42600.2020.00682 (cit. on pp. 19, 21).

[Xu23] Xu, Yihong; Ban, Yutong; Delorme, Guillaume; Gan, Chuang; Rus, Daniela and Alameda-Pineda, Xavier: “TransCenter: Transformers With Dense Representations for Multiple-Object Tracking”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.6 (2023), pp. 7820–7835. doi: 10.1109/TPAMI.2022.3225078 (cit. on p. 31).

[Yan14] Yan, Junjie; Lei, Zhen; Wen, Longyin and Li, Stan Z.: “The Fastest Deformable Part Model for Object Detection”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition.* 2014, pp. 2497–2504. doi: 10.1109/CVPR.2014.320 (cit. on p. 21).

[Yan16] Yang, Fan; Choi, Wongun and Lin, Yuanqing: “Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 2129–2137. doi: 10.1109/CVPR.2016.234 (cit. on p. 167).

[Yan23] Yang, Fan; Odashima, Shigeyuki; Masui, Shoichi and Jiang, Shan: “Hard to Track Objects with Irregular Motions and Similar Appearances? Make It Easier by Buffering the Matching Space”. In: *2023 IEEE/CVF Winter Conference on Applications of Computer*

Vision (WACV). 2023, pp. 4788–4797. DOI: 10.1109/WACV56688.2023.00478 (cit. on pp. 22, 27, 143, 169, 170).

[Ye22] Ye, Mang; Shen, Jianbing; Lin, Gaojie; Xiang, Tao; Shao, Ling and Hoi, Steven C. H.: "Deep Learning for Person Re-Identification: A Survey and Outlook". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.6 (2022), pp. 2872–2893. DOI: 10.1109/TPAMI.2021.3054775 (cit. on p. 35).

[Yi24] Yi, Kefu; Luo, Kai; Luo, Xiaolei; Huang, Jiangui; Wu, Hao; Hu, Rongdong and Hao, Wei: "UCMCTrack: Multi-Object Tracking with Uniform Camera Motion Compensation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.7 (2024), pp. 6702–6710. DOI: 10.1609/AAAI.V38I7.28493 (cit. on pp. 20, 22, 23, 25, 169, 170, 172).

[You23] You, Sisi; Yao, Hantao; Bao, Bing-kun and Xu, Changsheng: "UTM: A Unified Multiple Object Tracking Model with Identity-Aware Feature Enhancement". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 21876–21886. DOI: 10.1109/CVPR52729.2023.02095 (cit. on pp. 28–30, 169, 170, 172, 204).

[Yu23] Yu, En; Li, Zhuoling; Han, Shoudong and Wang, Hongwei: "RelationTrack: Relation-Aware Multiple Object Tracking With Decoupled Representation". In: *IEEE Transactions on Multimedia* 25 (2023), pp. 2686–2697. DOI: 10.1109/TMM.2022.3150169 (cit. on pp. 28, 32, 189).

[Yur20] Yurtsever, Ekim; Lambert, Jacob; Carballo, Alexander and Takeda, Kazuya: "A Survey of Autonomous Driving: Common Practices and Emerging Technologies". In: *IEEE Access* 8 (2020), pp. 58443–58469. DOI: 10.1109/ACCESS.2020.2983149 (cit. on p. 1).

[Zag16] Zagoruyko, Sergey and Komodakis, Nikos: "Wide Residual Networks". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Richard C. Wilson, Edwin R. Hancock and Smith, William A. P. Article no. 87. BMVA Press, 2016. URL:

https://bmva-archive.org.uk/bmvc/2016/papers/paper087 (cit. on p. 25).

[Zen22] Zeng, Fangao; Dong, Bin; Zhang, Yuang; Wang, Tiancai; Zhang, Xiangyu and Wei, Yichen: “MOTR: End-to-End Multiple-Object Tracking with Transformer”. In: *Computer Vision – ECCV 2022*. Ed. by Avidan, Shai; Brostow, Gabriel; Cissé, Moustapha; Farinella, Giovanni Maria and Hassner, Tal. Cham: Springer Nature Switzerland, 2022, pp. 659–675. doi: 10.1007/978-3-031-19812-0_38 (cit. on pp. 20, 22, 31, 65, 96, 167).

