



UNIVERSITY OF NIŠ
FACULTY OF SCIENCES AND
MATHEMATICS



Vukašin Stanojević

ADVANCED MULTIPLE OBJECT TRACKING ALGORITHMS

DOCTORAL DISSERTATION

Niš, 2026.



UNIVERZITET U NIŠU
PRIRODNO-MATEMATIČKI FAKULTET



Vukašin Stanojević

NAPREDNI ALGORITMI PRAĆENJA VEĆEG BROJA OBJEKATA

DOKTORSKA DISERTACIJA

Niš, 2026.

Data on Doctoral Dissertation

Doctoral Supervisor:	Branimir Todorović, Ph.D., associate professor, Faculty of Sciences and Mathematics, University of Niš
Title:	Advanced Multiple Object Tracking Algorithms
Abstract:	The dissertation studies online multiple object tracking (MOT) in the tracking-by-detection paradigm, focusing on the interaction between detection, motion estimation, data association, and tracklet management. Its objective is to develop tracking algorithms that are accurate, robust, adaptive, simple, and suitable for real-time use. It investigates how unreliable detections and uncertain motion predictions can be used more effectively. Detection-tracklet confidence scores are introduced to implicitly favor reliable detection-tracklet pairs in one-stage association. Additional mechanisms use tracklet information to select true-positive low-confidence detections and improve association through lightweight similarity cues. The dissertation further proposes a strictly online, motion-based framework that couples adaptive motion estimation, crowd-aware uncertainty modeling, uncertainty-guided bounding-box expansion, one-stage association, and tracklet management. Instead of relying on appearance features, post-processing, or complex association pipelines, this framework propagates uncertainty and reliability information between simple tracking components. The final framework achieves state-of-the-art results within its category and remains comparable to, or outperforms, more complex online appearance-based and offline motion-based methods on most evaluated benchmarks. Overall, the dissertation shows that robust online MOT can be achieved by carefully coupling simple components.
Scientific Field:	Computer science
Scientific Discipline:	Artificial intelligence
Key Words:	Multiple object tracking, tracking-by-detection, video analysis, visual tracking, motion-based tracking, data association, Kalman filtering, real-time tracking
UDC:	004.932:004.8(043)
CERIF Classification:	P 176 Artificial intelligence
Creative Commons License:	CC BY-NC-ND

Подаци о докторској дисертацији

Ментор: Др Бранимир Тодоровић, ванредни професор,
Природно-математички факултет, Универзитет у Нишу

Наслов: Напредни алгоритми праћења већег броја објеката

Резиме: Ова дисертација проучава онлајн праћење већег броја објеката у оквиру парадигме праћења заснованог на детекцијама, са фокусом на интеракцију између детекције, естимације кретања, придруживања података и управљања праћеним објектима. Циљ је развој алгоритама за праћење који су тачни, робусни, адаптивни, једноставни и погодни за рад у реалном времену. У дисертацији се истражује како се непоуздане детекције и неизвесне предикције кретања могу ефикасније користити. Уводе се мере поузданости које током придруживања података имплицитно фаворизују поуздане парове детекција и праћених објеката. Додатни механизми користе информације из трајекторија за избор тачно позитивних детекција ниске поузданости и побољшано преидруживање података помоћу једноставних мера сличности. У дисертацији се затим предлаже онлајн систем заснован на информацијама о кретању, који повезује адаптивну процену кретања, моделирање неизвесности у зависности од густине сцене, проширење граничних оквира на основу неизвесности, придруживање података у једној фази и управљање праћеним објектима. Уместо ослањања на визуелне карактеристике, постпроцесирање или сложене процедуре придруживања података, једноставне компоненте система деле информације о неизвесности и поузданости. Последњи метод у дисертацији постиже резултате на нивоу најбољих савремених метода у својој категорији и остаје упоредив са сложенијим онлајн методама заснованим на визуелном изгледу и офлајн методама заснованим на кретању, или их надмашује, на већини евалуираних скупова података. У целини, дисертација показује да се робусно онлајн праћење више објеката може постићи пажљивим повезивањем једноставних компоненти.

Научна област: Рачунарске науке

Научна дисциплина: Вештачка интелигенција

Кључне речи:	Праћење већег броја објеката, праћење засновано на детекцијама, анализа видеа, визуелно праћење, праћење засновано на кретању, придруживање података, Калманов филтер, праћење у реалном времену
УДК:	004.932:004.8(043)
CERIF класификација:	P 176 Вештачка интелигенција
Тип лиценце Креативне заједнице:	CC BY-NC-ND

Preface

Artificial intelligence has advanced rapidly in recent years and has become one of the most transformative areas of modern science and technology. Intelligent systems already influence many aspects of everyday life and have the potential to improve them further. Autonomous vehicles are becoming increasingly capable of understanding road scenes and planning safe movement through complex traffic. Robots are gradually gaining the ability to interact with people and operate in human environments. Drones and mobile agents can assist in inspection, delivery, rescue, and monitoring tasks. Intelligent video analysis systems can help observe public spaces, traffic flow, animal behavior, ecosystems, or sports events. In many of these applications, a system must understand not only what is present in the scene, but also how different objects move over time.

This is where multiple object tracking becomes an essential component. In order to act safely and effectively, an intelligent system must often locate several objects at once, distinguish them from one another, and preserve their identities over time. A self-driving car must reason about pedestrians, cyclists, vehicles, and other traffic participants, and be able to predict their movement. A robot moving among people must be aware of nearby humans and their motion. A drone or a surveillance system may need to follow several objects simultaneously while dealing with occlusions and changing scene conditions. In such cases, accurate tracking of all objects in the scene is a prerequisite for understanding dynamic scenes.

However, for these systems to be useful in the real world, accuracy alone is not sufficient. Tracking algorithms must also be fast enough to operate in real time, and robust enough to handle difficult and ambiguous scenes. In many applications, decisions must be made online, using only the current and previous observations, without access to future frames. These constraints make multiple object tracking a challenging and important research problem.

This dissertation, entitled *Advanced Multiple Object Tracking Algorithms*, contributes to the development of online multiple object tracking methods. The work presented here is motivated by the belief that tracking systems should be accurate, efficient, and practically applicable. The proposed methods are designed with the intention of improving real-time

tracking performance while avoiding unnecessary complexity whenever possible. More broadly, this research is guided by the hope that such algorithms will be used responsibly and for beneficial purposes: to improve safety, assist people, support scientific observation, and contribute to technologies that serve society.

The dissertation is organized into seven chapters. Chapter 1 introduces the multiple object tracking problem, motivates the research topic, and presents the main goals and contributions of the dissertation. It explains the tracking-by-detection paradigm and outlines the central idea followed throughout the dissertation: improving MOT by strengthening the interaction between detection, motion estimation, data association, and tracklet management.

Chapter 2 provides the necessary background for the rest of the dissertation. It introduces the standard tracking-by-detection workflow, object detection, Kalman-filter-based motion modeling, data association, tracklet management, and post-processing. It also presents the benchmark datasets and evaluation metrics used in the experimental analysis.

Chapter 3 presents BoostTrack, a method that improves one-stage multiple object tracking through detection confidence boosting and similarity boosting. This chapter introduces the first elements of cooperation between tracking components and shows that even lightweight components, when properly coupled, can reduce the need for computationally heavy additions such as ReID modules or complex association pipelines. This chapter is based on the following publication:

- Vukašin D. Stanojević and Branimir T. Todorović. BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking. *Machine Vision and Applications*, 35(3):53, 2024.

Chapter 4 presents BoostTrack++¹, which further investigates the use of tracklet information for recovering true-positive low-confidence detections. This chapter introduces more refined mechanisms for identifying true-positive low-confidence detections and provides additional insight into how unreliable observations can be used rather than discarded. In this way, BoostTrack++ further develops the idea that tracking information can support detection handling and improve association in challenging scenes. The chapter is based on the following publication:

- Vukašin Stanojević and Branimir Todorović. BoostTrack++: using tracklet information to detect more objects in multiple object tracking. *Filomat*, 39(16):5685–5702, 2025.

¹The source code for BoostTrack and BoostTrack++ is available at <https://github.com/vukasin-stanojevic/BoostTrack>.

Chapter 5 presents Crowd-Aware Tracker (CAT)², a strictly online, motion-based, and real-time tracking framework based on coupled motion estimates. CAT connects adaptive motion estimation, uncertainty-aware similarity computation, one-stage association, tracklet creation, and trajectory updates into a unified framework. This chapter represents the most complete realization of the main research direction of the dissertation: improving online MOT through cooperation between simple tracking components. The chapter is based on the following manuscript:

- Vukašin Stanojević. Towards Simple, Robust, and Real-Time Multiple Object Tracking with Coupled Motion Estimates. Manuscript under review in *International Journal of Computer Vision*.

Chapter 6 provides a general discussion of the proposed methods, followed by the main conclusions of the dissertation. It analyzes the evolution from BoostTrack to BoostTrack++ and CAT, discusses the common principles behind the methods, and explains the role of component coupling in tracking-by-detection. The chapter also discusses the limitations of the proposed approaches and outlines several directions in which the presented research can be further developed.

I would like to express my gratitude to the professors and teaching staff of the Department of Computer Science at the Faculty of Sciences and Mathematics, University of Niš, for the knowledge, support, and guidance they provided throughout my studies. I am especially grateful to my supervisor, Prof. Dr. Branimir Todorović, for his advice, encouragement, and support during the development of this research.

I would also like to thank my girlfriend, Mina Radiivojević, for her patience, understanding, and support during the years of research and writing.

²The source code for CAT is available at <https://github.com/vukasin-stanojevic/CAT>.

Contents

1	Introduction	1
1.1	Tracking-by-Detection Paradigm	1
1.1.1	Interdependence of Components in TbD MOT	2
1.2	Motivation of the Dissertation	2
1.3	Main Contributions	4
2	Background	7
2.1	MOT Problem Formulation and Notation	7
2.2	Object Detection	9
2.3	Kalman Filter for MOT	10
2.4	Data Association	11
2.5	Tracking-by-Detection Workflow and Tracklet Management	12
2.6	Post-processing	13
2.7	Classification of MOT methods	15
2.8	Evaluation Metrics	16
2.9	Benchmark Datasets	18
3	BoostTrack: Confidence and Similarity Boosting for One-Stage Multiple Object Tracking	21
3.1	Introduction	21
3.2	Related Work	24
3.2.1	Working with Unreliable Detections	24
3.2.2	Tracklet Confidence	24
3.2.3	Similarity Measures	25
3.2.4	Relation to the Proposed Method	25
3.3	Similarity Matrix Boosting Techniques	26
3.3.1	Detection-Tracklet Confidence Similarity Boost	27
3.3.2	Mahalanobis Distance Similarity Boost	28
3.3.3	Shape Similarity Boost	29

3.4	Detection Confidence Boosting Techniques	30
3.4.1	Detecting Likely Objects	30
3.4.2	Detecting “Unlikely” Objects	31
3.5	Experiments	32
3.5.1	Experimental Setup	32
3.5.2	Implementation Details	33
3.5.3	Ablation Study	35
3.5.4	Comparison with Contemporary Benchmark Methods	37
3.6	Conclusion	39
4	BoostTrack++: Tracklet-Guided Recovery of True-Positive Detections	41
4.1	Introduction	41
4.2	Limitations of BoostTrack	43
4.3	Proposed Method	46
4.3.1	Soft BIoU	46
4.3.2	Using an Improved Similarity Measure to Find Likely Objects	47
4.3.3	Soft Detection Confidence Boosting	48
4.3.4	Varying Similarity Threshold	49
4.3.5	Combining the Confidence Boost Components	51
4.3.6	KF-Based Trajectory Splitting as Post-Processing	51
4.4	Experiments and Results	54
4.4.1	Experimental Setup and Implementation Details	54
4.4.2	Ablation Study	55
4.4.3	Comparison with Contemporary Benchmark Methods	58
4.5	Conclusion	60
5	Crowd-Aware Tracker: Coupled Motion Estimates for Robust Real-Time Multiple Object Tracking	61
5.1	Introduction	61
5.2	Related Work	64
5.2.1	Kalman Filter for MOT	64
5.2.2	Expanding Bounding Boxes	64
5.2.3	Working with Low-Confidence Detections	65
5.3	Method	65
5.3.1	Adaptive Crowd-Aware Kalman Filter	66
5.3.2	Expanded IoU	68
5.3.3	Association Workflow	70
5.4	Experiments	77

5.4.1	Datasets and Metrics	77
5.4.2	Implementation Details	77
5.4.3	Baseline Setting	78
5.4.4	Comparison with State-of-the-Art Methods	79
5.4.5	Ablation Study	85
5.4.6	Component Analysis	86
5.5	Discussion	97
5.6	Conclusion	98
6	Discussion, Conclusion and Future Work	99
6.1	Evolution of the Proposed Methods	99
6.2	Towards an Ideal Online MOT Algorithm	100
6.3	Limitations	101
6.4	Conclusion	102
6.5	Future Work	103
	Bibliography	105
A	Additional Results for BoostTrack	121
A.1	Calculating Mahalanobis Distance Similarity	121
A.2	NMS Threshold	121
A.3	Influence of Detection-Tracklet Confidence Hyperparameters	123
B	Additional Results for BoostTrack++	125
B.1	Influence of Different Similarity Measures	125
B.2	Ablation Study on SORT and BoostTrack	126
C	Additional Results for CAT	127
C.1	Efficient Kalman Filter Computation	127
C.2	Ablation Study on MOT17 and DanceTrack	129
C.3	Additional adaptive CAKF Experiments	129
C.3.1	Additional Ablation Results	129
C.3.2	Estimated Covariance Matrices	131
C.4	Additional EIoU Experiments	133
C.5	Additional Association Workflow Analysis	135
C.5.1	KF-Based Tracklet Confidence	135
C.5.2	Association and Trajectory Initialization Criteria	136
C.5.3	Trajectory Update Criterion	137

Biography	141
Thesis Documentation	143

List of Figures

2.1	Example frames from the benchmark datasets	20
3.1	BoostTrack: method overview	23
3.2	BoostTrack: detecting likely objects using IoU	31
3.3	BoostTrack: detecting “unlikely” objects using Mahalanobis distance	32
4.1	IoU statistics after missed updates	45
4.2	BoostTrack++: confidence boosting using aggregate similarity	48
4.3	BoostTrack++: soft confidence boosting	50
4.4	BoostTrack++: confidence boosting with varying similarity threshold	51
4.5	BoostTrack++: trajectory splitting after occlusion	53
4.6	BoostTrack++: method comparison on MOT17 and MOT20	58
5.1	CAT: method overview	63
5.2	CAT: association ambiguity examples	66
5.3	Adaptive Crowd-Aware Kalman Filter overview	67
5.4	Expanded IoU calculation	69
5.5	Difference between IoU and EIoU	71
5.6	CAT: low-confidence true-positive detection	73
5.7	CATC criterion illustration	74
5.8	CAT: comparison across MOT benchmarks	84
5.9	Adaptive CAKF: per-sequence improvements on MOT20	87
5.10	EIoU scaling factors on MOT20 and SportsMOT	89
5.11	KFTC over time on MOT20 and SportsMOT	91
5.12	True-positive detection confidence on MOT20 and SportsMOT	93
5.13	CATC and AVC heatmap ablation	94
C.1	Adaptive CAKF: covariance distribution histograms	132
C.2	EIoU scaling factors on MOT17 and DanceTrack	134
C.3	True-positive detection confidence on MOT17 and DanceTrack	136
C.4	CATC and AVC grid search on all validation sets	137

List of Tables

3.1	BoostTrack: similarity boosting ablation on MOT17 and MOT20	35
3.2	BoostTrack: detection confidence boosting ablation on MOT17 and MOT20	36
3.3	BoostTrack: component ablation on MOT17 and MOT20	37
3.4	BoostTrack: comparison on MOT17 and MOT20	38
4.1	BoostTrack++: effect of DLO on IDs and IDSW	44
4.2	BoostTrack++: DLO similarity measures on MOT20	56
4.3	BoostTrack++: component ablation on MOT17 and MOT20	57
4.4	BoostTrack++: comparison on MOT17 and MOT20	59
5.1	CAT: dataset-specific hyperparameters	78
5.2	CAT: comparison on MOT20	80
5.3	CAT: comparison on SportsMOT	81
5.4	CAT: comparison on MOT17	82
5.5	CAT: comparison on DanceTrack	83
5.6	CAT: runtime comparison on MOT20	85
5.7	CAT: component ablation on MOT20 and SportsMOT	85
5.8	Adaptive CAKF: ablation on MOT20 and SportsMOT	87
5.9	EIoU: maximum scaling factor on SportsMOT	88
5.10	EIoU ablation on MOT20 and SportsMOT	90
5.11	KFTC ablation on MOT20 and SportsMOT	92
5.12	CATC and AVC: detection threshold ablation on MOT20	94
5.13	TUC: update threshold on MOT20 and SportsMOT	96
5.14	TUC compared with linear interpolation	96
A.1	BoostTrack: NMS threshold on MOT17	122
A.2	BoostTrack: tracklet confidence hyperparameters on MOT17	123
B.1	BoostTrack++: DLO similarity measures with BoostTrack baseline	125
B.2	BoostTrack++: component ablation with SORT and BoostTrack baselines	126

C.1	CAT: component ablation on MOT17 and DanceTrack	130
C.2	Adaptive CAKF: ablation on MOT17 and DanceTrack	130
C.3	Adaptive CAKF: covariance distribution divergence	133
C.4	EIoU ablation on MOT17 and DanceTrack	135
C.5	KFTC ablation on MOT17 and DanceTrack	135
C.6	TUC: update threshold on MOT17 and DanceTrack	138
C.7	TUC compared with linear interpolation on MOT17 and DanceTrack . . .	139

Chapter 1

Introduction

Multiple object tracking (MOT) is one of the fundamental open problems in computer vision. Its goal is to locate and identify every object of interest, such as a pedestrian, in each video frame while preserving its identity throughout the sequence. An MOT algorithm therefore produces a set of trajectories, each describing the position and size of an individual object over time.

Applications of MOT include autonomous driving [21, 108, 91, 42, 107], video surveillance [31, 120, 4], animal tracking [74, 143, 87], traffic analysis [83, 50], smart cities [107, 152, 13, 92], sports analytics [117, 102], robotics [35, 94], and behavior analysis [22, 6].

A tracker must account for temporary occlusions, missed and inaccurate detections, objects entering or leaving the observed scene, interactions among nearby objects, changes in object appearance, and object motion. Therefore, MOT methods are commonly evaluated not only according to their ability to locate objects, but also according to the accuracy and consistency of the resulting trajectories. An effective tracker must balance several potentially conflicting objectives: detection accuracy, identity preservation, robustness to uncertainty, computational efficiency, and applicability across different scenes and object dynamics.

1.1 Tracking-by-Detection Paradigm

Among the various proposed MOT methods, *tracking-by-detection* (TbD) methods have become a dominant choice, as they often provide a strong trade-off between accuracy and speed. Furthermore, TbD methods are modular and online by design (offline post-processing is a standard practice, but optional).

In TbD MOT, at every frame, the detected bounding boxes are matched with the currently tracked objects, computing the complete trajectory of each target over time across

the video sequence.

A typical online tracking-by-detection method consists of several components. A detector provides bounding boxes for the detected objects. A state estimation model, or *tracking module*, predicts the expected location and size of each tracked object (commonly referred to as tracklet) in the current frame. A similarity or association cost function evaluates the compatibility between tracklets and newly obtained detections. An assignment algorithm then determines the most likely correspondences. Finally, track management rules control tracklet initialization, update, and termination.

Some of the challenges that all MOT algorithms, TbD algorithms included, need to overcome include irregular motion [15, 118, 28], crowded scenes [111, 79, 119], occlusions [125, 16, 46], camera motion [137, 56, 44] and different filming settings [33, 150, 151], low visibility [128, 136, 57] and indistinguishable visual appearances [139, 104, 61]. However, these challenges only increase the difficulty of core tasks in TbD pipeline - namely, detection, tracking and data association.

1.1.1 Interdependence of Components in TbD MOT

All three tasks - detection, tracking, and association - are interconnected, and failure in any can lead to subsequent errors and degrade entire tracking performance.

The location information estimated by the tracking module provides the basis for extracting both motion and visual features (used in the association phase). Even with the perfect detector, the most sophisticated ReID module, motion features, and occlusion handling, inaccurate object state predictions can cause association errors and further degrade the performance of the entire feature extraction and association pipeline.

On the other hand, unlike in other tracking tasks, in MOT, tracking failure can occur even with a perfect motion model and noise-free observations due to association errors, which make tracking in MOT particularly challenging. We need accurate state predictions for better feature extraction and successful association, and we need successful association for more accurate predictions.

Furthermore, detector failures can also cause tracking and association errors by inducing false positive or inaccurate observations, or by failing to detect objects (false negative observations). Therefore, when the detector is imperfect, MOT also requires careful selection of reliable detections and effective handling of uncertain ones.

1.2 Motivation of the Dissertation

Many existing MOT methods address a specific failure case and focus on improving a particular part of the tracking pipeline. These directions have led to important improve-

ments, but they often treat detection, tracking, and association as separate components. As a result, the information produced by one component is not always fully exploited by the others, even though errors in MOT typically arise from their interaction.

The motivation behind this dissertation is to study MOT from a broader perspective. Instead of addressing only isolated difficulties, the goal is to investigate how an online MOT algorithm should be designed in order to be accurate, adaptive, simple, robust, and fast enough for real-time use. An ideal MOT algorithm should preserve object identities through occlusions and missed detections, avoid unnecessary identity switches, use true-positive detections even when their confidence score is low, reject unreliable detections, adapt to different scenes and motion patterns, and operate online without relying on future frames. At the same time, it should avoid unnecessary computational complexity and excessive dependence on heavy appearance models or large tracking architectures.

These requirements are often conflicting. Using more low-confidence detections can help recover missed objects, but may also introduce false positives and identity switches. Stronger appearance models can improve association but increase runtime and reduce real-time applicability. While offline post-processing can reconstruct fragmented trajectories, it violates the strict online processing requirements of many real-time applications. More complex association pipelines can recover difficult matches but may introduce additional thresholds, heuristics, and error propagation. Similarly, simple motion models are efficient, but their predictions may become unreliable in crowded scenes, under occlusion, or during irregular motion.

The central idea of this dissertation is that these trade-offs can be addressed not only by adding more complex components, but also by making better use of the relationships between existing ones. The identification and use of true-positive detections, tracklet information, motion uncertainty, similarity computation, and association should not be treated independently. Instead, they can provide useful information to one another. Tracklet information can help choose true-positive detections. Detection confidence can influence association. Motion uncertainty can guide similarity computation and trajectory updates. Association ambiguity can be used to adjust how much the tracker trusts new observations.

Following this idea, the dissertation investigates how the interdependence between detection, state estimation, and data association can be used as an advantage rather than treated only as a source of error propagation. This direction was reached gradually through the development of the methods presented in Chapters 3 to 5.

From this perspective, BoostTrack, presented in Chapter 3, represents the first step toward improving communication between the main components of the tracking pipeline. It uses tracklet information to improve the selection of true-positive detections, incorporates detection and tracklet confidence into association, and constructs a richer similarity

measure from lightweight components. In this way, BoostTrack shows that carefully coupled confidence and similarity cues can improve tracking performance while reducing the need for computationally heavy components, such as ReID modules, or more complex association pipelines.

BoostTrack++, presented in Chapter 4, further develops the idea that tracking information can support detection handling. The method introduces more refined mechanisms for identifying true-positive low-confidence detections by combining information from existing tracklets, motion-based similarity, shape consistency, and adaptive confidence boosting. It provides additional insight into how unreliable observations can be used in tracking-by-detection systems rather than removed before association. The chapter also extends this idea to post-processing through Kalman-filter-based trajectory splitting, where tracking predictions are used to decide whether fragmented trajectory parts should be interpolated or separated.

The method presented in Chapter 5 represents the culmination of this research direction. There, complementarity, adaptivity, robustness, and simplicity are explicitly used as design principles for an efficient real-time online tracker. Instead of relying primarily on computationally expensive ReID modules, large models, or increasingly complex multi-stage association procedures, the method investigates how far simple online tracking-by-detection systems can be improved by strengthening the communication between their components. The objective is to achieve robust and efficient tracking by allowing motion uncertainty, similarity computation, association decisions and tracklet management to support one another, while preserving the simplicity and speed required for practical real-time applications.

1.3 Main Contributions

The main contribution of this dissertation is the development of online tracking-by-detection methods that improve MOT performance by exploiting the interaction between detection, state estimation, and data association. The proposed methods use detection confidence, tracklet information, motion uncertainty, and association cues as complementary sources of information, while preserving the simplicity and efficiency required for real-time tracking.

The specific contributions of the dissertation are summarized as follows.

1. **Confidence-aware one-stage association.** In Chapter 3, we introduce a one-stage association framework that incorporates detection and tracklet confidence directly into the similarity measure. This allows reliable detection-tracklet pairs to be favored without explicitly dividing detections or tracklets into multiple association

stages.

2. **Improved similarity computation for association.** We propose lightweight additions to the standard IoU-based similarity measure, including Mahalanobis distance similarity and shape similarity. These additions reduce association ambiguity, especially in crowded scenes, and improve matching robustness without introducing computationally expensive appearance models.
3. **Detection confidence boosting for unreliable detections.** We introduce detection confidence boosting techniques that allow true-positive low-confidence detections to be retained before thresholding. The method increases the confidence of detections that are likely to correspond either to existing tracked objects or to previously undetected objects, enabling the tracker to use more true-positive detections in a one-stage association framework.
4. **Tracklet-guided recovery of true-positive detections.** In Chapter 4, we extend the detection recovery strategy by using richer tracklet-based information. The proposed method combines shape similarity, Mahalanobis distance similarity, and soft buffered IoU to identify low-confidence detections that are likely to correspond to real objects. Soft detection confidence boosting and varying thresholds are introduced to account for the original detection confidence and the reliability of tracklet predictions.
5. **Kalman-filter-based trajectory splitting.** We introduce a post-processing procedure that uses Kalman filter predictions to decide whether fragmented trajectory parts should be interpolated or separated. This reduces unreliable trajectory connections and shows that tracking information can also support post-processing.
6. **Crowd-Aware Kalman Filter.** In Chapter 5, we introduce the Crowd-Aware Kalman Filter (CAKF), an adaptive motion estimation method that estimates process and measurement uncertainty online and adjusts them according to local scene ambiguity. By incorporating both innovation statistics and crowd-aware uncertainty scaling, CAKF improves state estimation in crowded and ambiguous scenarios.
7. **Expanded Intersection over Union.** We propose the Expanded IoU (EIoU), an uncertainty-driven similarity measure that uses motion uncertainty to adapt the spatial matching region for each tracklet. Unlike fixed bounding-box expansion heuristics, EIoU performs object-specific expansion based on the uncertainty estimated during tracking, improving association while preserving a simple one-stage matching procedure.

8. **Improved uncertainty-aware association workflow.** We introduce an improved one-stage association workflow that uses all detections and applies uncertainty-aware criteria for association validity, tracklet initialization, and trajectory updates. The workflow reuses detection confidence, tracklet confidence, association information, and motion uncertainty to reduce the impact of unreliable detections and missed associations.
9. **Extensive experimental validation.** The proposed methods are evaluated on standard MOT benchmarks under different conditions, including crowded scenes, irregular motion, visually similar targets, and varying detection quality. BoostTrack and BoostTrack++ achieve competitive performance among contemporary MOT methods and become strong benchmark methods. CAT achieves state-of-the-art or near state-of-the-art performance among online motion-based methods on several datasets, while remaining strictly online, appearance-free, and real-time. The experiments also show that CAT is competitive with, and in some cases outperforms, more complex offline and appearance-based methods, demonstrating the effectiveness of carefully coupled motion-based components.

Together, these contributions support the central idea of the dissertation: strong MOT performance can be achieved not only by adding more complex components, but also by improving the communication between the existing components of a tracking-by-detection system.

Chapter 2

Background

This chapter establishes the common background and notation used throughout the dissertation. It first formulates the multiple object tracking problem and introduces the basic notation used to describe detections, tracklets and trajectories. It then reviews the main components of tracking-by-detection MOT: object detection, Kalman-filter-based state estimation, data association, association workflow and tracklet management.

In addition to the online tracking pipeline, the chapter briefly discusses post-processing techniques that are commonly used to refine tracking results after online inference. Finally, it introduces the benchmark datasets and evaluation metrics used in the experimental analysis of the proposed methods.

2.1 MOT Problem Formulation and Notation

Let

$$\mathcal{V} = \{I^1, I^2, \dots, I^N\} \quad (2.1)$$

denote a video sequence, where I^t is the frame at time step t , $t = 1, \dots, N$. For each object of interest, a trajectory is represented as a sequence of bounding boxes over the frames in which the object is present or annotated. Let

$$\mathcal{T}_j = \{T_j^{t_s}, T_j^{t_s+1}, \dots, T_j^{t_e}\} \quad (2.2)$$

be the trajectory of object j , which starts at frame t_s and ends at frame t_e , where $1 \leq t_s \leq t_e \leq N$.¹

The element T_j^t specifies the bounding box of object j in frame t . In this dissertation,

¹Annotation protocols may differ between datasets. For example, an object that leaves and later re-enters the scene may be assigned a new identity, and fully occluded intervals may be omitted from the annotated trajectory.

unless stated otherwise, bounding boxes are represented in the form

$$T_j^t = (u_j^t, v_j^t, w_j^t, h_j^t), \quad (2.3)$$

where (u_j^t, v_j^t) denotes the center of the bounding box, while w_j^t and h_j^t denote its width and height, respectively. If the video contains M objects of interest, the goal of MOT is to estimate the set of trajectories

$$\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_M\}, \quad (2.4)$$

while preserving a consistent identity for each object over time.

In tracking-by-detection MOT, detections are obtained independently for each frame. Let f_{det} denote an object detector. At frame t , the detector outputs

$$\mathcal{D}^t = f_{\text{det}}(I^t) = \{D_1^t, D_2^t, \dots, D_n^t\}, \quad (2.5)$$

where n is the number of detections in frame t . Each detection D_i^t has an associated confidence score $c_{D_i}^t$ indicating the reliability of the detection.

During online tracking, the algorithm maintains a set of active tracked objects. In tracking-by-detection literature, an active tracked object is commonly referred to as a tracklet.² Before association in frame t , the tracking module predicts the current bounding box of each tracklet. Let

$$\hat{T}^t = \{\hat{T}_1^t, \hat{T}_2^t, \dots, \hat{T}_m^t\}. \quad (2.6)$$

denote the set of predicted bounding boxes for the m tracklets in frame t .

In the association step, detections from $\mathcal{D}^t$ are matched with predicted bounding boxes from $\hat{T}^t$. When detection D_i^t is associated with predicted bounding box $\hat{T}_j^t$, the detection is used to update the corresponding tracklet and obtain the estimated bounding box T_j^t (which can be appended to $\mathcal{T}_j$). Unmatched detections may initialize new tracklets, while unmatched tracklets are either propagated using the tracking module or terminated according to tracklet management rules.

We use the notation introduced in this section throughout the dissertation. Whenever the frame index t is clear from the context or not relevant, it is omitted. The details of object detection, Kalman-filter-based state estimation, similarity computation, association, and tracklet management are discussed in the following sections.

²A possible source of confusion is that, in offline MOT, the term tracklet is often used to denote a short fragment of a trajectory.

2.2 Object Detection

Object detection serves as the perceptual foundation for tracking-by-detection systems. For each frame, the detector produces a set of candidate detections, where each detection usually consists of a bounding box, a class label, and a confidence score. In this dissertation, detections are represented by bounding boxes together with their confidence scores, since the considered benchmarks focus on single-class pedestrian or player tracking.

Modern object detectors are typically based on convolutional neural networks or related deep architectures that learn hierarchical visual representations directly from data, replacing hand-crafted visual features [149, 93]. Single-stage detectors are especially important for real-time video analysis because they predict object locations and classes in a single forward pass. Among them, detectors from the YOLO family are widely used due to their favorable balance between accuracy and speed [122, 27]. In this dissertation, we use YOLOX [38], a single-stage anchor-free detector with a decoupled head and dynamic label assignment.

In YOLO-style detectors, each candidate bounding box is associated with an objectness score and class probabilities. The objectness score estimates whether the predicted bounding box contains an object, while the class probabilities estimate the class of that object. The final detection confidence score is commonly obtained as the product of the objectness score and the corresponding class probability. For single-class MOT benchmarks, this score can be interpreted as the confidence that a detected bounding box corresponds to an object of interest.

Object detectors usually produce a large number of candidate bounding boxes. Many candidates have very low confidence scores, while multiple highly overlapping boxes may correspond to the same object. Therefore, two standard post-processing steps are commonly applied in detection step: confidence thresholding and non-maximum suppression (NMS).

Confidence thresholding removes detections whose confidence score is below a predefined threshold τ_{det} . This reduces the number of candidate detections and suppresses many low-confidence false positives.

Non-maximum suppression removes duplicate detections. In the standard NMS procedure, detections are sorted by confidence score. The highest-confidence detection is selected, and other detections that overlap with it more than a predefined threshold τ_{NMS} are removed. The overlap is usually measured by Intersection over Union (IoU). This procedure is repeated until all remaining detections have been processed. A typical value of τ_{NMS} in modern detection-based tracking systems is around 0.7.

Although these post-processing steps are effective, their parameters influence the detections available to the tracker and can therefore affect the final tracking performance.

2.3 Kalman Filter for MOT

The Kalman filter [53] is a common choice for state estimation in multiple object tracking (MOT). In tracking-by-detection systems, it is typically used to predict the state of each active tracklet and to update this prediction when a detection is associated with the tracklet.

The state of a tracked object at time step t , denoted by $\mathbf{x}_t$, can be defined as

$$\mathbf{x}_t = [u_t, v_t, w_t, h_t, \dot{u}_t, \dot{v}_t, \dot{w}_t, \dot{h}_t]^\top, \quad (2.7)$$

where (u_t, v_t) are the 2D coordinates of the bounding box center, w_t and h_t are the bounding box width and height, and $\dot{u}_t$, $\dot{v}_t$, $\dot{w}_t$, and $\dot{h}_t$ denote the corresponding velocities. A detected bounding box, represented as

$$\mathbf{z}_t = [u_t, v_t, w_t, h_t]^\top,$$

is used as an observation in the Kalman update step after it has been associated with a tracklet (i.e., $\mathbf{z}_t = D_i^t$ for some i).

Since video sequences are sampled at relatively short time intervals, object motion between consecutive frames is commonly approximated by a constant-velocity model. Assuming a unit time step between consecutive frames, the state-space model is given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{q}_t, \quad \mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{r}_t, \quad (2.8)$$

where $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, are process and measurement noise, respectively, and the state transition matrix $\mathbf{F}$ and the observation matrix $\mathbf{H}$ are defined as

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{0}_{4 \times 4} & \mathbf{I}_4 \end{bmatrix}, \quad \mathbf{H} = [\mathbf{I}_4 \quad \mathbf{0}_{4 \times 4}]. \quad (2.9)$$

State estimation with the Kalman filter consists of two phases: prediction and update. In the prediction phase, the state and covariance are predicted from the estimates obtained at the previous time step:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}, \\ \mathbf{P}_{t|t-1} &= \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q}_{t-1}. \end{aligned} \quad (2.10)$$

After receiving an associated observation $\mathbf{z}_t$, the predicted state and covariance are updated as follows:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}), \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}, \end{aligned} \quad (2.11)$$

where $\mathbf{K}_t$ is the Kalman gain defined as

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_{t|t-1} \mathbf{H}^\top + \mathbf{R}_t)^{-1}. \quad (2.12)$$

2.4 Data Association

Once we obtain detected bounding boxes $\mathcal{D}^t = \{D_1, D_2, \dots, D_n\}$ and predicted bounding boxes of active tracklets $\hat{T}^t = \{\hat{T}_1, \hat{T}_2, \dots, \hat{T}_m\}$, we need to find a matching between $\mathcal{D}^t$ and $\hat{T}^t$.

We can view the elements of $\mathcal{D}^t$ and $\hat{T}^t$ as two vertex sets of a bipartite graph, with edge weights $e_{i,j}$ that quantify the cost of associating D_i with $\hat{T}_j$, for every i and j . When formulated as a bipartite matching problem, the Hungarian algorithm [58] is typically used in MOT to find an optimal matching between detected bounding boxes and predicted bounding boxes of the currently tracked objects.

The main difficulty in association is how to define the association cost matrix $\mathbf{M}$, where $M_{i,j} = e_{i,j}$. The most similar bounding boxes should have the lowest cost. Therefore, throughout the dissertation, we focus on defining a similarity matrix $\mathbf{S}$ between elements of $\mathcal{D}^t$ and $\hat{T}^t$, and use $\mathbf{M} = -\mathbf{S}$ as the association cost.

In the simplest case, the similarity matrix can be defined using Intersection over Union (IoU) between bounding boxes, i.e.,

$$\mathbf{S} = \mathbf{S}_{\text{IoU}} = [\text{IoU}(D_i, \hat{T}_j)]_{n \times m}, \quad (2.13)$$

where

$$\text{IoU}(D_i, \hat{T}_j) = \frac{|D_i \cap \hat{T}_j|}{|D_i \cup \hat{T}_j|}. \quad (2.14)$$

However, IoU is often not discriminative enough in ambiguous or crowded scenarios. Visual appearance can also be used to construct a similarity matrix. A neural network f_{emb} can be used to extract visual embedding vectors from image regions corresponding to detections or tracklets.

The cosine similarity between a detection embedding $\mathbf{e}_{D_i}$ and a tracklet embedding $\mathbf{e}_{T_j}$ can be used to construct the appearance similarity matrix

$$\mathbf{S}_{\text{app}} = [S_{\text{cos}}(\mathbf{e}_{D_i}, \mathbf{e}_{T_j})]_{n \times m}, \quad (2.15)$$

where

$$S_{\text{cos}}(\mathbf{e}_{D_i}, \mathbf{e}_{T_j}) = \frac{\mathbf{e}_{D_i} \cdot \mathbf{e}_{T_j}}{\|\mathbf{e}_{D_i}\| \|\mathbf{e}_{T_j}\|}. \quad (2.16)$$

The embedding model f_{emb} can be a separate model from the detector [130, 129, 3, 30, 123], or the same model can perform both detection and embedding extraction [145, 19, 119].

An aggregate similarity matrix can then be defined as

$$\mathbf{S} = \lambda_{\text{IoU}} \mathbf{S}_{\text{IoU}} + \lambda_{\text{app}} \mathbf{S}_{\text{app}}, \quad (2.17)$$

where λ_{IoU} and λ_{app} control the influence of geometric and appearance similarity, respectively. In the same way, other motion, geometric, or visual similarity measures can be combined into an aggregate similarity matrix [15, 63, 60].

2.5 Tracking-by-Detection Workflow and Tracklet Management

Both the detector and the tracking module are inaccurate and error-prone, and a tracklet can be associated with an observation corresponding to a different object or with a false positive observation. False positive observations can also be used to initialize new tracklets, thus causing further association and tracking errors.

Therefore, TbD algorithms typically need to address several additional issues:

- How to reduce the impact of noisy observations?
- How to reduce the impact of incorrect associations?
- When should a tracklet be initialized or deleted?
- What criterion should be used to update a trajectory?

The detection confidence score can be interpreted as a measure of detection reliability. From the original set of detected bounding boxes, we can select only detections with confidence scores above a certain threshold τ_{det} (e.g., $\tau_{\text{det}} = 0.5$) and perform association using detections from $\mathcal{D}_{\text{high}}^t = \{D_i^t \mid D_i^t \in \mathcal{D}^t \wedge c_{D_i^t}^t \geq \tau_{\text{det}}\}$. The highly influential SORT method [10], which popularized the combination of the Kalman filter as a tracking module, IoU as a similarity measure, and the Hungarian algorithm as a matching algorithm, uses this simple thresholding strategy to select reliable observations.

However, low-confidence detections can also be true positives and may therefore be crucial for successful tracking. At the same time, lowering τ_{det} can degrade tracking performance by introducing false positives. Two-stage association, introduced by ByteTrack [144], offers a solution that has been adopted by many TbD MOT algorithms. First,

detections with confidence scores below τ_{low} are discarded, while the remaining detections are split into the sets $\mathcal{D}_{\text{high}}^t$ and $\mathcal{D}_{\text{low}}^t$.

$$\begin{aligned}\mathcal{D}_{\text{high}}^t &= \{D_i^t \mid D_i^t \in \mathcal{D}^t \wedge c_{D_i}^t \geq \tau_{\text{high}}\}, \\ \mathcal{D}_{\text{low}}^t &= \{D_i^t \mid D_i^t \in \mathcal{D}^t \wedge \tau_{\text{low}} \leq c_{D_i}^t < \tau_{\text{high}}\}.\end{aligned}\tag{2.18}$$

In the first association stage, tracklets $\hat{T}^t$ are matched with detections from $\mathcal{D}_{\text{high}}^t$. The remaining unmatched tracklets are then matched with detections from $\mathcal{D}_{\text{low}}^t$ in the second stage. The similarity measure used in the first and second stage can be different; for example, appearance similarity is often used only in the first stage.

To reduce the impact of incorrect associations, a minimum IoU threshold τ_{IoU} is commonly introduced. The pair $(D_i, \hat{T}_j)$ is considered a valid match only if

$$\text{IoU}(D_i, \hat{T}_j) \geq \tau_{\text{IoU}}.\tag{2.19}$$

Unmatched detections from $\mathcal{D}_{\text{high}}^t$ can be used to initialize new tracklets. Optionally, an additional higher threshold τ_{tc} can be used to select only more reliable detections for tracklet creation. A tracklet that is not successfully matched with any detection for A_{max} consecutive frames is usually terminated.

Finally, to reduce the effect of false positive detections and short-lived erroneous tracklets, size and position of a tracklet is often reported as valid and added to the trajectory only after the corresponding tracklet achieves A_{match} consecutive successful matches.

Algorithm 1 presents pseudocode for processing a single frame in a generic TbD algorithm with two-stage association.

2.6 Post-processing

The workflow described in the previous section can be used to estimate object trajectories in an online manner. However, in many MOT approaches, the online tracking stage is followed by an additional post-processing step that refines the initially estimated trajectories. Post-processing provides additional flexibility: the same online algorithm can be used when strictly online performance is required, while its output can be further refined when strictly online operation is not necessary. In this way, online tracking results can be improved and brought closer to the performance of offline methods.

Due to association failures and occlusions, the resulting trajectories are often fragmented. This fragmentation can be further increased by track confirmation rules that require A_{match} consecutive matches before an estimate is included in the final output.

Linear interpolation is commonly used as a post-processing step to fill gaps in frag-

Algorithm 1 General tracking-by-detection workflow and tracklet management

Inputs: Frame I^t , active trajectories $\mathcal{T}^{t-1}$, detector f_{det} , detection thresholds τ_{low} , τ_{high} , τ_{tc} , admissible match threshold τ_{IoU} , tracklet management hyperparameters A_{max} , A_{match} .

Output: Updated trajectories $\mathcal{T}^t$.

```
    /* Detection and prediction. */
1:  $\mathcal{D}^t \leftarrow f_{\text{det}}(I^t)$ 
2:  $\hat{T}^t \leftarrow \text{predict}(\mathcal{T}^{t-1})$  ▷ e.g., execute Kalman predict step
3:  $\mathcal{D}_{\text{high}}^t \leftarrow \{D_i^t \in \mathcal{D}^t \mid c_{D_i}^t \geq \tau_{\text{high}}\}$ 
4:  $\mathcal{D}_{\text{low}}^t \leftarrow \{D_i^t \in \mathcal{D}^t \mid \tau_{\text{low}} \leq c_{D_i}^t < \tau_{\text{high}}\}$ 
    /* First association stage. */
5:  $S_{\text{high}} \leftarrow \text{compute\_similarity\_I}(\mathcal{D}_{\text{high}}^t, \hat{T}^t)$  ▷ e.g., Eq. (2.17)
6:  $\mathcal{M}_{\text{high}}, \mathcal{D}_{\text{unmatched}}, \hat{T}_{\text{unmatched}} \leftarrow \text{association}(S_{\text{high}}, \tau_{\text{IoU}})$ 
    /* Second association stage. */
7:  $S_{\text{low}} \leftarrow \text{compute\_similarity\_II}(\mathcal{D}_{\text{low}}^t \cup \mathcal{D}_{\text{unmatched}}, \hat{T}_{\text{unmatched}})$  ▷ e.g., Eq. (2.13)
8:  $\mathcal{M}_{\text{low}}, \mathcal{D}_{\text{unmatched}}, \hat{T}_{\text{unmatched}} \leftarrow \text{association}(S_{\text{low}}, \tau_{\text{IoU}})$ 
    /* Update matched tracklets. */
9: for all  $(D_i^t, \hat{T}_j^t) \in \mathcal{M}_{\text{high}} \cup \mathcal{M}_{\text{low}}$  do
10:    $T_j^t \leftarrow \text{update}(\hat{T}_j^t, D_i^t)$  ▷ e.g., execute Kalman update step
11: end for
    /* Keep predictions of unmatched tracklets. */
12: for all  $\hat{T}_j^t \in \hat{T}_{\text{unmatched}}$  do
13:    $T_j^t \leftarrow \hat{T}_j^t$ 
14: end for
    /* Start new tracklets. */
15: for all  $D_i^t \in \mathcal{D}_{\text{unmatched}}$  do
16:   if  $c_{D_i}^t \geq \tau_{\text{tc}}$  then
17:      $\mathcal{T}^t \leftarrow \mathcal{T}^t \cup \{\text{new\_tracklet}(D_i^t)\}$ 
18:   end if
19: end for
    /* Delete lost tracklets. */
20: for all  $\hat{T}_j^t \in \hat{T}_{\text{unmatched}}$  do
21:   if  $\text{frames\_since\_update}(T_j) > A_{\text{max}}$  then
22:      $\text{remove}(T_j)$ 
23:   end if
24: end for
    /* Confirm and update trajectories. */
25:  $\mathcal{T}^t \leftarrow \mathcal{T}^{t-1}$ 
26: for all  $T_j \in \hat{T}^t$  do
27:   if  $\text{hit\_streak}(T_j) \geq A_{\text{match}}$  then
28:      $\mathcal{T}_j^t \leftarrow \mathcal{T}_j^t \cup \{T_j\}$ 
29:   end if
30: end for
31: return  $\mathcal{T}^t$ 
```

mented trajectories. Let $\mathcal{T}_j = \{T_j^{t_s}, T_j^{t_s+1}, \dots, T_j^{g_s-1}, T_j^{g_s}, T_j^{g_e}, T_j^{g_e+1}, \dots, T_j^{t_e}\}$, be a trajectory of object j with a gap between frames g_s and g_e , i.e., $g_e - g_s > 1$. Assuming linear motion of the bounding-box parameters, the missing estimates $T_j^{g_s+i}$, $i \in \{1, 2, \dots, g_e - g_s - 1\}$, can be interpolated as

$$T_j^{g_s+i} = T_j^{g_s} + \frac{i}{g_e - g_s} (T_j^{g_e} - T_j^{g_s}). \quad (2.20)$$

More advanced post-processing methods can be applied after missing frames have been initially filled by interpolation. Each bounding-box component $p \in \{u, v, w, h\}$ can then be represented as a time series $\{(t, p_j^t)\}$ for trajectory $\mathcal{T}_j$ and refined independently. Gaussian Process Regression smooths these temporal signals by imposing a smoothness prior [30], while a later gradient-boosting-based approach learns residual corrections for the initially interpolated trajectory and achieves more accurate nonlinear motion reconstruction [142]. Due to its stronger performance, the smoothing strategy based on gradient boosting is used in the post-processing configurations of the methods presented in Chapters 3 and 4.

2.7 Classification of MOT methods

MOT methods can be classified according to different criteria. One common distinction is based on the temporal information available to the tracker. A strictly online method processes frames sequentially and produces trajectories without modifying them after online inference. Offline methods use future frames or global sequence-level information to construct or refine trajectories [17, 37]. In a binary online/offline classification, most authors classify methods that apply post-processing to online tracking results as online methods [144, 75, 135], while some authors refer to them as offline methods [30, 59]. In this dissertation, we avoid this ambiguity and distinguish three types of methods based on the use of future information: strictly online methods, online methods with post-processing, and offline methods.

Another important distinction is based on the type of information used for association. Motion-based methods rely primarily on geometric and motion cues, such as bounding-box overlap and Kalman-filter predictions. Appearance-based methods additionally use visual features, most commonly in the form of embedding vectors extracted from image regions (see Section 2.4).

Many other criteria can be used to classify MOT methods. Some approaches are distinguished by the way they combine detection and appearance embedding [145, 19], some by whether the tracking pipeline is trained end-to-end [153, 147], and some by the use of

transformer-based architectures [141, 80, 55]. Methods can also be compared according to whether tracking is based primarily on deep learning or on Kalman-filter-based motion estimation [1, 34]. In this dissertation, we focus on the criteria most relevant to the proposed methods: the temporal processing type and the type of information used for association.

Combining these two criteria results in several possible categories, such as strictly online motion-based methods, strictly online appearance-based methods, etc. A complete classification along both axes is often unnecessarily detailed for experimental comparison, since the most relevant distinction depends on the focus of the method being analyzed. Therefore, in this dissertation, categories are grouped differently when needed.

In Chapters 3 and 4, where the focus is on detection confidence handling and association, methods are primarily discussed as online, online with post-processing, or offline. In Chapter 5, the focus shifts to strictly online motion-based tracking. Therefore, that chapter distinguishes methods according to the type of information used for association, i.e., motion-based or appearance-based, while using a stricter online/offline temporal classification. Under this classification, methods that use post-processing are treated as offline, even if their core tracker operates online.

2.8 Evaluation Metrics

Evaluating multiple object tracking is more challenging than evaluating object detection, as a tracker must not only accurately localize objects in individual frames but also maintain their identities consistently over time. An effective MOT method should detect as many objects as possible while minimizing false positives and false negatives, preserving continuous trajectories, and maintaining consistent identity assignments throughout the sequence. Consequently, MOT evaluation considers both detection performance and data association quality, combining detection-related errors, such as false positives and false negatives, with association-related errors, such as identity switches. In this dissertation, we primarily report MOTA, IDF1, and HOTA scores, which are the most commonly used MOT metrics.

An identity switch (IDSW) occurs when a ground-truth object that was previously associated with one predicted identity becomes associated with another. Since IDSW directly reflects failures in identity preservation, it is commonly reported together with aggregate MOT metrics.

Multiple Object Tracking Accuracy (MOTA) combines false negatives, false positives, and identity switches into a single score [9]. It is defined as

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t}, \quad (2.21)$$

where FN_t , FP_t , and $IDSW_t$ denote the number of false negatives, false positives, and identity switches in frame t , respectively, while GT_t is the number of ground-truth objects in that frame. Higher MOTA values indicate better tracking performance. However, MOTA is strongly influenced by detection errors, and may not fully reflect identity preservation. As such, MOTA is usually used to assess detector performance.

The IDF1 score focuses on identity preservation by measuring how well computed detections are assigned to the correct ground-truth identities [99]. It is based on identification precision and identification recall. Identification precision measures the fraction of computed detections that are correctly identified, while identification recall measures the fraction of ground-truth detections that are correctly identified. The IDF1 score is their harmonic mean and can be written as

$$IDF1 = \frac{2 IDTP}{2 IDTP + IDFP + IDFN}, \quad (2.22)$$

where IDTP, IDFP, and IDFN denote identity true positives, identity false positives, and identity false negatives, respectively. In this sense, IDF1 provides a single score that balances how many computed detections are assigned to the correct identities and how many ground-truth detections are successfully recovered with the correct identities. IDF1 is, therefore, used primarily as association performance measure.

Higher Order Tracking Accuracy (HOTA) was introduced to provide a more balanced evaluation of detection and association performance [72]. HOTA decomposes tracking quality into detection accuracy and association accuracy. In simplified form, for a localization threshold α , it can be written as

$$HOTA_\alpha = \sqrt{DetA_\alpha AssA_\alpha}, \quad (2.23)$$

where $DetA_\alpha$ measures detection accuracy and $AssA_\alpha$ measures association accuracy at threshold α . The final HOTA score is obtained by averaging over multiple localization thresholds. Detection accuracy reflects the ability of the tracker to correctly detect objects, while association accuracy evaluates how consistently matched detections are assigned to the same trajectories. HOTA is useful because it explicitly balances detection, tracking and localization into a single metric. In addition to HOTA, its components such as $DetA$, $AssA$, and $LocA$ are sometimes reported to analyze whether performance differences are caused primarily by detection, association, or localization quality.

In the experimental chapters of this dissertation, MOTA is used to assess the overall effect of false positives, false negatives, and identity switches; IDF1 is used to evaluate identity preservation; and HOTA is used as the main balanced metric that jointly reflects detection and association quality. Together, these metrics provide a more complete view

of MOT performance than any single metric alone. As a standard practice, we also report IDSW as an intuitive auxiliary measure, since it directly indicates identity changes caused by association failures.

2.9 Benchmark Datasets

The methods presented in this dissertation are evaluated on several standard MOT benchmark datasets that cover different tracking conditions, object densities, motion patterns, and association difficulties. Using multiple datasets is important because tracking performance can strongly depend on the type of scene. A method that performs well in crowded pedestrian sequences may not necessarily be equally effective in scenarios with fast camera motion, irregular object movement, or visually similar targets. In this dissertation, the main experimental evaluation is conducted on MOT17, MOT20, DanceTrack, and SportsMOT.

The choice of datasets follows the focus of each contribution. BoostTrack and BoostTrack++, presented in Chapters 3 and 4, primarily address low-confidence detections, and detection-to-tracklet association, and their evaluation is focused on MOT17 and MOT20, where these issues are especially relevant and where the methods were originally benchmarked. DanceTrack and SportsMOT are used, in addition to MOT17 and MOT20, in Chapter 5, where the focus is broader and includes robustness to irregular motion, dynamic scenes, and motion-based tracking without appearance features.

MOT17 [84] is one of the most widely used pedestrian tracking benchmarks. It contains video sequences recorded in different public environments, including indoor and outdoor scenes, static and moving cameras, and varying object densities. The benchmark is primarily focused on pedestrian tracking and has become a standard reference point for comparing MOT algorithms. MOT17 is especially useful for evaluating the general behavior of a tracker under common pedestrian tracking conditions, including moderate occlusions, camera motion, missed detections, and interactions among nearby people.

MOT20 [25] extends the MOTChallenge pedestrian tracking setting to significantly more crowded scenes. Compared with MOT17, MOT20 contains sequences with much higher pedestrian density, stronger occlusions, and more frequent overlaps between objects. These conditions make both detection and association more difficult. Since many identity switches occur when several pedestrians are close to one another or partially occluded, MOT20 is particularly important for evaluating association robustness, the use of low-confidence detections, and the ability of a tracker to preserve identities in dense scenes.

DanceTrack [118] was introduced to emphasize the association aspect of MOT. It con-

tains videos of dancers whose appearances are often similar and whose motions are complex, nonlinear, and highly coordinated. In such scenes, simple motion assumptions may fail, while appearance cues may be less discriminative than in standard pedestrian tracking datasets. DanceTrack is therefore useful for evaluating whether a tracker can maintain identities when objects move irregularly, frequently change relative positions, and exhibit similar visual appearance.

SportsMOT [23] focuses on tracking athletes in sports videos. Sports scenes introduce several challenges that differ from standard pedestrian tracking: fast and abrupt motion, frequent interactions between players, camera motion, changes in scale, and strong similarity in appearance due to team uniforms. These properties make SportsMOT a valuable benchmark for testing the robustness of MOT algorithms in dynamic environments. It is especially relevant for methods that aim to remain accurate and real-time without relying heavily on computationally expensive appearance models.

Throughout the dissertation, validation sets are used for ablation studies, parameter analysis, and component-level evaluation, while test sets are used for final comparison with existing methods. For MOT17 and MOT20, following common MOT practice, validation sets are constructed from the second half of each training sequence. The remaining official test sets are used for benchmark comparison. For datasets with official validation splits, such as DanceTrack and SportsMOT, the official validation and test splits are used. This protocol allows method development and parameter selection to be separated from final test evaluation.

Together, these datasets provide a diverse evaluation setting. MOT17 and MOT20 test the performance of tracking methods in standard and highly crowded pedestrian scenarios, while DanceTrack and SportsMOT introduce more challenging motion patterns and stronger appearance ambiguities. This combination allows a more complete analysis of the proposed methods with respect to detection quality, data association, identity preservation, robustness, and real-time applicability.



(a) MOT17.



(b) MOT20.



(c) DanceTrack.



(d) SportsMOT.

Figure 2.1: Example frames from the datasets used in the dissertation.

Chapter 3

BoostTrack: Confidence and Similarity Boosting for One-Stage Multiple Object Tracking

3.1 Introduction

As discussed in the previous chapters, tracking-by-detection MOT relies on three tightly connected components: object detection, state prediction, and data association. A failure in any of these components can affect the others and degrade the quality of the resulting trajectories. In this chapter, we focus on two common sources of such errors: detection handling and data association.

Throughout the dissertation, we use YOLOX [38] for object detection, and our goal is not to improve or develop a new deep learning model for object detection, nor to *directly* improve the performance of the detector. Instead, our goal is to improve the selection of true-positive detected bounding boxes and minimize the impact of false-positive and noisy detections in multiple object tracking.

A standard way to reduce false positives is to discard detections whose confidence score is below a predefined threshold. However, low-confidence detections are not necessarily false positives. They may correspond to partially occluded objects, objects entering the scene, objects near image boundaries, or detections affected by poor visibility. Removing all such detections can interrupt trajectories and degrade tracking performance. On the other hand, using them without additional control can introduce false trajectories and identity switches.

A common solution is to use two-stage or, more generally, multi-stage association, where high-confidence detections are matched first and low-confidence detections are considered only in a later stage. While this strategy can recover some true-positive de-

tections, it also changes the structure of the assignment problem. Decisions made in an earlier association stage may constrain later stages and can lead to identity switches [110], especially when multiple objects are close to one another. This motivates the question of whether low-confidence detections can be used effectively while preserving a one-stage association framework.

The second issue addressed in this chapter is the limited discriminative power of simple overlap-based association. Intersection over Union (IoU) is efficient and widely used, but it may be insufficient in crowded scenes, under occlusion, or when several objects have highly overlapping bounding boxes. Appearance features can improve association, but they require additional visual feature extraction and may reduce real-time applicability. Therefore, a lightweight alternative is needed to improve association before relying on computationally expensive ReID modules.

To address these issues, this chapter presents BoostTrack, a tracking-by-detection method built on top of SORT [10]. BoostTrack introduces two groups of lightweight additions: detection confidence boosting and similarity boosting.

Detection confidence boosting allows useful low-confidence detections to be retained before thresholding, without introducing a separate low-confidence association stage. BoostTrack increases the confidence of detections that are likely to correspond either to existing tracklets or to new objects not explained by current tracklet predictions. The first case helps preserve trajectory continuity during occlusion or partial visibility, while the second helps initialize trajectories of newly visible or previously undetected objects.

To preserve benefits of multi-stage association while avoiding its drawbacks, BoostTrack introduces detection-tracklet confidence. This score scales the similarity measure and implicitly favors reliable detection-tracklet pairs within a single association stage.

Finally, BoostTrack improves simple IoU-based association by introducing lightweight similarity boosting components. In addition to IoU, the aggregate similarity measure incorporates detection-tracklet confidence, Mahalanobis distance similarity, and shape similarity. These additions reduce association ambiguity while preserving the simplicity and efficiency of the tracking pipeline. In the rest of this chapter, the proposed methods for improving detection selection are referred to as detection confidence boosting, while the proposed additions to the association score are referred to as similarity boosting.

In addition to the appearance-free BoostTrack variant, we also consider BoostTrack+, which incorporates appearance similarity into the aggregate association score. This extension achieves stronger accuracy, but at the cost of increased runtime due to visual feature extraction.

The resulting method improves tracking performance while preserving a simple and efficient tracking pipeline. The appearance-free BoostTrack variant emphasizes real-time

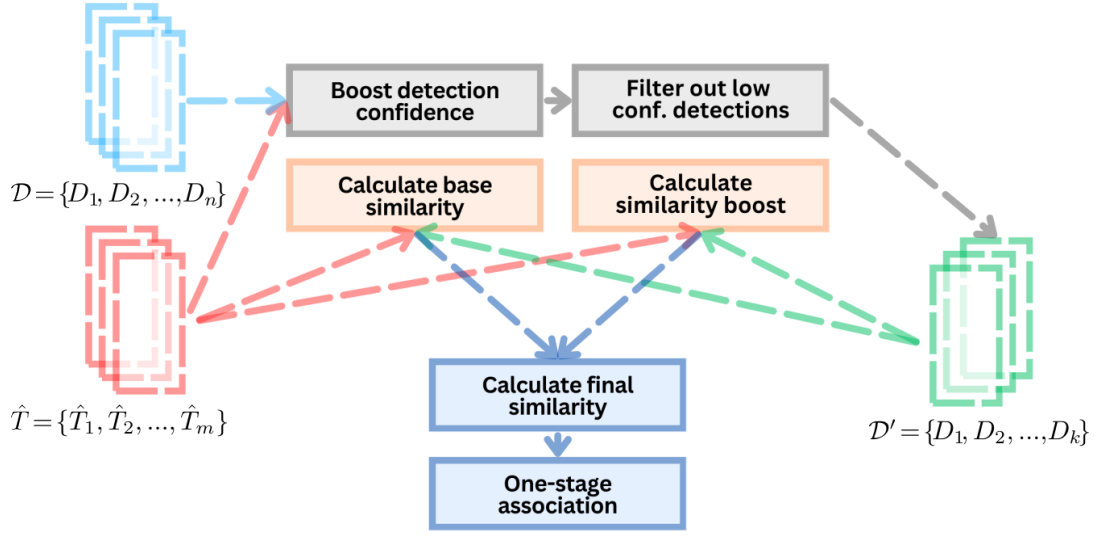


Figure 3.1: Overview of the BoostTrack method. Tracklet predictions and detections are used to boost selected detection confidence scores before filtering, while the remaining detections $\mathcal{D}'$ and tracklets are associated using a boosted similarity measure.

operation without ReID features, while BoostTrack+ demonstrates that the proposed components can also be combined with appearance similarity when higher accuracy is preferred over speed. An overview of BoostTrack is shown in Figure 3.1.

The main contributions presented in this chapter are:

- two detection confidence boosting techniques for using low-confidence detections in one-stage association;
- detection-tracklet confidence, used to favor reliable detection-tracklet pairs without explicit multi-stage matching;
- lightweight similarity based on Mahalanobis distance and bounding box shape;
- an experimental analysis demonstrating the effect of the proposed components on MOT17 and MOT20, including both the appearance-free BoostTrack variant and the appearance-based BoostTrack+ extension.

The rest of the chapter is organized as follows. First, in Section 3.2 we review related work on unreliable detections, multi-stage association, tracklet confidence, and similarity measures. We then introduce the proposed similarity boosting and detection confidence boosting components in Sections 3.3 and 3.4. We present the experimental evaluation, ablation study and compare our method with other methods in Section 3.5. Section 3.6 concludes the chapter.

3.2 Related Work

This section reviews prior work related to the main components of BoostTrack: handling unreliable detections, tracklet confidence, and similarity measures for data association.

3.2.1 Working with Unreliable Detections

Various strategies have been used to deal with unreliable detections, i.e., to identify and discard false positive detections [96]. In [131], an SVM [12] model was trained to classify detections into tracked or inactive classes. In [88], all detections are used for association, but the association is performed in a two-stage manner, associating first detections and tracklets with greater similarity and associating the rest in the second stage. Filtering out unreliable detections based on detection confidence, i.e., thresholding, is the most common practice [130, 140, 30]. However, some low-confidence detections can correspond to partially occluded objects, and using these detections can improve tracking performance.

In [101], high-confidence detections are used for trajectory initialization and tracking, while low-confidence detections are used for tracking only. As discussed in Section 2.5, ByteTrack [144] popularized the use of low-confidence detections through two-stage association, where high-confidence detections are matched first and the remaining detections are considered in the second stage. This strategy has since been adopted or extended in several tracking methods (e.g., [125, 3, 75, 133, 62]). In [77], the authors proposed an offline tracking algorithm that uses all detection boxes. LG-Track [81] uses localization and classification confidence scores from the detector and divides detections into four groups based on thresholds for the two scores. Association is performed in four stages using different cost matrices, which are scaled differently by detection confidence scores in different association stages. ImprAsso [110] splits detections into high- and low-confidence sets and calculates association distances for both sets, which are combined into a single matrix and used in a single association step.

3.2.2 Tracklet Confidence

The notion of tracklet confidence is not standardized and has been defined and used in different ways in prior work [7, 20, 88, 8, 135, 24]. In [7], tracklet confidence is defined as an intuitive measure of similarity between the constructed trajectory and the true object trajectory. It is used to split tracklets into low- and high-confidence groups, which are then associated using different strategies.

In [20], the scene is divided into a $k \times k$ grid, and both detections and predicted bounding boxes are used as candidates for association. Tracklet confidence is defined and used

to estimate the probability of an object being located in a given image region and to filter out unreliable predictions.

In [8], the tracklet score is calculated from detection scores and used as a tracklet termination criterion. In [88], tracklet confidence is designed to detect occlusions. Since a low-confidence true-positive detection often indicates that the object is only partially visible or occluded, Hybrid-SORT [135] defines tracklet confidence as a measure of object visibility and predicts it as part of the object state. The difference between detection confidence and predicted tracklet confidence is then used as an additional similarity cue.

3.2.3 Similarity Measures

Various improvements and additions to IoU have been proposed to improve matching performance. In [62], Generalized IoU (GIoU) [98] is used. Normalized IoU is proposed in [88] to include differences between bounding box size and center. To account for object motion, a momentum term was added to IoU in [15]. Width and height information are used in several previous works, e.g., [66, 140, 71], and in [135], Height Modulated IoU is introduced to explicitly incorporate height similarity into the IoU matrix.

Since DeepSORT [130], using appearance features to associate detections with tracklets has become a popular approach in tracking-by-detection MOT [129, 3, 30, 123]. Specifically, the cosine distance between visual embedding vectors is used to construct the association cost matrix. Mahalanobis distance is used as a gating mechanism to discard inadmissible associations. Since uncertainty increases when a tracklet is not updated due to occlusion or missing detections, association is performed in a cascade, in increasing order of the number of steps since the last update [130]. Following DeepSORT, a weighted sum of Mahalanobis distance and cosine distance is used in several other works, e.g., [30, 3, 145]. In [70], Mahalanobis distance is smoothed by adding $\alpha \cdot \mathbf{I}$ to the covariance matrix when calculating the distance.

3.2.4 Relation to the Proposed Method

In [130, 71], recently updated tracklets, i.e., more reliable tracklets, are explicitly prioritized through cascade matching. On the other hand, methods that follow ByteTrack line of research (e.g., [125, 3, 75, 133, 62, 88]) use two-stage association strategies that prioritize high-confidence detections.

Our use of detection-tracklet confidence can be viewed as a combination of these two ideas. BoostTrack prioritizes both high-confidence detections and reliable tracklets, but does so implicitly within a simpler one-stage association framework. Instead of first

matching selected groups of detections or tracklets, detection-tracklet confidence is incorporated directly into the similarity score.

A closely related idea was independently explored in FastTrack [64], where detection and tracklet confidence are also used to prioritize reliable associations. BoostTrack follows a similar motivation of favoring reliable detection-tracklet pairs in one-stage association. However, the role of detection-tracklet confidence is different. In BoostTrack, the similarity of reliable detection-tracklet pairs is only moderately increased, i.e., boosted, and the full aggregate similarity score is not scaled entirely by detection-tracklet confidence.

We adopt and modify shape similarity from [66]. In [66] and [140], shape similarity, i.e., similarity between height and width, is used together with Euclidean distance and visual similarity to construct an association cost matrix, and it cannot be used as a standalone metric or as a direct addition to the association cost. In our work, shape similarity is used together with detection-tracklet confidence to create a standalone addition to the similarity matrix, designed to reduce possible ambiguity arising from the IoU measure.

Prior works used Mahalanobis distance directly to define the similarity matrix [30, 145] or to discard inadmissible associations [130]. In our work, we convert Mahalanobis distance into probabilities, creating a more intuitive, robust, and general similarity measure.

In [135], Height Modulated IoU, velocity cost, and confidence cost are added to the appearance cost to create the final cost matrix used for association. In our terminology, this can be seen as boosting the appearance similarity by adding additional lightweight measures. In our work, we propose and use different similarity measures.

In [110], one-stage association is performed using all detection boxes, and a normalization parameter β is used to control the association cost of low-confidence detections. Our detection confidence boosting strategy does not use all detection boxes in the association step or require any change in the construction of the assignment cost matrix. Instead, it increases the confidence of selected boxes before filtering out low-confidence detections.

To the best of our knowledge, no prior work explicitly targeted and used detections that *seem to be* outliers. Our detection confidence boosting method attempts to use more true-positive detection bounding boxes without the need for sequence-specific hyperparameter tuning used in some previous works, e.g., [144, 3, 110].

3.3 Similarity Matrix Boosting Techniques

In this section, we introduce our similarity matrix boosting techniques. The proposed improvements are orthogonal to existing approaches and can be added to any base similarity

matrix $\mathbf{S}_{\text{base}}$ to improve association performance. For example, $\mathbf{S}_{\text{base}}$ can be calculated as IoU between detected bounding boxes and existing tracklets, as in Eq. (2.13), or it can combine IoU and appearance similarity, as in Eq. (2.17). For simplicity, we omit the frame index t throughout this section.

3.3.1 Detection-Tracklet Confidence Similarity Boost

To benefit from hierarchical association strategies that favor high-confidence detections [144, 125, 3, 75, 133, 62] or recently updated tracklets [130, 71], while avoiding their drawbacks, we design detection-tracklet confidence as a scaling factor that favors high-confidence detection, high-confidence tracklet pairs in one-stage association.

Let $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ and $\hat{\mathcal{T}} = \{\hat{T}_1, \hat{T}_2, \dots, \hat{T}_m\}$ be the set of detections and the set of predicted tracklet bounding boxes, respectively, and let $c_{D_1}, c_{D_2}, \dots, c_{D_n}$ and $c_{T_1}, c_{T_2}, \dots, c_{T_m}$ denote their corresponding confidence scores. The predicted bounding boxes $\hat{T}_1, \hat{T}_2, \dots, \hat{T}_m$ are obtained as outputs of the Kalman prediction step.

We assume that recently updated tracklets, i.e., active tracklets that were recently assigned a detection and updated using the Kalman filter, should have more reliable predictions and higher confidence. Due to initially noisy predictions, new tracklets should also have lower confidence. Let $a_j = \text{age}(T_j)$ and $u_j = \text{last_update}(T_j)$ be the number of steps since creation and the number of steps since the last update of tracklet T_j , respectively. We define tracklet confidence c_{T_j} as

$$c_{T_j} = \begin{cases} \beta^{s_{\text{init}} - a_j}, & \text{if } a_j < s_{\text{init}}, \\ \beta^{u_j - 1}, & \text{otherwise,} \end{cases} \quad (3.1)$$

for $j \in 1, 2, \dots, m$. Here, $\beta \in (0, 1)$ is the tracklet confidence decay hyperparameter, and s_{init} is the number of steps during which a tracklet is considered new, i.e., during which its predictions are considered initially unreliable.

We define detection-tracklet confidence of detection D_i and tracklet T_j , denoted by c_{D_i, T_j} , as the product $c_{D_i} \cdot c_{T_j}$. To encourage admissible associations only, e.g., $\text{IoU}(D_i, \hat{T}_j) \geq \tau_{\text{IoU}}$, we set c_{D_i, T_j} to 0 for inadmissible associations. In summary, we define c_{D_i, T_j} as

$$c_{D_i, T_j} = \begin{cases} c_{D_i} \cdot c_{T_j}, & \text{if } (D_i, \hat{T}_j) \text{ is admissible,} \\ 0, & \text{otherwise.} \end{cases} \quad (3.2)$$

We define the confidence matrix $\mathbf{C}$ as $\mathbf{C} = [c_{D_i, T_j}]_{n \times m}$.

Using the detection-tracklet confidence scores, we can increase the similarity between high-confidence detections and high-confidence tracklets, i.e., boost the similarity score.

This is done by adding a confidence-scaled IoU similarity term to an existing similarity matrix $\mathbf{S}_{\text{base}}$. Let $\mathbf{S}_{\text{IoU}} = \text{IoU}(\mathcal{D}, \hat{T})$ denote the IoU similarity matrix. The resulting boosted similarity matrix $\mathbf{S}_{\text{boost}}$ is obtained as

$$\mathbf{S}_{\text{boost}} = \mathbf{S}_{\text{base}} + \lambda_{\text{IoU}} \cdot \mathbf{C} \odot \mathbf{S}_{\text{IoU}}, \quad (3.3)$$

where $\odot$ denotes the element-wise matrix product, λ_{IoU} is a hyperparameter. The same boosting strategy can be applied to any similarity matrix, not only to IoU.

3.3.2 Mahalanobis Distance Similarity Boost

Mahalanobis distance [76] is used as a similarity measure in several previous MOT works, e.g., [30, 145, 3].

The Kalman filter, used here as a state estimator, provides both a state estimate and the corresponding covariance estimate. If process and observation noise are assumed to be Gaussian, the predicted state and covariance define a Gaussian distribution. Mahalanobis distance then provides a natural way to measure the compatibility between a predicted tracklet state and a detection.

However, raw Mahalanobis distance values are not directly normalized as similarity scores. Their admissible range depends on the number of degrees of freedom and the chosen confidence interval. For example, with 4 degrees of freedom and a 95% confidence interval, the usual gating threshold is 9.4877 [30, 145, 3]. Changing the state representation or the confidence interval changes this range and may require different weighting when Mahalanobis distance is combined with other similarity terms.

For association, only the relative differences between detections for a given tracklet are important. Therefore, we convert Mahalanobis distances into normalized similarity scores. For each tracklet, distances are first clipped to a maximum admissible value. Then, each distance is subtracted from this limit, so that smaller distances obtain larger scores. Finally, softmax is applied separately for each column of the similarity matrix. Detections whose Mahalanobis distance exceeds the limit are assigned similarity 0, preventing inadmissible associations. Pseudocode of this procedure is given in Algorithm 4 in Appendix A.

After obtaining the $\mathbf{S}_{\text{M}}$ similarity matrix in the described way, we can boost the initial similarity measure $\mathbf{S}_{\text{base}}$ by

$$\mathbf{S}_{\text{boost}} = \mathbf{S}_{\text{base}} + \lambda_{\text{M}} \cdot \mathbf{S}_{\text{M}}, \quad (3.4)$$

where λ_{M} denotes the weight of the Mahalanobis distance similarity boost.

Normalizing Mahalanobis distances makes the similarity term less sensitive to changes in dimensionality, since only the admissible distance threshold needs to be adjusted. It also

makes the resulting scores easier to combine with other similarity measures. When several detections have similar Mahalanobis distances to a given tracklet, the softmax normalization reduces the influence of Mahalanobis distance similarity and allows $\mathbf{S}_{\text{base}}$ to remain decisive. Conversely, when $\mathbf{S}_{\text{base}}$ is ambiguous, $\mathbf{S}_M$ can provide additional information for correct association. The softmax temperature could be used for further control in ambiguous cases, but we keep it set to 1 for simplicity.

3.3.3 Shape Similarity Boost

To reduce possible ambiguity in other similarity metrics, such as IoU, we reintroduce the shape similarity metric (shape-based similarities have been used previously, e.g., in [140, 66]). Consider a scenario where two objects are highly overlapped, e.g., pedestrians passing by one another. The corresponding tracklets can have greater IoU with the wrong detections, leading to identity switches. However, in a short time frame, the shape of an object, represented by its width and height, should remain relatively stable, and using shape information can reduce association ambiguity.

However, it is our premise that we should not rely too much on the shape information from a tracklet that was not recently updated. For example, a person may move their hands and appear wider, but only temporarily. Even height can change due to the object moving closer to or farther from the camera. On the other hand, detection confidence should also be considered because comparing shapes with unreliable detections can reduce the reliability of the shape similarity. To account for this, we scale shape similarity by the detection-tracklet confidence scores introduced in Section 3.3.1.

Let $\Delta_{i,j}^{\text{shape}}$ be the shape difference between detection D_i and predicted tracklet $\hat{T}_j$, defined as

$$\Delta_{i,j}^{\text{shape}} = \frac{|w_{D_i} - w_{\hat{T}_j}|}{\max(w_{D_i}, w_{\hat{T}_j})} + \frac{|h_{D_i} - h_{\hat{T}_j}|}{\max(h_{D_i}, h_{\hat{T}_j})}, \quad (3.5)$$

We define the shape similarity measure between detection D_i and tracklet $\hat{T}_j$ as

$$S_{\text{shape}}(D_i, \hat{T}_j) = c_{D_i, \hat{T}_j} \cdot \exp(-\Delta_{i,j}^{\text{shape}}). \quad (3.6)$$

We can boost the similarity measure $\mathbf{S}_{\text{base}}$ by

$$\mathbf{S}_{\text{boost}} = \mathbf{S}_{\text{base}} + \lambda_{\text{shape}} \cdot \mathbf{S}_{\text{shape}}, \quad (3.7)$$

where $\mathbf{S}_{\text{shape}} = [S_{\text{shape}}(D_i, \hat{T}_j)]_{n \times m}$, and λ_{shape} is a hyperparameter used as the weight of shape similarity.

Combining the three proposed similarity boosting techniques, we obtain

$$\begin{aligned} \mathbf{S}_{\text{boost}} = & \mathbf{S}_{\text{base}} + \lambda_{\text{IoU}} \cdot \mathbf{C} \odot \mathbf{S}_{\text{IoU}} \\ & + \lambda_{\text{M}} \cdot \mathbf{S}_{\text{M}} \\ & + \lambda_{\text{shape}} \cdot \mathbf{S}_{\text{shape}}. \end{aligned} \quad (3.8)$$

3.4 Detection Confidence Boosting Techniques

Not all low-confidence detections are false positives. In this section, we describe our proposed methods for utilizing two groups of low-confidence detections by boosting their confidence scores.

3.4.1 Detecting Likely Objects

When an object is partially occluded, it can sometimes still be detected. The detection confidence of such partially visible objects can be low, and their detected bounding boxes can be discarded for having confidence scores below the threshold τ_{det} . However, having a tracking module, i.e., a Kalman filter, enables us to predict where the tracked objects should be.

If the IoU score between the tracklet and the detection box is high, we propose to increase the confidence of that detection box, enabling it to be used for subsequent association. For each detection box D_i , we calculate IoU between the detection box and all tracklets and increase the confidence of detection D_i using the maximum value of the calculated IoUs. The boosted confidence $\hat{c}_{D_i}$ is obtained as

$$\hat{c}_{D_i} = \max \left(c_{D_i}, \beta_c \cdot \max_j (\text{IoU}(D_i, \hat{T}_j)) \right). \quad (3.9)$$

Hyperparameters β_c and τ_{det} implicitly define the IoU threshold for detections to be used for association, even if the original detection confidence c_{D_i} is low. In this way, we also increase the confidence of some detections for which $c_{D_i} > \tau_{\text{det}}$, which we found to slightly improve performance when combined with detection-tracklet confidence from Section 3.3.1.

Figure 3.2 shows an example of a highly occluded person with a low-confidence detected bounding box, shown in blue, with $c_D = 0.175$, that has high IoU, 0.95, with a predicted bounding box. Our method increases the confidence of this detection and uses it for association.



Figure 3.2: Detecting likely objects based on IoU value. Predicted tracklet bounding boxes are shown in red. The blue bounding box is the detection with the low confidence score, which is increased based on high IoU with the predicted bounding box.

3.4.2 Detecting “Unlikely” Objects

The previous confidence boosting strategy aims to increase the detection confidence for detections located where a tracked object is likely to be. However, some objects may not have been detected before. They can be partially occluded throughout the video or positioned at the edge of the scene and only partially visible. We propose a method to detect some of these objects. False-positive low-confidence detections often occur near tracked objects. Therefore, if a low-confidence detection is located *far* from where any tracked object is expected to be, we can assume that it is unlikely to be a duplicate or noisy detection associated with a currently tracked object. Such outliers may correspond to previously undetected objects.

Mahalanobis distance can be used to detect outliers [39]. As previously noted, Mahalanobis distance values are chi-squared distributed, and values greater than a certain threshold τ_{MhD} are considered outliers. We set the threshold τ_{MhD} to 13.2767, corresponding to the 99% confidence interval boundary for the chi-squared distribution with 4 degrees of freedom.

For a given detection D_i , we compute the distance between D_i and every tracklet $\hat{T}_j$ and consider D_i an outlier if the distance between D_i and the closest tracklet is greater than τ_{MhD} , i.e., if

$$\min_j (\text{MhD}(D_i, \hat{T}_j)) > \tau_{\text{MhD}}. \quad (3.10)$$

If a given detection D_i is an outlier with respect to the state distributions of all currently

tracked objects, we consider it a candidate for a previously undetected object.

However, some of these outliers can still be false positives. For all detections satisfying Eq. (3.10), we apply non-maximum suppression with a threshold of $\tau_{\text{NMS}} = 0.3$ to remove overlapping detections and reduce the number of used false-positive detections. We provide details on the influence of τ_{NMS} in Appendix A.2. We set the detection confidence of the remaining detections to τ_{det} , allowing them to be used in the association step.

Figure 3.3 shows an example of a low-confidence detection with original detection confidence $c_D = 0.225$. The closest Mahalanobis distance to the existing tracklets is 2178.23, and we assume that this detection is unlikely to be a false positive generated by currently tracked objects. Therefore, we increase its detection confidence.