[Zha08] Zhang, Li; Li, Yuan and Nevatia, Ramakant: “Global Data Association for Multi-Object Tracking Using Network Flows”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008. doi: 10.1109/CVPR.2008.4587584 (cit. on pp. 18, 19).

[Zha17] Zhang, Shanshan; Benenson, Rodrigo and Schiele, Bernt: “CityPersons: A Diverse Dataset for Pedestrian Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4457–4465. doi: 10.1109/CVPR.2017.474 (cit. on p. 54).

[Zha18] Zhang, Shifeng; Wen, Longyin; Bian, Xiao; Lei, Zhen and Li, Stan Z.: “Occlusion-Aware R-CNN: Detecting Pedestrians in a Crowd”. In: *Computer Vision – ECCV 2018*. Ed. by Ferrari, Vittorio; Hebert, Martial; Sminchisescu, Cristian and Weiss, Yair. Cham: Springer International Publishing, 2018, pp. 657–674. doi: 10.1007/978-3-030-01219-9_39 (cit. on p. 36).

[Zha19] Zhang, Wenwei; Zhou, Hui; Sun, Shuyang; Wang, Zhe; Shi, Jianping and Loy, Chen Change: “Robust Multi-Modality Multi-Object Tracking”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2365–2374. doi: 10.1109/ICCV.2019.00245 (cit. on p. 22).

[Zha21] Zhang, Yifu; Wang, Chunyu; Wang, Xinggang; Zeng, Wenjun and Liu, Wenyu: “FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking”. In: *International Journal of Computer Vision* 129.11 (2021), pp. 3069–3087. doi:

10.1007/S11263-021-01513-4 (cit. on pp. 18, 21, 28, 79, 146, 167, 189).

[Zha22a] Zhang, Hang; Wu, Chongruo; Zhang, Zhongyue; Zhu, Yi; Lin, Haibin; Zhang, Zhi; Sun, Yue; He, Tong; Mueller, Jonas; Manmatha, R.; Li, Mu and Smola, Alexander: "ResNeSt: Split-Attention Networks". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 2022, pp. 2735–2745. doi: 10.1109/CVPRW56347.2022.00309 (cit. on pp. 86, 190).

[Zha22b] Zhang, Tianhao; Aftab, Waqas; Mihaylova, Lyudmila; Langran-Wheeler, Christian; Rigby, Samuel; Fletcher, David I.; Maddock, Steve and Bosworth, Garry: "Recent Advances in Video Analytics for Rail Network Surveillance for Security, Trespass and Suicide Prevention - A Survey". In: *Sensors* 22.12 (2022). Article no. 4324. doi: 10.3390/S22124324 (cit. on p. 2).

[Zha22c] Zhang, Yifu; Sun, Peize; Jiang, Yi; Yu, Dongdong; Weng, Fucheng; Yuan, Zehuan; Luo, Ping; Liu, Wenyu and Wang, Xinggang: "ByteTrack: Multi-object Tracking by Associating Every Detection Box". In: *Computer Vision – ECCV 2022.* Ed. by Avidan, Shai; Brostow, Gabriel; Cissé, Moustapha; Farinella, Giovanni Maria and Hassner, Tal. Cham: Springer Nature Switzerland, 2022, pp. 1–21. doi: 10.1007/978-3-031-20047-2_1 (cit. on pp. 3, 18–20, 22, 23, 26, 27, 37, 39, 41, 44, 50, 65, 66, 69, 79, 94, 96, 97, 104, 111, 115, 119, 143, 158, 167–169, 171, 172, 186).

[Zha23] Zhang, Shilong; Wang, Xinjiang; Wang, Jiaqi; Pang, Jiangmiao; Lyu, Chengqi; Zhang, Wenwei; Luo, Ping and Chen, Kai: "Dense Distinct Query for End-to-End Object Detection". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2023, pp. 7329–7338. doi: 10.1109/CVPR52729.2023.00708 (cit. on p. 36).