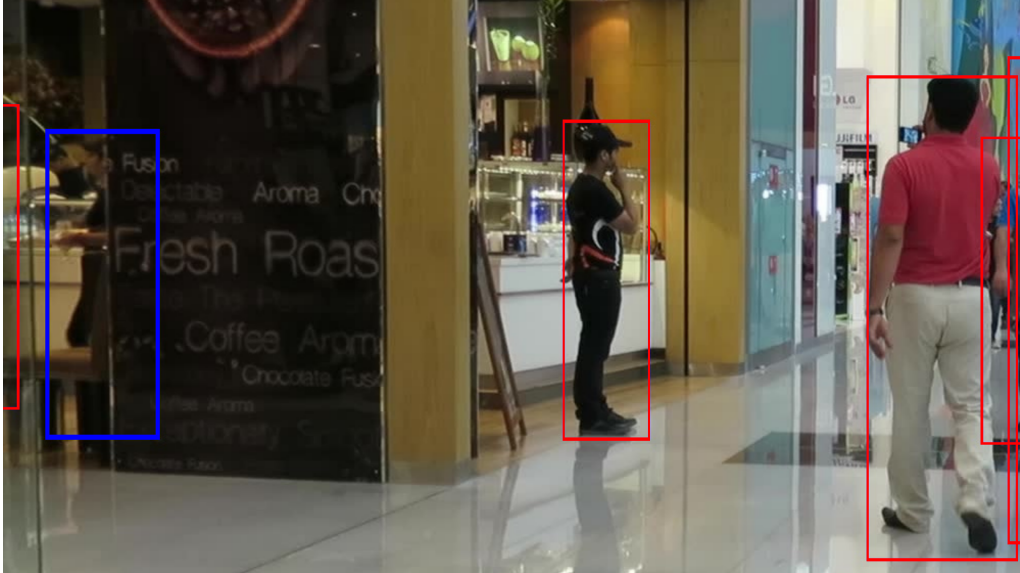


Figure 3.3: Detecting “unlikely” objects based on Mahalanobis distance. Red bounding boxes represent tracklets. The blue bounding box is a detection with the low confidence score. However, its confidence is increased based on its high Mahalanobis distance to all existing tracklets.

3.5 Experiments

3.5.1 Experimental Setup

We evaluate BoostTrack on MOT17 [84] and MOT20 [25]. As discussed in Section 2.9, these datasets are suitable for evaluating detection confidence handling, association robustness, and tracking in crowded pedestrian scenes. Ablation studies are performed on the validation splits, while final comparisons are reported on the official test sets.

As in several popular tracking methods [144, 3, 30, 75], we replace the original dataset detections with detections from YOLOX-X [38] and conduct experiments under the private detection protocol.

The evaluation metrics used in this chapter are introduced in Section 2.8. We evaluate our method using HOTA, IDF1, and MOTA metrics, and we show IDSW as an additional and more intuitive measure of association performance.

3.5.2 Implementation Details

Kalman filter. As in [130, 145, 30, 144], we define the Kalman filter state as the eight-dimensional vector

$$\mathbf{x} = [u, v, h, r, \dot{u}, \dot{v}, \dot{h}, \dot{r}]^T, \quad (3.11)$$

where (u, v) denote the coordinates of the bounding box center, h is the height, and r is the aspect ratio, while $\dot{u}, \dot{v}, \dot{h}, \dot{r}$ represent the corresponding velocities. We retain the constant process and measurement noise from [10].¹

MOT-specific settings. As in [10], we report a tracklet only after 3 consecutive matches, i.e., Kalman update steps, and use $\tau_{\text{IoU}} = 0.3$ as the criterion for discarding inadmissible associations. As in [15, 75], we set the detection confidence threshold τ_{det} to 0.6 for MOT17 and 0.4 for MOT20.

Since our detection confidence boosting techniques rely on predicted tracklet positions, we keep unassociated tracklets alive longer than in several previous works, where $A_{\text{max}} = 30$ frames is commonly used [130, 144, 3]. Because different sequences have different frame rates, we use the sequence-specific value

$$A_{\text{max}} = \max(30, 2 \cdot \text{FPS}), \quad (3.12)$$

i.e., we keep unassociated tracklets alive for 2 seconds.

BoostTrack-specific settings. As the base similarity measure, we use IoU, i.e., $S_{\text{base}} = \text{IoU}$ in Eq. (3.8).

We run a grid search to find values of λ_{IoU} , λ_{M} , and λ_{shape} . For each λ , we test values from the set $\{0, 0.25, 0.5, 0.75, 1\}$. As the best trade-off between different metrics on MOT17 and MOT20, we choose

$$\lambda_{\text{IoU}} = 0.5, \quad \lambda_{\text{M}} = 0.25, \quad \lambda_{\text{shape}} = 0.25.$$

¹SORT uses a 7-dimensional state, and we adapt it to the 8-dimensional setting. We lower the variance of the aspect ratio measurement noise to 0.01, as in [130], and increase the process noise variance corresponding to $\dot{r}$ to 0.01.

We observed that any setting where λ_M is not dominant improves performance compared to the baseline.

We test various (β, s_{init}) settings for $(\beta, s_{\text{init}}) \in [0.7, 0.95] \times \{0, 1, \dots, 10\}$ on the MOT17 validation set. We found that all tested settings improve the baseline performance and set $\beta = 0.9$ and $s_{\text{init}} = 7$ as the best trade-off between different metrics. More details on the influence of β and s_{init} are provided in Appendix A.3.

We set the limit value for Mahalanobis distance similarity boosting and outlier detection to 13.2767, corresponding to the 99% confidence interval boundary for the chi-squared distribution with 4 degrees of freedom. For boosting the detection confidence of likely objects, we set $\beta_c = 0.65$ for MOT17 and $\beta_c = 0.5$ for MOT20, corresponding to the required IoU values of 0.923 and 0.8 for a low-confidence detection to surpass the threshold τ_{det} .

We evaluate the results using TrackEval [51].

Additional modules. For YOLOX-X [38], we use the weights from [144]. As in [144, 75, 3], in the post-processing step of the detection, detections with confidence score below 0.1 are discarded, and Non-Maximum Suppression with $\tau_{\text{NMS}} = 0.7$ is applied to detector output to remove duplicate detections. We use Enhanced Correlation Coefficient maximization [32] for camera motion compensation (CMC) and rely on the implementation from [30]. We keep the same settings as in [30], but resize images to 350 pixels in width, with proportional height.

As in [144] we use linear interpolation, but we do not set a maximum interval for interpolation. Tracks with fewer than 25 detections are not interpolated, following the default value from [144]. We further improve interpolation results by applying gradient boosting smoothing from [142].

As in [144, 3, 75], we use FastReID [45] as a visual embedding model and apply Dynamic Appearance embedding update from [75]. When using appearance similarity, we add $\lambda_{\text{app}} \cdot \mathbf{S}_{\text{app}}$ (see Eq. (2.15)) to the overall similarity measure proposed in Eq. (3.8). Since the total weight of the non-appearance similarity components is 2, we set $\lambda_{\text{app}} = 3$ to give more weight to appearance similarity. In addition to the admissibility condition $\text{IoU}(D_i, \hat{T}_j) \geq \tau_{\text{IoU}}$, we allow associations between detection box D_i and tracklet $\hat{T}_j$ if

$$\text{IoU}(D_i, \hat{T}_j) \geq \frac{\tau_{\text{IoU}}}{2} \wedge S_{\text{app}}(D_i, \hat{T}_j) \geq \frac{3}{4}. \quad (3.13)$$

Hardware and software. We run all experiments on a desktop computer with an AMD Ryzen 9 5950X 16-Core CPU and an NVIDIA GeForce RTX 3090 GPU. Our implementation is developed on top of publicly available codebases [75, 10, 144, 30, 142].

3.5.3 Ablation Study

Similarity boost To test the impact of each component of the proposed similarity boosting technique, we conduct a detailed ablation study on the MOT17 and MOT20 validation sets. We evaluate all combinations of the proposed components: detection-tracklet confidence (DTC) boost, Mahalanobis distance (MhD) boost, and shape similarity boost. The baseline corresponds to the case where $\mathbf{S} = \text{IoU}(\mathcal{D}, \hat{T})$. We set $\lambda_{\text{IoU}} = 0.5$, $\lambda_{\text{MhD}} = 0.25$, and $\lambda_{\text{shape}} = 0.25$, as described in Section 3.5.2.

The results show that DTC and shape similarity improve performance both as standalone additions and when combined. Adding the MhD similarity boost alone slightly decreases the overall performance, but when combined with DTC and shape similarity, it contributes to a significant performance gain on both MOT17 and MOT20. Since similarity boosting primarily targets association quality, IDF1 is the most relevant metric for evaluating its effect. Compared to the IoU-only baseline, the full similarity boosting method improves IDF1 by +1.22 on MOT17 and +0.54 on MOT20, while also improving the other main metrics.

Table 3.1: Ablation study on the MOT17 and MOT20 validation sets for different similarity boosting settings. Best results are shown in bold.

Setting			MOT17				MOT20			
DTC	MhD	Shape	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
$\times$	$\times$	$\times$	66.04	77.15	74.72	231	56.65	73.60	69.91	1127
$\times$	$\times$	$\checkmark$	66.69	78.34	75.00	181	56.92	74.00	69.96	1095
$\times$	$\checkmark$	$\times$	65.92	77.14	74.50	250	55.97	72.63	69.62	1521
$\times$	$\checkmark$	$\checkmark$	66.32	77.73	74.84	208	56.99	74.14	69.85	1262
$\checkmark$	$\times$	$\times$	66.21	77.90	74.87	189	56.96	73.92	69.96	1110
$\checkmark$	$\times$	$\checkmark$	66.28	78.21	74.99	180	57.02	74.00	69.96	1117
$\checkmark$	$\checkmark$	$\times$	66.39	77.87	74.90	202	57.20	74.40	69.93	1149
$\checkmark$	$\checkmark$	$\checkmark$	66.70	78.37	74.95	174	57.19	74.26	69.96	1128

Detection confidence boost. We also test the effectiveness of the proposed detection confidence boosting (DCB) techniques. We evaluate detecting likely objects (DLO) and detecting “unlikely” objects (DUO), both independently and in combination with the similarity boosting (SB) components. When SB is used, all three similarity boosting components are included. We present results in Table 3.2.

The results show that both DCB techniques improve MOT performance, both as standalone additions and when combined with SB. Since DCB introduces additional detections, while SB improves association, HOTA is especially useful for summarizing their joint effect. Without SB, DCB improves HOTA by +0.56 on MOT17 and +1.96 on

MOT20. When combined with SB, the overall gain becomes +1.42 HOTA on MOT17 and +2.32 HOTA on MOT20. This indicates that the proposed boosting techniques complement each other and can be used jointly. The combination also leads to clear improvements in IDF1 and MOTA.

Table 3.2: Ablation study on the MOT17 and MOT20 validation sets for different detection confidence boost settings (best in bold).

Setting			MOT17				MOT20			
SB	DLO	DUO	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
X	X	X	66.04	77.15	74.72	231	56.65	73.60	69.91	1127
X	X	✓	66.27	77.44	74.82	227	56.66	73.62	69.92	1130
X	✓	X	66.31	77.40	74.89	225	58.58	75.09	73.28	1259
X	✓	✓	66.60	77.67	74.93	221	58.61	75.12	73.25	1258
✓	X	X	66.70	78.37	74.95	174	57.19	74.26	69.96	1128
✓	X	✓	66.98	78.81	75.03	169	57.20	74.29	69.96	1128
✓	✓	X	66.94	78.65	75.14	173	58.98	75.77	73.36	1227
✓	✓	✓	67.46	79.42	74.99	168	58.97	75.72	73.32	1228

Additional modules. Finally, we evaluate the effect of additional modules that are commonly used in modern MOT systems. Starting from the proposed similarity and detection confidence boosting components (SB+DCB), we additionally consider camera motion compensation (CMC), appearance similarity (AS), and post-processing (PP). For post-processing, we use linear interpolation and gradient boosting smoothing from [142]. The results are shown in Table 3.3.

Runtime was not reported in the previous ablation tables because the corresponding settings have comparable speed. However, CMC and AS introduce additional computational cost, and we therefore also report FPS in this experiment. The reported FPS corresponds to online inference and does not include offline post-processing.

The first two rows of Table 3.3 summarize the effect of the proposed BoostTrack components before adding external modules. We already discussed the influence of these components and presented ablation details in Tables 3.1 and 3.2.

Adding CMC further improves performance on MOT17, where camera motion is more common, increasing HOTA from 67.46 to 68.37 and IDF1 from 79.42 to 81.35. The improvement on MOT20 is smaller, which is expected because MOT20 sequences are filmed mostly with static cameras. However, CMC reduces runtime substantially, from 340.00 FPS to 65.45 FPS on MOT17 and from 49.79 FPS to 32.79 FPS on MOT20.

Adding appearance similarity improves association accuracy but significantly reduces runtime. Without post-processing, AS improves IDF1 from 81.35 to 82.42 on MOT17 and from 75.96 to 77.20 on MOT20. However, FPS drops to 15.35 on MOT17 and 3.05 on

Table 3.3: Ablation study on the MOT17 and MOT20 validation sets for additional modules. The fifth row corresponds to BoostTrack, while the last row corresponds to BoostTrack+. Best results are shown in bold.

Setting				MOT17					MOT20				
SB+DCB	CMC	AS	PP	HOTA	IDF1	MOTA	IDSW	FPS	HOTA	IDF1	MOTA	IDSW	FPS
×	×	×	×	66.04	77.15	74.72	231	359.2	56.65	73.60	69.91	1127	52.7
✓	×	×	×	67.46	79.42	74.99	168	340.00	58.97	75.72	73.32	1228	49.79
✓	✓	×	×	68.37	81.35	75.56	118	65.45	59.16	75.96	73.37	1237	32.79
✓	✓	✓	×	69.18	82.42	75.53	119	15.35	59.82	77.20	73.42	1118	3.05
✓	✓	×	✓	71.33	83.84	80.55	106	65.45	61.76	77.49	77.43	903	32.79
✓	✓	✓	✓	72.31	85.18	80.90	81	15.35	62.59	78.95	77.66	801	3.05

MOT20, showing that appearance extraction is the main computational bottleneck.

These results suggest that, on MOT17, the proposed boosting components reduce the dependence on appearance similarity, as strong association performance is achieved without AS. Adding AS further improves accuracy, but at a substantial runtime cost. On MOT20, where crowded scenes make association more ambiguous, appearance information remains more beneficial.

Post-processing provides a large performance gain, especially in MOTA and HOTA. For the appearance-free setting, adding PP improves HOTA from 68.37 to 71.33 on MOT17 and from 59.16 to 61.76 on MOT20. It also reduces IDSW from 1237 to 903 on MOT20. Since PP is applied after online inference, it does not affect the reported FPS.

The fifth row of Table 3.3 corresponds to the appearance-free BoostTrack variant, while the last row corresponds to BoostTrack+, which additionally uses appearance similarity. BoostTrack preserves real-time speed, reaching 65.45 FPS on MOT17 and 32.79 FPS on MOT20. BoostTrack+ achieves the best accuracy, with 72.31 HOTA and 85.18 IDF1 on MOT17, and 62.59 HOTA and 78.95 IDF1 on MOT20, but at a much lower speed. This highlights the trade-off between accuracy and efficiency: BoostTrack is suitable when real-time appearance-free tracking is required, while BoostTrack+ is preferable when higher accuracy is prioritized over runtime.

3.5.4 Comparison with Contemporary Benchmark Methods

We compare BoostTrack with strong contemporary MOT methods on the MOT17 and MOT20 test sets under the private detection protocol. Following the classification introduced in Section 2.7, we group the compared methods into strictly online methods, online methods with post-processing, and offline methods. The comparison follows the experimental context of the original BoostTrack study and includes strong representative methods that were available at the time of its publication. Later methods, including the

subsequent contributions presented in the following chapters of this dissertation, are not included in the comparison.

In Table 3.4 we report the results of BoostTrack, BoostTrack+ and contemporary benchmark methods on MOT17 and MOT20 test sets.

Table 3.4: Comparison with contemporary benchmark methods on the MOT17 and MOT20 test sets under the private detection protocol. Methods are grouped according to their temporal processing type.

Method	MOT17				MOT20			
	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
<i>Strictly online methods</i>								
FairMOT [145]	59.3	72.3	73.7	3303	54.6	67.3	61.8	5243
MOTR [141]	62.0	75.0	78.6	2619	/	/	/	/
GHOST [103]	62.8	77.1	78.7	2325	61.2	75.2	73.7	1264
QuoVadis [26]	63.1	77.7	80.3	2103	61.5	75.7	77.8	1187
FineTrack [97]	64.3	79.5	80.0	1272	63.6	79.0	77.9	980
MotionTrack [95]	65.1	80.1	81.1	1140	62.8	76.5	78.0	1165
SuppTrack [146]	/	/	/	/	61.9	75.5	78.2	1325
UTM [138]	/	/	/	/	62.5	76.9	78.2	1228
<i>Online methods with post-processing</i>								
ByteTrack [144]	63.1	77.3	80.3	2196	61.3	75.2	77.8	1223
OC-SORT [15]	63.2	77.5	78.0	1950	62.1	75.9	75.5	913
StrongSORT++ [30]	64.4	79.5	79.6	1194	62.6	77.0	73.8	770
BASE [59]	64.5	78.6	81.9	1281	63.5	77.6	78.2	984
Deep OC-SORT [75]	64.9	80.6	79.4	1023	63.9	79.2	75.6	779
BoT-SORT [3]	65.0	80.2	80.5	1212	63.3	77.5	77.8	1313
SparseTrack [69]	65.1	80.1	81.0	1170	63.5	77.6	78.1	1120
ConfTrack [52]	65.4	80.0	81.2	1155	64.8	80.2	77.2	702
LG-Track [81]	65.4	80.4	81.4	1125	63.4	77.4	77.8	1161
StrongTBD [109]	65.6	80.8	81.6	954	63.6	77.0	78.0	1101
PIA2 [112]	66.0	81.1	82.2	1026	64.7	79.0	78.5	1023
ImprAsso [110]	66.4	82.1	82.2	924	64.6	78.8	78.6	992
BoostTrack	65.4	80.2	80.5	1104	63.0	76.5	76.4	992
BoostTrack+	66.4	81.8	80.6	1086	66.2	81.5	77.2	899
<i>Offline methods</i>								
SUSHI [17]	66.5	83.1	81.1	1149	64.3	79.8	74.3	706
CoNo-Link [37]	67.1	83.7	82.7	1092	65.9	81.8	77.5	956

Table 3.4 shows that BoostTrack achieves competitive performance among online methods with post-processing while remaining appearance-free. On MOT17, BoostTrack obtains 65.4 HOTA and 80.2 IDF1, which is comparable to several strong post-processed trackers. On MOT20, where scenes are more crowded and association is more difficult,

the appearance-free variant is less competitive than the strongest methods, suggesting that dense scenes benefit from additional association cues.

BoostTrack+ further improves the results by incorporating appearance similarity into the aggregate association score. On MOT17, it reaches 66.4 HOTA, matching the best online+PP result in the table, and obtains 81.8 IDF1. Although offline methods still achieve slightly higher scores on MOT17, BoostTrack+ remains competitive with both online+PP and offline methods.

The improvement is more pronounced on MOT20. BoostTrack+ achieves the best HOTA among all listed methods, and the best IDF1 among online+PP methods. It also outperforms the listed offline methods in HOTA and is only slightly below CoNo-Link in IDF1. These results indicate that the proposed boosting components are compatible with appearance similarity and that their combination is especially effective in crowded scenes.

3.6 Conclusion

In this chapter, we presented BoostTrack, a tracking-by-detection method designed to improve detection handling and data association while preserving a simple one-stage association framework. The method introduces two groups of additions. The first group consists of similarity boosting components: detection-tracklet confidence, Mahalanobis distance similarity, and shape similarity. The second group consists of detection confidence boosting techniques, which increase the confidence of selected low-confidence detections before thresholding.

The proposed method follows the broader idea of this dissertation: the components of a tracking-by-detection pipeline should not be treated as independent modules, but as mutually informative parts of the same system. In BoostTrack, tracking information is used to reinterpret detection confidence and recover useful low-confidence detections. At the same time, detection confidence and tracklet confidence influence association through detection-tracklet confidence. In this way, detection, tracking, and association exchange information through lightweight mechanisms without introducing a complex multi-stage pipeline.

The experimental results on MOT17 and MOT20 show that the proposed components improve both association quality and overall tracking performance. Without appearance similarity, BoostTrack achieves 71.33 HOTA, 83.84 IDF1, and 80.55 MOTA on MOT17, while running at 65.45 FPS. On MOT20, it achieves 61.76 HOTA, 77.49 IDF1, and 77.43 MOTA, with a runtime of 32.79 FPS.

Adding appearance similarity further improves accuracy, resulting in BoostTrack+. Compared with the appearance-free variant, BoostTrack+ improves HOTA/IDF1 by +0.98/+

1.34 on MOT17 and by +0.83/ + 1.46 on MOT20. However, this accuracy gain comes at a clear computational cost, reducing the runtime to 15.35 FPS on MOT17 and 3.05 FPS on MOT20, thus limiting real-time usage.

The next chapter extends this direction by further investigating how tracklet information can be used to recover true-positive low-confidence detections. BoostTrack++ develops a more detailed tracklet-guided strategy for detection recovery.

Chapter 4

BoostTrack++: Tracklet-Guided Recovery of True-Positive Detections

4.1 Introduction

In the previous chapter, we presented BoostTrack, which showed that low-confidence detections should not always be discarded before association. Instead, selected low-confidence detections can be recovered by using information from existing tracklets. This idea follows the broader direction of the dissertation: the selection of true-positive detections should not be performed in isolation, but together with tracking and association information accumulated over time.

BoostTrack uses detection confidence boosting to increase the confidence of detections that are likely to correspond either to already tracked objects or to previously undetected objects. In particular, the detecting likely objects (DLO) component uses the overlap between a low-confidence detection and predicted tracklet bounding boxes. If a detection has sufficiently high IoU with an existing tracklet prediction, its confidence is increased before thresholding, allowing it to be used in the same one-stage association procedure as high-confidence detections.

This chapter builds on that idea and investigates how the recovery of true-positive low-confidence detections can be made more reliable. BoostTrack uses IoU only for DLO confidence boost. However, IoU can be unreliable due to inaccurate motion prediction, irregular motion, or partial visibility. Moreover, the original detection confidence and the reliability of the tracklet prediction should also influence the recovery decision. A detection whose confidence is only slightly below the threshold should not necessarily require the same similarity to a tracklet as a detection whose confidence is close to zero. Similarly, the reliability of the tracklet prediction should influence how much similarity is required before the detection confidence is boosted.

BoostTrack++ addresses these issues by introducing a more detailed tracklet-guided strategy for recovering true-positive low-confidence detections. Instead of relying only on IoU, it combines multiple cues that were shown to be useful for association: Mahalanobis distance similarity, shape similarity, and a new soft Buffered IoU similarity. In addition, BoostTrack++ introduces soft detection confidence boosting, which takes the original detection confidence into account, and a varying boost threshold, which depends on the number of frames since the last tracklet update.

In addition to confidence boosting, this chapter also introduces a small offline post-processing component for trajectory splitting.¹ Since BoostTrack-based methods use interpolation to fill trajectory gaps, we introduce a KF-based trajectory splitting procedure that checks whether the two fragments around a gap are likely to correspond to the same object before interpolation is applied.

BoostTrack++ provides a more explicit example of the interaction between tracking and detection. Information accumulated during tracking, together with richer similarity cues, is used to improve detection handling and identify true-positive observations more reliably. The additional trajectory splitting component extends this idea to post-processing, where motion predictions are used to decide whether fragmented trajectories should be connected or separated.

The main contributions presented in this chapter are:

- a soft Buffered IoU similarity measure designed for comparing detections and tracklet predictions under uncertainty;
- a tracklet-guided detection recovery strategy that combines Mahalanobis distance similarity, shape similarity, and soft Buffered IoU;
- soft detection confidence boosting, which accounts for the original detection confidence score;
- a varying boost threshold based on the number of frames since the last update of a tracklet;
- a KF-based offline trajectory splitting procedure used as post-processing before interpolation and trajectory smoothing;
- an experimental analysis showing the effect of the proposed components on MOT17 and MOT20.

¹This post-processing component was not included in the original BoostTrack++ paper and is presented here as an additional dissertation contribution.

Since BoostTrack is described in detail in Chapter 3, and the DLO confidence boosting component in particular is introduced in Section 3.4.1, the rest of this chapter focuses only on the extensions proposed in BoostTrack++. In Section 4.2, we first discuss the limitations of the original DLO strategy. Then, in Section 4.3, we introduce the proposed confidence boosting components and the KF-based trajectory splitting procedure. Experimental results are presented in Section 4.4, and the chapter is concluded in Section 4.5.

4.2 Limitations of BoostTrack

This section summarizes the limitations of the DLO confidence boosting component introduced in Section 3.4.1. These limitations motivate the more detailed tracklet-guided recovery strategy proposed in this chapter.

DLO confidence boosting was shown to be useful for recovering low-confidence detections that likely correspond to already tracked objects. Its effect is especially visible on crowded sequences, where objects are frequently occluded and detection confidence can be unreliable. For example, on the MOT20 validation set, adding DLO to the SORT baseline increases MOTA from 69.91 to 73.28, which corresponds to a relative improvement of approximately 4.8% (see Table 4.1).

However, the original DLO strategy also has several limitations. To achieve the best results in BoostTrack, a different value of the hyperparameter β_c had to be specified depending on the dataset. In particular, β_c was set to 0.65 for MOT17 and 0.5 for MOT20, which implicitly corresponds to different required IoU values for a low-confidence detection to surpass the detection threshold. This indicates that the original confidence boosting rule is sensitive to the dataset and that a more adaptive mechanism is needed.

The most important practical issue is that DLO can also increase the number of produced identities and identity switches. Ideally, if a boosted detection truly corresponds to an existing tracklet, DLO should help update that tracklet without introducing new identities or additional identity switches. In practice, however, some boosted detections may be unreliable, may not be correctly associated, or may lead to additional tracklets. This issue is illustrated in Table 4.1, which shows the influence of the original DLO component on the MOT20 validation set for several baseline methods. The effect is particularly visible on MOT20 because it contains many more identities than the MOT17 validation set and includes significantly denser scenes.

For example, when DLO is added to SORT, HOTA, MOTA, and IDF1 all improve, but the number of identities increases by 151 (+8%) and the number of identity switches by 132 (+12%). Similar behavior can be observed for BoostTrack and BoostTrack+. This shows that the original DLO strategy is useful, but not sufficiently selective.

Table 4.1: Influence of DLO confidence boosting on the MOT20 validation set for different baseline methods.

Method	HOTA	MOTA	IDF1	IDSW	IDs
SORT	56.65	69.91	73.60	1127	1894
SORT + DLO	58.58	73.28	75.09	1259 (+132)	2045 (+151)
BoostTrack - DLO	61.39	77.02	77.23	803	1867
BoostTrack	61.74	77.46	77.45	898 (+95)	2008 (+141)
BoostTrack+ - DLO	62.59	77.16	79.29	730	1852
BoostTrack+	62.58	77.70	78.93	794 (+64)	2019 (+167)

Since DLO confidence boosting relies on selecting detections whose similarity to an existing tracklet is above a certain threshold, two natural sources of improvement are the similarity measure itself and the thresholding rule used for boosting.

First, the original boost decision is based only on IoU. This can be restrictive when the predicted tracklet position is inaccurate or when the object is only partially visible. In such cases, a true-positive low-confidence detection may not overlap sufficiently with the predicted bounding box, even though other cues, such as shape similarity or Mahalanobis distance, may suggest that it corresponds to the tracklet. If IoU alone can cause ambiguities during association, it can also be unreliable as the only criterion for detection confidence boosting.

Second, the original detection confidence is not explicitly considered when determining how much similarity is required for a successful boost. A detection with confidence close to zero and a detection with confidence slightly below the threshold are treated in the same way. In practice, the latter should require weaker tracklet support to be recovered, because the detector already provides partial evidence that the detection may be reliable.

Third, the reliability of the tracklet prediction is not explicitly reflected in the boost threshold. A recently updated tracklet usually has a more reliable prediction, while a tracklet that has not been updated for several frames may have accumulated prediction error. Therefore, requiring the same similarity for all tracklets may be too strict for less reliable predictions and may prevent useful detections from being recovered.

This limitation is also supported by the Kalman filter prediction model. When a tracklet is not matched with a detection, the observation is missed and the Kalman update step is not executed. As discussed in Section 2.3, the predicted covariance is computed as $\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^T + \mathbf{Q}_{t-1}$. For a constant velocity model, if we omit process noise for simplicity, after n prediction steps without an update, the variance of a state component

x , where $x \in \{u, v, h, r\}$, is given by

$$\text{Var}(x_{t+n}) = \text{Var}(x_t) + n^2 \text{Var}(\dot{x}_t) + 2n \text{Cov}(x_t, \dot{x}_t). \quad (4.1)$$

Prediction uncertainty increases with the number of missed updates, and the velocity uncertainty contributes quadratically with respect to the number of prediction steps. Therefore, the similarity required for confidence boosting should depend on how recently the corresponding tracklet was updated.

We empirically verify this behavior in Figure 4.1. The left part of the figure shows the average IoU value between the detections and predicted bounding boxes of the tracklets on the MOT17 and MOT20 validation sets. The values are shown as a function of the number of frames since the last successful tracklet update. The right part shows the corresponding 0.9 quantile of IoU values. The 0.9 quantile indicates what can be considered high IoU for tracklets with different numbers of missed updates. As the number of frames since the last update increases, both the average IoU and, more importantly, the 0.9 quantile decrease.

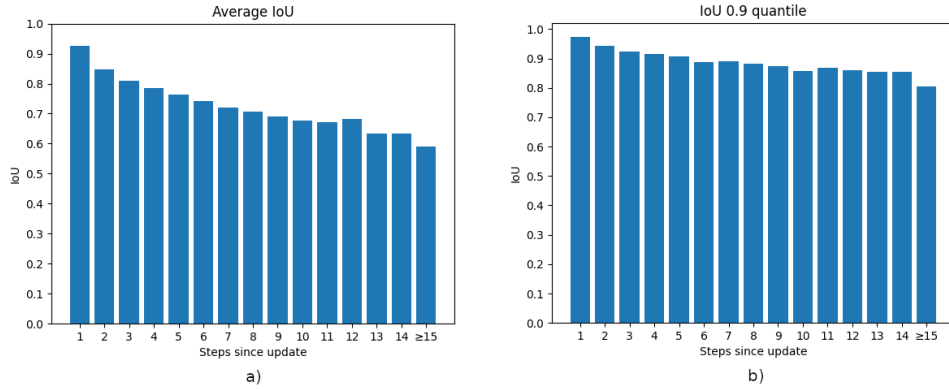


Figure 4.1: IoU statistics calculated on the MOT17 and MOT20 validation sets for tracklets with different numbers of frames since the last successful update. Left: average IoU. Right: 0.9 quantile of IoU values.

The data in Table 4.1, together with the covariance analysis and the empirical IoU statistics in Figure 4.1, indicate that a more sophisticated DLO confidence boosting strategy is needed. BoostTrack++ therefore aims to preserve the ability to recover true-positive low-confidence detections while reducing the risk of introducing unreliable detections, new identities, and identity switches. The following section introduces this strategy by improving the similarity measure, incorporating the original detection confidence, and using tracklet-specific varying thresholds.

4.3 Proposed Method

4.3.1 Soft BIoU

The aim of DLO confidence boosting is to locate low-confidence detections that are likely to correspond to existing tracklets. Matching unreliable detections with predicted tracklet bounding boxes can be improved by expanding the bounding boxes and thus increasing the IoU value between boxes that would otherwise have little or no overlap.

This idea was introduced in [133], where the resulting IoU between expanded bounding boxes is referred to as Buffered IoU (BIOU). We are interested in using the BIOU idea for DLO confidence boosting. However, BIOU was designed for two-stage association and cannot be directly adopted without modification. In the first stage, all bounding boxes are scaled by the value b_1 . The underlying assumption is that less ambiguous, “easier” pairs can be matched using a smaller scale b_1 , while the remaining, more “difficult” pairs are associated in the second stage, where a larger expansion scale, b_2 , is used.

A straightforward way of using BIOU in a one-stage association setting would be to use a single scale value b . However, not only do “easy” and “difficult” tracklets require different scale values, but not all “easy” tracklets are equally easy to match, and not all “difficult” tracklets are equally difficult.

Each predicted bounding box should be enlarged according to the uncertainty of the prediction. In BoostTrack, we introduced tracklet confidence as a measure of the reliability of the predicted tracklet state. We propose to use tracklet confidence to calculate a tracklet-specific scale without the need for two-stage association. However, if different scales are used for different tracklets in one-stage association, we cannot simply scale all tracklets and detections by the same parameter b and calculate IoU between all pairs, as done in [133]. Instead, every detection-tracklet pair $(D_i, \hat{T}_j)$ should use a scale based on the tracklet confidence c_{T_j} . In other words, instead of a discrete, “hard” choice of the expansion scale, as in the original BIOU [133], we use a continuous, “soft” criterion. The greater the tracklet confidence c_{T_j} , the lower the expansion scale should be when calculating similarity between $\hat{T}_j$ and a detection D_i .

Let $B = (u, v, w, h)$ be a bounding box with center (u, v) , width w , and height h . By $E(B, s)$, we denote the scaled bounding box with the same center (u, v) , width $s \cdot w$, and height $s \cdot h$, i.e.,

$$E(B, s) = (u, v, s \cdot w, s \cdot h). \quad (4.2)$$

We propose to use tracklet-specific scales s_j^D and s_j^T , defined as

$$\begin{aligned} s_j^D &= 1 + \frac{1 - c_{T_j}}{2}, \\ s_j^T &= 1 + (1 - c_{T_j}), \end{aligned} \quad (4.3)$$

where $c_{T_j} \in [0, 1]$ is the confidence of tracklet T_j .

We define Soft BIoU (SBIoU) between detection D_i and predicted tracklet bounding box $\hat{T}_j$ as

$$\text{SBIoU}(D_i^t, \hat{T}_j^t) = \text{IoU}(E(D_i^t, s_j^D), E(\hat{T}_j^t, s_j^T)), \quad (4.4)$$

for $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$.

Note that the range of values of s_j^T matches the range of scale values used in [133]. When scaling the detection box, we use a smaller scale because we are more uncertain about the predicted tracklet state than about the detected bounding box position. When $c_{T_j} = 1$, both scale values are equal to 1, and SBIoU reduces to standard IoU.

4.3.2 Using an Improved Similarity Measure to Find Likely Objects

DLO confidence boosting relies on IoU similarity. We propose to replace IoU in Eq. (3.9) with a more sophisticated similarity measure S , which gives

$$\hat{c}_{D_i} = \max(c_{D_i}, \beta_c \cdot \max_j(S(D_i, \hat{T}_j))). \quad (4.5)$$

A straightforward choice would be to set $S = \text{SBIoU}$. However, we propose to use a richer similarity measure, where SBIoU is only one component.

In BoostTrack, a combination of IoU, shape similarity, and Mahalanobis distance similarity was used to construct a boosted similarity measure that improved association performance. If such a similarity measure helps to better distinguish objects during association, it should also be useful for detecting likely true-positive low-confidence detections.

We define the similarity between a detected bounding box D_i and a predicted tracklet bounding box $\hat{T}_j$, denoted by $S(D_i, \hat{T}_j)$, as the average of the selected similarity measures:

$$S(D_i, \hat{T}_j) = \frac{S_1(D_i, \hat{T}_j) + \dots + S_p(D_i, \hat{T}_j)}{p}. \quad (4.6)$$

Intuitively, using the average of multiple similarity measures for DLO confidence boosting encourages agreement among different cues before increasing the confidence score of a

given detection. In our implementation, we use

$$\mathbf{S} = \frac{\mathbf{S}_{\text{BIOU}} + \mathbf{S}_{\text{M}} + \mathbf{S}_{\text{shape}}}{3}. \quad (4.7)$$

Note that detection-tracklet confidence scores c_{D_i, T_j} used to calculate $\mathbf{S}_{\text{shape}}$ decrease when detection confidence is low. This would suppress the shape similarity exactly in the case where it is needed for detection confidence boosting, since the detections considered here are low-confidence detections. To avoid this issue, we set $c_{D_i, T_j} = 1$ when using shape similarity for detection confidence boosting.

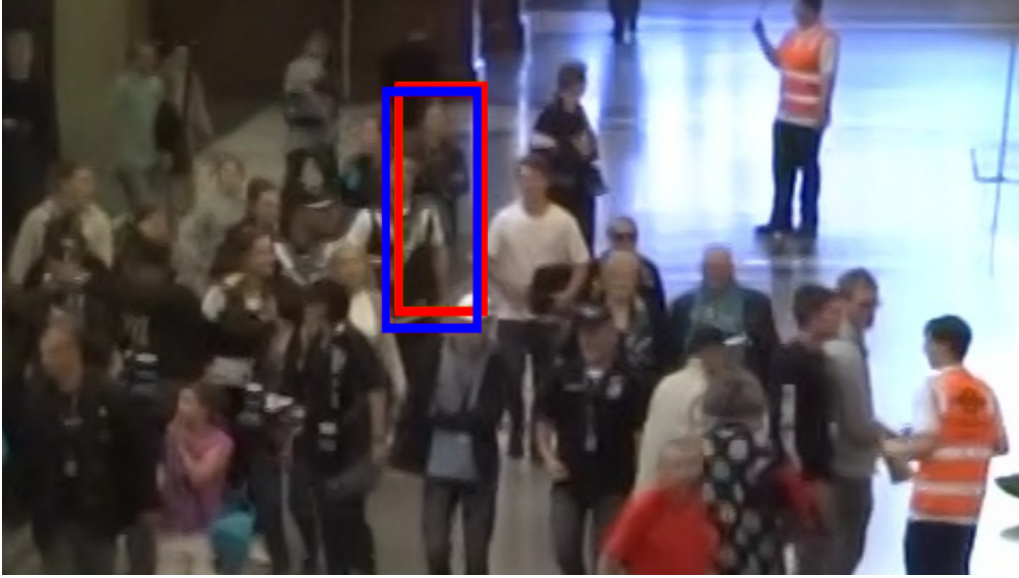


Figure 4.2: Example of a low-confidence detection, shown in blue, and the corresponding tracklet prediction, shown in red. Although the IoU between them is not sufficient for the original DLO confidence boosting criterion, the average similarity obtained from SBIOU, S_{shape} , and S_{M} is high enough to retain the detection.

Figure 4.2 shows an example from the MOT20 dataset of a low-confidence detection D_i with $c_{D_i} = 0.16$ and insufficient IoU with the corresponding predicted tracklet bounding box $\hat{T}_j$, where $\text{IoU}(D_i, \hat{T}_j) = 0.74$. However, $\text{SBIOU}(D_i, \hat{T}_j) = 0.77$, $S_{\text{M}}(D_i, \hat{T}_j) = 0.96$, and $S_{\text{shape}}(D_i, \hat{T}_j) = 0.82$. The average similarity from Eq. (4.6) is therefore high enough to keep D_i for the association step.

4.3.3 Soft Detection Confidence Boosting

BoostTrack increases the detection confidence score based only on the IoU between a detection and predicted tracklet bounding boxes (see Eq. (3.9)). If the IoU between a given tracklet and a low-confidence detection is high enough, e.g., 0.8 for MOT20, the detection confidence is increased enough for the detection to be used in association.

However, the original confidence score of the detection is not considered in this decision. A detection with a very low confidence score, e.g., 0.1, and a detection with a confidence score only slightly below the threshold τ_{det} , e.g., 0.35, require the same high similarity to be boosted above τ_{det} . This is undesirable because detections with lower confidence are more likely to be false positives and should require stronger tracklet support than detections whose confidence is already close to the threshold.

Furthermore, previous work has shown a positive correlation between detection confidence and the IoU between detected and ground-truth bounding boxes [52]. If the tracking module provides accurate predictions, the similarity between a detection and a predicted tracklet bounding box can also be interpreted as an indicator of detection reliability. Motivated by this, we introduce soft detection confidence boosting, which combines the original detection confidence with the similarity-based tracklet support.

For every detected bounding box D_i , $D_i \in \{D_1, D_2, \dots, D_n\}$, we compute the boosted confidence score $\hat{c}_{D_i}$ as

$$\hat{c}_{D_i} = \max \left(c_{D_i}, \alpha \cdot c_{D_i} + (1 - \alpha) \cdot \left(\max_j S(D_i, \hat{T}_j) \right)^q \right), \quad (4.8)$$

where $\alpha \in [0, 1]$ and $q \geq 1$ are hyperparameters, and S is a similarity measure between detections and predicted tracklet bounding boxes. Following the discussion in Section 4.3.2, S can be defined as in Eq. (4.6). The exponent q provides additional control over the boost: larger values suppress medium similarity values and make the boost more selective. All hyperparameter values are discussed in Section 4.4.1.

Figure 4.3 shows an example from MOT17. The detected bounding box, shown in blue, has confidence score 0.56, which is slightly below the standard MOT17 threshold $\tau_{\text{det}} = 0.6$. The IoU between the detection and the corresponding tracklet is 0.82, which is not sufficient for the original hard DLO confidence boosting criterion on MOT17. However, because the original detection confidence is already close to the threshold, soft detection confidence boosting increases the confidence score above τ_{det} and allows the detection to be used in association.

4.3.4 Varying Similarity Threshold

All the similarity measures discussed previously depend upon tracking performance, i.e., the quality of predictions. As discussed in Section 4.2, the reliability of a tracklet prediction decreases with the increase of the number of steps since the last tracklet update. Reduced quality of predictions leads to lower IoU value when objects get matched (see Figure 4.1) and indicates that a single threshold value for DLO confidence boost cannot provide the best results. The threshold should be higher for recently updated tracklets, whose

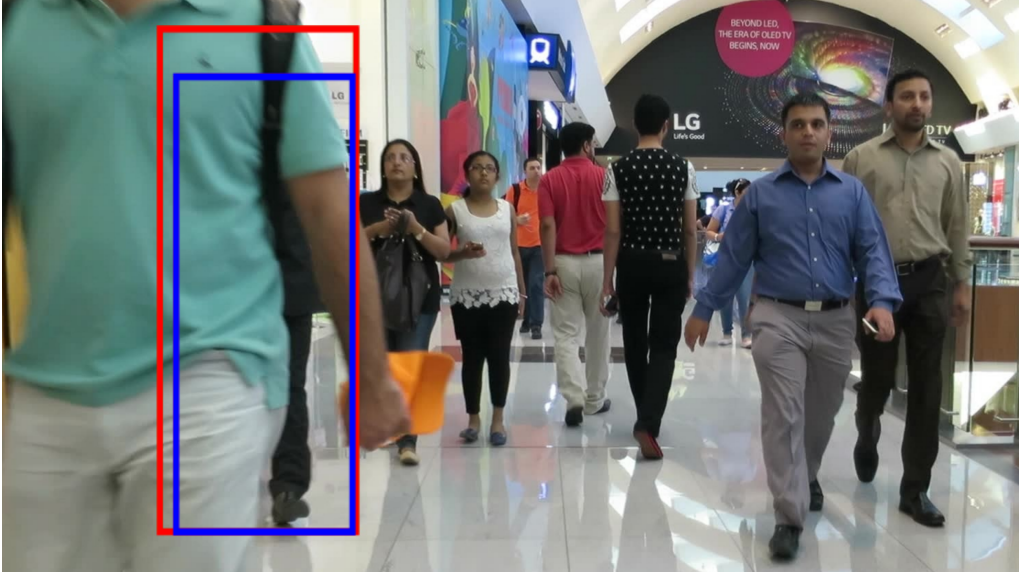


Figure 4.3: Example of soft detection confidence boosting. The detected bounding box, shown in blue, has a confidence score slightly below the detection threshold. The corresponding tracklet is shown in red. Although the IoU between the detection and the tracklet is not sufficient for the original hard DLO criterion, soft detection confidence boosting increases the confidence score above the threshold.

predictions are more reliable, and lower for tracklets that have not been updated for several frames.

Let $u_j = \text{last_update}(T_j)$ be the number of frames since the last update of tracklet T_j , for $j \in \{1, 2, \dots, m\}$. We decrease the threshold β_j corresponding to tracklet T_j linearly from the starting value β_{high} to the final value β_{low} and define

$$\beta_j = \max(\beta_{\text{low}}, \beta_{\text{high}} - \gamma \cdot (u_j - 1)), \quad (4.9)$$

where γ controls how quickly the threshold decreases.

Using the varying similarity threshold, the boosted confidence of detection D_i , $i \in \{1, 2, \dots, n\}$, is obtained as

$$\hat{c}_{D_i} = \begin{cases} \max(c_{D_i}, \tau_{\text{det}}), & \text{if } S(D_i, \hat{T}_j) \geq \beta_j \text{ for some } j \in \{1, 2, \dots, m\}, \\ c_{D_i}, & \text{otherwise,} \end{cases} \quad (4.10)$$

where S is a similarity measure between detections and predicted tracklet bounding boxes, and τ_{det} is the threshold used for discarding low-confidence detections.

Figure 4.4 shows an example of a low-confidence detection D_i whose similarity with a predicted tracklet bounding box $\hat{T}_j$ is high enough with respect to the varying threshold β_j .



Figure 4.4: Example of varying similarity threshold. The detected bounding box D_i , shown in blue, has original confidence score 0.35. The predicted bounding box, shown in red, corresponds to a tracklet that has not been matched for 27 frames. The threshold for this tracklet is $\beta_j = 0.8$. Since $\text{IoU}(D_i, \hat{T}_j) = 0.83 \geq \beta_j$, the confidence score c_{D_i} is increased, and D_i is retained for association.

4.3.5 Combining the Confidence Boost Components

The proposed enhancements are modular and do not interfere with each other, allowing them to be applied individually or in combination. In the final BoostTrack++ configuration, all proposed components are integrated together. For the ablation study, we also consider partial configurations controlled by the flags `useS`, `useSB`, and `useVT`, which determine whether the improved similarity measure, soft detection confidence boosting, and varying similarity threshold are used, respectively.

Algorithm 2 shows the combined procedure. The algorithm can be used as a replacement for the DLO confidence boosting component introduced in BoostTrack.

4.3.6 KF-Based Trajectory Splitting as Post-Processing

The confidence boosting techniques described above operate during online tracking by modifying detection confidence scores before association. In contrast, the method introduced in this subsection is an offline post-processing step. Therefore, it does not preserve strict online operation, but can improve the final online+PP tracking results.

When an object does not receive observations for several consecutive frames, its trajectory becomes fragmented. Longer periods without observations make predictions less reliable and increase the risk of an identity switch once a detection is associated again. Linear interpolation is commonly used as post-processing to fill such gaps. However, be-

Algorithm 2 Improved detection confidence boosting

Inputs: Detections $\mathcal{D} = \{D_1, \dots, D_n\}$ with confidence scores $c_{D_1}, \dots, c_{D_n}$; predicted tracklet bounding boxes $\hat{T} = \{\hat{T}_1, \dots, \hat{T}_m\}$; flags useS, useSB, useVT; hyperparameters $\alpha, q, \beta_c, \beta_{\text{low}}, \beta_{\text{high}}, \gamma, \tau_{\text{det}}$.

Output: Boosted confidence scores $\hat{c}_{D_1}, \dots, \hat{c}_{D_n}$.

```
/* Initialize boosted confidence scores. */
1: for  $i = 1$  to  $n$  do
2:    $\hat{c}_{D_i} \leftarrow c_{D_i}$ 
3: end for
/* Compute similarity matrix. */
4: if useS then
5:    $S \leftarrow \text{compute\_similarity}(\mathcal{D}, \hat{T})$  ▷ Using Eq. (4.7)
6: else
7:    $S \leftarrow \text{IoU}(\mathcal{D}, \hat{T})$ 
8: end if
/* Original hard confidence boosting. */
9: if not useSB and not useVT then
10:   for  $i = 1$  to  $n$  do
11:      $\hat{c}_{D_i} \leftarrow \max(c_{D_i}, \beta_c \cdot \max_j S(D_i, \hat{T}_j))$  ▷ Using Eq. (4.5)
12:   end for
13: else
/* Soft detection confidence boosting. */
14:   if useSB then
15:     for  $i = 1$  to  $n$  do
16:        $\hat{c}_{D_i} \leftarrow \max(\hat{c}_{D_i}, \alpha \hat{c}_{D_i} + (1 - \alpha)(\max_j S(D_i, \hat{T}_j))^q)$  ▷ Using Eq. (4.8)
17:     end for
18:   end if
/* Varying similarity threshold. */
19:   if useVT then
20:     for  $j = 1$  to  $m$  do
21:        $\beta_j \leftarrow \max(\beta_{\text{low}}, \beta_{\text{high}} - \gamma(\text{last\_update}(\hat{T}_j) - 1))$  ▷ Using Eq. (4.9)
22:     end for
23:     for  $i = 1$  to  $n$  do
24:       for  $j = 1$  to  $m$  do
25:         if  $S(D_i, \hat{T}_j) \geq \beta_j$  then
26:            $\hat{c}_{D_i} \leftarrow \max(\hat{c}_{D_i}, \tau_{\text{det}})$ 
27:           break
28:         end if
29:       end for
30:     end for
31:   end if
32: end if
33: return  $\hat{c}_{D_1}, \hat{c}_{D_2}, \dots, \hat{c}_{D_n}$ 
```



Figure 4.5: Example of an identity switch after occlusion between frames 139 and 176. The person moving to the right is incorrectly replaced by a person moving to the left. The gap should therefore not be interpolated, and the trajectory should be split.

fore interpolation is applied, one should decide whether the two ends of the gap actually belong to the same object. Existing approaches often reduce the risk of identity switches by manually setting the maximum gap length for interpolation [144, 110] or by using deep learning to split or reconnect trajectories [124, 134]. Since deep learning solutions require pretraining, we propose a simpler method for splitting trajectories based on Kalman filter predictions.

We argue that an object reappearing after a long occlusion is unlikely to immediately exhibit a substantially different motion pattern. If such a change is detected, the trajectory should be split, and the fragment after the gap should be assigned a new identity. To detect such cases, we apply the Kalman filter from both ends of the trajectory gap. If the two trajectory fragments correspond to the same object, the states predicted from the left and from the right side of the gap should be similar.

More specifically, let

$$\mathcal{T}_j = \{T_j^{t_s}, \dots, T_j^{g_s}, T_j^{g_e}, \dots, T_j^{t_e}\}$$

be an online computed trajectory with identity j , with the gap between frames g_s and g_e , i.e., $g_e - g_s > 1$. The goal is to determine whether the fragment $\{T_j^{g_e}, \dots, T_j^{t_e}\}$ actually belongs to the same object or should be assigned a new identity.

We use the available trajectory states as observations and run the Kalman filter from both directions. The first Kalman filter is initialized from the left part of the trajectory and predicts forward through the gap, while the second is initialized from the right part of the trajectory and predicts backward in time. Let

$$\mathcal{T}_L^{g_s:g_e} = \{\hat{T}_L^{g_s+1}, \hat{T}_L^{g_s+2}, \dots, \hat{T}_L^{g_e-1}\}$$

and

$$\mathcal{T}_R^{g_s:g_e} = \{\hat{T}_R^{g_s+1}, \hat{T}_R^{g_s+2}, \dots, \hat{T}_R^{g_e-1}\}$$

denote the predicted bounding boxes inside the gap obtained from the left and from the

right trajectory fragments, respectively. If the two fragments correspond to the same object, the predictions from $\mathcal{T}_L^{g_s:g_e}$ and $\mathcal{T}_R^{g_s:g_e}$ should not differ significantly.

We compare the two predicted fragments using the average IoU:

$$\overline{\text{IoU}}(\mathcal{T}_L^{g_s:g_e}, \mathcal{T}_R^{g_s:g_e}) = \frac{1}{g_e - g_s - 1} \sum_{i=g_s+1}^{g_e-1} \text{IoU}(\hat{T}_L^i, \hat{T}_R^i). \quad (4.11)$$

If $\overline{\text{IoU}}(\mathcal{T}_L^{g_s:g_e}, \mathcal{T}_R^{g_s:g_e}) < \tau_{\text{split}}$, we split the original trajectory into

$$\mathcal{T}_j = \{T_j^{t_s}, \dots, T_j^{g_s}\}$$

and

$$\mathcal{T}_k = \{T_j^{g_e}, \dots, T_j^{t_e}\},$$

where k is a new identity.

Since the average IoU can be unstable for very short gaps, we apply the method only to gaps of length at least n_g . In addition, one end of a trajectory may correspond to a ghost track, which is typically short and only rarely matched. Therefore, after applying the KF-based splitting procedure, we remove short trajectories $\mathcal{T}_i$ for which $|\mathcal{T}_i| < n_{\text{rm}}$.

Figure 4.5 shows an example of an identity switch after occlusion. The tracked person moving to the right is occluded from frame 139 to frame 176. At frame 177, an identity switch occurs, and the same identity is assigned to a different person entering the scene and moving to the left. Running the Kalman filter backward in time predicts motion to the right during the gap and does not match the left end of the trajectory. We therefore conclude that the gap between frames 139 and 176 should not be interpolated and that the two trajectory fragments correspond to different objects.

After splitting trajectories and removing short ones, we apply the same interpolation and smoothing procedure as in BoostTrack, namely linear interpolation followed by gradient boosting smoothing [142].

4.4 Experiments and Results

4.4.1 Experimental Setup and Implementation Details

BoostTrack++ extends BoostTrack+ from Chapter 3, and, unless stated otherwise, we use the same detector, Kalman filter setup, camera motion compensation, appearance model, post-processing, and tracking hyperparameters as in BoostTrack+. Experiments are performed on MOT17 and MOT20, using the same validation and test protocol as in the previous chapter. In addition to the metrics used to evaluate BoostTrack, we also monitor

the number of produced identities (IDs), since the goal is to improve tracking performance without initializing unnecessary tracklets from false positive detections. The number of IDs should not be minimized directly, but should remain reasonably close to the number of ground-truth identities: 339 on MOT17 validation set and 1418 on MOT20 validation set.

BoostTrack++-specific settings. We run a grid search to select the parameters q and α used in soft detection confidence boosting introduced in Section 4.3.3. Specifically, we test

$$(q, \alpha) \in \{1, 1.25, 1.5, 1.75, 2\} \times \{0, 0.05, 0.1, \dots, 0.95, 1\}$$

on the MOT17 validation set. For simplicity, this search is performed using $S = \text{IoU}$. Based on the best trade-off between different metrics and different baseline settings, including the use of camera motion compensation, post-processing, and appearance similarity, we set $q = 1.5$ and $\alpha = 0.65$.

For the varying similarity threshold introduced in Section 4.3.4, we use the empirical analysis from Figure 4.1 and set

$$\beta_{\text{high}} = 0.95, \quad \beta_{\text{low}} = 0.8.$$

The threshold is decreased from β_{high} to β_{low} over 20 frames, giving

$$\gamma = \frac{\beta_{\text{high}} - \beta_{\text{low}}}{20} = 0.0075.$$

For KF-based trajectory splitting, we set $\tau_{\text{split}} = 0.15$, $n_g = 25$, and $n_{\text{rm}} = 10$.

Software and evaluation. The implementation is built on top of the BoostTrack code-base. We use TrackEval [51] to evaluate validation results, while test-set results are evaluated on the official MOTChallenge server [86].

Hardware. All experiments are performed on a desktop computer with a 13th Gen Intel(R) Core(TM) i9-13900K CPU and an NVIDIA GeForce RTX 3080 GPU.

4.4.2 Ablation Study

The goal of the proposed method is to improve the DLO confidence boosting strategy used in BoostTrack. Ideally, the improved method should recover true-positive low-confidence detections without increasing the number of produced identities compared to a baseline without DLO, because the boosted detections should correspond to already existing tracklets. At the same time, a successful DLO strategy should improve not only MOTA, which is strongly affected by false positives and false negatives, but also HOTA, IDF1, and IDSW, since the quality of the selected detections directly affects association.

Influence of different similarity measures. Since DLO confidence boosting has the strongest effect on MOT20, we study the effect of different similarity measures from Eq. (4.7) on the MOT20 validation set. We perform experiments using three baselines: SORT+DLO, BoostTrack, and BoostTrack+. Table 4.2 shows the results for the SORT and BoostTrack+ baselines, representing the simplest and the strongest baseline settings, respectively. Results obtained using the BoostTrack baseline are provided in Appendix B.1.

Table 4.2: Effect of different similarity measures used for DLO confidence boosting on the MOT20 validation set. Best results among DLO-based variants are shown in bold.

Setting				SORT baseline					BoostTrack+ baseline				
DLO	SBIoU	S_M	S_{shape}	HOTA	IDF1	MOTA	IDSW	IDs	HOTA	IDF1	MOTA	IDSW	IDs
$\times$	$\times$	$\times$	$\times$	56.65	73.6	69.91	1127	1894	62.59	79.29	77.16	730	1852
$\checkmark$	$\times$	$\times$	$\times$	58.58	75.09	73.28	1259	2045	62.58	78.93	77.7	794	2019
$\checkmark$	$\times$	$\times$	$\checkmark$	57.55	72.59	73.4	2008	2729	61.99	77.51	75.63	1081	2736
$\checkmark$	$\times$	$\checkmark$	$\times$	57.76	74.87	71.3	1188	1932	62.79	79.37	77.71	739	1917
$\checkmark$	$\times$	$\checkmark$	$\checkmark$	57.73	74.81	71.33	1186	1926	62.8	79.35	77.75	723	1914
$\checkmark$	$\checkmark$	$\times$	$\times$	58.52	74.94	73.19	1228	1981	62.53	78.77	77.84	780	1964
$\checkmark$	$\checkmark$	$\times$	$\checkmark$	58.5	74.97	73.22	1273	2040	62.63	78.82	77.7	782	2038
$\checkmark$	$\checkmark$	$\checkmark$	$\times$	57.87	75.08	71.3	1158	1934	62.84	79.49	77.78	736	1928
$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	57.95	75.2	71.38	1162	1929	62.82	79.42	77.8	729	1918

The first two rows of Table 4.2 define the reference points. The first row shows the baseline without DLO confidence boosting, while the second row uses the original IoU-based DLO strategy from BoostTrack. A desirable, improved DLO method should preserve the performance gains of the original DLO strategy while keeping the number of identities and identity switches closer to the no-DLO baseline.

The results show that using a richer similarity measure changes the behavior of DLO confidence boosting. Shape similarity alone is too restrictive and leads to a large increase in IDs and IDSW, especially for the SORT baseline. Mahalanobis distance similarity, on the other hand, helps reduce the number of produced identities and identity switches. SBIoU preserves much of the original DLO behavior, but by itself does not sufficiently reduce the number of identities.

The combination of SBIoU, S_M , and S_{shape} provides the best overall trade-off. It substantially reduces IDs and IDSW compared to the original DLO strategy, while preserving or improving the main tracking metrics, especially when applied to the stronger BoostTrack+ baseline.

Influence of other components. We study the influence of the proposed components on the BoostTrack+ baseline on the MOT17 and MOT20 validation sets. Additional ablation results using SORT [10] as a more general baseline are provided in Appendix B.2.

Table 4.3 shows the effect of adding the proposed components to BoostTrack+. The first row corresponds to BoostTrack+ without DLO confidence boosting, while the second row corresponds to the original IoU-based DLO strategy. The remaining rows show different combinations of the proposed additions: the improved similarity measure (S), soft detection confidence boosting (SB), varying similarity threshold (VT), and the KF-based trajectory splitting post-processing component (TS). When SB or VT is used without S, the similarity measure is standard IoU (the full procedure is summarized in Algorithm 2).

Table 4.3: Ablation study on the MOT17 and MOT20 validation sets for different additional components added to the BoostTrack+ baseline. The first row corresponds to BoostTrack+ without DLO, while the second row uses the original IoU-based DLO strategy. Best values for standard tracking metrics are shown in bold.

Setting						MOT17					MOT20				
DLO	S	SB	VT	TS		HOTA	IDF1	MOTA	IDSW	IDs	HOTA	IDF1	MOTA	IDSW	IDs
$\times$	$\times$	$\times$	$\times$	$\times$		72.17	84.86	81.02	79	388	62.59	79.29	77.16	730	1852
$\checkmark$	$\times$	$\times$	$\times$	$\times$		72.41	85.09	80.95	83	402	62.58	78.93	77.7	794	2019
$\checkmark$	$\times$	$\times$	$\checkmark$	$\times$		72.13	84.76	80.88	84	395	62.8	79.56	77.3	712	1852
$\checkmark$	$\times$	$\checkmark$	$\times$	$\times$		72.05	84.91	80.67	80	402	62.86	79.23	77.84	812	2043
$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\times$		72.02	85.08	80.58	81	412	62.82	79.16	77.86	813	2042
$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$		71.88	84.82	80.96	89	398	62.82	79.42	77.8	729	1918
$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\times$		72.17	84.75	81.15	83	398	62.75	79.44	77.35	722	1848
$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\times$		72.24	85.44	81.23	81	396	63.2	80.05	78.03	734	1912
$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\times$		72.22	85.45	81.33	84	396	63.16	79.97	78.02	732	1906
$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$		72.48	85.72	81.96	64	331	63.18	80.0	77.97	632	1691

The results in Table 4.3 show that using SB or VT with IoU alone does not provide a stable improvement. In particular, on MOT20 these settings increase the number of produced identities compared to the original DLO strategy, which indicates that soft boosting and varying thresholds require a more selective similarity measure.

When combined with the proposed similarity measure S, both SB and VT become more effective. The combination of S and SB gives the best HOTA, MOTA, and IDF1 on MOT20 before trajectory splitting, while adding VT produces very similar results and slightly reduces the number of identities. On MOT17, the differences between these settings are small, which is expected because the effect of DLO is less pronounced on this dataset.

The KF-based trajectory splitting component (TS) has a different role, since it is applied as post-processing. It improves the final online+PP result mostly by reducing identity switches and unnecessary identities. On MOT17, adding TS to the full configuration improves HOTA, MOTA, and IDF1 and reduces IDSW from 84 to 64. On MOT20, TS reduces IDSW from 732 to 632 and the number of IDs from 1906 to 1691, with almost

unchanged HOTA and IDF1. This confirms that TS is useful for a more reliable interpolation.

Overall, the full configuration provides the most balanced behavior across datasets and metrics. Although some partial configurations achieve slightly higher scores on individual metrics, using all proposed components gives stable improvements while reducing the risk of introducing additional identities and identity switches.

4.4.3 Comparison with Contemporary Benchmark Methods

We evaluate the final BoostTrack++ configuration on the MOT17 [84] and MOT20 [25] test sets under the private detection protocol. The final configuration uses the proposed similarity measure, soft detection confidence boosting, varying similarity threshold, and KF-based trajectory splitting before post-processing.

The comparison follows the experimental context of the original BoostTrack study and includes strong contemporary benchmark methods. The results are shown in Table 4.4.

To complement the numerical comparison in Table 4.4, Figure 4.6 visualizes the relative position of BoostTrack++ with respect to contemporary benchmark methods (we mark our BoostTrack methods in bold). We plot HOTA and IDF1 scores on the x- and y-axes, respectively, with marker size corresponding to MOTA and colors indicating the temporal processing category.

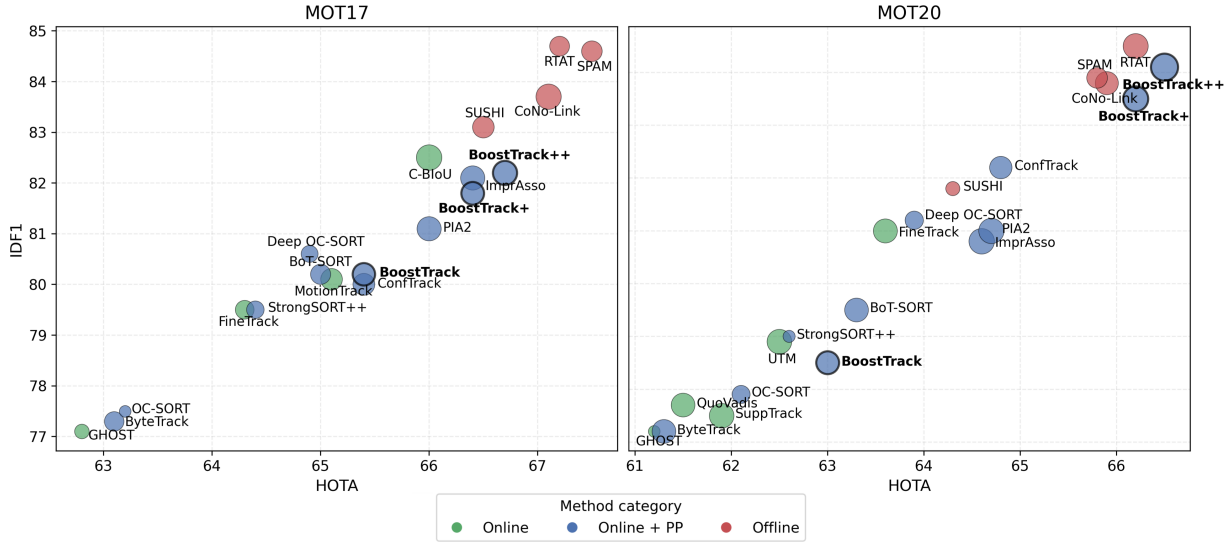


Figure 4.6: HOTA-IDF1 comparison of selected contemporary MOT methods on the MOT17 and MOT20 test sets under the private detection protocol. Marker size is proportional to MOTA, while colors indicate the temporal processing category: strictly online, online with post-processing, and offline. Methods closer to the upper-right corner achieve better HOTA and IDF1 scores.

Table 4.4: Comparison with contemporary benchmark methods on the MOT17 and MOT20 test sets under the private detection protocol. Methods are grouped according to their temporal processing type. Best results are shown in bold.

Method	MOT17				MOT20			
	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
<i>Strictly online methods</i>								
FairMOT [145]	59.3	72.3	73.7	3303	54.6	67.3	61.8	5243
MOTR [141]	62.0	75.0	78.6	2619	/	/	/	/
GHOST [103]	63.1	77.1	78.7	2325	61.2	75.2	73.7	1264
QuoVadis [26]	63.1	77.7	80.3	2103	61.5	75.7	77.8	1187
BPMTrack [36]	63.6	78.1	81.3	2010	62.3	76.7	78.3	1314
UTM [138]	64.0	78.7	81.8	1431	62.5	76.9	78.2	1228
FineTrack [97]	64.3	79.5	80.0	1272	63.6	79.0	77.9	980
MotionTrack [95]	65.1	80.1	81.1	1140	62.8	76.5	78.0	1165
C-BIoU [133]	66.0	82.5	82.8	1194	/	/	/	/
SuppTrack [146]	/	/	/	/	61.9	75.5	78.2	1325
<i>Online methods with post-processing</i>								
ByteTrack [144]	63.1	77.3	80.3	2196	61.3	75.2	77.8	1223
OC-SORT [15]	63.2	77.5	78.0	1950	62.1	75.9	75.5	913
StrongSORT++ [30]	64.4	79.5	79.6	1194	62.6	77.0	73.8	770
BASE [59]	64.5	78.6	81.9	1281	63.5	77.6	78.2	984
Deep OC-SORT [75]	64.9	80.6	79.4	1023	63.9	79.2	75.6	779
BoT-SORT [3]	65.0	80.2	80.5	1212	63.3	77.5	77.8	1313
SparseTrack [69]	65.1	80.1	81.0	1170	63.5	77.6	78.1	1120
ConfTrack [52]	65.4	80.0	81.2	1155	64.8	80.2	77.2	702
LG-Track [81]	65.4	80.4	81.4	1125	63.4	77.4	77.8	1161
StrongTBD [109]	65.6	80.8	81.6	954	63.6	77.0	78.0	1101
PIA2 [112]	66.0	81.1	82.2	1026	64.7	79.0	78.5	1023
ImprAsso [110]	66.4	82.1	82.2	924	64.6	78.8	78.6	992
BoostTrack	65.4	80.2	80.5	1104	63.0	76.5	76.4	992
BoostTrack+	66.4	81.8	80.6	1086	66.2	81.5	77.2	827
BoostTrack++	66.7	82.2	81.1	792	66.5	82.1	78.0	661
<i>Offline methods</i>								
SUSHI [17]	66.5	83.1	81.1	1149	64.3	79.8	74.3	706
CoNo-Link [37]	67.1	83.7	82.7	1092	65.9	81.8	77.5	956
RTAT [43]	67.2	84.7	80.4	912	66.2	82.5	78.4	787
SPAM [18]	67.5	84.6	80.7	1161	65.8	81.9	76.5	961

As shown in Table 4.4 and Figure 4.6, BoostTrack++ improves the BoostTrack+ baseline on both test sets. On MOT17, it improves HOTA from 66.4 to 66.7, MOTA from 80.6 to 81.1, and IDF1 from 81.8 to 82.2, while reducing IDSW from 1086 to 792. On MOT20, BoostTrack++ improves HOTA from 66.2 to 66.5, MOTA from 77.2 to 78.0, and

IDF1 from 81.5 to 82.1, while reducing IDSW from 827 to 661.

Among non-offline methods, BoostTrack++ achieves the best HOTA on MOT17 and the lowest number of identity switches. On MOT20, BoostTrack++ achieves the best HOTA and IDSW among all listed methods, including offline methods, the best IDF1 score among non-offline methods, and achieves close to the best MOTA result. These results show that the proposed tracklet-guided detection recovery and KF-based trajectory splitting improve the BoostTrack+ baseline, especially in terms of identity preservation.

4.5 Conclusion

In this chapter, we presented BoostTrack++, which extends the detection confidence boosting strategy introduced in BoostTrack. The goal was to improve the recovery of true-positive low-confidence detections that correspond to already tracked objects. Instead of relying only on IoU, BoostTrack++ combines Mahalanobis distance similarity, shape similarity, and the newly introduced soft Buffered IoU. It further improves the selection of true-positive detections through soft detection confidence boosting and a varying similarity threshold, which together provide more reliable control over the recovery of low-confidence detections.

These components further support the central idea of the dissertation: detection confidence, tracklet information, and association cues should be used jointly rather than independently. In BoostTrack++, tracklets help decide which low-confidence detections should be retained before association, while detection confidence influences how much tracklet support is required.

In addition, this chapter introduced a KF-based trajectory splitting procedure as an offline post-processing component. This method uses Kalman filter predictions from both sides of a trajectory gap to decide whether interpolation should be applied or the trajectory should be split. From the perspective of component cooperation, it can be viewed as a way of using the tracking module to improve post-processing. In the dissertation version of BoostTrack++, this component further improves identity preservation and slightly improves the final MOTA score compared to the original BoostTrack++ paper.

The experimental results on MOT17 and MOT20 show that BoostTrack++ improves the BoostTrack+ baseline, especially on MOT20, where crowded scenes make detection recovery and identity preservation more difficult. At the same time, the method still relies on hand-designed thresholds and part of the final improvement comes from post-processing. The next chapter therefore shifts the focus from detection confidence boosting, hand-tuning, heuristics, and offline refinement toward a strictly online framework that couples motion estimation, uncertainty modeling, similarity computation, and association.

Chapter 5

Crowd-Aware Tracker: Coupled Motion Estimates for Robust Real-Time Multiple Object Tracking

This chapter is adapted from a manuscript "Towards Simple, Robust, and Real-Time Multiple Object Tracking with Coupled Motion Estimates" currently under review in the *International Journal of Computer Vision* (submission ID VISI-D-26-01251).

5.1 Introduction

The previous two chapters studied how tracking-by-detection MOT can be improved by strengthening the interaction between detection handling, tracklet information, and data association. BoostTrack introduced confidence and similarity boosting as lightweight components that improve one-stage association without relying on complex tracking pipelines. BoostTrack++ further developed this idea by using richer tracklet-guided information to recover true-positive low-confidence detections, but it also increased the complexity of the overall system. Furthermore, both methods used post-processing to obtain the final benchmark results.

In this chapter, we take the next step and revisit the design of online MOT under stricter constraints. Instead of adding further appearance cues, detection confidence boosting rules, multi-stage association steps, or offline refinement, we ask whether a simple motion-based tracker can remain robust and competitive if its basic components are coupled more carefully. In other words, the goal is not to improve tracking by adding more external modules, but by making motion estimation, uncertainty modeling, similarity computation, association, and tracklet management work together more effectively.

This question is motivated by a key property of tracking-by-detection systems: their

components are tightly interconnected and mutually dependent. Inaccurate state predictions can lead to wrong associations; wrong associations corrupt the estimated motion state; unreliable detections affect both tracklet creation and trajectory continuity; and missed detections increase uncertainty over time. A central theme of this dissertation, fully developed in this chapter, is that the core components of a TbD MOT pipeline can also complement each other.

The method introduced in this chapter, Crowd-Aware Tracker (CAT), is a strictly online, appearance-free, and real-time MOT algorithm. It does not use ReID features, large pretrained appearance models, offline post-processing, explicit detection confidence boosting, or multi-stage association. Instead, CAT relies on adaptive motion estimation, uncertainty-aware bounding-box expansion, one-stage association, and confidence-based tracklet management. The central idea is that even simple motion-based components can produce robust tracking when they are designed to be adaptive and to support one another.

The design of CAT follows four principles that summarize this direction: adaptivity, simplicity, robustness, and complementarity. The tracker adapts to the current scene using online uncertainty and density estimates, preserves a simple one-stage motion-based association pipeline, explicitly accounts for noisy detections and missed associations, and couples its components so that information estimated in one part of the pipeline can guide decisions in another.

CAT is built around three tightly connected components. First, the adaptive Crowd-Aware Kalman Filter (CAKF) estimates process and measurement noise online and additionally adapts the prediction uncertainty according to local scene density. This allows the tracker to dynamically adjust its trust in predictions and observations, especially in crowded or ambiguous scenarios.

Second, we introduce Expanded Intersection over Union (EIoU), an uncertainty-driven similarity measure. Instead of using fixed hand-tuned expansion scales, EIoU expands detection and tracklet bounding boxes according to the uncertainty estimated by the Kalman filter. In this way, the similarity measure becomes object-specific and directly connected to the tracking state.

Third, we propose an improved one-stage association workflow that uses all detections produced by the detector, without prior selection or explicit confidence boosting, and applies uncertainty-aware criteria for association validity, tracklet creation, and trajectory updates. In contrast to conventional tracklet management based on consecutive matches and fixed detection confidence thresholds, CAT uses detection reliability, uncertainty estimates, and scene density information to achieve a robust association workflow. This also reduces the need for offline interpolation while preserving strictly online operation.

An overview of the proposed CAT framework is shown in Figure 5.1.

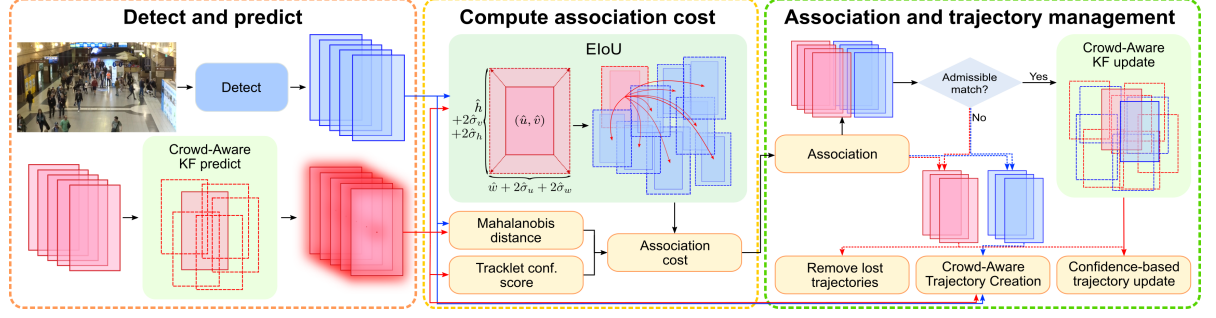


Figure 5.1: Overview of the proposed CAT framework.

The experiments show that CAT achieves state-of-the-art performance among online motion-based trackers on MOT20, SportsMOT, and DanceTrack, while maintaining real-time speed and remaining competitive with more complex appearance-based and offline approaches. These results make CAT the most complete realization of the main research direction of this dissertation: improving online MOT through cooperation between simple tracking components rather than through increased pipeline complexity.

In summary, the main contributions presented in this chapter are:

- **Crowd-Aware Kalman Filter (CAKF):** an adaptive Kalman filter that estimates process and measurement noise online and adjusts uncertainty using local crowd density;
- **Expanded IoU (EIoU):** an uncertainty-driven similarity measure that computes object-specific bounding-box expansions and enables effective one-stage association without hand-tuned expansion heuristics;
- **Improved association workflow:** a strictly online one-stage association framework that uses all detections and introduces uncertainty-aware criteria for association validation, tracklet creation, and trajectory updates;
- **Robust real-time motion-based tracking:** an experimental evaluation showing that CAT achieves state-of-the-art results among strictly online motion-based trackers on multiple MOT benchmarks while remaining real-time and conceptually simple. At the time of writing, CAT also outperforms the best online appearance-based and offline motion-based methods on MOT20 dataset.

The rest of this chapter is organized as follows. Section 5.2 briefly discusses related work specific to this chapter, focusing on Kalman-filter-based MOT, unreliable detections, and bounding-box expansion strategies. Section 5.3 introduces the proposed method, including CAKF, EIoU, and the improved association workflow. Section 5.4 presents the experimental evaluation, while Section 5.5 discusses the main properties and limitations of the method. Finally, Section 5.6 concludes the chapter.

5.2 Related Work

Since the general tracking-by-detection workflow and the main MOT components were introduced in Chapter 2, this section focuses only on the topics most directly related to CAT.

5.2.1 Kalman Filter for MOT

Noise covariance matrices. Existing methods typically use constant process and measurement noise covariance matrices, as in SORT [10], or scale them according to object height, as in [130, 3]. A recent trend in MOT is to use the detection confidence score to scale otherwise constant measurement noise covariance matrices [29, 142, 78, 52, 48, 90]. In [65, 68], process noise is also refined based on tracklet confidence [65] or the IoU value between the predicted and associated bounding boxes [68].

Only a few methods have addressed the estimation of noise covariance matrices for MOT applications. In [40], an artificial neural network is used to estimate noise and state covariance matrices, while these matrices are estimated using a modified Sage–Husa algorithm [100] in [49].

Unlike these methods, we argue that even noise-free observations in MOT should not be fully trusted because of possible matching errors. To the best of our knowledge, CAT is the first TbD MOT method to explicitly address uncertainty caused by crowd density and use it to adjust process and measurement noise covariance estimates.

Using a Kalman filter beyond tracking. In MOT, the state covariance estimated by the Kalman filter is used in probabilistic data association frameworks, e.g., [11, 14]. In the TbD setting, its use usually focuses on calculating the Mahalanobis distance or negative log-likelihood between tracked objects and observations [30, 130, 82, 116].

Following the approach in BoostTrack, described in Chapter 3, CAT computes Mahalanobis-distance-based similarity using state covariance estimates and does not claim novelty in this aspect. However, CAT also uses these statistics to compute the tracklet confidence score and the proposed Expanded IoU similarity measure.

5.2.2 Expanding Bounding Boxes

To account for irregular motion or frequent occlusions, which degrade the performance of tracklet state prediction, recent works increase the “search” area by expanding bounding boxes before calculating IoU [133, 148, 41, 47].

In [133, 41], two-stage association is adopted, and the expansion scale is manually tuned and fixed for all detection-tracklet pairs in a given association phase. A smaller scale

is used in the first stage and a larger scale in the second stage for the remaining unmatched pairs. Similarly, in [148], the width of the bounding boxes is expanded using a fixed hand-tuned scale. In [47], iterative multi-stage association is applied, and the expansion scale is increased in each iteration. In BoostTrack++, we introduced soft BIoU that uses tracklet information to define a more flexible expansion strategy, but we used soft BIoU only in the pre-association, detection confidence boosting phase.

Unlike these approaches, EIoU expands bounding boxes according to uncertainty estimated by the Kalman filter and therefore requires no hand-tuned expansion scale, heuristic stage design, or complex multi-stage association.

5.2.3 Working with Low-Confidence Detections

The importance of using low-confidence detections was discussed in the previous chapters. CAT’s detection handling and tracklet management are partially inspired by the insights from BoostTrack and BoostTrack++, described in Chapters 3 and 4. As in [64, 67, 114, 115], CAT uses the tracklet confidence score to scale the similarity matrix, but it computes tracklet confidence from Kalman filter covariance estimates.

The recent tracker TrackTrack [106] uses fixed penalty terms to prioritize matching with high-confidence detections and applies NMS-based tracklet initialization. Unlike [106], which uses NMS in the tracklet creation process, the proposed Crowd-Aware Tracklet Creation uses no explicit NMS or thresholding other than a minimum confidence threshold.

While existing approaches improve individual components, they rarely exploit the interactions between them. In contrast, CAT explicitly propagates uncertainty between tracking and association, while the association workflow and tracklet management leverage uncertainty from tracking and detection, as well as additional information available from association.

5.3 Method

In this section, we present the proposed Crowd-Aware Tracker (CAT). CAT is built from three main components: an adaptive Crowd-Aware Kalman Filter, Expanded IoU, and an improved one-stage association workflow. The adaptive Crowd-Aware Kalman Filter estimates motion uncertainty while taking local crowd density into account. Expanded IoU uses the estimated uncertainty to adapt the spatial overlap between detections and predicted tracklets. Finally, the improved association workflow uses detection reliability, uncertainty estimates, crowd density information, and tracklet confidence to control

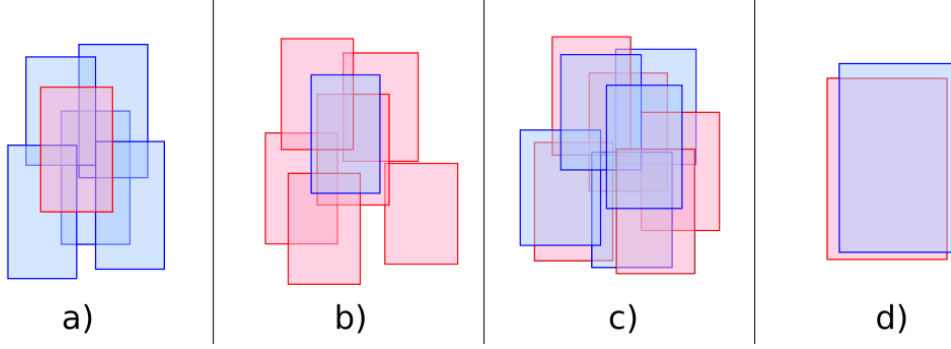


Figure 5.2: Illustrative tracklet and detection configurations. Predicted tracklet bounding boxes are shown with dashed red rectangles, while detections are shown with dotted blue rectangles. Association ambiguity is minimal only in scenario d.

association validity, tracklet creation, and trajectory updates.