[Zha24] Zhang, Yu; Chen, Huaming; Lai, Zhongzheng; Zhang, Zao and Yuan, Dong: "Handling Heavy Occlusion in Dense Crowd Tracking by Focusing on the Heads". In: *AI 2023: Advances in*

Artificial Intelligence. Ed. by Liu, Tongliang; Webb, Geoff; Yue, Lin and Wang, Dadong. Singapore: Springer Nature Singapore, 2024, pp. 79–90. doi: 10.1007/978-981-99-8388-9_7 (cit. on pp. 170, 172, 173, 204).

[Zhe20] Zheng, Zhaohui; Wang, Ping; Liu, Wei; Li, Jinze; Ye, Rongguang and Ren, Dongwei: “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.7 (2020), pp. 12993–13000. doi: 10.1609/AAAI.V34I07.6999 (cit. on p. 148).

[Zhe21] Zheng, Linyu; Tang, Ming; Chen, Yingying; Zhu, Guibo; Wang, Jinqiao and Lu, Hanqing: “Improving Multiple Object Tracking with Single Object Tracking”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2453–2462. doi: 10.1109/CVPR46437.2021.00248 (cit. on p. 35).

[Zhe22] Zheng, Anlin; Zhang, Yuang; Zhang, Xiangyu; Qi, Xiaojuan and Sun, Jian: “Progressive End-to-End Object Detection in Crowded Scenes”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 847–856. doi: 10.1109/CVPR52688.2022.00093 (cit. on p. 36).

[Zho19a] Zhou, Kaiyang; Yang, Yongxin; Cavallaro, Andrea and Xiang, Tao: “Omni-Scale Feature Learning for Person Re-Identification”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 3701–3711. doi: 10.1109/ICCV.2019.00380 (cit. on pp. 189, 190).

[Zho19b] Zhou, Xingyi; Wang, Dequan and Krähenbühl, Philipp: Objects as Points. 2019. arXiv: 1904.07850 (cit. on pp. 20, 21, 28, 31).

[Zho20] Zhou, Xingyi; Koltun, Vladlen and Krähenbühl, Philipp: “Tracking Objects as Points”. In: *Computer Vision – ECCV 2020*. Ed. by Vedaldi, Andrea; Bischof, Horst; Brox, Thomas and Frahm, Jan-Michael. Cham: Springer International Publishing, 2020, pp. 474–490. DOI: 10.1007/978-3-030-58548-8_28 (cit. on pp. 20, 25, 50, 65, 96, 143).

[Zhu21] Zhu, Xizhou; Su, Weijie; Lu, Lewei; Li, Bin; Wang, Xiaogang and Dai, Jifeng: "Deformable DETR: Deformable Transformers for End-to-End Object Detection". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=gZ9hCDWe6ke (cit. on p. 31).

[Zhu23] Zhu, Tianyu; Hiller, Markus; Ehsanpour, Mahsa; Ma, Rongkai; Drummond, Tom; Reid, Ian and Rezatofighi, Hamid: "Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.11 (2023), pp. 12783–12797. DOI: 10.1109/TPAMI.2022.3213073 (cit. on pp. 20, 31, 167).

[Zie12] Ziegler, Andrew; Christiansen, Eric; Kriegman, David and Belongie, Serge: "Locally Uniform Comparison Image Descriptor". In: *Advances in Neural Information Processing Systems*. Ed. by Pereira, F.; Burges, C. J.; Bottou, L. and Weinberger, K. Q. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/hash/c20ad4d76fe97759aa27a0c99bff6710-Abstract.html (cit. on p. 163).