5.3.1 Adaptive Crowd-Aware Kalman Filter

The accuracy of Kalman filter estimates depends on the suitability of the motion model, the quality of the observations provided by the detector, and the subsequent data association process. Controlling the process noise covariance matrix $\mathbf{Q}$ and the measurement noise covariance matrix $\mathbf{R}$ is crucial for mitigating the effects of noisy observations and changes in object motion. In particular, measurement noise variance controls how much a new observation affects the state estimate: the larger the measurement noise variance, the less influence the observation has, and the more the estimate relies on the Kalman filter prediction. Process noise variance, on the other hand, reflects uncertainty in the motion dynamics and model mismatch. However, as noted in Section 5.2.1, existing methods usually rely on predefined noise covariance models instead of estimating them online.

We estimate $\mathbf{Q}$ and $\mathbf{R}$ online using the method from [5]. After receiving the observation $\mathbf{z}_t$ and computing the updated state estimate $\hat{\mathbf{x}}_{t|t}$, the estimates $\mathbf{Q}_t$ and $\mathbf{R}_{t+1}$, used in the following time step, are computed as

$$\begin{aligned}\mathbf{R}_{t+1} &= \alpha \mathbf{R}_t + (1 - \alpha) (\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t^\top + \mathbf{H} \mathbf{P}_{t|t-1} \mathbf{H}^\top), \\ \mathbf{Q}_t &= \alpha \mathbf{Q}_{t-1} + (1 - \alpha) (\mathbf{K}_t \mathbf{d}_t \mathbf{d}_t^\top \mathbf{K}_t^\top),\end{aligned}\tag{5.1}$$

where $\boldsymbol{\varepsilon}_t = \mathbf{z}_t - \mathbf{H} \hat{\mathbf{x}}_{t|t}$ and $\mathbf{d}_t = \mathbf{z}_t - \mathbf{H} \hat{\mathbf{x}}_{t|t-1}$ are the residual and innovation vectors, respectively, while α , $0 < \alpha \leq 1$, is a forgetting factor used instead of averaging over a moving window.

However, adaptive Kalman filtering alone does not address MOT-specific uncertainty. In tracking-by-detection, the observations assigned to a tracklet depend on the association result, which can be error-prone, especially in crowded scenes. Even if the detector pro-

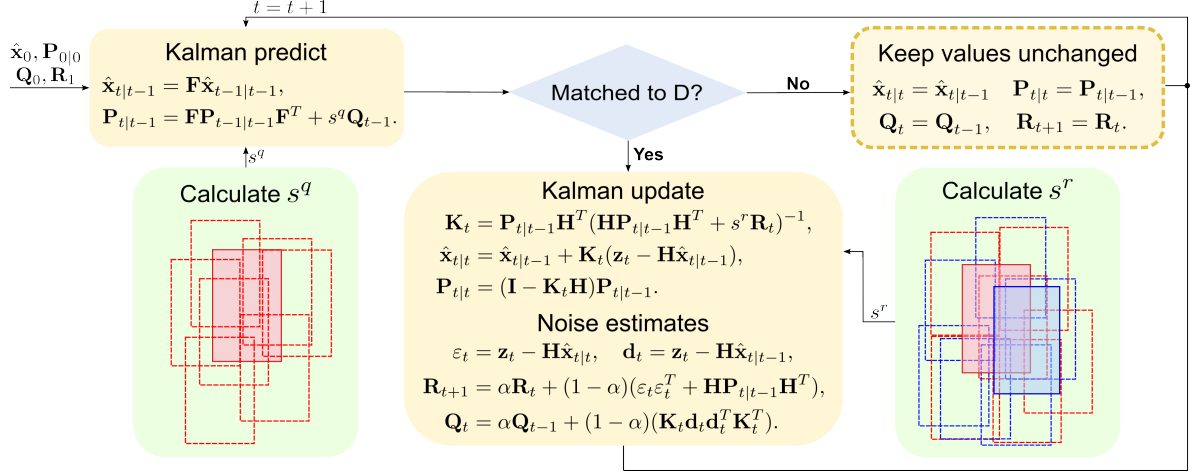


Figure 5.3: Overview of the Adaptive Crowd-Aware Kalman Filter. The filled red rectangle represents the predicted bounding box of the object whose state is being estimated, while the filled blue rectangle represents the associated detection D_i . Dashed rectangles correspond to other tracklets and detections.

vides high-quality observations, an association error can cause the Kalman filter to update the tracklet state using an incorrect observation.

Figure 5.2 illustrates several illustrative scenarios. In case a, multiple detections overlap with a single predicted tracklet bounding box, and the tracked object may be associated with an incorrect or false-positive detection. In case b, multiple predicted tracklets overlap with a single detection, and the detection may be assigned to the wrong tracklet. Case c represents a typical crowded scenario, where multiple tracklets and detections overlap, making association ambiguous. Only case d corresponds to a low-ambiguity configuration, where detection confidence can be treated as the primary indicator of observation reliability.

Suppose that the predicted tracklet bounding box $\hat{T}_j$ is matched with detection D_i . If there exists another detection D_l , $l \neq i$, or another predicted tracklet bounding box $\hat{T}_k$, $k \neq j$, such that $\text{IoU}(D_i, D_l)$ or $\text{IoU}(\hat{T}_j, \hat{T}_k)$ is high, the observation D_i should be treated as less reliable because of possible association error.

To mitigate this problem, we replace $\mathbf{R}$ in Eq. (2.12) with the scaled measurement noise covariance matrix $\tilde{\mathbf{R}} = s_{i,j}^r \mathbf{R}$, where the scaling factor is defined as

$$s_{i,j}^r = 1 - c_{D_i} + \max \left(\max_{k \neq j} \text{IoU}(\hat{T}_k, \hat{T}_j), \max_{l \neq i} \text{IoU}(D_l, D_i) \right). \quad (5.2)$$

Since $\mathbf{R}$ is estimated online, this scaling further adapts the measurement noise in dense and ambiguous association scenarios. When the maximum IoU terms in Eq. (5.2) are equal to zero, the scaling reduces to the NSA Kalman filter strategy from [29]. In general, this formulation increases measurement uncertainty in ambiguous spatial configurations,

reducing the influence of potentially incorrect associations.

Similarly, we refine the process noise estimate based on local crowd density. If the predicted bounding box of an object strongly overlaps with the predicted bounding box of another object, possible interaction between the objects may lead to changes in motion. During the prediction step for object j , we compute $\text{IoU}(\hat{T}_j, \hat{T}_k)$ between $\hat{T}_j$ and all other predicted tracklet bounding boxes $\hat{T}_k, k \neq j$, and define

$$s_j^q = 1 + \max_{k \neq j} \text{IoU}(\hat{T}_k, \hat{T}_j). \quad (5.3)$$

In the Kalman filter prediction step from Eq. (2.10), we use the scaled process noise covariance matrix $\tilde{\mathbf{Q}} = s_j^q \mathbf{Q}$ instead of $\mathbf{Q}$.

To reduce computational complexity, we assume diagonal noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, which implies independence of noise across state variables. In Appendix C.1, we reformulate the Kalman prediction, update, and noise estimation equations for improved computational efficiency.

5.3.2 Expanded IoU

Several recent trackers have adopted the strategy of calculating IoU between tracklets and detections using expanded bounding boxes in order to account for uncertainty in the tracklet state [133, 148, 41, 47]. However, existing methods use various heuristics or complex multi-stage association schemes to determine appropriate expansion scales in different scenarios, as discussed in Section 5.2.2.

Uncertainty about the position and size of a tracklet is naturally encoded in the state covariance estimated by the Kalman filter. This uncertainty typically increases during prediction steps when the object is not matched with an observation. When using the adaptive CAKF introduced in Section 5.3.1, the state covariance matrix encodes not only uncertainty caused by changes in motion or noisy observations, but also uncertainty caused by crowd density.

Using the tracked object state covariance, we can calculate an object-specific expansion scale instead of using a uniform scale for all objects, as commonly done in previous works [133, 148, 41, 47]. Furthermore, object-specific expansion scales allow us to compute and use EIoU in a one-stage association framework.

Similarly to the scaling operator used for soft BIoU in Section 4.3.1, we define a bounding-box expansion operator. Let $B = (u, v, w, h)$ be a bounding box with center (u, v) , width w , and height h . By $E(B, s_w, s_h)$, we denote the scaled bounding box

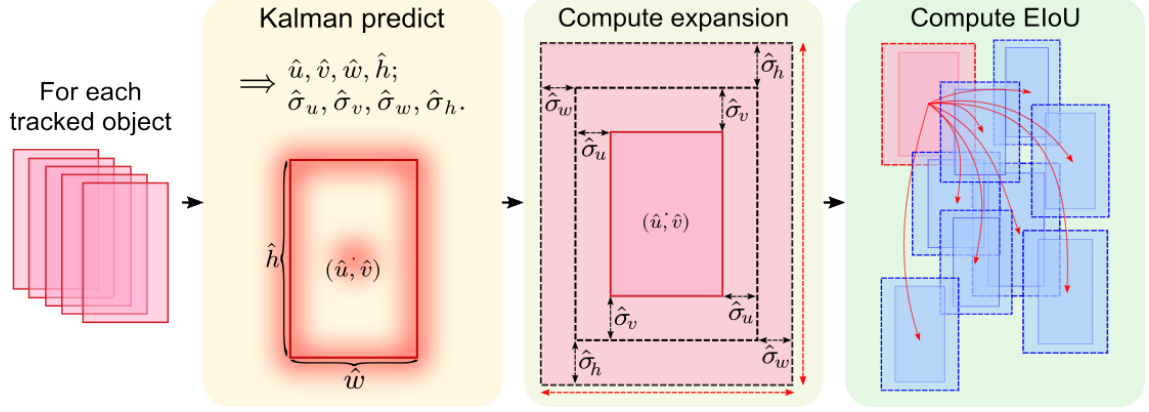


Figure 5.4: Calculation of EIou between tracklets, shown as red rectangles, and detections, shown as blue rectangles. IoU is calculated between expanded bounding boxes.

with the same center (u, v) , width $s_w \cdot w$, and height $s_h \cdot h$, i.e.,

$$E(B, s_w, s_h) = (u, v, s_w \cdot w, s_h \cdot h). \quad (5.4)$$

Let $\hat{\sigma}^2(u)_j^t$, $\hat{\sigma}^2(v)_j^t$, $\hat{\sigma}^2(w)_j^t$, and $\hat{\sigma}^2(h)_j^t$ be the predicted variance values of the position and size of object j in frame t , obtained from the diagonal of the predicted covariance matrix $\mathbf{P}_{t|t-1}^j$. Since the exact position of the object is uncertain, we expand the width by adding $\hat{\sigma}(u)_j^t$ to both sides of the predicted bounding box. Similarly, since the size of the object is uncertain, we also add $\hat{\sigma}(w)_j^t$ to both sides. Analogously, we increase the height of $\hat{T}_j^t$ by adding $\hat{\sigma}(v)_j^t$ and $\hat{\sigma}(h)_j^t$.

If the Kalman update is not executed for several consecutive steps, the predicted standard deviations may become too large and allow incorrect associations. To avoid this, we impose a maximum expansion scale.

In summary, for object j in frame t , we calculate the expansion scales s_w^j and s_h^j as

$$\begin{aligned} s_w^j &= \min \left(1 + \frac{2 \cdot (\hat{\sigma}(u)_j^t + \hat{\sigma}(w)_j^t)}{\hat{w}_j^t}, s_{\max} \right), \\ s_h^j &= \min \left(1 + \frac{2 \cdot (\hat{\sigma}(v)_j^t + \hat{\sigma}(h)_j^t)}{\hat{h}_j^t}, s_{\max} \right). \end{aligned} \quad (5.5)$$

The maximum scale $s_{\max}$ is the only hyperparameter required. We calculate EIou between $\hat{T}_j^t$ and D_i^t , for $i = 1, \dots, n$, as

$$\text{EIou}(\hat{T}_j^t, D_i^t) = \text{IoU} \left(E(\hat{T}_j^t, s_w^j, s_h^j), E(D_i^t, s_w^j, s_h^j) \right). \quad (5.6)$$

The process is repeated for each tracked object j to obtain the complete similarity matrix. The computational complexity of calculating EIou between n detections and m tracklets

is the same as that of IoU, namely $\mathcal{O}(n \cdot m)$. Unlike heuristic expansion strategies, EIoU directly reflects uncertainty estimated by the tracking model.

We illustrate the computation of EIoU in Figure 5.4. In Figure 5.5, we show an example where IoU and EIoU are computed between two tracklets, $\hat{T}_a$ and $\hat{T}_b$, shown as yellow and magenta rectangles, respectively, and a detected bounding box D , shown as a blue rectangle.¹

In the left part of Figure 5.5, we calculate $\text{IoU}(\hat{T}_a, D)$ and $\text{IoU}(\hat{T}_b, D)$. The overlap between $\hat{T}_a$ and D is almost zero, while the overlap between $\hat{T}_b$ and D is low but higher. However, the corresponding EIoU values differ significantly, as illustrated in the right part of the figure.

The predicted uncertainty of $\hat{T}_b$, shown in magenta, is low because it corresponds to an object that is clearly visible and not located in a crowded area. On the other hand, the predicted uncertainty of $\hat{T}_a$, which corresponds to an object in the crowd, is significantly higher. Therefore, the bounding boxes $\hat{T}_b$ and D are only slightly expanded when computing $\text{EIoU}(\hat{T}_b, D)$, while a larger expansion scale is used when computing $\text{EIoU}(\hat{T}_a, D)$. Intuitively, the tracker is more certain about the position and size of $\hat{T}_b$. In contrast, given the history of previous observations, motion changes, and crowd density, D may be more likely to correspond to $\hat{T}_a$ even if $\text{IoU}(\hat{T}_a, D) < \text{IoU}(\hat{T}_b, D)$.

Note, a non-zero EIoU between distant bounding boxes does not necessarily imply that these boxes will be matched. Closer bounding boxes will typically still have higher EIoU values, while the association workflow described in the next subsection further constrains the matching process and improves overall robustness.

5.3.3 Association Workflow

We aim to use all detected bounding boxes in one-stage association and replace commonly used complex multi-stage association methods. However, one-stage association requires rethinking several processing steps that are often overlooked: criteria for selecting true-positive detected bounding boxes, an association method that implicitly prioritizes high-confidence detections (see Chapters 3 and 4), and criteria for initializing new tracklets [110, 106].

In the rest of this subsection, we describe the association workflow used in CAT.

Calculating the similarity matrix. Instead of IoU, we use Expanded IoU, introduced in Section 5.3.2, to construct the similarity matrix between predicted bounding boxes $\hat{T} = \{\hat{T}_1, \hat{T}_2, \dots, \hat{T}_m\}$ and detected bounding boxes $D = \{D_1, D_2, \dots, D_n\}$. We adopt

¹We choose the detected bounding box D to be almost equally distant from both tracked objects in order to avoid cluttered visualization and to better highlight the difference between IoU and EIoU. The detected bounding box D does not correspond to either $\hat{T}_a$ or $\hat{T}_b$.

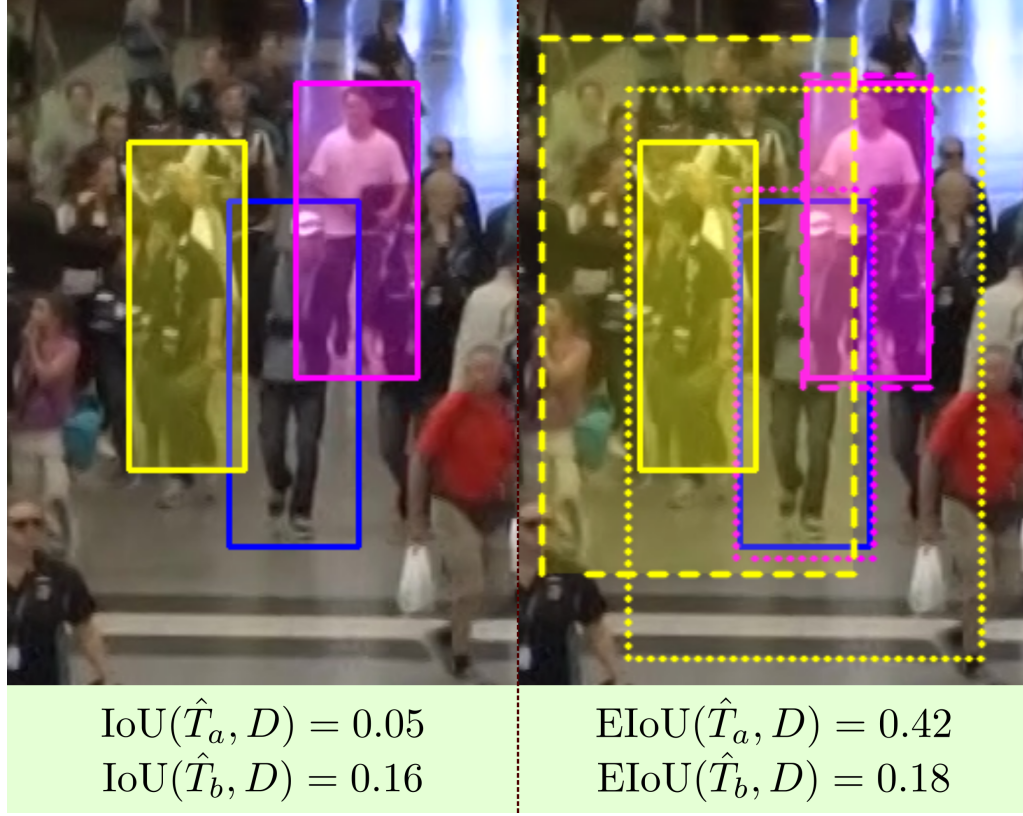


Figure 5.5: IoU and EIoU between the tracklets $\hat{T}_a$ and $\hat{T}_b$, shown in yellow and magenta, and the detected bounding box D , shown in blue. Dashed rectangles denote the expanded predicted bounding boxes, in colors corresponding to the tracked objects, while dotted rectangles represent the expansion of D according to the expansion scales of the corresponding tracked object. The large predicted variance of $\hat{T}_a$ leads to a relatively high value of $\text{EIoU}(\hat{T}_a, D)$.

similarity boosting and Mahalanobis distance similarity from BoostTrack and define the similarity matrix $\mathbf{S}$ as

$$\mathbf{S} = \frac{\mathbf{S}_{\text{EIoU}} + \lambda_{\text{boost}} \mathbf{C} \odot \mathbf{S}_{\text{EIoU}} + \lambda_{\text{M}} \mathbf{S}_{\text{M}}}{1 + \lambda_{\text{boost}} + \lambda_{\text{M}}}, \quad (5.7)$$

where $\mathbf{S}_{\text{EIoU}}$ and $\mathbf{S}_{\text{M}}$ represent the EIoU and Mahalanobis distance similarity matrices, respectively, while $\mathbf{C} = [c_{D_i, T_j}]_{n \times m}$, with $c_{D_i, T_j} = c_{D_i} \cdot c_{T_j}$, denotes the detection-tracklet confidence score matrix. Multiplying $\mathbf{S}_{\text{EIoU}}$ by confidence scores gives more weight to high-confidence detection-tracklet pairs and allows us to utilize some of the benefits of multi-stage association within a one-stage association framework (see Chapter 3).

Unlike BoostTrack, which defines tracklet confidence as a function of the number of steps since the last Kalman update, CAT uses second-order statistics estimated by the Kalman filter. Since we use the adaptive Crowd-Aware Kalman Filter, the score c_{T_j} encodes uncertainty caused by irregular motion, noisy observations, unsuccessful matches,

crowd density, and possible identity switches. Specifically, we use the predicted standard deviation of position as a measure of uncertainty and define the tracklet confidence of object j as

$$c_{T_j} = \max \left(1 - \max \left(\frac{\hat{\sigma}(u)_j^t}{\hat{w}_j^t}, \frac{\hat{\sigma}(v)_j^t}{\hat{h}_j^t} \right), 0 \right). \quad (5.8)$$

Since u and v represent coordinates, $\hat{\sigma}(u)$ and $\hat{\sigma}(v)$ are meaningful only relative to the size of the object. Therefore, we use the maximum of the uncertainties normalized by the width and height. If this value is greater than 1, we set $c_{T_j} = 0$.

Association and trajectory initialization criteria. Our association workflow is partially inspired by the insights from BoostTrack++. Informally, the association guidelines can be summarized as follows: even very low-confidence detections should be used in association; low-confidence detections should require a higher similarity score to be accepted as true-positive observations; and a detection used to start a new trajectory does not need to have a very high confidence score if it does not correspond to any already tracked object and is not a noisy duplicate of another detection.

We do not discard any detected bounding box before association, and even detections with very low confidence can be used for matching. However, low-confidence detections should require stronger similarity confirmation in order to be used as true-positive observations. To this end, we consider the pair $(D_i, \hat{T}_j)$ a valid match only if

$$S(D_i, \hat{T}_j) \geq \tau_s + \alpha_s(1 - c_{D_i}), \quad (5.9)$$

where τ_s is the minimum similarity required for an admissible match of a noise-free detection D_i with $c_{D_i} = 1$, and α_s controls the impact of c_{D_i} on the final similarity threshold.

In Figure 5.6, we show an example of using a low-confidence detection D_i , represented by the green rectangle. The detection corresponds to a partially occluded person who is almost blended with the background. Hence, the confidence score c_{D_i} is only 0.04. However, D_i is not discarded a priori, but is used in association. Its similarity with the predicted bounding box, shown as the red rectangle, is above the detection-specific threshold from Eq. (5.9), and D_i is therefore accepted and used as a true-positive observation.

Unmatched detections $\mathcal{D}_{\text{remain}}$ are candidates for starting new trajectories. Only true-positive detections that do not correspond to already tracked objects should be used to initialize new trajectories.

If we can ensure that $D_i, D_i \in \mathcal{D}_{\text{remain}}$, is not a duplicate of another detection D_k , $k \neq i$, and does not correspond to an already tracked object, then c_{D_i} does not have to be

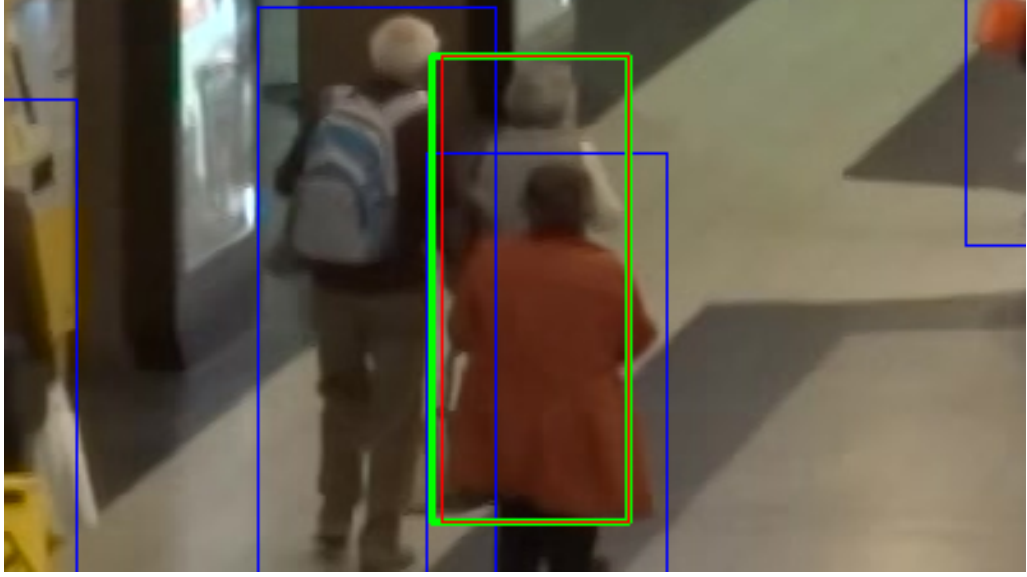


Figure 5.6: True-positive detection, shown in green, with a low confidence score but high overlap with a tracked object, shown in red.

very high for D_i to start a new trajectory. We propose to use D_i to start a new trajectory if

$$c_{D_i} \geq \tau_{tc} + \frac{\max_j \text{IoU}(D_i, \hat{T}_j) + \max_{k \neq i} \text{IoU}(D_i, D_k)}{2}. \quad (5.10)$$

We refer to the condition in Eq. (5.10) as the Crowd-Aware Tracklet Creation (CATC) criterion. The threshold τ_{tc} should be relatively low, e.g., $\tau_{tc} = 0.2$, while the final detection-specific threshold depends on the maximum IoU values. If $\max_j \text{IoU}(D_i, \hat{T}_j)$ is high, D_i may correspond to an existing tracklet. If $\max_{k \neq i} \text{IoU}(D_i, D_k)$ is high, D_i may be a duplicate of another detected bounding box, which introduces an implicit NMS effect. If both values are high, the resulting threshold is high, and D_i will not be used to start a new trajectory. High-confidence detections are implicitly prioritized through similarity boosting in Eq. (5.7), while remaining ambiguous detections are more likely to be rejected by the CATC criterion.

In Figure 5.7, we present two examples of unmatched detections, shown as green rectangles, which serve as candidates for initializing new trajectories. Blue rectangles correspond to the remaining detections, while red rectangles indicate existing tracklets.

In the left part of the figure, the unmatched detection D has a confidence score of 0.35 and would normally be discarded as a false positive. However, due to its low overlap with other detections and tracklets, D is used as a true-positive observation to initialize a trajectory.

In the second scenario, shown on the right side of the figure, a high-confidence detection D remains unmatched. However, D has high overlap with another tracklet and an-

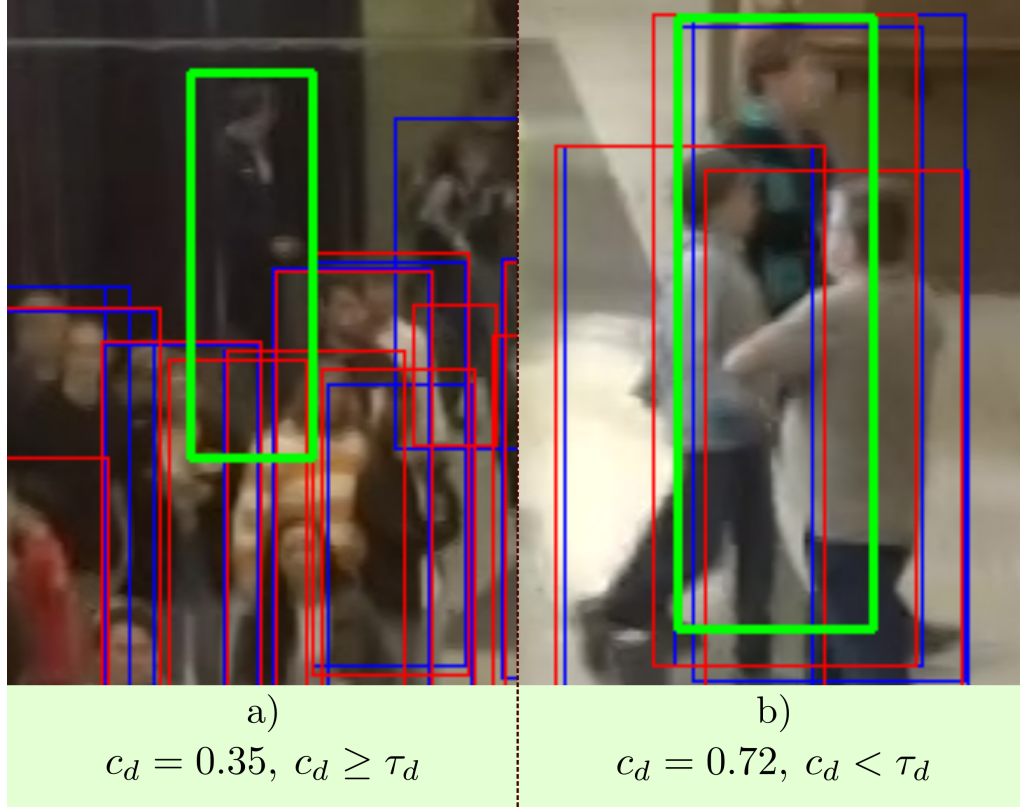


Figure 5.7: Illustration of the CATC criterion. (a) A low-confidence detection, shown in green, is used to start a trajectory due to low overlap with other detections and tracklets. (b) A high-confidence detection, shown in green, is not used to start a trajectory due to high overlap with an existing tracklet and another detection.

other detection, which leads to a high detection-specific threshold. Hence, D is not used to initialize a trajectory. In this case, D corresponds to an already tracked object and is a less accurate duplicate of another detection D_k , with $c_D > c_{D_k}$ and $\text{IoU}(D, D_k) = 0.75$. A standard NMS procedure would discard the better fitting but lower-confidence detection D_k . In contrast, the proposed association workflow uses D_k in association, but does not initialize a new trajectory from the higher-confidence duplicate D .

Trajectory update criterion. An important tracklet management question is which state estimates should be added to the final trajectories. Existing approaches often count the number of successive matches to decide whether the estimated state should be appended to the trajectory [130, 75]. Let $\mathcal{T}_j$ be the trajectory of object j , and let t_e be the last time step when $\mathcal{T}_j$ was updated, i.e., $T_j^{t_e}$ is the last state added to the trajectory.

At time step t , we use the tracklet confidence score and the last update time t_e to decide whether to add T_j^t to $\mathcal{T}_j$. We update $\mathcal{T}_j$ if

$$\tilde{c}_{T_j} \geq \tau_u \cdot (t - t_e), \quad (5.11)$$

where τ_u is a hyperparameter and $\tilde{c}_{T_j}$ is the tracklet confidence score calculated from $\mathbf{P}_{t|t}$ after the Kalman update step. This criterion allows unmatched tracklets to be appended to trajectories when their confidence score is high enough. We set $\tau_u = 0.25$ and discuss its impact in Section 5.4.6.

Unmatched tracklets are kept for at most $A_{\max}$ frames, or for A_{os} frames if we predict that the object has left the scene.

The pseudocode of the proposed association and tracklet management procedure is given in Algorithm 3. To simplify the pseudocode, we assume that Kalman-filter-related functions internally store all required data that are not passed as arguments. Similarly, we assume that the KF_update function also updates the noise estimates.

Algorithm 3 CAT: Association and tracklet management in CAT

Inputs: Frame I^t , active trajectories $\mathcal{T}^{t-1}$, detector f_{det} ,

association hyperparams. τ_s, α_s , tracklet management hyperparams. $\tau_{\text{tc}}, \tau_{\text{u}}, A_{\text{max}}, A_{\text{os}}$.

Output: Updated trajectories $\mathcal{T}^t$.

/ Detection, prediction and IoU computation. */*

- 1: $\mathcal{D}^t \leftarrow f_{\text{det}}(I^t)$
- 2: $\hat{T}^t \leftarrow \text{KF_predict_x}()$ ▷ Eq. (2.10)
- 3: $iou_{tt} \leftarrow \text{IoU}(\hat{T}^t, \hat{T}^t)$
- 4: $iou_{dt} \leftarrow \text{IoU}(\mathcal{D}^t, \hat{T}^t)$
- 5: $iou_{dd} \leftarrow \text{IoU}(\mathcal{D}^t, \mathcal{D}^t)$
- 6: $s^q \leftarrow \text{Q_scale}(\hat{T}^t, iou_{tt})$ ▷ Eq. (5.3)
- 7: $\hat{P}^t \leftarrow \text{KF_predict_P}(s^q)$
- 8: $c_T \leftarrow \text{tracklet_conf}(\hat{T}^t, \hat{P}^t)$ ▷ Eq. (5.8)

/ One-stage association */*

- 9: $S \leftarrow \text{compute_similarity}(\mathcal{D}^t, \hat{T}^t, \hat{P}^t, c_D, c_T)$ ▷ Eq. (5.7)
- 10: $\mathcal{M}, \mathcal{D}_{\text{unmatched}}, \hat{T}_{\text{unmatched}} \leftarrow \text{association}(S)$ ▷ Hungarian algorithm
- 11: $\mathcal{M} \leftarrow \text{valid_pairs}(\mathcal{M}, c_D, \tau_s, \alpha_s)$ ▷ Eq. (5.9)

/ Kalman update */*

- 12: **for all** $(D_i, \hat{T}_j) \in \mathcal{M}$ **do**
- 13: $s \leftarrow \text{R_scale}(iou_{dd}, iou_{tt}, c_{D_i}, i, j)$ ▷ Eq. (5.2)
- 14: $\hat{T}_j, \hat{P}_j \leftarrow \text{KF_update}(\hat{T}_j, D_i, s)$ ▷ Eq. (2.11)
- 15: **end for**

/ Start new trajectories */*

- 16: **for all** $D \in \mathcal{D}_{\text{unmatched}}$ **do**
- 17: **if** $\text{valid}(D, iou_{dt}, iou_{dd}, \tau_{tc})$ **then** ▷ Eq. (5.10)
- 18: $\mathcal{T}^t \leftarrow \mathcal{T}^t \cup \{\text{new_trajectory}(D)\}$
- 19: **end if**
- 20: **end for**

/ Delete lost tracks */*

- 21: **for all** $\hat{T}_j \in \hat{T}_{\text{unmatched}}$ **do**
- 22: **if** $\text{last_update}(\mathcal{T}_j) \geq A_{\text{max}}$ or $(\text{is_out}(\hat{T}_j) \text{ and } \text{last_update}(\mathcal{T}_j) \geq A_{\text{os}})$ **then**
- 23: $\text{remove}(\mathcal{T}_j)$
- 24: **end if**
- 25: **end for**

/ Update trajectories */*

- 26: $\mathcal{T}^t \leftarrow \mathcal{T}^{t-1}$
 - 27: **for all** $\hat{T}_j \in \hat{T}$ **do**
 - 28: **if** $\text{eligible}(\hat{T}_j)$ **then** ▷ Eq. (5.11)
 - 29: $\mathcal{T}_j^t \leftarrow \mathcal{T}_j^t \cup \{\hat{T}_j\}$
 - 30: **end if**
 - 31: **end for**
 - 32: **return** $\mathcal{T}^t$
-

5.4 Experiments

In this section, we evaluate the proposed method on standard MOT benchmarks. We first briefly describe the experimental setup and implementation details. We then compare CAT with existing methods, followed by an ablation study and a component analysis.

5.4.1 Datasets and Metrics

We evaluate CAT on MOT17 [84], MOT20 [25], DanceTrack [118], and SportsMOT [23] in order to test its robustness across different tracking scenarios, including crowded pedestrian scenes, dynamic motion, visually similar targets, and frequent motion changes.

In the ablation study and component analysis, we primarily report results on MOT20 and SportsMOT. Due to its high crowd density, MOT20 is particularly suitable for analyzing the effect of CAKF, while SportsMOT, characterized by strong motion variations, is better suited for evaluating the impact of EIoU. Additional results on MOT17 and DanceTrack are provided in Appendix C.

The datasets and evaluation metrics are described in Sections 2.8 and 2.9. We report HOTA, IDF1, MOTA, and IDSW on all datasets. For DanceTrack, we additionally report AssA and DetA, following the standard evaluation protocol for this benchmark. We use TrackEval [51] for validation-set evaluation, while test-set results are evaluated on the official benchmark servers.

5.4.2 Implementation Details

As in BoostTrack, we use YOLOX-X [38] as the detector. For MOT17 and MOT20, we use the weights from [144]; for DanceTrack, the weights from [15]; and for SportsMOT, the weights from [23]. We apply Enhanced Correlation Coefficient maximization [32] for camera motion compensation, using the implementation from [30]. For the Kalman filter, we use the same state-space model as introduced in Section 2.3. The initial state covariance matrix and the initial noise covariance matrices are set to constant values scaled by object height, following the standard approach from [130]. However, CAT uses these heuristic values only for initialization, since the noise covariance matrices are subsequently estimated online.

We discard only detections with a very low confidence score below $\tau_{\text{det}} = 0.01$ and apply NMS with threshold $\tau_{\text{NMS}} = 0.8$. Compared to common settings used in previous works, such as $\tau_{\text{det}} = 0.1$ and $\tau_{\text{NMS}} = 0.7$ [110, 144], this allows CAT to work with a larger set of detections. We set the minimum similarity threshold to $\tau_s = 0.2$. The value $\lambda_{\text{boost}} = 0.5$ is adopted from BoostTrack. Since CAT does not use the shape similarity

component from BoostTrack, we add its weight to the Mahalanobis distance similarity term and set $\lambda_M = 0.5$.

Unmatched objects are kept for at most $A_{\max} = 75$ frames before being removed, or for $A_{\text{os}} = 10$ frames if we predict that the object has left the scene. For SportsMOT, we set $A_{\text{os}} = A_{\max} = 75$ due to dataset-specific characteristics. The forgetting factor used for noise estimation is set to $\alpha = 0.3$, as in [5]. We set the maximum expansion scale for EIoU to $s_{\max} = 4$ and the trajectory update threshold to $\tau_u = 0.25$.

Datasets differ in lighting, resolution, camera viewpoint, object scale, occlusions, and motion patterns, which leads to different detector behavior across datasets. In practice, this is reflected in different detection confidence distributions and different numbers of false-positive detections. Therefore, we use dataset-specific values for the detection-confidence-related hyperparameters α_s and τ_{tc} . These values are shown in Table 5.1. The influence of these hyperparameters is analyzed in Section 5.4.6.

Table 5.1: Dataset-specific values of α_s and τ_{tc} , used in Eqs. (5.9) and (5.10).

Dataset	α_s	τ_{tc}
MOT17	0.55	0.25
MOT20	0.40	0.20
DanceTrack	0.55	0.50
SportsMOT	0.15	0.40

Experiments were conducted on a system with an Intel Core i9-13900K CPU and an NVIDIA GeForce RTX 3080 GPU.

5.4.3 Baseline Setting

For a fair comparison, the baseline method follows the same general tracking framework as CAT, but without the proposed adaptive CAKF, EIoU, and improved association workflow. It uses a Kalman filter with constant height-scaled noise covariance matrices. Detections with confidence score below $\tau_{\text{det}} = 0.1$ are discarded. Association is performed in one stage using the combined similarity from Eq. (5.7), but with IoU instead of EIoU and with tracklet confidence computed as in BoostTrack (see Chapter 3). A matched detection-tracklet pair is considered valid if its similarity is greater than $\tau_s = 0.2$. Unmatched detections with confidence scores greater than 0.5 are used to initialize new trajectories. The trajectory of a given object is updated only after three consecutive matches.

5.4.4 Comparison with State-of-the-Art Methods

We evaluate CAT on the MOT20, SportsMOT, MOT17, and DanceTrack test sets. For MOT17 and MOT20, we select strong methods ranked by HOTA on the official MOTChallenge [86] and Codabench [132] benchmark pages. For DanceTrack and SportsMOT, we collect results from the top-ranked methods on MOT17 and MOT20, as well as from widely used benchmark methods for highly dynamic tracking scenarios.

CAT is an online motion-based tracking method, and its primary comparison is therefore with other online motion-based trackers. However, since our goal is to develop a reliable tracker that remains competitive with offline and appearance-based methods, we also compare CAT with methods from these categories.

Following the classification introduced in Section 2.7, we distinguish methods along two axes: the type of information used for association, i.e., motion-based or appearance-based, and the temporal processing type, i.e., online or offline. Since the comparison in this chapter focuses on a stricter online setting, we consider a method offline if it uses any post-processing of the tracking results, such as linear interpolation of fragmented trajectories.²

Table 5.2 presents the comparison on the MOT20 test set. CAT outperforms all online motion-based methods in HOTA and IDF1, improving over the second-best online motion-based result by +1.8 HOTA and +3.9 IDF1. It also achieves the second-best MOTA and IDSW scores among online motion-based methods. Furthermore, although CAT does not use appearance features or post-processing, it achieves higher HOTA and IDF1 than all offline motion-based methods and all online appearance-based methods listed in the table. It also outperforms several widely used offline appearance-based trackers, such as BoT-SORT and Deep OC-SORT.

The comparison on SportsMOT is shown in Table 5.3. CAT outperforms all online motion-based methods in HOTA, IDF1, and IDSW. It also outperforms OC-SORT, a standard offline motion-based benchmark, and achieves performance comparable to that of MoveSORT, a more recent offline motion-based method. However, appearance-based methods still outperform CAT by a noticeable margin, indicating that ReID features remain important for handling highly dynamic motion patterns in sports scenarios.

In Table 5.4, we compare CAT with existing methods on the MOT17 test set. Although CAT does not achieve the best performance among online motion-based methods, it still ranks among the top methods in this category, achieving the second-best HOTA score. Its

²Throughout this section, we group methods according to online/offline operation and motion/appearance-based association. The authors of SparseTrack [69] and Hybrid-SORT [135] describe their methods as online and do not explicitly discuss post-processing in the corresponding papers. However, their publicly available implementations include post-processing steps. Therefore, for strict online comparison, we classify the reported post-processed results as offline and use the results without post-processing when comparing online methods.

Table 5.2: Comparison with state-of-the-art methods on the MOT20 test set. Methods are grouped into offline/online and appearance/motion-based categories. Best results in each category are shown in bold.

Method	HOTA	IDF1	MOTA	IDSW
<i>Offline appearance-based</i>				
RTU++ [126]	62.8	76.8	76.5	971
BoT-SORT [3]	63.3	77.5	77.8	1257
FineTrack [97]	63.6	79.0	77.9	980
Hybrid-SORT* [135]	63.9	78.4	76.7	1621
Deep OC-SORT [75]	63.9	79.2	75.6	779
SUSHI [17]	64.3	79.8	74.3	706
ConfTrack [52]	64.8	80.2	77.2	702
StableTrack [105]	65.1	80.2	76.9	830
TrackTrack [106]	65.7	80.9	78.0	741
SPAM [18]	65.8	81.9	76.5	961
CoNo-Link [37]	65.9	81.8	77.5	956
RTAT [43]	66.2	82.5	78.4	787
BoostTrack++ [115]	66.4	82.0	77.7	762
<i>Offline motion-based</i>				
OC-SORT [15]	62.1	75.9	75.5	913
Hybrid-SORT [135]	62.5	76.2	76.4	/
BoT-SORT [3]	62.6	76.3	77.7	1212
UCMCTrack [137]	62.8	77.4	75.7	1335
BoostTrack [114]	63.0	76.5	76.4	992
SparseTrack [69]	63.4	77.3	78.2	1116
BASE [59]	63.5	77.6	78.2	984
<i>Online appearance-based</i>				
UTM [138]	53.3	65.9	64.4	2592
GHOST [103]	61.2	75.2	73.7	1264
QuoVadis [26]	61.5	75.7	77.8	1187
BUSCA [121]	61.8	76.3	72.7	1006
SAT [127]	62.6	76.6	75.0	816
LG-Track [81]	63.4	77.4	77.8	1161
<i>Online motion-based</i>				
Hybrid-SORT [135]	57.8	70.2	72.2	1621
SFSORT [85]	60.9	73.5	75.0	1322
FOR_Tracking [89]	61.4	76.4	76.8	1443
SparseTrack [69]	62.4	76.5	75.1	935
MotionTrack [95]	62.8	76.5	78.0	1165
CAT	64.6	80.4	77.1	1037

Table 5.3: Comparison with state-of-the-art methods on the SportsMOT test set. We mark the only two offline motion-based methods with * instead of using a separate category. Best results in each category are shown in bold.

Method	HOTA	IDF1	MOTA	IDSW
<i>Offline</i>				
ByteTrack [144]	64.1	71.4	95.9	3089
SPAM [18]	67.5	84.6	80.7	1161
BoT-SORT [3]	68.7	70.0	94.5	6729
OC-SORT* [15]	73.7	74.0	96.5	2728
MixSort-OC [23]	74.1	74.4	96.5	2781
MoveSORT* [2]	74.5	76.2	96.8	N/A
Deep EIoU [47]	77.2	79.8	96.3	2659
<i>Online appearance-based</i>				
WDTtrack [148]	73.8	80.5	95.3	N/A
DiffMOT [73]	76.2	76.1	97.1	N/A
McByte [113]	76.9	77.5	97.2	N/A
SportMamba [54]	77.3	77.7	96.9	N/A
Deep HM-SORT [41]	80.1	85.2	96.6	1703
<i>Online motion-based</i>				
Hybrid-SORT [135]	71.3	70.6	95.5	3526
SFSORT [85]	72.3	71.2	96.9	4400
SparseTrack [69]	73.2	74.4	97.0	2639
MotionTrack [95]	74.0	74.0	96.6	N/A
CAT	74.4	75.0	96.5	2307

performance is also comparable to that of offline motion-based and online appearance-based methods. However, the best offline appearance-based approaches outperform CAT by a noticeable margin.

Finally, in Table 5.5, we compare CAT with existing methods on the DanceTrack test set. Among online motion-based methods, CAT achieves the best performance on most metrics. It also outperforms all offline motion-based methods except UCMCTrack, which achieves slightly better results. However, appearance-based methods generally outperform CAT on this benchmark.

To provide a compact visual summary of the benchmark comparison, selected methods from the previous tables are shown in Figure 5.8.

Overall, the results show that CAT is particularly strong in crowded scenes and remains competitive in dynamic scenarios, despite using only motion information. The remaining gap on SportsMOT and DanceTrack suggests that appearance information is still needed when visually similar objects undergo frequent and abrupt motion changes.

Table 5.4: Comparison with state-of-the-art methods on the MOT17 test set. Methods are grouped into offline/online and appearance/motion-based categories. Best results in each category are shown in bold.

Method	HOTA	IDF1	MOTA	IDSW
<i>Offline appearance-based</i>				
FineTrack [97]	64.3	79.5	80.0	1272
StrongSORT++ [30]	64.4	79.5	79.6	1194
Deep OC-SORT [75]	64.9	80.6	79.4	1023
BoT-SORT [3]	65.0	80.2	80.5	1212
ConfTrack [52]	65.4	81.2	80.0	1155
SUSHI [17]	66.5	83.1	81.1	1149
BoostTrack++ [115]	66.6	82.2	80.7	1062
StableTrack [105]	67.1	83.0	82.0	921
TrackTrack [106]	67.1	83.1	81.8	822
CoNo-Link [37]	67.1	83.7	82.7	1092
RTAT [43]	67.2	84.7	80.4	912
SPAM [18]	67.5	84.6	80.7	1161
<i>Offline motion-based</i>				
FastTrack [64]	63.4	77.6	80.5	2013
Hybrid-SORT [135]	63.6	78.4	79.3	N/A
BASE [59]	64.5	78.6	81.9	1281
BoT-SORT [3]	64.6	79.5	80.6	1257
SparseTrack [69]	65.1	80.1	81.0	1170
BoostTrack [114]	65.4	80.2	80.5	1104
UCMCTrack [137]	65.8	81.1	80.5	1689
<i>Online appearance-based</i>				
GHOST [103]	62.8	77.1	78.7	2325
QuoVadis [26]	63.1	77.7	80.3	2103
BUSCA [121]	63.9	79.2	78.6	1428
Hybrid-SORT [135]	64.0	78.7	79.9	N/A
SAT [127]	64.4	79.8	80.0	1356
LG-Track [81]	65.4	80.4	81.4	1125
<i>Online motion-based</i>				
Hybrid-SORT [135]	60.7	74.7	77.1	2559
SFSORT [85]	61.7	74.4	78.8	3264
FOR_Tracking [89]	63.3	77.7	80.4	2298
SparseTrack [69]	64.0	79.4	79.0	1359
C-BIoU [133]	64.1	79.7	81.1	/
MotionTrack [95]	65.1	80.1	81.1	1140
CAT	64.4	79.6	79.4	1887

Table 5.5: Comparison with state-of-the-art methods on the DanceTrack test set. Methods are grouped into offline/online and appearance/motion-based categories. Best results in each category are shown in bold.

Method	HOTA	IDF1	MOTA	AssA	DetA
<i>Offline appearance-based</i>					
MixSort-Byte [23]	46.7	53.0	85.8	31.9	68.6
ByteTrack [144]	47.1	51.9	88.2	31.5	70.5
FineTrack [97]	52.7	59.8	89.9	38.5	72.4
Deep OC-SORT [75]	61.3	61.5	92.3	45.8	82.2
SUSHI [17]	63.3	63.4	88.7	50.1	80.1
CoNo-Link [37]	63.8	64.1	89.7	50.7	80.2
SPAM [18]	64.0	63.4	89.2	N/A	N/A
StableTrack [105]	66.3	67.8	92.0	53.0	N/A
TrackTrack [106]	66.5	67.8	93.6	52.9	N/A
<i>Offline motion-based</i>					
OC-SORT [15]	55.1	54.9	92.2	40.4	80.4
SparseTrack [69]	55.5	58.3	91.3	39.1	78.9
MoveSORT [2]	56.1	56.0	91.8	38.7	81.6
BASE [59]	56.4	/	91.7	/	/
FastTrack [64]	57.4	58.2	92.8	40.7	81.1
UCMCTrack [137]	63.6	65.0	88.9	51.3	/
<i>Online appearance-based</i>					
BUSCA [121]	55.5	56.1	91.5	38.0	81.4
GHOST [103]	56.7	57.7	91.3	39.8	81.1
DiffMOT [73]	62.3	63.0	92.8	47.2	82.5
Hybrid-SORT [135]	65.7	67.4	91.8	/	/
WDTtrack [148]	66.8	72.8	91.5	55.9	80.0
McByte [113]	67.1	68.1	92.9	/	/
<i>Online motion-based</i>					
SFSORT [85]	52.5	50.5	91.7	33.9	81.4
SparseTrack [69]	53.1	52.6	92.0	35.1	80.7
OC-SORT [15]	54.6	54.6	89.6	40.2	80.4
C-BIoU [133]	60.6	61.6	91.6	45.4	81.3
Hybrid-SORT [135]	62.2	63.0	91.6	/	/
CAT	61.7	64.2	92.1	47.2	80.9

Runtime Performance

Since MOT20 contains the most crowded scenes and the largest number of objects per frame among the evaluated datasets, a method that achieves real-time performance on MOT20 can be expected to operate in real time on the other datasets as well. In Table 5.6,

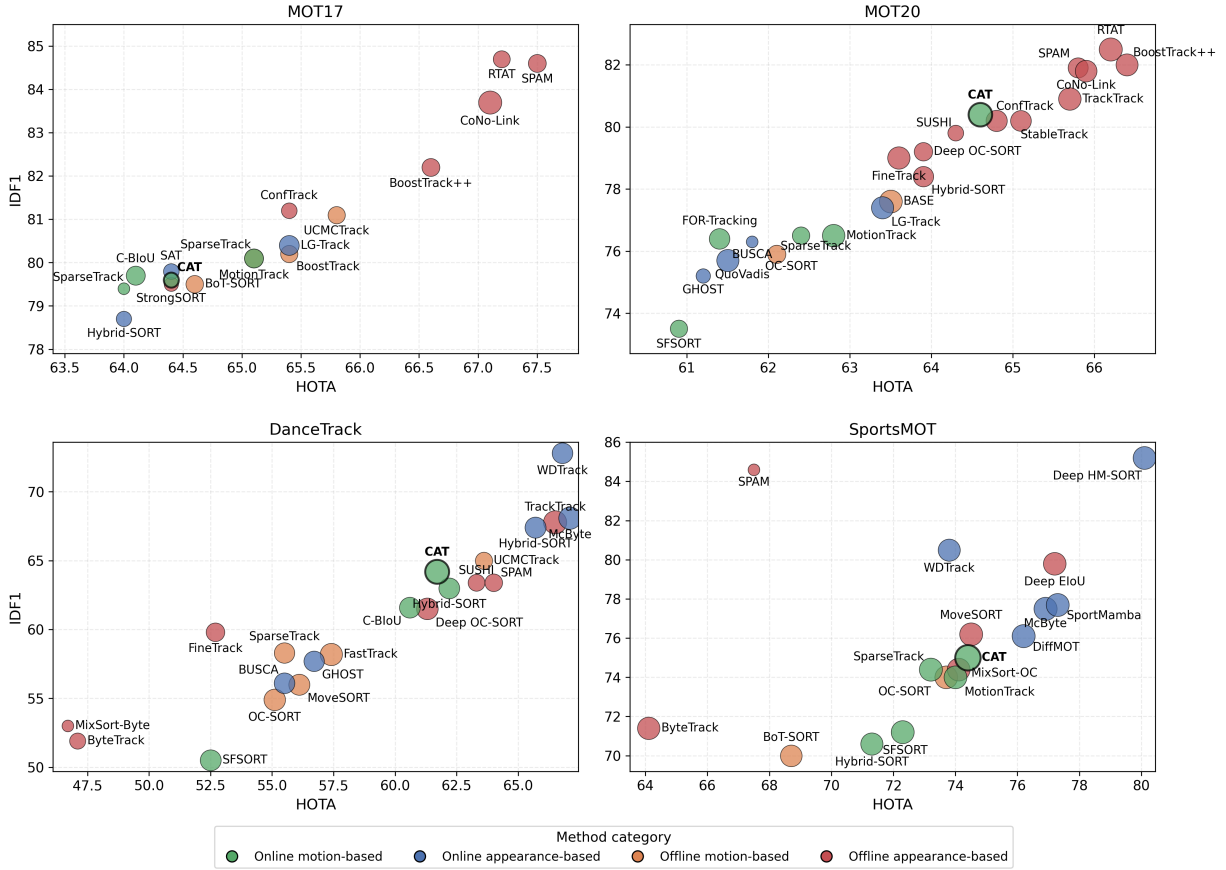


Figure 5.8: Comparison of CAT with selected MOT methods on MOT17, MOT20, DanceTrack, and SportsMOT. HOTA and IDF1 are shown on the horizontal and vertical axes, respectively, while marker size reflects MOTA. Colors denote method categories. For clarity, only selected methods from the complete comparison tables are shown.

we report the runtime of different online motion-only methods for each sequence of the MOT20 test set. We measure the speed of online motion-only methods with publicly available GitHub implementations. All measurements are performed on the same hardware, described in Section 5.4.2. Following common practice in the MOT literature, all reported speeds exclude detector runtime and refer only to the tracking pipeline.

CAT achieves real-time performance on all MOT20 test sequences, with the lowest speed of 60.3 FPS on MOT20-4, the most crowded sequence in the test set. Due to the efficient Kalman filter implementation described in Appendix C.1, the main computational bottleneck of CAT is camera motion compensation (see Section 5.4.2).

SparseTrack and SFSORT are faster than CAT. SFSORT achieves the highest runtime performance among all compared methods, with FPS values ranging from 129.2 to 691.9 across different sequences. However, as shown in Tables 5.2 to 5.5, CAT achieves stronger tracking performance than these methods across the evaluated benchmarks, while still maintaining real-time speed even on the most crowded MOT20 sequences.

Table 5.6: Runtime comparison of different online motion-only methods in terms of frames per second (FPS) on the MOT20 test set.

Sequence	MOT20-4	MOT20-6	MOT20-7	MOT20-8
FOR_Tracking [89]	40.4	52.2	170.3	69
Hybrid-SORT [135]	54.5	52.8	114.8	70.8
SparseTrack [69]	87.6	111.6	154.4	127.8
SFSORT [85]	129.2	407.1	798.9	691.9
CAT	60.3	65.4	107.5	76.2

5.4.5 Ablation Study

In this subsection, we provide an ablation study and report the results of different combinations of the proposed components. To avoid combinatorial explosion, we group the proposed additions into three components: adaptive Crowd-Aware Kalman Filter (Adapt. CAKF), Expanded IoU (EIoU), and Improved Association Workflow (IAW). IAW includes the usage of KF-based tracklet confidence, Crowd-Aware Tracklet Creation, association validity, and trajectory update criteria, i.e., all components proposed in Section 5.3.3, together with a lowered detection threshold $\tau_{\text{det}} = 0.01$. We analyze these components in detail in Section 5.4.6, while Table 5.7 presents the overall ablation results on the MOT20 and SportsMOT validation sets.

Table 5.7: Ablation study CAT components on the MOT20 and SportsMOT validation sets. Best results are shown in bold.

Setting			MOT20				SportsMOT			
Adapt. CAKF	EIoU	IAW	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
$\times$	$\times$	$\times$	60.20	74.62	82.67	2021	72.84	74.42	94.66	870
$\times$	$\times$	$\checkmark$	62.15	76.64	83.35	1412	74.76	77.05	95.16	788
$\times$	$\checkmark$	$\times$	57.67	71.16	82.39	2657	71.66	72.92	94.68	876
$\times$	$\checkmark$	$\checkmark$	62.64	77.82	84.13	1305	74.83	77.51	95.20	664
$\checkmark$	$\times$	$\times$	63.62	80.71	82.91	1402	74.54	76.11	94.41	881
$\checkmark$	$\times$	$\checkmark$	66.17	83.63	85.01	1072	76.16	77.79	95.00	886
$\checkmark$	$\checkmark$	$\times$	63.00	79.78	82.57	1696	75.13	76.95	94.53	779
$\checkmark$	$\checkmark$	$\checkmark$	66.79	85.12	85.27	975	77.15	79.33	95.02	696

When used without other newly introduced components, Adapt. CAKF and IAW consistently improve tracking performance, while adding only EIoU to the baseline degrades association and overall performance. On both datasets, using IAW improved MOTA score (+0.68 on MOT20 and +0.5 on SportsMOT) suggesting better use of true-positive detections. Adaptive CAKF improves HOTA and IDF1, with only minor changes in MOTA (+0.24 on MOT20 and -0.25 on SportsMOT).

Using any two components improves performance relative to the baseline, and, with the only exception being Adapt. CAKF+EIoU relative to Adapt. CAKF on MOT20, any two component setting outperforms one component setting, i.e., even though adding EIoU to the baseline degrades tracking performance, EIoU combined with adaptive CAKF or IAW generally improves the performance.

Adapt. CAKF+IAW is the best two-component setting on both datasets. Compared to the baseline, it improves MOT20 by +5.97 HOTA, +9.01 IDF1, +2.34 MOTA, and reduces IDSW by 949. On SportsMOT, it improves HOTA by +3.32, IDF1 by +3.37, and MOTA by +0.34, with an increase of 16 IDSW.

When all components are combined, the method achieves the best overall performance on both datasets. Compared to the best two-component setting, the full configuration, i.e., adding EIoU to Adapt. CAKF+IAW, further improves MOT20 by +0.62 HOTA, +1.39 IDF1, +0.26 MOTA, and reduces IDSW by 97. On SportsMOT, the gains are +0.99 HOTA, +1.54 IDF1, +0.02 MOTA, and −190 IDSW. These results confirm that the proposed components are complementary and that their combination leads to consistent improvements, supporting the design principle of component coupling.

Ablation results on DanceTrack and MOT17 show similar trends and are provided in Appendix C.2.

5.4.6 Component Analysis

In this subsection, we provide a detailed analysis of the proposed components. The components are analyzed in the same order in which they are introduced in Section 5.3.

Adaptive CAKF

To evaluate the effectiveness of adaptive CAKF, we separate the effects of crowd-aware scaling and adaptive estimation of noise covariance matrices. The results on the MOT20 and SportsMOT validation sets are shown in Table 5.8.

The baseline uses a Kalman filter with constant height-scaled noise covariance matrices. We then evaluate the effect of adding crowd-aware scaling alone, denoted as CAKF, and adaptive noise estimation alone, denoted as Adapt. KF. Finally, the last row of Table 5.8 shows the effect of combining both components, resulting in Adapt. CAKF.

On MOT20, CAKF alone results in only a slight improvement. Adapt. KF improves IDF1 compared to the baseline by +0.85, but it also decreases MOTA by −1.34. However, when combined, these two additions significantly improve overall tracking performance, resulting in +3.42 HOTA, +6.09 IDF1, +0.24 MOTA, and −619 IDSW. On SportsMOT, using only Adapt. KF degrades performance. However, when combined with crowd-

Table 5.8: Ablation study of adaptive covariance estimation and crowd-aware scaling on MOT20 and SportsMOT validation sets. Best results are shown in bold.

Dataset	Adapt. KF	CAKF	HOTA	IDF1	MOTA	IDSW
MOT20	$\times$	$\times$	60.20	74.62	82.67	2021
	$\times$	$\checkmark$	60.39	74.88	82.68	2005
	$\checkmark$	$\times$	60.04	75.47	81.33	2072
	$\checkmark$	$\checkmark$	63.62	80.71	82.91	1402
SportsMOT	$\times$	$\times$	72.84	74.42	94.66	870
	$\times$	$\checkmark$	73.88	74.07	94.62	894
	$\checkmark$	$\times$	36.65	35.03	69.41	3015
	$\checkmark$	$\checkmark$	74.54	76.11	94.41	881

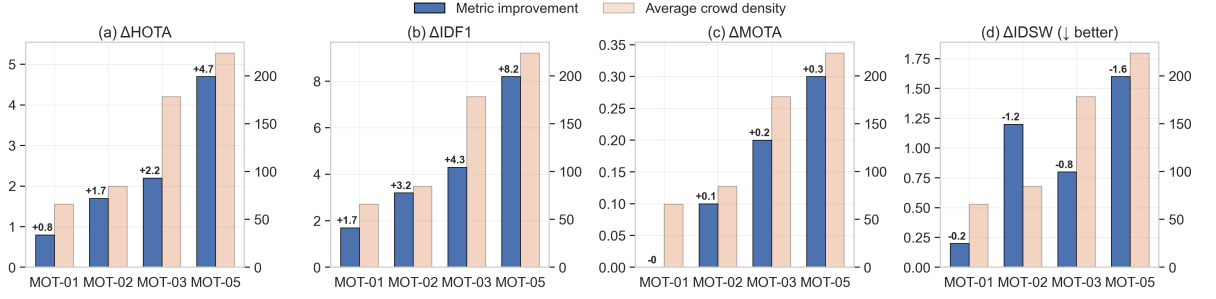


Figure 5.9: Per-sequence improvements in HOTA, IDF1, MOTA, and density-relative IDSW achieved by Adapt. CAKF on the MOT20 validation set. For each metric, we show the performance improvement together with the average crowd density of the corresponding sequence.

aware scaling, Adapt. CAKF achieves the best HOTA and IDF1 scores, improving over the baseline by +1.70 HOTA and +1.69 IDF1.

In Figure 5.9, we present the per-sequence performance gains achieved by adaptive CAKF on the MOT20 validation set. We compare these gains with the average number of tracked objects per frame, computed from the ground-truth data. Since IDSW represents the total number of identity switches and is expected to increase with the number of objects in the scene, we divide IDSW by the average scene density to obtain a more meaningful density-relative value. If the number of objects per frame is used as a quantitative indicator of scene density, the improvements achieved by Adapt. CAKF increase monotonically as the scene density (i.e., crowdedness) increases.

Additional experiments on the MOT17, and DanceTrack validation sets, as well as a detailed noise covariance analysis, are provided in Appendix C.3.

Table 5.9: Influence of the maximum scaling factor $s_{\max}$ on tracking performance on SportsMOT validation set. Best results are shown in bold.

$s_{\max}$	HOTA	IDF1	MOTA	IDSW
1	74.54	76.11	94.41	881
2	75.14	76.95	94.57	769
3	75.12	76.90	94.54	787
4	75.13	76.95	94.53	779
5	74.95	76.71	94.53	799
10	75.14	76.98	94.51	801
∞	75.12	77.00	94.46	790

EIoU

The scaling factors s_w and s_h used to calculate EIoU are determined by the predicted state covariance, which in turn depends on the underlying Kalman filter. Therefore, we analyze the effect of using EIoU under two tracking settings: the baseline Kalman filter and the proposed adaptive CAKF.

Figure 5.10 illustrates how the mean values of s_w and s_h change as the number of steps since the last successful association increases. Due to noisy observations and occasional tracking or association errors, some objects may produce extremely large covariance values, which can disproportionately affect the mean. To better show the overall trend, we exclude the top 5% of values when computing the mean.

The figure shows that the horizontal expansion factor s_w is consistently higher than the vertical expansion factor s_h across both Kalman filter settings and both datasets. This can be attributed to the camera perspective and typical motion patterns in the evaluated sequences: object height and vertical position are usually more stable than horizontal position. As a result, CAT expands bounding boxes more in width than in height.

The mean scaling factors s_w and s_h increase approximately linearly as the number of steps since the last successful association increases. For example, when using adaptive CAKF on SportsMOT, the mean s_w increases from 1.31, corresponding to a 31% width expansion, to 9.33, corresponding to an 833% width expansion, as the number of unmatched steps increases from 1 to 8. This corresponds to an average increase of approximately 1.15 per step.

Table 5.9 reports the influence of the maximum scaling factor $s_{\max}$ on tracking performance on the SportsMOT validation set. We use adaptive CAKF as the Kalman filter setting. For $s_{\max} = 1$, there is no bounding-box expansion and EIoU reduces to standard IoU. Therefore, the first row of Table 5.9 represents the corresponding baseline setting.

We observe that performance remains stable across a wide range of $s_{\max}$ values, in-

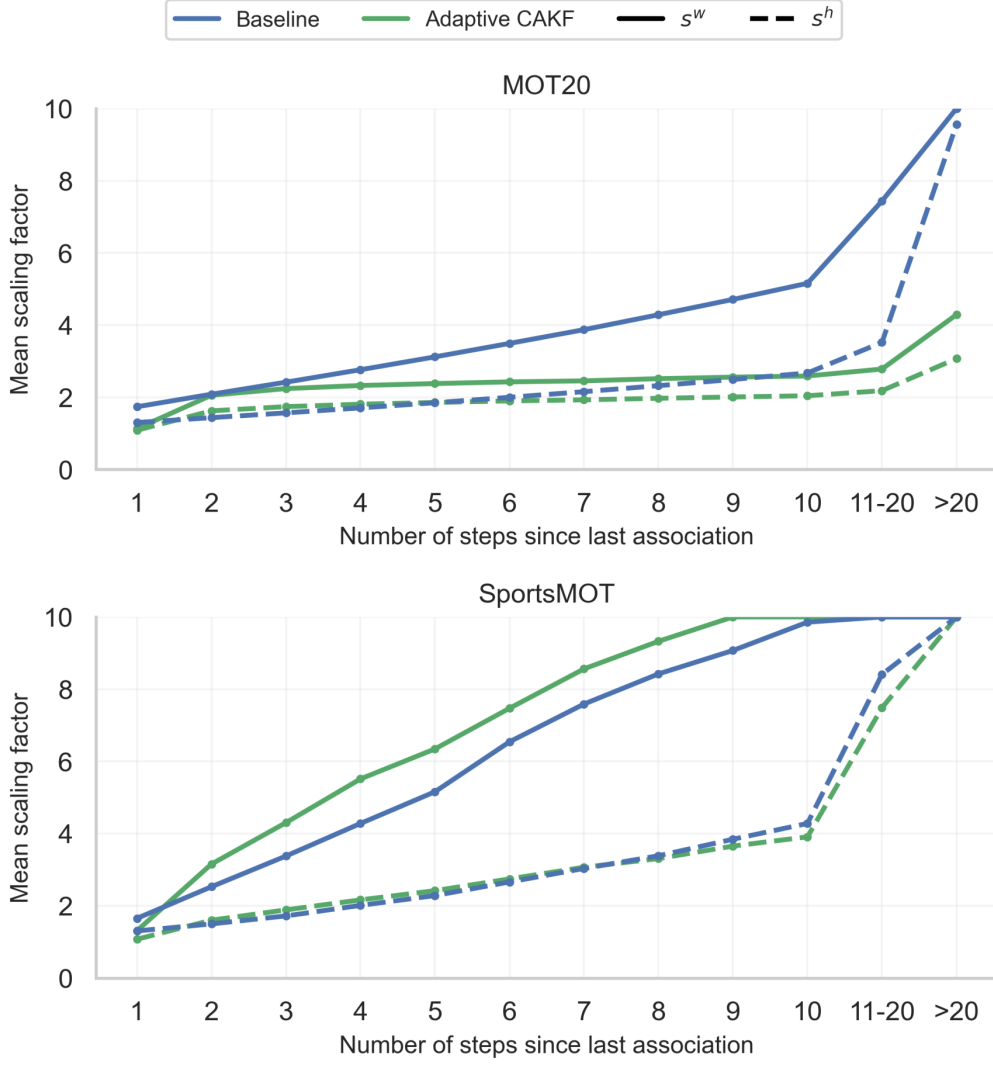


Figure 5.10: Mean values of the scaling factors s_w and s_h over time on the MOT20 and SportsMOT validation sets. Solid lines correspond to s_w , while dashed lines correspond to s_h . The scaling factors are calculated using either the baseline Kalman filter or adaptive CAKF. Values above 10 are clipped for visualization purposes.

dicating that the proposed method is not overly sensitive to this parameter. As the best trade-off for the most relevant metrics on all datasets, we use $s_{\max} = 4$ in all experiments.

Table 5.10 summarizes the influence of incorporating EIoU on the MOT20 and SportsMOT validation sets under both the baseline Kalman filter and adaptive CAKF settings.

When combined with the baseline Kalman filter, adding EIoU degrades performance on both datasets. On SportsMOT, however, integrating EIoU with adaptive CAKF yields consistent improvements, including +0.59 HOTA, +0.84 IDF1, +0.12 MOTA, and -102 IDSW. The improvements in IDF1 and IDSW indicate improved association in motion-dominated scenarios.

However, on MOT20, the incorporation of EIoU even within the adaptive CAKF

Table 5.10: Effect of EIou on MOT20 and SportsMOT validation sets with different KF settings. Best results are shown in bold.

Dataset	Adapt. CAKF	EIoU	HOTA	IDF1	MOTA	IDSW
MOT20	$\times$	$\times$	60.20	74.62	82.67	2021
	$\times$	$\checkmark$	57.67	71.16	82.39	2657
	$\checkmark$	$\times$	63.62	80.71	82.91	1402
	$\checkmark$	$\checkmark$	63.00	79.78	82.57	1696
SportsMOT	$\times$	$\times$	72.84	74.42	94.66	870
	$\times$	$\checkmark$	71.66	72.92	94.68	876
	$\checkmark$	$\times$	74.54	76.11	94.41	881
	$\checkmark$	$\checkmark$	75.13	76.95	94.53	779

framework leads to performance degradation. The most pronounced drops are observed in IDF1 (-0.93) and IDSW (+294), which suggest reduced association performance. This effect is mitigated by the improved association workflow. In Section 5.4.6, we show that when EIou is integrated within the proposed association workflow, it also improves association performance on MOT20.

Improved Association Workflow

We test the influence of the four additions introduced in Section 5.3.3, following the order in which they were introduced.

KF-based tracklet confidence. Our KF-based tracklet confidence (KFTC) encodes the estimated uncertainty of the predicted state. KFTC decreases when the Kalman update step is not performed, i.e., when a successful association is not obtained. In Figure 5.11, we show how KFTC values decrease as the number of steps since the last successful association increases. Since different tracked objects can have different confidence values for the same number of steps since the last association, we show the 25% quantile, median, and 75% quantile of KFTC values on MOT20 and SportsMOT. The baseline uses the tracklet confidence formulation from BoostTrack, which is calculated only as a function of the number of steps since the last tracklet update. We also show the corresponding baseline tracklet confidence decay in Figure 5.11.

On MOT20, where object motion is easier to predict, the confidence of approximately 25% of the least challenging objects, i.e., the 75% quantile, remains greater than 0.8 even after 10 consecutive frames without a successful association, while the 25% quantile drops to zero after 6 steps. The median confidence remains above the baseline for 10 steps.

On the more dynamic SportsMOT dataset, the 25% quantile and median values drop to zero after 2 and 5 consecutive association failures, respectively, while the 75% quantile

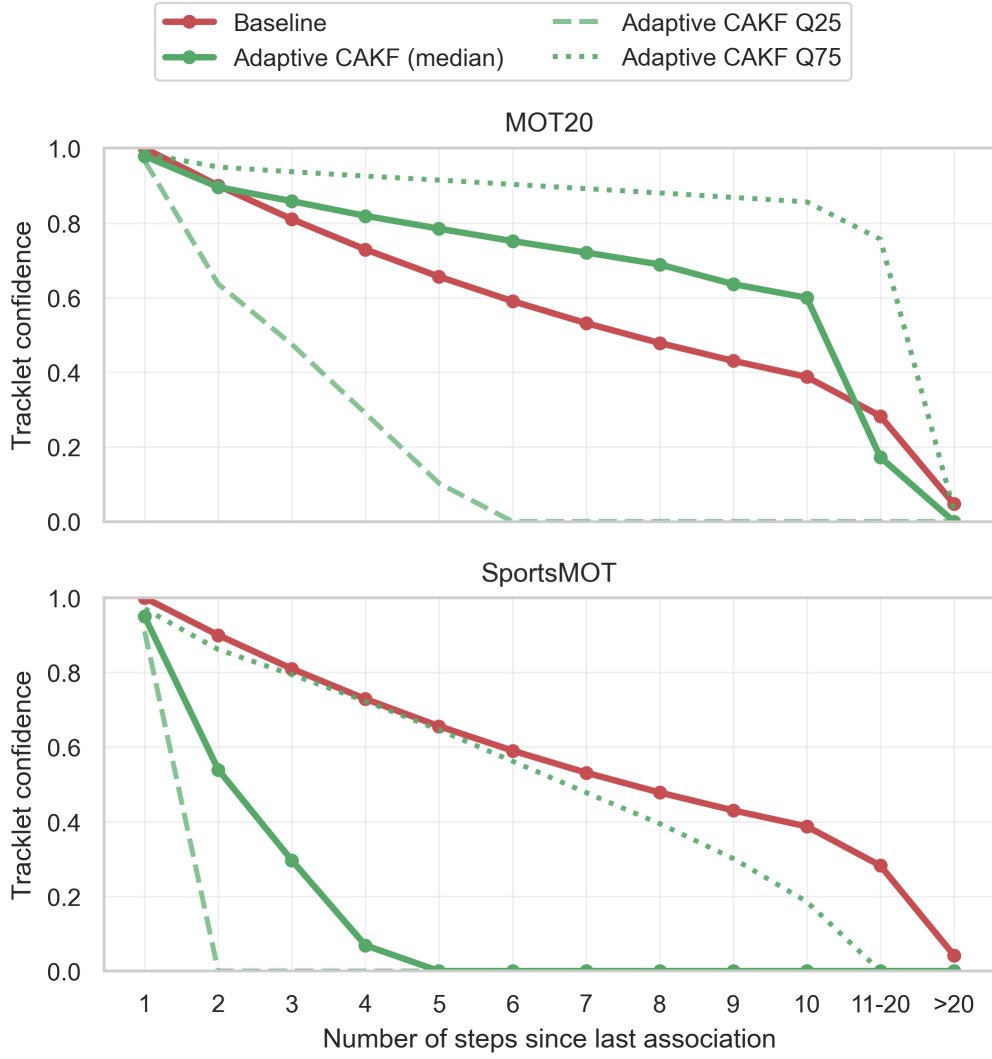


Figure 5.11: Median, 25% quantile, and 75% quantile of KFTC over time on the MOT20 and SportsMOT validation sets, shown in green. The red line denotes the baseline tracklet confidence formulation from BoostTrack.

remains approximately the same as the baseline confidence.

The results illustrate that KFTC adapts the confidence decay to the estimated tracking uncertainty, rather than enforcing a fixed decay schedule as in the baseline approach.

In Table 5.11, we report the results of using KFTC on the MOT20 and SportsMOT validation sets. Since KFTC depends on the Kalman filter setting, we test it with two baselines: the standard KF and adaptive CAKF. All settings in Table 5.11 use EIoU.

With both baseline configurations, KFTC provides only a marginal benefit on MOT20. However, on SportsMOT, KFTC significantly improves association. With the standard KF baseline, it improves IDF1 by +0.87 and HOTA by +0.62, while reducing IDSW by 37. With adaptive CAKF as the baseline, the improvements are more notable: +2.04 IDF1, +1.25 HOTA, and -67 IDSW.

Table 5.11: Influence of KF-based tracklet confidence (KFTC) under the baseline KF and adaptive CAKF settings on the MOT20 and SportsMOT validation sets. Best results are shown in bold.

Dataset	Adapt. CAKF	KFTC	HOTA	IDF1	MOTA	IDSW
MOT20	$\times$	$\times$	57.67	71.16	82.39	2657
	$\times$	$\checkmark$	57.72	71.23	82.41	2684
	$\checkmark$	$\times$	63.00	79.78	82.57	1696
	$\checkmark$	$\checkmark$	62.97	79.86	82.58	1576
SportsMOT	$\times$	$\times$	71.66	72.92	94.68	876
	$\times$	$\checkmark$	72.28	73.79	94.72	839
	$\checkmark$	$\times$	75.13	76.96	94.53	779
	$\checkmark$	$\checkmark$	76.38	79.00	94.61	712

Association and trajectory initialization criteria. To evaluate the proposed Crowd-Aware Tracklet Creation (CATC) and the new Association Validity Criterion (AVC), which aim to improve association by using all detected bounding boxes, we reduce the detection confidence threshold τ_{det} from 0.1 to 0.01. Unmatched detections are treated as candidates for tracklet creation, and newly created tracklets can subsequently influence the association of existing tracklets. As a result, CATC and AVC are interdependent and should be analyzed jointly.

Since CATC and AVC depend on detection confidence scores, it is important to examine how these scores are distributed across datasets. Figure 5.12 shows the distribution of detection confidence scores for true-positive detections on the MOT20 and SportsMOT validation sets, where a detection is considered true positive if it is matched with a ground-truth object. Due to occlusions, crowding, object size variations, and challenging lighting conditions in some scenes, detection on MOT20 is more difficult, which is reflected in lower confidence scores. As a result, 4.82% of true-positive detections in MOT20 and 0.43% in SportsMOT have confidence score $c_D < 0.1$. In the standard setting, such detections would be discarded. Similarly, 7.88% of true-positive detections in MOT17 and 2.94% in DanceTrack have $c_d < 0.1$ (see Appendix C.5.2).

The differences in detection confidence score distributions across datasets suggest that the crowd-aware tracklet creation threshold τ_{tc} (see Eq. (5.10)) and the admissible match hyperparameter α_s (see Eq. (5.9)), used in CATC and AVC, should be tuned for each dataset. To analyze their influence, we perform a grid search and determine the optimal values of τ_{tc} and α_s for each dataset.

For a fair comparison, because the optimal values of τ_{tc} and α_s for one baseline configuration may not be optimal for another, we report grid-search results for multiple settings in Figure 5.13 to evaluate both the impact of τ_{tc} and α_s and the effectiveness of our new

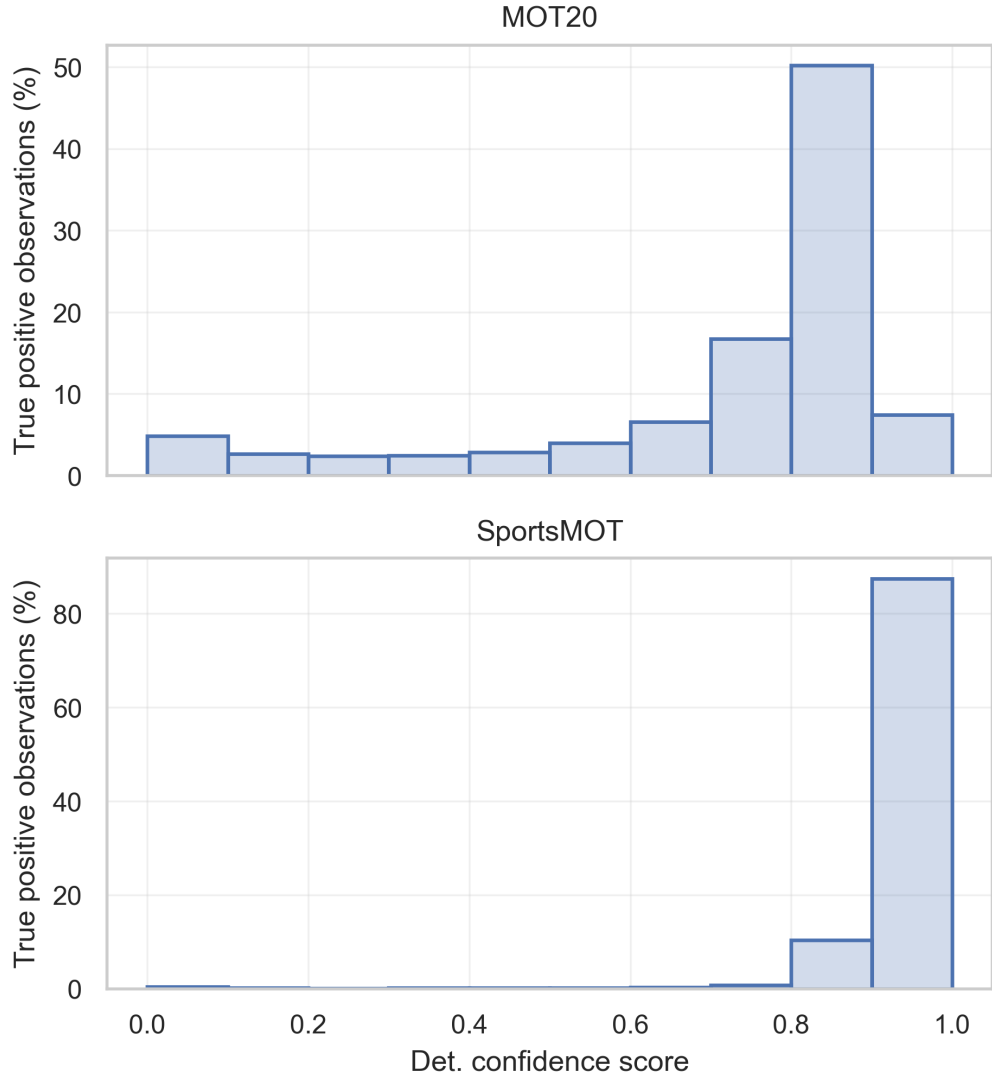


Figure 5.12: Histogram of detection confidence scores for true-positive detections on the MOT20 and SportsMOT validation sets. Each bin represents a confidence interval of width 0.1.

association method across different baselines. For clarity, we restrict the comparison to the HOTA metric, which reflects overall tracking performance. All tested settings use adaptive CAKF. Above each heatmap, we indicate the additional setting, namely the use of KFTC or EIoU, as well as the value of τ_{det} . For readability, we show only the 4×4 heatmap patch containing the optimum for each setting.