Publications

[1] Du, Dawei; Wen, Longyin; Zhu, Pengfei; Fan, Heng; Hu, Qinghua; Ling, Haibin; Shah, Mubarak; Pan, Junwen; Axenopoulos, Apostolos; Schumann, Arne; Psaltis, Athanasios; Jain, Ayush; Dong, Bin; Li, Changlin; Chen, Chen; Duan, Chengzhen; Zhang, Chongyang; Stadler, Daniel et al.: “VisDrone-DET2020: The Vision Meets Drone Object Detection in Image Challenge Results”. In: *Computer Vision – ECCV 2020 Workshops*. Ed. by Bartoli, Adrien and Fusiello, Andrea. Cham: Springer International Publishing, 2020, pp. 692–712. doi: 10.1007/978-3-030-66823-5_42.

[2] Fan, Heng; Du, Dawei; Wen, Longyin; Zhu, Pengfei; Hu, Qinghua; Ling, Haibin; Shah, Mubarak; Pan, Junwen; Schumann, Arne; Dong, Bin; Stadler, Daniel et al.: “VisDrone-MOT2020: The Vision Meets Drone Multiple Object Tracking Challenge Results”. In: *Computer Vision – ECCV 2020 Workshops*. Ed. by Bartoli, Adrien and Fusiello, Andrea. Cham: Springer International Publishing, 2020, pp. 713–727. doi: 10.1007/978-3-030-66823-5_43.

[3] Stadler, Daniel; Sommer, Lars Wilko and Beyerer, Jürgen: “PAS Tracker: Position-, Appearance- and Size-Aware Multi-object Tracking in Drone Videos”. In: *Computer Vision – ECCV 2020 Workshops*. Ed. by Bartoli, Adrien and Fusiello, Andrea. Cham: Springer International Publishing, 2020, pp. 604–620. doi: 10.1007/978-3-030-66823-5_36.

[4] Specker, Andreas; Stadler, Daniel; Florin, Lucas and Beyerer, Jürgen: “An Occlusion-aware Multi-target Multi-camera Tracking System”. In: *2021 IEEE/CVF Conference on Computer Vision and*

Pattern Recognition Workshops (CVPRW). 2021, pp. 4168–4177. DOI: 10.1109/CVPRW53098.2021.00471.

[5] STADLER, Daniel: “Multi-Object Tracking in Drone Videos”. In: *Proceedings of the 2020 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Ed. by BEYERER, Jürgen and ZANDER, Tim. KIT Scientific Publishing, 2021, pp. 123–133. DOI: 10.5445/IR/1000135221.

[6] STADLER, Daniel and BEYERER, Jürgen: “Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 10953–10962. DOI: 10.1109/CVPR46437.2021.01081.

[7] STADLER, Daniel and BEYERER, Jürgen: “Multi-Pedestrian Tracking with Clusters”. In: *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2021. DOI: 10.1109/AVSS52988.2021.9663829.

[8] STADLER, Daniel and BEYERER, Jürgen: “On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking”. In: *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2021. DOI: 10.1109/AVSS52988.2021.9663836.

[9] STADLER, Daniel: “Multi-Person Tracking with a Multi-Hypothesis Approach for Ambiguous Assignments”. In: *Proceedings of the 2021 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Ed. by BEYERER, Jürgen and ZANDER, Tim. KIT Scientific Publishing, 2022, pp. 153–167. DOI: 10.5445/IR/1000148359.

[10] STADLER, Daniel and BEYERER, Jürgen: “Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds”. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2022, pp. 133–142. DOI: 10.1109/WACVW54805.2022.00019.

[11] Eisemann, Leon; Fehling-Kaschek, Mirjam; Gommel, Henrik; Hermann, David; Klemp, Marvin; Lauer, Martin; Lickert, Benjamin; Lüttner, Florian; Moss, Robin; Neis, Nicole; Pohle, Maria; Romanski, Simon; Stadler, Daniel; Stolz, Alexander; Ziehn, Jens and Zhou, Jingxing: "An Approach to Systematic Data Acquisition and Data-Driven Simulation for the Safety Testing of Automated Driving Functions". In: *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. 2023, pp. 2440–2447. doi: 10.1109/ITSC57777.2023.10422676.