Adding CATC and AVC to the EIoU baseline results in a +1.63 HOTA improvement on MOT20 when $\tau_{\text{det}} = 0.1$, using the best combination of τ_{tc} and α_s . On SportsMOT, the corresponding improvement is +1.73 HOTA compared to the results from Table 5.11. The addition of KFTC provides only a negligible improvement over the best EIoU+CATC+AVC setting on both datasets: +0.01 HOTA on MOT20 and +0.09 HOTA on SportsMOT.

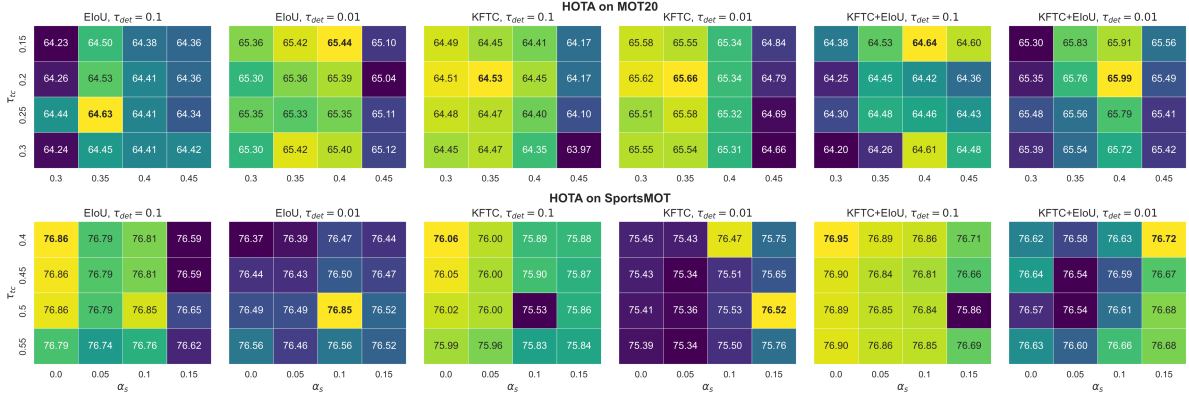


Figure 5.13: Ablation study of adding CATC and the AVC to different baseline settings. Since different baseline settings yield different optimal values of τ_{tc} and α_s , we present the results as heatmaps. For clarity, we limit the comparison to the HOTA metric.

Lowering τ_{det} from 0.1 to 0.01 improved the results on MOT20 for all baseline methods resulting in the best HOTA score of 65.99 for EIoU+KFTC setting (+3.02 HOTA). Since only 0.43% of true-positive detections in SportsMOT have confidence scores below 0.1, setting τ_{det} to 0.01 slightly degrades performance on SportsMOT. Complete HOTA, IDF1, MOTA, and IDSW results for the EIoU+KFTC setting with $\tau_{det} = 0.01$ on all evaluated datasets are provided in Appendix C.5.2.

In Table 5.12, we compare the effect of lowering τ_{det} on the MOT20 validation set under two settings. The first setting, denoted as the baseline, uses adaptive CAKF, EIoU, KFTC, and the standard association and tracklet creation criteria described in Section 5.4.3. The second setting uses CATC and AVC instead of the standard criteria.

Since both CATC and decreasing τ_{det} could, in principle, increase the number of tracked objects substantially, we monitor the number of initialized trajectories, i.e., the number of IDs, instead of IDSW.

Table 5.12: Influence of hyperparameter τ_{det} with and without the use of CATC and AVC on MOT20 validation set. Best results are shown in bold.

τ_{det}	Baseline				Baseline+CATC+AVC			
	HOTA	IDF1	MOTA	IDs	HOTA	IDF1	MOTA	IDs
0.001	53.23	67.50	53.48	2145	65.93	84.22	83.65	1694
0.005	59.51	75.09	73.69	1912	65.90	84.45	83.76	1678
0.01	61.23	77.59	78.07	1836	65.99	84.63	83.87	1679
0.025	62.37	78.71	81.47	1755	65.46	83.75	83.70	1675
0.05	63.24	80.20	82.60	1681	64.89	82.85	83.42	1687
0.1	62.97	79.86	82.58	1619	64.42	82.32	82.72	1711

Reducing τ_{det} from 0.1 to 0.05 slightly improves tracking performance in the baseline

setting. However, a further decrease of τ_{det} leads to performance degradation. The decreasing MOTA score and the increasing number of IDs indicate that the baseline method does not sufficiently distinguish between true-positive and false-positive detections. As a result, it may use unreliable low-confidence detections in association, while leaving some high-confidence detections unmatched, which can then be used to initialize new trajectories.

Adding CATC and AVC makes the method more robust to a decrease in τ_{det} . The best results are obtained for $\tau_{\text{det}} = 0.01$, yielding improvements of +2.31 IDF1 and +1.15 MOTA, which indicates improved association and better use of true-positive detections. Consequently, HOTA also improves by +1.57. With CATC and AVC, introducing low-confidence detections does not increase the number of tracked objects, i.e., IDs. On the contrary, the number of IDs decreases, which suggests that improved association and tracklet creation reduce unnecessary trajectory initialization.

Trajectory update criterion. All previously introduced components were added to the baseline that updates trajectories only after an object has been matched with a detection in three consecutive frames. In Table 5.13, we study the influence of the proposed trajectory update criterion (TUC) for different values of the hyperparameter τ_u . TUC is added to the setting that already incorporates all previously introduced components.

The first row of the table corresponds to the setting without TUC and represents the baseline update strategy.

On MOT20, the best results are obtained for $\tau_u \in [0.15, 0.30]$, while $\tau_u \in [0.35, 0.45]$ yields the best results on SportsMOT. As a compromise across datasets (including MOT17 and DanceTrack), we use $\tau_u = 0.25$. This setting improves performance on MOT20 by +0.80 HOTA, +0.49 IDF1, and +1.40 MOTA, with a slight increase of +7 IDSW. On SportsMOT, it improves performance by +0.43 HOTA, +0.05 IDF1, and +0.64 MOTA.

TUC allows confident estimates to be added to trajectories independently of the hit streak. Even if we fail to associate an object with the corresponding detection, the predicted state can still be added to the trajectory if the tracklet confidence is sufficiently high. Compared to the standard baseline, which appends estimates only after at least three consecutive matches, using TUC results in a +5.23% increase in the total length of all trajectories on MOT20 and a +2.17% increase on the SportsMOT validation set.

For example, on MOT20, +2.19% corresponds to predicted states without successful association, while +3.12% corresponds to state estimates with successful association but fewer than three consecutive matches. On the other hand, 0.08% of estimates had more than three consecutive matches but tracklet confidence that was too low to be included in the final trajectories. Together, these changes result in the total trajectory-length increase of +5.23%.

Table 5.13: Influence of the trajectory update hyperparameter τ_u on the MOT20 and SportsMOT validation sets. In the first row, the standard update criterion based on three consecutive matches is used. Best results are shown in bold.

τ_u	MOT20				SportsMOT			
	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
/	65.99	84.63	83.87	968	76.72	79.28	94.38	663
0	64.54	82.08	78.63	1001	74.41	76.86	88.60	701
0.05	66.56	84.73	84.63	887	76.72	78.93	94.06	695
0.10	66.70	84.95	85.06	882	76.91	79.10	94.48	699
0.15	66.80	85.07	85.27	900	77.02	79.20	94.73	697
0.20	66.81	85.11	85.30	934	77.09	79.28	94.90	696
0.25	66.79	85.12	85.27	975	77.15	79.33	95.02	696
0.30	66.77	85.12	85.21	974	77.18	79.37	95.09	693
0.35	66.67	85.07	85.02	1035	77.19	79.40	95.13	692
0.40	66.60	85.04	84.89	1021	77.19	79.42	95.14	692
0.45	66.52	84.99	84.73	1020	77.18	79.44	95.14	693
0.50	66.22	84.80	84.16	1286	77.17	79.44	95.13	696

Since TUC produces longer trajectories online, an important question is whether it can replace offline linear interpolation. In Table 5.14, we compare results with and without TUC, as well as the standard update criterion followed by linear interpolation as post-processing.

On MOT20, the post-processed results obtained with the standard trajectory update criterion outperform our online method by only +0.28 HOTA, +0.21 IDF1, and +0.47 MOTA, with the most notable difference in IDSW, where linear interpolation reduces IDSW by 295. However, on SportsMOT, which exhibits more dynamic motion, linear interpolation degrades HOTA, IDF1, and MOTA, while TUC outperforms both the online baseline and the post-processed baseline.

Table 5.14: Comparison between the proposed trajectory update criterion (TUC) and the standard update criterion with linear interpolation as post-processing. Results are reported on the MOT20 and SportsMOT validation sets, with the best results shown in bold.

Dataset	TUC	Lin. inter.	HOTA	IDF1	MOTA	IDSW
MOT20	X	X	65.99	84.63	83.87	968
	X	✓	67.07	85.33	85.74	680
	✓	X	66.79	85.12	85.27	975
SportsMOT	X	X	76.72	79.28	94.38	663
	X	✓	76.53	79.06	93.76	644
	✓	X	77.15	79.33	95.02	696

In Appendix C.5.3, we further analyze the influence of τ_u and TUC on MOT17 and DanceTrack.

5.5 Discussion

Our study focuses on improving tracking in multiple object scenarios, improving association by designing a robust expanded IoU that can be used without manual tuning or multi-stage association, and minimizing the effect of detector and association failures by improving tracklet management and the overall association workflow. The proposed components – CAKF, EIoU, and IAW – were designed according to four principles: adaptivity, simplicity, robustness, and complementarity. In the previous section, we presented detailed ablations and component analyses evaluating each of them.

The results across all four datasets consistently support our design philosophy. Adaptive noise estimation, uncertainty-driven expansion of the bounding boxes, soft single-stage association, and confidence-based trajectory updates all contribute to adaptive, robust tracking without complex heuristics or multi-stage pipelines. Crucially, the full CAT configuration outperforms any two-component setting on all four benchmarks (see Table 5.7 for MOT20 and SportsMOT; the same pattern holds for MOT17 and DanceTrack, as shown in Appendix C.2). This consistency across datasets with very different characteristics demonstrates the robustness and adaptivity of our tracker.

Online and real-time performance. CAT operates strictly online and does not use post-processing. Although offline interpolation can further improve results on datasets with regular motion, CAT achieves competitive results without modifying trajectories after online inference. In particular, it reaches strong performance on MOT20 and remains real-time even on the most crowded MOT20 sequences, showing that the proposed uncertainty-aware components are computationally efficient.

Hyperparameters and robustness. CAT introduces four CAT-specific hyperparameters: $s_{\max}$, α_s , τ_{tc} , and τ_u . The component analysis shows that the method is relatively robust to changes in $s_{\max}$ and τ_u , and the same values are used across all datasets. However, α_s and τ_{tc} are inherently tied to detector performance on different datasets. Dataset specific differences, such as object size, illumination and resolution, affect detection and resulting detection confidence scores. Consequently, α_s and τ_{tc} have to be tuned for each dataset. While such tuning is common practice in tracking (and even performed at the sequence level in some methods [144, 110]), we consider this a limitation of our method.

The role of appearance features. CAT achieves state-of-the-art performance among online motion-based trackers on several benchmarks and remains competitive with more complex methods. However, recent appearance-based approaches still outperform CAT

on some dynamic datasets, especially SportsMOT and DanceTrack. This indicates that appearance cues provide additional discriminative information when visually similar objects exhibit complex and irregular motion. Integrating lightweight appearance features into the same uncertainty-aware framework is therefore a promising direction for future work.

5.6 Conclusion

In this chapter, we presented CAT, a strictly online and motion-based MOT method built around adaptive uncertainty estimation and component coupling. Through adaptive CAKF, EIoU, and IAW, CAT connects motion estimation, similarity computation, association, tracklet creation, and trajectory updates within a unified real-time tracking framework.

Experimental results on MOT17, MOT20, DanceTrack, and SportsMOT show that CAT achieves strong performance among motion-based methods while avoiding appearance features, multi-stage association, detection confidence boosting, and post-processing. These results demonstrate that carefully coupled motion-based components can provide robust online tracking and represent the most complete realization of the main research direction developed throughout this dissertation.

Chapter 6

Discussion, Conclusion and Future Work

6.1 Evolution of the Proposed Methods

The three methods presented in this dissertation follow a common research direction: improving online tracking-by-detection by strengthening the interaction between detection, motion estimation, and association, while avoiding unnecessary pipeline complexity.

BoostTrack represents the first step in this direction. It improves one-stage association through lightweight confidence and similarity boosting components. The method shows that tracking information can improve the selection of true-positive detections, i.e., that successful tracking can contribute to successful detection handling. The implicit prioritization of high-confidence detections and reliable tracklets shows that coupling detection and tracking information in association can simplify the tracking pipeline by avoiding multi-stage association, while also improving performance. In addition, the lightweight additions to the similarity measure show that additional motion-based information can improve association without computationally heavy components such as ReID modules.

BoostTrack++ further develops the idea of recovering true-positive low-confidence detections. It uses richer tracklet-guided information, including soft BIoU, Mahalanobis distance similarity, and shape similarity, to decide which low-confidence detections should be retained before association. However, this improvement also relies on additional hand-designed rules and therefore moves away from the simplicity that we ultimately aim to achieve. Furthermore, the final benchmark configurations of BoostTrack and BoostTrack++ use post-processing to improve tracking results, which is useful in practice but undesirable from the perspective of strictly online tracking.

CAT takes the final step by returning to a stricter real-time online setting. It removes appearance similarity, explicit detection confidence boosting, and post-processing. It re-

lies on adaptive motion estimation, uncertainty-aware bounding-box expansion, one-stage association, and tracklet management based on estimated uncertainty, scene density, and detection reliability. In this sense, CAT represents the culmination of the dissertation: it shows that simple motion-based components can achieve robust real-time tracking when they are designed to cooperate and improve each other’s performance.

This evolution leads to a broader conclusion: tracking-by-detection components should not be treated as independent modules. The selection of true-positive observations, tracklet reliability, motion uncertainty, and association decisions influence one another throughout the pipeline. BoostTrack and BoostTrack++ exploit this interaction by using tracking information to improve association and selection of true-positive detections, while CAT extends the same principle by propagating uncertainty estimates between motion estimation, similarity computation, association validation, tracklet creation, and trajectory updates. This progression supports one of the main conclusions of the dissertation: cooperation between components can be as important as the design of each individual component.

6.2 Towards an Ideal Online MOT Algorithm

A broader view of the methods presented in this dissertation can be taken from the perspective of an ideal MOT algorithm, or at least the direction in which we believe such an algorithm should develop. In addition to accurately tracking all objects of interest without identity switches, an MOT algorithm should operate online, run in real time, adapt to different scenes, require little or no additional pretraining, and remain robust and simple. Instead of relying on manually designed multi-stage association rules and hard confidence thresholds, reliability or uncertainty estimates should be encoded directly into similarity measures, association validity criteria, and tracklet management rules. Such a method should also use a small number of hyperparameters and be robust to changes in their values.

Through the development of the methods presented in this dissertation, we gradually arrived at the conclusion that coupling and cooperation between components can bring an MOT algorithm closer to this ideal. This direction is different from relying primarily on heavier components, scenario-specific designs, complex heuristics, or increasingly complicated association rules.

BoostTrack and BoostTrack++ show that lightweight components and the coupling of tracking and detection information can result in simpler methods that outperform many trackers in their category. At the same time, they still rely on additional heuristics, appearance features in their stronger variants, post-processing in their final benchmark configurations, and several hyperparameters.

CAT represents a more significant step toward the described ideal. Adaptive CAKF and EIoU enable robustness and adaptivity while retaining real-time speed and simplicity. Detection-tracklet confidence scores, first introduced in BoostTrack, enable one-stage association, while the improved association workflow further increases adaptivity and robustness without introducing multi-stage association. Although post-processing can still improve results in some scenarios, especially when motion is regular, CAT demonstrates that longer and more reliable trajectories can also be produced online through confidence-based trajectory updates.

Taken together, these methods suggest that stronger interaction between components can be an effective alternative to increasing the complexity of individual components. In such a system, detection information can help update tracklets, tracklet information can help interpret detections, motion estimation can provide information about association reliability, and association results can influence tracking and subsequent trajectory updates. This creates a form of cooperation in which information provided by one component can compensate for the limitations of another.

At the same time, reaching the described ideal would require further reduction of dependence on detector-specific behavior, hand-designed criteria, and manually selected hyperparameters. A more general online MOT algorithm could therefore adapt reliability estimates and association criteria directly from the observed tracking behavior while retaining real-time operation and avoiding unnecessary complexity. Such an approach would preserve the main advantages of the methods presented in this dissertation while further increasing their robustness across different scenes and tracking conditions.

6.3 Limitations

Although the methods presented in this dissertation improve online tracking-by-detection in several important aspects, a number of limitations remain.

First, all three methods still depend on the quality and behavior of the object detector. BoostTrack and BoostTrack++ explicitly modify or reinterpret detection confidence scores, while CAT uses detection confidence in association validation and tracklet creation. Therefore, changes in detector calibration, object size, illumination, camera viewpoint, and false-positive statistics can affect the optimal detection-related hyperparameter values. This is most visible in CAT, where α_s and τ_{tc} are dataset-specific. Although dataset-level tuning is common in MOT, reducing this dependence remains an important direction for future work.

Second, the proposed methods are primarily designed around bounding-box-level motion cues. CAT intentionally avoids appearance features in order to preserve a strictly on-

line, motion-based, and real-time design. This makes the method simple and efficient, but it also limits performance in scenarios with highly similar objects and irregular motion, such as SportsMOT and DanceTrack. The results suggest that motion information can be pushed surprisingly far, but appearance cues still provide useful discriminative information in the most ambiguous cases.

Third, while CAT reduces the need for post-processing and multi-stage association, it still relies on hand-designed criteria for association validity, tracklet creation, and trajectory updates. These criteria are more adaptive than many multi-stage association heuristics, but they are not learned automatically from data. A more general solution would adapt these criteria online or learn them in a lightweight way while preserving real-time operation.

Finally, the evaluation is based on 2D bounding-box MOT benchmarks. Although these benchmarks cover crowded scenes, dynamic motion, and visually similar targets, they do not fully capture all real-world tracking conditions. Future work should therefore evaluate the proposed ideas in broader settings, including multi-class tracking, and possibly 3D or multi-sensor tracking scenarios.

These limitations do not undermine the main findings of the dissertation, but rather indicate directions in which the proposed ideas could be further extended. In particular, reducing detector dependence, incorporating additional sources of information when necessary, and learning or adapting the remaining hand-designed criteria could bring online MOT closer to the ideal described in the previous section.

6.4 Conclusion

This dissertation studied how online tracking-by-detection MOT can be improved by strengthening the interaction between detection, motion estimation, data association, and tracklet management. Instead of treating these components as independent modules, the main idea was to make information produced by one component useful for the others. This idea was developed through three methods: BoostTrack, BoostTrack++, and CAT.

BoostTrack showed that lightweight confidence and similarity boosting can improve one-stage association without relying on complex multi-stage pipelines. In particular, it demonstrated that detection confidence should not be interpreted independently from tracklet information, that lightweight motion cues can improve association, and that reliable detection-tracklet pairs can be prioritized implicitly in one-stage association setting.

BoostTrack++ further developed this direction by focusing on the recovery of true-positive low-confidence detections. It used richer tracklet-guided information, including soft BIoU, Mahalanobis distance similarity, and shape similarity, to decide which detec-

tions should be retained before association. This showed that tracking information can help improve detection handling, especially in crowded scenes where true-positive detections often receive low confidence scores.

CAT represents the final and most complete step in this research direction. It removes appearance similarity, explicit detection confidence boosting, and post-processing, and instead relies on adaptive uncertainty estimation and component coupling. Through adaptive Crowd-Aware Kalman Filter, Expanded IoU, and Improved Association Workflow, CAT connects motion estimation, similarity computation, association validation, tracklet creation, and trajectory updates into a strictly online real-time tracking framework.

The experimental results show that the proposed methods achieve strong performance on standard MOT benchmarks. BoostTrack and BoostTrack++ improve association and detection handling in one-stage tracking-by-detection, while CAT represents the most complete realization of the dissertation’s main direction. CAT achieves the best HOTA score among strictly online motion-based methods on MOT20, DanceTrack, and SportsMOT. Moreover, on MOT20, CAT also outperforms all online appearance-based and offline motion-based methods available at the time of writing, while preserving real-time performance and avoiding appearance features and post-processing.

Overall, the results support the central conclusion of this dissertation: robust online MOT can be achieved not only by adding stronger individual modules, but also by carefully coupling simple components so that they compensate for each other’s weaknesses.

6.5 Future Work

Several directions remain open for future research.

The first direction is reducing the dependence on dataset-specific hyperparameters. In CAT, the values of α_s and τ_{tc} depend on detector confidence statistics, which vary across datasets. Future work should investigate automatic calibration of these parameters, either through online statistics, detector uncertainty estimation, or lightweight adaptation.

The second direction is integrating appearance information without abandoning the principles developed in this dissertation. The results on SportsMOT and DanceTrack show that appearance cues are still useful in highly dynamic scenarios with visually similar targets. A promising direction is to incorporate lightweight appearance features into the same uncertainty-aware framework, so that appearance similarity is used only when motion information is insufficient or unreliable, while preserving real-time performance.

The third direction is improving robustness under camera motion. In CAT, camera motion compensation represents the main computational bottleneck of the tracking pipeline. At the same time, accurate compensation of camera motion is important for reliable mo-

tion prediction and association, especially in dynamic scenes. Future work should therefore investigate faster and more robust camera motion compensation strategies that preserve real-time execution speed while maintaining strong tracking performance. Reducing the computational cost of this component could also leave more room for incorporating lightweight appearance cues without losing real-time operation.

The fourth direction is investigating how the proposed ideas generalize to broader tracking scenarios. A particularly important direction is autonomous driving, where tracking systems must operate online and in real time while handling occlusions, crowded traffic, changing viewpoints, uncertain detections, and objects observed at different distances. In this context, component coupling, uncertainty reuse, and confidence-based tracklet management could be combined with 3D tracking and multi-sensor perception to improve robustness. The same principles could also be useful in robotic perception, multi-camera tracking, and animal tracking, where reliable identity preservation is needed under varying appearance, motion, and detection conditions.

In summary, the methods presented in this dissertation show that simple online MOT systems can be made significantly more robust when their components are designed to cooperate. Future work should continue in this direction by making this cooperation more adaptive, more general, and less dependent on manually selected parameters.

Bibliography

- [1] Momir Adžemović. “Deep learning-based multi-object tracking: a comprehensive survey from foundations to state-of-the-art.” In: *The Visual Computer* 42.9 (2026), p. 368. DOI: 10.1007/s00371-026-04607-7.
- [2] Momir Adžemović et al. “Beyond Kalman filters: deep learning-based filters for improved object tracking.” In: *Machine Vision and Applications* 36.1 (2025), p. 20. DOI: 10.1007/s00138-024-01644-x.
- [3] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. “BoT-SORT: Robust Associations Multi-Pedestrian Tracking.” In: *arXiv preprint abs/2206.14651* (2022). DOI: 10.48550/arXiv.2206.14651.
- [4] Hyochang Ahn and Han-Jin Cho. “Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system.” In: *Personal and Ubiquitous Computing* 26.2 (2022), pp. 385–394. DOI: 10.1007/s00779-019-01296-z.
- [5] Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. “Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation.” In: *2017 IEEE power & energy society general meeting*. 2017, pp. 1–5. DOI: 10.1109/PESGM.2017.8273755.
- [6] Somaieh Amraee et al. “Advancing Multi-Person Tracking for Autism Behavior Analysis: Challenges, Opportunities, and Future Directions in Clinical Settings.” In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2025, pp. 73–82. DOI: 10.1109/WACVW65960.2025.00013.
- [7] Seung-Hwan Bae and Kuk-Jin Yoon. “Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning.” In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1218–1225. DOI: 10.1109/CVPR.2014.159.

- [8] Nuri Benbarka, Jona Schröder, and Andreas Zell. “Score refinement for confidence-based 3D multi-object tracking.” In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 8083–8090. DOI: 10.1109/IROS51168.2021.9636032.
- [9] Keni Bernardin and Rainer Stiefelhagen. “Evaluating multiple object tracking performance: the clear mot metrics.” In: *EURASIP Journal on Image and Video Processing* 2008 (2008), pp. 1–10. DOI: 10.1155/2008/246309.
- [10] Alex Bewley et al. “Simple online and realtime tracking.” In: *2016 IEEE International Conference on Image Processing (ICIP)* (2016), pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.
- [11] Borna Bićanić et al. “Pedestrian Tracking by Probabilistic Data Association and Correspondence Embeddings.” In: *2019 22th International Conference on Information Fusion (FUSION)*. 2019, pp. 1–6. DOI: 10.23919/FUSION43075.2019.9011317.
- [12] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers.” In: *Proceedings of the fifth annual workshop on Computational learning theory*. Pittsburgh, Pennsylvania, USA, 1992, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401.
- [13] Nikolajs Bumanis et al. “Multi-object Tracking for Urban and Multilane Traffic: Building Blocks for Real-World Application.” In: *ICEIS (I)*. Jan. 2021, pp. 729–736. DOI: 10.5220/0010467807290736.
- [14] Hanwen Cao, George J. Pappas, and Nikolay Atanasov. “PKF: Probabilistic Data Association Kalman Filter for Multi-Object Tracking.” In: *IEEE Robotics and Automation Letters* 10.11 (2025), pp. 11506–11513. DOI: 10.1109/LRA.2025.3608646.
- [15] Jinkun Cao et al. “Observation-centric sort: Rethinking sort for robust multi-object tracking.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 9686–9696. DOI: 10.1109/CVPR52729.2023.00934.
- [16] Zhengcai Cao et al. “A multi-object tracking algorithm with center-based feature extraction and occlusion handling.” In: *IEEE Transactions on Intelligent Transportation Systems* 24.4 (2022), pp. 4464–4473. DOI: 10.1109/TITS.2022.3229978.

- [17] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. “Unifying Short and Long-Term Tracking with Graph Hierarchies.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 22877–22887. DOI: 10.1109/CVPR52729.2023.02191.
- [18] Orcun Cetintas et al. “SPAMming Labels: Efficient Annotations for the Trackers of Tomorrow.” In: (Nov. 2024), pp. 377–395. DOI: 10.1007/978-3-031-73254-6_22.
- [19] Sixian Chan et al. “Online multiple object tracking using joint detection and embedding network.” In: *Pattern Recognition* 130 (2022), p. 108793. DOI: 10.1016/j.patcog.2022.108793.
- [20] Long Chen et al. “Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification.” In: *2018 IEEE International Conference on Multimedia and Expo (ICME)* (2018), pp. 1–6. DOI: 10.48550/arXiv.1809.04427.
- [21] Hsu-Kuang Chiu et al. “Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving.” In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 14227–14233. DOI: 10.1109/ICRA48506.2021.9561754.
- [22] Wongun Choi and Silvio Savarese. “A unified framework for multi-target tracking and collective activity recognition.” In: *European conference on computer vision*. Springer. 2012, pp. 215–230. DOI: 10.1007/978-3-642-33765-9_16.
- [23] Yutao Cui et al. “Sportsmot: A large multi-object tracking dataset in multiple sports scenes.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9921–9931. DOI: 10.1109/ICCV51070.2023.00910.
- [24] Minh-Quan Dao and Vincent Frémont. “A two-stage data association approach for 3D Multi-object Tracking.” In: *Sensors* 21.9 (2021), p. 2894. DOI: 10.3390/s21092894.
- [25] Patrick Dendorfer et al. “MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking.” In: *International Journal of Computer Vision* 129.4 (Apr. 2021), pp. 845–881. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01393-0.
- [26] Patrick Dendorfer et al. “Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?” In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15657–15671. DOI: 10.52202/068431-1139.

- [27] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. “Object detection using YOLO: challenges, architectural successors, datasets and applications.” In: *Multimedia Tools and Applications* 82.6 (2023), pp. 9243–9275. DOI: 10 . 1007 / s11042-022-13644-y.
- [28] Chenjie Du et al. “Exploring the state-of-the-art in multi-object tracking: A comprehensive survey, evaluation, challenges, and future directions.” In: *Multimedia tools and applications* 83.29 (2024), pp. 73151–73189. DOI: 10 . 1007 / s11042-023-17983-2.
- [29] Yunhao Du et al. “GiaoTracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021.” In: *Proceedings of the IEEE/CVF International conference on computer vision*. 2021, pp. 2809–2819. DOI: 10 . 1109 / ICCVW54120 . 2021 . 00315.
- [30] Yunhao Du et al. “StrongSORT: Make deepSORT great again.” In: *IEEE Transactions on Multimedia* (2023), pp. 8725–8737. DOI: 10 . 1109 / TMM . 2023 . 3240881.
- [31] Mohamed Elhoseny. “Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems.” In: *Circuits, systems, and signal processing* 39.2 (2020), pp. 611–630. DOI: 10 . 1007 / s00034-019-01234-7.
- [32] Georgios D. Evangelidis and Emmanouil Z. Psarakis. “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.
- [33] Weitao Feng et al. “Towards frame rate agnostic multi-object tracking.” In: *International Journal of Computer Vision* (2023), pp. 1–20. DOI: 10 . 1007 / s11263-023-01943-2.
- [34] Ian Christian B. Fernandez et al. “Review of Kalman Filters in Multiple Object Tracking Algorithms.” In: *2024 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. 2024, pp. 1–4. DOI: 10 . 1109 / COINS61597 . 2024 . 10622143.
- [35] Abdalla Gad et al. “Multiple Object Tracking in Robotic Applications: Trends and Challenges.” In: *Applied Sciences* 12.19 (2022). ISSN: 2076-3417. DOI: 10 . 3390 / app12199408.
- [36] Yan Gao et al. “BPMTrack: Multi-Object Tracking with Detection Box Application Pattern Mining.” In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* PP (Feb. 2024), pp. 1508–1521. DOI: 10 . 1109 / TIP . 2024 . 3364828.

- [37] Yan Gao et al. “Multi-Scene Generalized Trajectory Global Graph Solver with Composite Nodes for Multiple Object Tracking.” In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.3 (2024), pp. 1842–1850. DOI: 10.1609/aaai.v38i3.27953.
- [38] Zheng Ge et al. “Yolox: Exceeding yolo series in 2021.” In: *arXiv preprint* (2021). DOI: 10.48550/arXiv.2107.08430.
- [39] Hamid Ghorbani. “Mahalanobis distance and its application for detecting multivariate outliers.” In: *Facta Universitatis, Series: Mathematics and Informatics* (2019), pp. 583–595. ISSN: 2406-047X. DOI: 10.22190/FUMI1903583G.
- [40] Yan Gong et al. “Stable at any speed: Speed-driven multi-object tracking with learnable kalman filtering.” In: *arXiv preprint* (2025). DOI: 10.48550/arXiv.2508.00358.
- [41] Matias Gran-Henriksen et al. “Deep HM-SORT: Enhancing Multi-Object Tracking in Sports with Deep Features, Harmonic Mean, and Expansion IOU.” In: *arXiv preprint arXiv:2406.12081* (2024). DOI: 10.48550/arXiv.2406.12081.
- [42] Shuman Guo et al. “A Review of Deep Learning-Based Visual Multi-Object Tracking Algorithms for Autonomous Driving.” In: *Applied Sciences* (2022). DOI: 10.3390/app122110741.
- [43] Song Guo, Rujie Liu, and Narishige Abe. “Rtat: A robust two-stage association tracker for multi-object tracking.” In: *International Conference on Pattern Recognition*. 2024, pp. 432–447. DOI: 10.1007/978-3-031-78444-6_28.
- [44] Shoudong Han et al. “Mat: Motion-aware multi-object tracking.” In: *Neurocomputing* 476 (2022), pp. 75–86. DOI: 10.1016/j.neucom.2021.12.104.
- [45] Lingxiao He et al. “FastReID: A Pytorch Toolbox for General Instance Re-identification.” In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 9664–9667. DOI: 10.1145/3581783.3613460.
- [46] Yaoqi Hu et al. “Dynamic center point learning for multiple object tracking under severe occlusions.” In: *Knowledge-Based Systems* 300 (2024), p. 112130. DOI: 10.1016/j.knosys.2024.112130.
- [47] Hsiang-Wei Huang et al. “Iterative scale-up expansioniou and deep features association for multi-object tracking in sports.” In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2024, pp. 163–172. DOI: 10.1109/WACVW60836.2024.00024.

- [48] Xi Huang and Yinwei Zhan. “Multi-object tracking with adaptive measurement noise and information fusion.” In: *Image and Vision Computing* 144 (2024), p. 104964. ISSN: 0262-8856. DOI: 10.1016/j.imavis.2024.104964.
- [49] Chao Jiang et al. “A Novel Adaptive Noise Covariance Matrix Estimation and Filtering Method: Application to Multiobject Tracking.” In: *IEEE Transactions on Intelligent Vehicles* 9.1 (2024), pp. 626–641. DOI: 10.1109/TIV.2023.3286979.
- [50] Diego M Jiménez-Bravo et al. “Multi-object tracking in traffic environments: A systematic literature review.” In: *Neurocomputing* 494 (2022), pp. 43–55. DOI: 10.1016/j.neucom.2022.04.087.
- [51] Arne Hoffhues Jonathon Luiten. *TrackEval*. <https://github.com/JonathonLuiten/TrackEval>. 2020.
- [52] Hyeonchul Jung et al. “ConfTrack: Kalman Filter-Based Multi-Person Tracking by Utilizing Confidence Score of Detection Box.” In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 6583–6592. DOI: 10.1109/WACV57701.2024.00645.
- [53] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems.” In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552.
- [54] Dheeraj Khanna et al. “Sportmamba: Adaptive non-linear multi-object tracking with state space models for team sports.” In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 6143–6153. DOI: 10.1109/CVPRW67362.2025.00611.
- [55] Seyed Alireza Khoshnevis and Abdollah Amirkhani. “Tracking with attention: A review of transformer-based object tracking.” In: *Engineering Science and Technology, an International Journal* 73 (2026), p. 102263. ISSN: 2215-0986. DOI: 10.1016/j.jestch.2025.102263.
- [56] Tarasha Khurana, Achal Dave, and Deva Ramanan. “Detecting Invisible People.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 3174–3184. DOI: 10.1109/ICCV48922.2021.00316.
- [57] Nadezda Kirillova et al. “Into the Fog: Evaluating Robustness of Multiple Object Tracking.” In: *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA, 2024.
- [58] H. W. Kuhn. “The Hungarian method for the assignment problem.” In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97. DOI: 10.1002/nav.3800020109.

- [59] Martin Larsen. et al. “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*. 2024, pp. 110–121. DOI: 10.5220/0012386600003660.
- [60] Bingyu Li et al. “Multiple object tracking under occlusions based on the stage-wise association strategy with weak cues.” In: *International Journal of Multimedia Information Retrieval* 15.1 (2026), p. 1. DOI: 10.1007/s13735-025-00392-6.
- [61] Hui Li et al. “Multiobject Tracking via Discriminative Embeddings for the Internet of Things.” In: *IEEE Internet of Things Journal* 10.12 (2023), pp. 10532–10546. DOI: 10.1109/JIOT.2023.3242739.
- [62] Jiaxin Li, Yan Ding, and Hua-Liang Wei. “SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking.” In: *Sensors (Basel, Switzerland)* 22 (2022). DOI: 10.3390/s22155863.
- [63] Hong Liang et al. “Multi-cue SORT: integrating weak cues with appearance and motion for multi-object tracking: H. Liang et al.” In: *The Journal of Supercomputing* 81.13 (2025), p. 1251. DOI: 10.1007/s11227-025-07730-9.
- [64] Chongwei Liu, Haojie Li, and Zhihui Wang. “Fasttrack: A highly efficient and generic gpu-based multi-object tracking method with parallel kalman filter.” In: *International Journal of Computer Vision* 132.5 (2024), pp. 1463–1483. DOI: 10.1007/s11263-023-01933-4.
- [65] Junting Liu et al. “Adaptive multi-object tracking based on sensors fusion with confidence updating.” In: *International Journal of Applied Earth Observation and Geoinformation* 125 (2023), p. 103577. DOI: 10.1016/j.jag.2023.103577.
- [66] Mingjie Liu et al. “Online multiple object tracking using confidence score-based appearance model learning and hierarchical data association.” In: *IET Computer Vision* 13.3 (2019), pp. 312–318. DOI: 10.1049/iet-cvi.2018.5499.
- [67] Youyu Liu et al. “A full-detection association tracker with confidence optimization for real-time multi-object tracking.” In: *Journal of Real-Time Image Processing* 21.4 (2024), pp. 1–15. DOI: 10.1007/s11554-024-01513-w.
- [68] Youyu Liu et al. “Adaptive Kalman Filter with power transformation for online multi-object tracking.” In: *Multimedia Systems* 29.3 (2023), pp. 1231–1244. DOI: 10.1007/s00530-023-01052-7.

- [69] Zelin Liu et al. “Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 35.5 (2025), pp. 4870–4882. DOI: 10.1109/TCSVT.2024.3524670.
- [70] Zhongji Liu et al. “Robust Movement-Specific Vehicle Counting at Crowded Intersections.” In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2617–2625. DOI: 10.1109/CVPRW50498.2020.00315.
- [71] Jincheng Lu et al. “Robust and Online Vehicle Counting at Crowded Intersections.” In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021, pp. 3997–4003. DOI: 10.1109/CVPRW53098.2021.00451.
- [72] Jonathon Luiten et al. “Hota: A higher order metric for evaluating multi-object tracking.” In: *International journal of computer vision* 129 (2021), pp. 548–578. ISSN: 0920-5691. DOI: 10.1007/s11263-020-01375-2.
- [73] Weiyi Lv et al. “Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 19321–19330. DOI: 10.1109/CVPR52733.2024.01828.
- [74] Hao Ma et al. “Automatic multiple-object tracking of farm livestock with improved MOTR.” In: *Expert Systems with Applications* 281 (2025), p. 127605. DOI: 10.1016/j.eswa.2025.127605.
- [75] Gerard Maggolino et al. “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.
- [76] Prasanta Chandra Mahalanobis. “On the generalized distance in statistics.” In: *Proceedings of the National Institute of Sciences (Calcutta)* 2 (1936), pp. 49–55.
- [77] Travis Mandel et al. “Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections.” In: *Pattern Recognition* 135 (2023), p. 109107. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2022.109107.
- [78] Huan Mao et al. “SCTracker: Multi-object tracking with shape and confidence constraints.” In: *IEEE Sensors Journal* (2023). DOI: 10.1109/JSEN.2023.3339212.

- [79] Rhys Martin and Ognjen Arandjelović. “Multiple-object tracking in cluttered and crowded public spaces.” In: *International Symposium on Visual Computing*. Springer. 2010, pp. 89–98. DOI: 10.1007/978-3-642-17277-9_10.
- [80] Tim Meinhardt et al. “Trackformer: Multi-object tracking with transformers.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 8844–8854. DOI: 10.1109/CVPR52688.2022.00864.
- [81] Ting Meng et al. “Localization-guided track: A deep association multiobject tracking framework based on localization confidence of camera detections.” In: *IEEE Sensors Journal* 25.3 (2024), pp. 5282–5293. DOI: 10.1109/JSEN.2024.3522021.
- [82] Weijun Meng et al. “Motion-Perception Multi-Object Tracking (MPMOT): Enhancing Multi-Object Tracking Performance via Motion-Aware Data Association and Trajectory Connection.” In: *Journal of Imaging* 11.5 (2025). ISSN: 2313-433X. DOI: 10.3390/jimaging11050144.
- [83] Ala Mhalla et al. “An embedded computer-vision system for multi-object detection in traffic surveillance.” In: *IEEE Transactions on Intelligent Transportation Systems* 20.11 (2018), pp. 4006–4018. DOI: 10.1109/TITS.2018.2876614.
- [84] Anton Milan et al. “MOT16: A Benchmark for Multi-Object Tracking.” In: *arXiv preprint* (2016). DOI: 10.48550/arXiv.1603.00831.
- [85] M Mehrdad Morsali et al. “SFSORT: Scene Features-based Simple Online Real-Time Tracker.” In: *arXiv preprint arXiv:2404.07553* (2024).
- [86] *MOT Challenge*. <https://motchallenge.net/>. Accessed: 2026-03-02.
- [87] Hemal Naik et al. “BuckTales: A multi-UAV dataset for multi-object tracking and re-identification of wild antelopes.” In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 81992–82009. DOI: 10.52202/079017-2606.
- [88] Mohammad Hossein Nasserli et al. “Online relational tracking with camera motion suppression.” In: *Journal of Visual Communication and Image Representation* 90 (2023), p. 103750. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2022.103750.
- [89] Mohammad Hossein Nasserli et al. “Online relational tracking with camera motion suppression.” In: *Journal of Visual Communication and Image Representation* 90 (2023), p. 103750. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2022.103750.
- [90] Zhixiang Ni et al. “A Multi-Object Tracking Method With Adaptive Dual Decoder and Better Motion Affinity.” In: *IEEE Access* (2024). DOI: 10.1109/ACCESS.2024.3362673.