[12] Stadler, Daniel: "A Detailed Study of the Association Task in Tracking-by-Detection-based Multi-Person Tracking". In: *Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Ed. by Beyerer, Jürgen and Zander, Tim. KIT Scientific Publishing, 2023, pp. 59–85. doi: 10.5445/IR/1000161972.

[13] Stadler, Daniel and Beyerer, Jürgen: "An Improved Association Pipeline for Multi-Person Tracking". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2023, pp. 3170–3179. doi: 10.1109/CVPRW59228.2023.00319.

[14] Stadler, Daniel and Beyerer, Jürgen: "BYTEv2: Associating More Detection Boxes Under Occlusion for Improved Multi-person Tracking". In: *Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges*. Ed. by Rousseau, Jean-Jacques and Kapralos, Bill. Cham: Springer Nature Switzerland, 2023, pp. 79–94. doi: 10.1007/978-3-031-37660-3_6.

[15] Stadler, Daniel and Beyerer, Jürgen: "Past Information Aggregation for Multi-Person Tracking". In: *2023 IEEE International Conference on Image Processing (ICIP)*. 2023, pp. 321–325. doi: 10.1109/ICIP49359.2023.10223159.

[16] Beyerer, Jürgen; Hagmanns, Raphael and Stadler, Daniel: Pattern Recognition: Introduction, Features, Classifiers and Principles, 2nd Edition. Berlin, Boston: De Gruyter Oldenbourg, 2024. doi: 10.1515/9783111339207.

[17] Eisemann, Leon; Fehling-Kaschek, Mirjam; Forkert, Silke; Forster, Andreas; Gommel, Henrik; Günther, Susanne; Hammer, Stephan; Hermann, David; Klemp, Marvin; Lickert, Benjamin; Lüttner, Florian; Moss, Robin; Neis, Nicole; Pohle, Maria; Schreiber, Dominik; Sowa, Cathrina; Stadler, Daniel; Stompe, Janina; Strobelt, Michael; Unger, David and Ziehn, Jens: A Joint Approach Towards Data-Driven Virtual Testing for Automated Driving: The AVEAS Project. 2024. arXiv: 2405.06286.

[18] Kiefer, Benjamin; Žust, Lojze; Kristan, Matej; Perš, Janez; Teršek, Matija; Wiliem, Arnold; Messmer, Martin; Yang, Cheng-Yen; Huang, Hsiang-Wei; Jiang, Zhongyu; Kuo, Heng-Cheng; Mei, Jie; Hwang, Jenq-Neng; Stadler, Daniel et al.: “2nd Workshop on Maritime Computer Vision (MaCVi) 2024: Challenge Results”. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2024, pp. 869–891. doi: 10.1109/WACVW60836.2024.00099.

List of Figures

List of Tables

Acronyms

AssA	association accuracy
AssPr	association precision
AssRe	association recall
BN	batch normalization
BoT	bag of tricks
BRIEF	binary robust independent elementary features
CH	CrowdHuman
CMC	camera motion compensation
CNN	convolutional neural network
CPU	central processing unit
DetA	detection accuracy
DetPr	detection precision
DetRe	detection recall
DETR	detection transformer

DIoU	distance IoU
ECC	enhanced correlation coefficient
EMA	exponential moving average
FAST	features from accelerated segment test
FC	fully-connected
FN	false negative
FNA	false negative association
FP	false positive
FPA	false positive association
FPS	frames per second
GAP	global average pooling
GIoU	generalized IoU
GNN	graph neural network
GPU	graphics processing unit
GSI	Gaussian smoothed interpolation
GT	ground truth
HOTA	higher order tracking accuracy
HP	height preservation
IDF1	identity F1

IDFN	identity false negative
IDFP	identity false positive
IDPr	identity precision
IDRe	identity recall
IDSW	identity switch
IDTP	identity true positive
IoU	intersection over union
JDE	joint detection and embedding
JV	Jonker–Volgenant
LI	linear interpolation
LSA	linear sum assignment
MOT	multiple object tracking
MOTA	multiple object tracking accuracy
MPN	message passing network
MPT	multi-person tracking
NMS	non-maximum suppression
NSA	noise scale adaptive
OAI	occlusion-aware initialization
OOM	out of memory

ORB	oriented FAST and rotated BRIEF
OSNet	omni-scale network
PP22	PersonPath22
PTZ	pan–tilt–zoom
RANSAC	random sample consensus
REID	re-identification
RNN	recurrent neural network
SBS	stronger baseline
SDE	separate detection and embedding
SORT	simple online and real-time tracking
SOT	single object tracking
SOTA	state of the art
TBA	tracking-by-attention
TBD	tracking-by-detection
TBR	tracking-by-regression
TP	true positive
TPA	true positive association
TWC	tracking with clusters
YOLO	you only look once

Symbols

a	aspect ratio of bounding box
A	bounding box area
α	position parameter in Kalman filter covariance matrices
B	batch size
β	velocity parameter in Kalman filter covariance matrices
$\mathbf{b}$	bounding box
$\mathbf{B}$	control matrix in Kalman filter system equation
c	number of object categories
c_f	number of channels in feature volume
$\mathcal{C}$	track cluster
$\widetilde{\mathcal{C}}$	track cluster without detections
$\boldsymbol{\mathfrak{C}}$	set of track clusters
$\widetilde{\boldsymbol{\mathfrak{C}}}$	set of track clusters without detections
d	distance

d_{app}	appearance distance
$\tilde{d}_{\text{app}}$	scaled appearance distance
d_{BoT}	combined distance from BoT-SORT
d_{comb}	proposed combined distance
$\tilde{d}_{\text{comb}}$	combined distance with additional overlap requirement
$d_{\text{comb,IoU}}$	combined distance with IoU for motion information
$d_{\text{comb,DIoU}}$	combined distance with DIoU for motion information
$d_{\text{comb,GIoU}}$	combined distance with GIoU for motion information
d_{cos}	cosine distance
d_C	diagonal length of box C
d_{DIoU}	DIoU distance
d_{DS}	combined distance from DeepSORT
d_{EMA}	distance used together with EMA feature update
d_{GIoU}	GIoU distance
d_{IoU}	IoU distance
d_{JDE}	combined distance from JDE
d_{L2}	Euclidean distance
d_{max}	maximum distance threshold
d_{mean}	mean distance

d_{min}	minimum distance
d_{mot}	motion distance
d_{Mah}	squared Mahalanobis distance
D	detection
D_{GT}	GT detection
$\mathrm{D}_{\mathrm{pred}}$	predicted detection
$\mathrm{D}_{\mathrm{norm}}$	normal detection
$\mathrm{D}_{\mathrm{occ}}$	occluded detection
D^{u}	unassigned detection
δ	Kronecker delta
$\mathbf{d}$	detection in Kalman filter measurement space
$\mathbf{D}$	distance matrix
$\mathcal{D}$	set of detections
$\widetilde{\mathcal{D}}$	set of raw detections
$\mathcal{D}_{\mathrm{high}}$	set of high-confidence detections
$\mathcal{D}_{\mathrm{init}}$	set of detections for track initialization
$\mathcal{D}_{\mathrm{low}}$	set of low-confidence detections
$\mathcal{D}^{\mathrm{norm}}$	set of normal detections
$\mathcal{D}_{\mathrm{occ}}$	set of occluded detections

$\widetilde{\mathcal{D}}_{\text{occ}}$	set of confident occluded detections
$\mathcal{D}^{\text{u}}$	set of unassigned detections
$\widetilde{\mathcal{D}^{\text{u}}}$	preliminary set of unassigned detections
ϵ	small constant
η	localization threshold
f	frame number of video
F	list of feature vectors, feature bank
$\mathbf{f}$	feature vector
$\mathbf{f}_{\text{a}}$	anchor feature vector in triplet loss
$\mathbf{f}_{\text{ID}}$	feature vector on which identity loss is applied
$\mathbf{f}_{\text{KP}}$	feature vector of keypoint
$\mathbf{f}_{\text{n}}$	negative feature vector in triplet loss
$\mathbf{f}_{\text{p}}$	positive feature vector in triplet loss
$\mathbf{f}_{\text{triplet}}$	feature vector on which triplet loss is applied
$\mathbf{F}$	transition matrix in Kalman filter system equation
$\mathcal{F}$	set of visual descriptors
g	GT label
γ	margin of triplet loss
h	height of bounding box

h_f	height of feature volume
$\mathbf{H}$	observation matrix in Kalman filter observation equation
i	time period a track has been inactive
i_max	inactive patience
$\mathbf{I}$	image
$\mathbf{I}_n$	identity matrix
$\mathbf{I}_\mathrm{reg}$	image region
K	number of different images per person
$\varkappa$	large constant
$\mathbf{K}$	Kalman gain matrix of Kalman filter update step
l	video length, number of video frames
l_f	length of feature vector
L	list of detections
λ	weighting factor in combined distance functions
$\mathcal{L}_\mathrm{ID}$	identity loss
$\mathcal{L}_\mathrm{triplet}$	triplet loss
n_D	number of detections
n_init	number of consecutive detections for track initialization
n_T	number of tracks

N_F	size of feature bank
N_I	number of image pixels
N_ID	number of person identities
N_p	number of transformation parameters
$\mathcal{N}$	normal distribution
o	overlap (IoU)
o_cluster	overlap threshold of TWC
o_init	overlap threshold in the OAI
o_NMS	overlap threshold of the standard NMS
o_NMS1	first overlap threshold of the adapted NMS
o_NMS2	second overlap threshold of the adapted NMS
p	output of FC layer
$\tilde{p}$	normalized output of FC layer
P	number of different persons
ϕ	weighting factor in EMA update
$\mathbf{p}$	keypoint on image
$\mathbf{P}$	covariance matrix of Kalman filter state
$\mathcal{P}$	set of keypoints
q	scale parameter in transformation matrix

$\mathbf{Q}$	covariance matrix of Kalman filter process noise
$\mathbf{R}$	covariance matrix of Kalman filter measurement noise
s	detection confidence
s_{init}	confidence threshold for track initialization
s_{IoU}	IoU regression score
s_{min}	minimum confidence of detections used in tracking
s_{occ}	min. confidence of occluded detections used in tracking
s_{person}	person classification score
s_{track}	track confidence threshold
$\mathbf{S}$	projected covariance matrix in measurement space
t	time index
t_x	x-translation parameter in transformation matrix
t_y	y-translation parameter in transformation matrix
T	track
T^{a}	active track
T_{GT}	GT track
T^{i}	inactive track
T^{n}	new track
$\mathrm{T}_{\mathrm{pred}}$	predicted track

θ	rotation parameter in transformation matrix
$\mathcal{T}$	set of tracks
$\widetilde{\mathcal{T}}$	set of propagated/updated tracks
$\mathcal{T}^{\mathrm{a}}$	set of active tracks
$\mathcal{T}_{\mathrm{GT}}$	set of GT tracks
$\mathcal{T}^{\mathrm{i}}$	set of inactive tracks
$\mathcal{T}^{\mathrm{m}}$	set of matched tracks
$\mathcal{T}^{\mathrm{n}}$	set of new tracks
$\mathcal{T}_{\mathrm{pred}}$	set of predicted tracks
$\mathcal{T}^{\mathrm{u}}$	set of unassigned tracks
$\mathbf{u}$	control vector in Kalman filter system equation
V	video, sequence of images
$\mathbf{v}$	measurement noise in Kalman filter observation equation
w	width of bounding box
w_{f}	width of feature volume
$\mathbf{w}$	process noise in Kalman filter system equation
$\mathbf{W}$	transformation matrix for CMC
x	x-coordinate in image
$\mathbf{x}$	Kalman filter state

y	y-coordinate in image
$\mathbf{y}$	projected mean in measurement space
$\tilde{\mathbf{y}}$	innovation in Kalman filter update step
$\mathbf{z}$	measurement in Kalman filter observation equation