- [91] Ziqi Pang, Zhichao Li, and Naiyan Wang. “SimpleTrack: Understanding And Re-thinking 3D Multi-Object Tracking.” In: *Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*. 2023, pp. 680–696. DOI: 10.1007/978-3-031-25056-9_43.
- [92] Jihun Park et al. “Multi-Object Tracking on SWIR Images for City Surveillance in an Edge-Computing Environment.” In: *Sensors* 23 (2023). ISSN: 1424-8220. DOI: 10.3390/s23146373.
- [93] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. “Application of deep learning for object detection.” In: *Procedia computer science* 132 (2018), pp. 1706–1717. DOI: 10.1016/j.procs.2018.05.144.
- [94] Ricardo Pereira et al. “Sort and deep-sort based multi-object tracking for mobile robotics: Evaluation with new data association metrics.” In: *Applied Sciences* 12.3 (2022), p. 1319. DOI: 10.3390/app12031319.
- [95] Zheng Qin et al. “Motiontrack: Learning robust short-term and long-term motions for multi-object tracking.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 17939–17948. DOI: 10.1109/CVPR52729.2023.01720.
- [96] Lionel Rakai et al. “Data association in multiple object tracking: A survey of recent techniques.” In: *Expert Systems with Applications* 192 (2022), p. 116300. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2021.116300.
- [97] Hao Ren et al. “Focus On Details: Online Multi-object Tracking with Diverse Fine-grained Representation.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 11289–11298. DOI: 10.1109/CVPR52729.2023.01086.
- [98] Seyed Hamid Rezatofighi et al. “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.
- [99] Ergys Ristani et al. “Performance measures and a data set for multi-target, multi-camera tracking.” In: *European conference on computer vision*. 2016, pp. 17–35. DOI: 10.1007/978-3-319-48881-3_2.
- [100] Andrew P Sage and Gary W Husa. “Adaptive filtering with unknown prior statistics.” In: *Joint Automatic Control Conference*. 7. 1969, pp. 760–769.

- [101] Ricardo Sánchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. “Online Multi-target Tracking with Strong and Weak Detections.” In: *Computer Vision – ECCV 2016 Workshops*. 2016, pp. 84–99. DOI: 10.1007/978-3-319-48881-3_7.
- [102] Atom Scott et al. “Teamtrack: A dataset for multi-sport multi-object tracking in full-pitch videos.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 3357–3366. DOI: 10.1109/CVPRW63382.2024.00340.
- [103] Jenny Seidenschwarz et al. “Simple cues lead to a strong multi-object tracker.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 13813–13823. DOI: 10.1109/CVPR52729.2023.01327.
- [104] Yuqing Shao et al. “From Detection to Association: Learning Discriminative Object Embeddings for Multi-Object Tracking.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2026, pp. 6878–6888.
- [105] Matvei Shelukhan, Timur Z. Mamedov, and Karina Kvanchiani. “StableTrack: Stabilizing Multi-Object Tracking on Low-Frequency Detections.” In: *ArXiv* (2025). DOI: 10.48550/arXiv.2511.20418.
- [106] Kyujin Shim et al. “Focusing on tracks for online multi-object tracking.” In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 11687–11696. DOI: 10.1109/CVPR52734.2025.01091.
- [107] Divya Singh, Ashish Kumar, and Roshan Singh. “Multiple Object Tracking of Autonomous Vehicles for Sustainable and Smart Cities.” In: *Object Tracking Technology: Trends, Challenges and Applications*. Springer Nature Singapore, 2023, pp. 201–219. DOI: 10.1007/978-981-99-3288-7_9.
- [108] Sayanan Sivaraman and Mohan Manubhai Trivedi. “Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis.” In: *IEEE Transactions on Intelligent Transportation Systems* 14.4 (2013), pp. 1773–1795. DOI: 10.1109/TITS.2013.2266661.
- [109] Daniel Stadler. “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.: J. Beyerer. Vol. 62. 2023, pp. 59–85. ISBN: 978-3-7315-1304-9.
- [110] Daniel Stadler and Jürgen Beyerer. “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.

- [111] Daniel Stadler and Jürgen Beyerer. “Modelling ambiguous assignments for multi-person tracking in crowds.” In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 133–142. DOI: 10.1109/wacvw54805.2022.00019.
- [112] Daniel Stadler and Jürgen Beyerer. “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.
- [113] Tomasz Stanczyk, Seongro Yoon, and Francois Bremond. “No train yet gain: Towards generic multi-object tracking in sports and beyond.” In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 6039–6048. DOI: 10.1109/CVPRW67362.2025.00601.
- [114] Vukasin D. Stanojevic and Branimir T. Todorovic. “BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking.” In: *Machine Vision and Applications* 35.3 (2024), p. 53. ISSN: 1432-1769. DOI: 10.1007/s00138-024-01531-5.
- [115] Vukašin Stanojević and Branimir Todorović. “BoostTrack++: using tracklet information to detect more objects in multiple object tracking.” In: *Filomat* 39.16 (2025), pp. 5685–5702. DOI: 10.2298/FIL2516685S.
- [116] Sanbao Su et al. “Collaborative multi-object tracking with conformal uncertainty propagation.” In: *IEEE Robotics and Automation Letters* 9.4 (2024), pp. 3323–3330. DOI: 10.1109/LRA.2024.3364450.
- [117] Jiacheng Sun et al. “GTA: Global Tracklet Association for Multi-Object Tracking in Sports.” In: *Computer Vision – ACCV 2024 Workshops: 17th Asian Conference on Computer Vision, Hanoi, Vietnam, December 8–12, 2024, Revised Selected Papers, Part II*. 2024, pp. 74–87. DOI: 10.1007/978-981-96-2644-1_6.
- [118] Peize Sun et al. “Dancetrack: Multi-object tracking in uniform appearance and diverse motion.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 20993–21002. DOI: 10.1109/CVPR52688.2022.02032.
- [119] Chi-Yi Tsai, Guan-Yu Shen, and Humaira Nisar. “Swin-jde: Joint detection and embedding multi-object tracking in crowded scenes based on swin-transformer.” In: *Engineering Applications of Artificial Intelligence* 119 (2023), p. 105770. DOI: 10.1016/j.engappai.2022.105770.

- [120] Tsung-Han Tsai and Ching-Chin Yang. “A real-time surveillance system with multi-object tracking.” In: *Multidimensional Systems and Signal Processing* 34.4 (2023), pp. 767–791. DOI: 10.1007/s11045-023-00883-x.
- [121] Lorenzo Vaquero et al. “Lost and found: Overcoming detector failures in on-line multi-object tracking.” In: *European Conference on Computer Vision*. 2024, pp. 448–466. DOI: 10.1007/978-3-031-73464-9_27.
- [122] Ajantha Vijayakumar and Subramaniaswamy Vairavasundaram. “Yolo-based object detection models: A review and its applications.” In: *Multimedia Tools and Applications* 83.35 (2024), pp. 83535–83574. DOI: 10.1007/s11042-024-18872-y.
- [123] Gaoang Wang, Mingli Song, and Jenq-Neng Hwang. “Recent advances in embedding methods for multi-object tracking: a survey.” In: *arXiv preprint arXiv:2205.10766* (2022). DOI: 10.48550/arXiv.2205.10766.
- [124] Gaoang Wang et al. “Split and connect: A universal tracklet booster for multi-object tracking.” In: *IEEE Transactions on Multimedia* 25 (2022), pp. 1256–1268. DOI: 10.1109/TMM.2022.3140919.
- [125] Yu-Hsiang Wang et al. “SMILEtrack: similarity learning for occlusion-aware multiple object tracking.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 6. 2024, pp. 5740–5748. DOI: 10.1609/aaai.v38i6.28386.
- [126] Shuai Wang et al. “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.
- [127] Shuai Wang et al. “Tracking game: Self-adaptive agent based multi-object tracking.” In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 1964–1972. DOI: 10.1145/3503161.3548231.
- [128] Xinzhe Wang et al. “Multi-Object Tracking in the Dark.” In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 382–392. DOI: 10.1109/CVPR52733.2024.00044.
- [129] Zhongdao Wang et al. “Towards real-time multi-object tracking.” In: *European Conference on Computer Vision*. 2020, pp. 107–122. DOI: 10.1007/978-3-030-58621-8_7.
- [130] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. “Simple Online and Realtime Tracking with a Deep Association Metric.” In: *2017 IEEE International Conference on Image Processing (ICIP)*. Beijing, China, 2017, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.

- [131] Yu Xiang, Alexandre Alahi, and Silvio Savarese. “Learning to Track: Online Multi-object Tracking by Decision Making.” In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4705–4713. DOI: 10.1109/ICCV.2015.534.
- [132] Zhen Xu et al. “Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform.” In: *Patterns* 3.7 (2022), p. 100543. ISSN: 2666-3899. DOI: 10.1016/j.patter.2022.100543.
- [133] F. Yang et al. “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)* (2022), pp. 4788–4797. DOI: 10.1109/wacv56688.2023.00478.
- [134] Fan Yang et al. “ReMOT: A model-agnostic refinement for multiple object tracking.” In: *Image and Vision Computing* 106 (2021), p. 104091. DOI: 10.1016/j.imavis.2020.104091.
- [135] Mingzhan Yang et al. “Hybrid-sort: Weak cues matter for online multi-object tracking.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 2024, pp. 6504–6512. DOI: 10.1609/aaai.v38i7.28471.
- [136] Anqi Yi and Nantheera Anantrasirichai. “A Comprehensive Study of Object Tracking in Low-Light Environments.” In: *Sensors* 24.13 (2024). DOI: 10.3390/s24134359.
- [137] Kefu Yi et al. “Ucmctrack: Multi-object tracking with uniform camera motion compensation.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 2024, pp. 6702–6710. DOI: 10.1609/aaai.v38i7.28493.
- [138] Sisi You et al. “UTM: A Unified Multiple Object Tracking Model With Identity-Aware Feature Enhancement.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 21876–21886. DOI: 10.1109/cvpr52729.2023.02095.
- [139] En Yu, Zhuoling Li, and Shoudong Han. “Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 8834–8843. DOI: 10.1109/CVPR52688.2022.00863.
- [140] Fengwei Yu et al. “POI: Multiple Object Tracking with High Performance Detection and Appearance Feature.” In: *Computer Vision – ECCV 2016 Workshops*. 2016, pp. 36–42. DOI: 10.1007/978-3-319-48881-3_3.

- [141] Fangao Zeng et al. “Motr: End-to-end multiple-object tracking with transformer.” In: *European Conference on Computer Vision*. 2022, pp. 659–675. DOI: 10 . 1007/978-3-031-19812-0_38.
- [142] Kai Zeng et al. “NCT: noise-control multi-object tracking.” In: *Complex & Intelligent Systems* 9 (2023), pp. 4331–4347. DOI: 10 . 1007/s40747-022-00946-9.
- [143] Libo Zhang et al. “Animaltrack: A benchmark for multi-animal tracking in the wild.” In: *International Journal of Computer Vision* 131.2 (2023), pp. 496–513. DOI: 10 . 1007/s11263-022-01711-8.
- [144] Yifu Zhang et al. “ByteTrack: Multi-Object Tracking By Associating Every Detection Box.” In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*. Tel Aviv, Israel, 2022, pp. 1–21. ISBN: 978-3-031-20046-5. DOI: 10 . 1007/978-3-031-20047-2_1.
- [145] Yifu Zhang et al. “FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking.” In: *International Journal of Computer Vision* 129 (2020), pp. 3069–3087. DOI: 10 . 1007/s11263-021-01513-4.
- [146] Yu Zhang et al. “Handling heavy occlusion in dense crowd tracking by focusing on the heads.” In: *Australasian Joint Conference on Artificial Intelligence*. Springer. 2023, pp. 79–90. DOI: 10 . 1007/978-981-99-8388-9_7.
- [147] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. “MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 22056–22065. DOI: 10 . 1109/CVPR52729 . 2023 . 02112.
- [148] Zeyong Zhao, Jingyi Wu, and Ruicong Zhi. “WDTtrack: tracking multiple objects with indistinguishable appearance and irregular motion.” In: *Applied Intelligence* (2024), pp. 1–21. DOI: 10 . 1007/s10489-024-05682-w.
- [149] Zhong-Qiu Zhao et al. “Object detection with deep learning: A review.” In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232. DOI: 10 . 1109/TNNLS . 2018 . 2876865.
- [150] Tao Zhou et al. “Apptracker: Improving tracking multiple objects in low-frame-rate videos.” In: *Proceedings of the 30th ACM international conference on multimedia*. 2022, pp. 6664–6674. DOI: 10 . 1145/3503161 . 3548162.
- [151] Tao Zhou et al. “APPTTracker+: displacement uncertainty for occlusion handling in low-frame-rate multiple object tracking.” In: *International Journal of Computer Vision* 133.4 (2025), pp. 2044–2069. DOI: 10 . 1007/s11263-024-02237-x.

- [152] Xiaolong Zhou et al. “Multi-object tracking based on attention networks for Smart City system.” In: *Sustainable Energy Technologies and Assessments* 52 (2022), p. 102216. ISSN: 2213-1388. DOI: 10.1016/j.seta.2022.102216.
- [153] Tianyu Zhu et al. “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.

Appendix A

Additional Results for BoostTrack

This appendix provides additional implementation details and parameter analyses related to the BoostTrack method presented in Chapter 3. In particular, we describe the computation of the Mahalanobis distance similarity matrix and report validation experiments used to select several hyperparameters.

A.1 Calculating Mahalanobis Distance Similarity

In Section 3.3.2, we described the Mahalanobis distance similarity used in BoostTrack. The corresponding procedure for calculating the Mahalanobis distance similarity matrix is given in Algorithm 4.

A.2 NMS Threshold

As noted in Section 3.4.2, BoostTrack applies non-maximum suppression to detected bounding boxes for which Eq. (3.10) holds. We tested different values of the NMS threshold τ_{NMS} on the MOT17 validation set, and the results are shown in Table A.1. The first row corresponds to the baseline without the DUO component, while the remaining rows show the results obtained by applying DUO with different values of τ_{NMS} .

The results show that $\tau_{\text{NMS}} = 0.3$ provides a good trade-off between tracking accuracy, the number of produced identities, and identity switches. Therefore, we use $\tau_{\text{NMS}} = 0.3$ in the BoostTrack experiments in Section 3.5. We also observe that all tested values of τ_{NMS} improve MOTA compared to the baseline, which indicates that the DUO component helps retain additional true-positive detections.

Algorithm 4 Calculating the Mahalanobis distance similarity matrix

Inputs: Detections $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$; tracklets $\hat{T} = \{\hat{T}_1, \hat{T}_2, \dots, \hat{T}_m\}$; maximum distance $d_{\max}$.

Output: Mahalanobis distance similarity matrix $\mathbf{S}_M \in \mathbb{R}^{n \times m}$.

```
1:  $\mathbf{M}_{\text{raw}} \leftarrow \text{compute\_mahalanobis\_dist}(\mathcal{D}, \hat{T})$ 
2:  $\mathbf{G} \leftarrow \mathbf{0}_{n \times m}$  ▷ Gating mask
3: for  $i = 1$  to  $n$  do
4:   for  $j = 1$  to  $m$  do
5:     if  $\mathbf{M}_{\text{raw}}[i, j] > d_{\max}$  then
6:        $\mathbf{G}[i, j] \leftarrow 1$ 
7:        $\mathbf{M}_{\text{raw}}[i, j] \leftarrow d_{\max}$ 
8:     end if
9:   end for
10: end for
11:  $\mathbf{M}_{\text{inv}} \leftarrow d_{\max} \mathbf{1}_{n \times m} - \mathbf{M}_{\text{raw}}$ 
12:  $\mathbf{S}_M \leftarrow \mathbf{0}_{n \times m}$ 
13: for  $j = 1$  to  $m$  do
14:    $\mathbf{S}_M[:, j] \leftarrow \text{softmax}(\mathbf{M}_{\text{inv}}[:, j])$ 
15: end for
16: for  $i = 1$  to  $n$  do
17:   for  $j = 1$  to  $m$  do
18:     if  $\mathbf{G}[i, j] = 1$  then
19:        $\mathbf{S}_M[i, j] \leftarrow 0$ 
20:     end if
21:   end for
22: end for
23: return  $\mathbf{S}_M$ 
```

Table A.1: Effect of τ_{NMS} on the MOT17 validation set.

τ_{NMS}	HOTA	IDF1	MOTA	IDs	IDSW
–	66.13	77.30	74.74	440	227
0.0	66.03	77.00	74.80	441	226
0.1	66.03	77.00	74.80	441	226
0.2	66.34	77.51	74.79	443	225
0.3	66.34	77.58	74.82	443	223
0.4	66.38	77.58	74.81	448	225
0.5	66.38	77.58	74.83	448	225
0.6	66.34	77.49	74.81	453	229
0.7	66.06	77.10	74.78	449	232

A.3 Influence of Detection-Tracklet Confidence Hyperparameters

We select the tracklet confidence decay parameter β and the initial score s_{init} based on experiments on the MOT17 validation set. We test $\beta \in [0.7, 0.95]$ and $s_{\text{init}} \in \{0, 1, \dots, 10\}$. Table A.2 shows the results for $s_{\text{init}} \in \{0, 7\}$.

The results are reported without detection confidence boosting, camera motion compensation, and interpolation, using $\lambda_{\text{IoU}} = 0.5$, $\lambda_{\text{M}} = 0$, and $\lambda_{\text{shape}} = 0.25$. We set $\lambda_{\text{M}} = 0$ because the Mahalanobis distance similarity is not affected by β or s_{init} . The first row corresponds to the baseline without detection-tracklet confidence.

Table A.2: Effect of β and s_{init} on the MOT17 validation set.

β	$s_{\text{init}} = 0$			$s_{\text{init}} = 7$		
	HOTA	IDF1	MOTA	HOTA	IDF1	MOTA
0	65.96	77.05	74.75	65.96	77.05	74.75
0.70	66.30	78.40	75.02	66.38	78.57	75.06
0.75	66.41	78.47	75.03	66.39	78.53	75.06
0.80	66.47	78.48	75.03	66.44	78.56	75.05
0.85	66.44	78.29	75.03	66.52	78.50	75.06
0.90	66.38	78.16	74.96	66.53	78.56	74.99
0.95	66.48	78.24	74.90	66.61	78.56	74.92

Appendix B

Additional Results for BoostTrack++

This appendix provides additional ablation results for the BoostTrack++ method presented in Chapter 4. The results complement the main experimental analysis by evaluating the proposed components with additional baseline trackers.

B.1 Influence of Different Similarity Measures

In Table B.1, we show the results of applying different similarity measures instead of IoU for DLO confidence boosting on MOT20 validation set. BoostTrack is used as the baseline method. As in the experiments with SORT and BoostTrack+ baselines in Table 4.2, using the average of all three similarity measures provides the best trade-off between introducing additional identities and improving overall tracking performance.

Table B.1: Effect of different similarity measures used for DLO confidence boosting on the MOT20 validation set. BoostTrack is used as the baseline. Best values among DLO-based variants for the main tracking metrics and IDSW are shown in bold.

Setting				BoostTrack baseline				
DLO	SBIoU	S_M	S_{shape}	HOTA	IDF1	MOTA	IDSW	IDs
$\times$	$\times$	$\times$	$\times$	61.39	77.23	77.02	803	1867
$\checkmark$	$\times$	$\times$	$\times$	61.74	77.45	77.46	898	2008
$\checkmark$	$\times$	$\times$	$\checkmark$	59.64	73.85	75.34	1252	2637
$\checkmark$	$\times$	$\checkmark$	$\times$	61.70	77.52	77.38	856	1911
$\checkmark$	$\times$	$\checkmark$	$\checkmark$	61.69	77.57	77.52	862	1911
$\checkmark$	$\checkmark$	$\times$	$\times$	61.74	77.58	77.65	865	1949
$\checkmark$	$\checkmark$	$\times$	$\checkmark$	61.17	76.63	77.50	915	2026
$\checkmark$	$\checkmark$	$\checkmark$	$\times$	61.80	77.70	77.60	834	1917
$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	61.88	77.89	77.57	825	1902

B.2 Ablation Study on SORT and BoostTrack

To show the robustness of the proposed components, we perform additional ablation experiments using SORT and BoostTrack as baseline methods. Table B.2 shows the results on the MOT20 validation set. The meaning of the setting columns follows Section 4.4.2 and Algorithm 2. The TS column denotes the KF-based trajectory splitting component, which is applied as post-processing before interpolation-based refinement.

The results show that individual components may lead to different trade-offs. For example, using only VT with BoostTrack reduces IDSW, but provides only a limited improvement in the main tracking metrics. In contrast, settings based on the proposed similarity measure, especially S and S+SB+VT, provide a better overall balance between tracking accuracy and identity preservation. The final row additionally includes TS and corresponds to the full BoostTrack++ configuration with trajectory split before post-processing.

Table B.2: Ablation study on the MOT20 validation set for different additional components added to the SORT and BoostTrack baselines. Best values for the main tracking metrics and IDSW are shown in bold.

Setting						SORT baseline					BoostTrack baseline				
DLO	S	SB	VT	TS		HOTA	IDF1	MOTA	IDSW	IDs	HOTA	IDF1	MOTA	IDSW	IDs
X	X	X	X	X		56.65	73.60	69.91	1127	1894	61.39	77.23	77.02	803	1867
✓	X	X	X	X		58.58	75.09	73.28	1259	2045	61.74	77.45	77.46	898	2008
✓	X	X	✓	X		56.62	73.60	70.13	1130	1901	61.53	77.46	77.20	801	1865
✓	X	✓	X	X		58.07	74.55	73.06	1315	2059	61.29	76.88	77.61	902	2052
✓	X	✓	✓	X		58.03	75.00	73.06	1325	2060	61.25	76.84	77.59	904	2048
✓	✓	X	X	X		57.95	75.20	71.38	1162	1929	61.88	77.89	77.57	825	1902
✓	✓	X	✓	X		56.90	73.97	70.41	1148	1891	61.54	77.48	77.23	819	1862
✓	✓	✓	X	X		57.61	74.53	71.65	1192	1933	61.57	77.43	77.67	866	1901
✓	✓	✓	✓	X		57.65	74.60	71.64	1200	1935	61.56	77.38	77.71	864	1895
✓	✓	✓	✓	✓		61.2	76.89	77.58	789	1748	61.62	77.4	77.65	782	1734

Appendix C

Additional Results for CAT

This appendix provides supplementary details and results for the CAT method presented in Chapter 5. The main chapter focuses on the most representative analyses on MOT20 and SportsMOT, since these datasets emphasize crowded scenes and dynamic motion, respectively. Here, we provide additional implementation details and extend analyses from the main chapter to MOT17 and DanceTrack for completeness.

In particular, we first describe the efficient Kalman filter computation used in CAT. We then report additional ablation studies and component analyses on MOT17 and DanceTrack, following the same structure as in the main chapter. These results support the conclusions drawn in Chapter 5.

C.1 Efficient Kalman Filter Computation

We assume diagonal noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, i.e., independent noise across state variables. The covariance matrices $\mathbf{P}$ and $\mathbf{Q}$ can be written in block form as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xv} \\ \mathbf{P}_{vx} & \mathbf{P}_{vv} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{xx} & \mathbf{0}_{4 \times 4} \\ \mathbf{0}_{4 \times 4} & \mathbf{Q}_{vv} \end{bmatrix}. \quad (\text{C.1})$$

Here, $\mathbf{P}_{xx}$ denotes the covariance matrix associated with position and size, $\mathbf{P}_{vv}$ corresponds to the velocity covariance, while $\mathbf{P}_{xv}$ and $\mathbf{P}_{vx}$ represent the cross-covariance matrices between position, size, and velocity. Throughout this section, we use $\mathbf{P}_{vx} = \mathbf{P}_{xv}^\top$.

If $\mathbf{P}_{0|0}$ is initialized as a diagonal matrix (see Section 5.4.2) and

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{0}_{4 \times 4} & \mathbf{I}_4 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{I}_4 & \mathbf{0}_{4 \times 4} \end{bmatrix}$$

are used (see Section 2.3), the matrices $\mathbf{P}_{xx}$, $\mathbf{P}_{xv}$, and $\mathbf{P}_{vv}$ remain diagonal, with $\mathbf{P}_{vx} = \mathbf{P}_{xv}$.

Therefore, the Kalman filter equations can be written in a more computationally efficient form.

For a diagonal matrix $\mathbf{D}$, we use the corresponding lowercase symbol to denote the vector of elements on its diagonal, i.e., $\mathbf{D} = \text{diag}(\mathbf{d})$. We use the notation $\mathbf{v}_{[i:j]}$ to represent the vector consisting of the entries $v_i, v_{i+1}, \dots, v_{j-1}, v_j$. To reduce clutter in subscripts, the time step is denoted using superscripts in the following equations.

Under the stated assumptions, the Kalman prediction equations can be rewritten as

$$\begin{aligned}
\hat{\mathbf{x}}_{[1:4]}^{t|t-1} &= \hat{\mathbf{x}}_{[1:4]}^{t-1|t-1} + \hat{\mathbf{x}}_{[5:8]}^{t-1|t-1}, \\
\hat{\mathbf{x}}_{[5:8]}^{t|t-1} &= \hat{\mathbf{x}}_{[5:8]}^{t-1|t-1}, \\
\mathbf{p}_{xx}^{t|t-1} &= \mathbf{p}_{xx}^{t-1|t-1} + 2\mathbf{p}_{xv}^{t-1|t-1} + \mathbf{p}_{vv}^{t-1|t-1} + \mathbf{q}_{xx}^{t-1}, \\
\mathbf{p}_{xv}^{t|t-1} &= \mathbf{p}_{xv}^{t-1|t-1} + \mathbf{p}_{vv}^{t-1|t-1}, \\
\mathbf{p}_{vv}^{t|t-1} &= \mathbf{p}_{vv}^{t-1|t-1} + \mathbf{q}_{vv}^{t-1}.
\end{aligned} \tag{C.2}$$

More importantly, under the diagonal noise assumption and diagonal initial state covariance, the Kalman update equations can also be rewritten in vector form. The Kalman gain is calculated as

$$\begin{aligned}
\mathbf{K}^t &= \begin{bmatrix} \mathbf{P}_{xx}^{t|t-1} \left(\mathbf{P}_{xx}^{t|t-1} + \mathbf{R}^t \right)^{-1} \\ \mathbf{P}_{vx}^{t|t-1} \left(\mathbf{P}_{xx}^{t|t-1} + \mathbf{R}^t \right)^{-1} \end{bmatrix} \\
&= \begin{bmatrix} \text{diag} \left(\mathbf{p}_{xx}^{t|t-1} \odot \mathbf{s}^{-1} \right) \\ \text{diag} \left(\mathbf{p}_{xv}^{t|t-1} \odot \mathbf{s}^{-1} \right) \end{bmatrix},
\end{aligned} \tag{C.3}$$

where $\mathbf{s} = \mathbf{p}_{xx}^{t|t-1} + \mathbf{r}^t$, and $\mathbf{s}^{-1}$ denotes the element-wise reciprocal of $\mathbf{s}$.

The Kalman update equations can then be written as

$$\begin{aligned}
\mathbf{d}^t &= \mathbf{z}^t - \hat{\mathbf{x}}_{[1:4]}^{t|t-1}, \\
\hat{\mathbf{x}}_{[1:4]}^{t|t} &= \hat{\mathbf{x}}_{[1:4]}^{t|t-1} + \mathbf{p}_{xx}^{t|t-1} \odot \mathbf{s}^{-1} \odot \mathbf{d}^t, \\
\hat{\mathbf{x}}_{[5:8]}^{t|t} &= \hat{\mathbf{x}}_{[5:8]}^{t|t-1} + \mathbf{p}_{xv}^{t|t-1} \odot \mathbf{s}^{-1} \odot \mathbf{d}^t, \\
\mathbf{p}_{xx}^{t|t} &= \mathbf{p}_{xx}^{t|t-1} - \mathbf{s}^{-1} \odot \left(\mathbf{p}_{xx}^{t|t-1} \right)^2, \\
\mathbf{p}_{xv}^{t|t} &= \mathbf{p}_{xv}^{t|t-1} - \mathbf{s}^{-1} \odot \mathbf{p}_{xx}^{t|t-1} \odot \mathbf{p}_{xv}^{t|t-1}, \\
\mathbf{p}_{vv}^{t|t} &= \mathbf{p}_{vv}^{t|t-1} - \mathbf{s}^{-1} \odot \left(\mathbf{p}_{xv}^{t|t-1} \right)^2.
\end{aligned} \tag{C.4}$$

Finally, the noise covariance matrices are estimated as

$$\begin{aligned}
\boldsymbol{\varepsilon}^t &= \mathbf{z}^t - \hat{\mathbf{x}}_{[1:4]}^{t|t}, \\
\mathbf{r}^{t+1} &= \alpha \mathbf{r}^t + (1 - \alpha) \left((\boldsymbol{\varepsilon}^t)^2 + \mathbf{p}_{xx}^{t|t-1} \right), \\
\mathbf{q}_{xx}^t &= \alpha \mathbf{q}_{xx}^{t-1} + (1 - \alpha) \left(\mathbf{p}_{xx}^{t|t-1} \odot \mathbf{s}^{-1} \odot \mathbf{d}^t \right)^2, \\
\mathbf{q}_{vv}^t &= \alpha \mathbf{q}_{vv}^{t-1} + (1 - \alpha) \left(\mathbf{p}_{vv}^{t|t-1} \odot \mathbf{s}^{-1} \odot \mathbf{d}^t \right)^2.
\end{aligned} \tag{C.5}$$

Here, all squares and products in vector expressions are computed element-wise.

With the diagonal noise assumption and by directly using the structure of $\mathbf{F}$ and $\mathbf{H}$, the Kalman prediction Eq. (C.2), Kalman update Eq. (C.4), and noise estimation Eq. (C.5) can be computed with $\mathcal{O}(n)$ operations per tracked object instead of the usual $\mathcal{O}(n^3)$ cost of general matrix operations.

C.2 Ablation Study on MOT17 and DanceTrack

In Table C.1, we present ablation results on the MOT17 and DanceTrack validation sets, using the same settings and notation as in Table 5.7 (see Section 5.4.5). The results are consistent with the ablation results on MOT20 and SportsMOT discussed in the main chapter.

Adding EIoU without the other proposed components degrades tracking performance on both datasets. Among the two-component settings, the best results are obtained by combining Adapt. CAKF and IAW. Adding EIoU to this setting results in further improvement: +0.63 HOTA, +1.30 IDF1, +0.29 MOTA, and -67 IDSW on MOT17, and +1.51 HOTA, +3.06 IDF1, +0.36 MOTA, and -2 IDSW on DanceTrack.

Compared to the baseline, the full CAT configuration improves the results by +7.25 HOTA, +11.42 IDF1, +8.70 MOTA, and -279 IDSW on the MOT17 validation set. On DanceTrack, the corresponding improvements are +10.78 HOTA, +16.69 IDF1, +2.49 MOTA, and -1918 IDSW.

C.3 Additional adaptive CAKF Experiments

C.3.1 Additional Ablation Results

We further evaluate the proposed adaptive CAKF on the MOT17 and DanceTrack validation sets. The ablation results are summarized in Table C.2.

Crowd-aware scaling provides a small improvement over the baseline on DanceTrack, while on MOT17 it slightly improves MOTA and IDSW but slightly decreases HOTA

Table C.1: Ablation study of CAT components on the MOT17 and DanceTrack validation sets. Best results are shown in bold.

Setting			MOT17				DanceTrack			
Adapt.	CAKF	Elou IAW	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
$\times$	$\times$	$\times$	62.70	72.32	69.84	456	48.41	45.20	87.08	3164
$\times$	$\times$	$\checkmark$	67.42	78.28	78.24	243	54.93	54.57	88.24	1333
$\times$	$\checkmark$	$\times$	60.80	69.30	68.14	529	47.33	44.00	86.56	3360
$\times$	$\checkmark$	$\checkmark$	67.34	78.64	78.04	234	54.16	53.83	88.52	1444
$\checkmark$	$\times$	$\times$	65.73	77.19	70.47	305	53.81	53.54	87.22	2463
$\checkmark$	$\times$	$\checkmark$	69.32	82.44	78.25	244	57.68	58.83	89.21	1248
$\checkmark$	$\checkmark$	$\times$	64.20	74.56	68.79	355	52.99	53.39	86.24	2628
$\checkmark$	$\checkmark$	$\checkmark$	69.95	83.74	78.54	177	59.19	61.89	89.57	1246

Table C.2: Ablation study of adaptive covariance estimation and crowd-aware scaling on the MOT17 and DanceTrack validation sets. Best results are shown in bold.

Dataset	Adapt. KF	CAKF	HOTA	IDF1	MOTA	IDSW
MOT17	$\times$	$\times$	62.70	72.32	69.84	456
	$\times$	$\checkmark$	62.67	72.08	70.03	440
	$\checkmark$	$\times$	61.13	72.48	67.84	369
	$\checkmark$	$\checkmark$	65.73	77.19	70.47	305
DanceTrack	$\times$	$\times$	48.41	45.20	87.08	3164
	$\times$	$\checkmark$	49.10	45.34	87.20	3112
	$\checkmark$	$\times$	36.69	35.73	81.60	3929
	$\checkmark$	$\checkmark$	53.81	53.54	87.22	2463

and IDF1. When used without crowd-aware scaling, adaptive covariance estimation is unstable: it substantially degrades performance on DanceTrack and, as discussed in the main chapter, also on SportsMOT. On MOT17, it improves IDF1 and IDSW but decreases HOTA and MOTA.

When adaptive covariance estimation and crowd-aware scaling are combined, the resulting adaptive CAKF achieves the best results across all metrics on both MOT17 and DanceTrack. These results support the conclusion from the main chapter that adaptive covariance estimation and crowd-aware scaling are complementary: adaptive estimation alone may be unreliable, but crowd-aware scaling stabilizes it and improves overall tracking performance.

C.3.2 Estimated Covariance Matrices

To assess the accuracy of the estimated process noise covariance $\mathbf{Q}$, measurement noise covariance $\mathbf{R}$, and state covariance $\mathbf{P}$, we compute offline reference estimates for each object.

Since the true object states are not directly observed, the reference estimates are computed from sequences of detections. In an ideal setting, the tracker should receive, for each object, only observations that correspond to that object, without false positives or detections belonging to other objects. To approximate such an observation sequence, we match detected bounding boxes to the ground-truth annotations and keep only the matched detections for each object. These detection sequences, rather than the ground-truth annotations themselves, are then used to estimate $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{P}$ using the expectation–maximization algorithm. Using matched detections instead of ground-truth annotations preserves the measurement noise introduced by the detector, while removing association errors from the analysis.

If the ground-truth states were used directly, $\mathbf{Q}$ and $\mathbf{R}$ could instead be estimated from the corresponding state-transition and measurement residuals using direct maximum-likelihood estimates. We deliberately do not follow this procedure because the objective is to estimate the noise characteristics from detector observations rather than from ground-truth states, while using ground truth only to eliminate association errors.

In Figure 5.9 we show a visual comparison of the distributions of the diagonal entries of the estimated covariance matrices of all tracked objects in the MOT20 validation set. The blue histograms represent the offline reference estimates. For comparison, we also show the corresponding distributions obtained from the baseline method, shown in orange, and from adaptive CAKF, shown in green. For adaptive CAKF, $\mathbf{Q} = \tilde{\mathbf{Q}}$ and $\mathbf{R} = \tilde{\mathbf{R}}$. In both the baseline and adaptive settings, mean values are computed using state estimates with at least three consecutive successful matches and for the trajectories with more than 10 frames.

Overall, the covariance estimates produced by adaptive CAKF are closer to the reference values than those produced by the baseline. For several components, such as Q_u , Q_v , $P_{\dot{u}}$, $P_{\dot{v}}$, and P_h , the distributions match the reference estimates almost perfectly.

We observe a systematic shift in the estimated $\mathbf{R}$ variances, where adaptive CAKF produces slightly larger values than the reference estimates. This effect is not caused by crowd-aware scaling, since the average scaling factor s^r on MOT20 is 0.58, i.e., scaling actually reduces the estimated variances. Instead, the mismatch can be attributed to imperfections in tracking and association, which lead to larger residuals and therefore larger estimated measurement noise variances.

To quantitatively assess distribution similarity, we compute the Jensen–Shannon (JS)

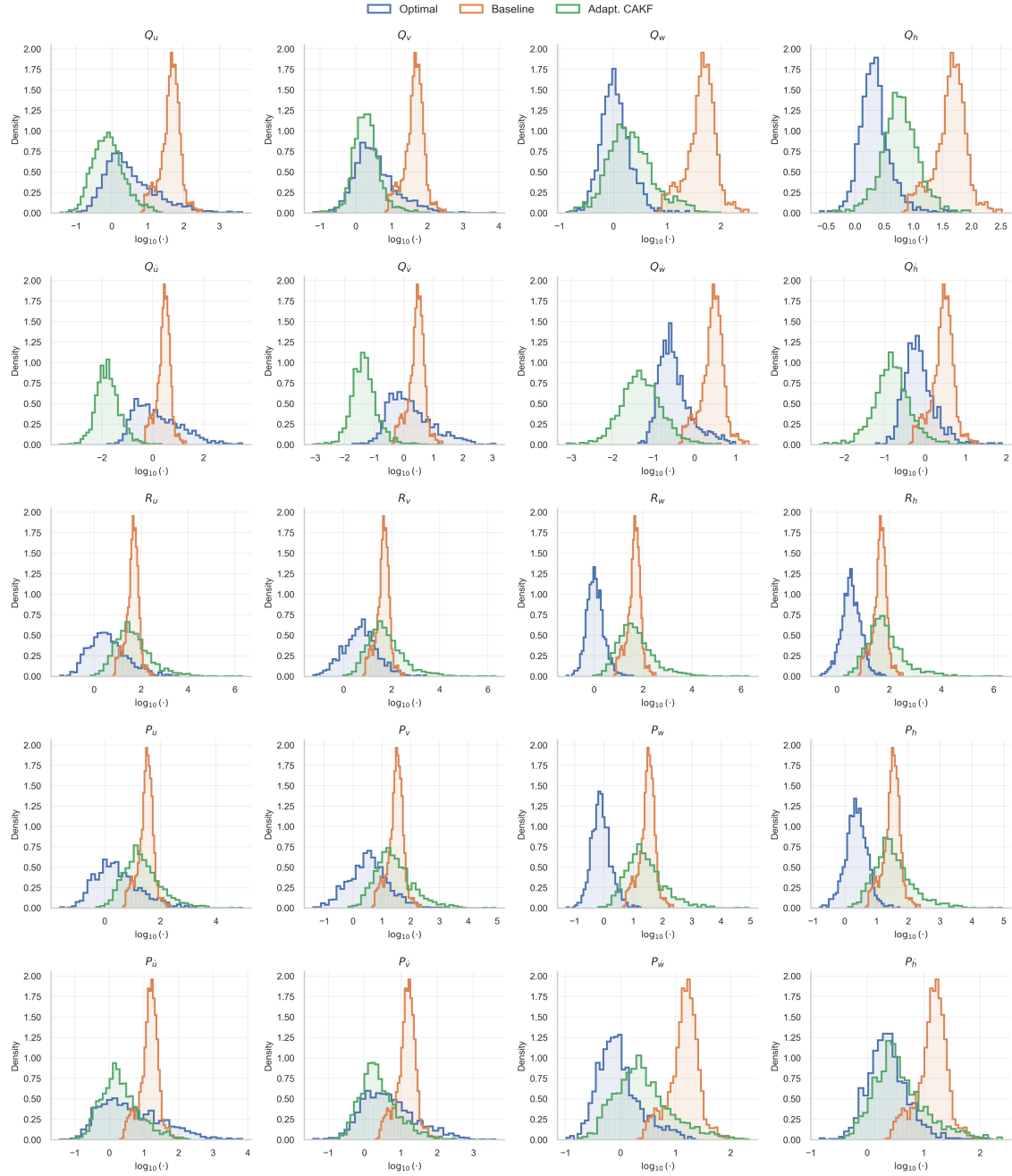


Figure C.1: Histograms of diagonal elements of the process noise covariance $\mathbf{Q}$, measurement noise covariance $\mathbf{R}$, and estimated state covariance $\mathbf{P}$. For each covariance component, the histograms show values obtained from ground-truth trajectories, shown in blue, the baseline KF setting, shown in orange, and adaptive CAKF, shown in green.

divergence between the reference and estimated covariance distributions. In addition to the baseline and adaptive CAKF settings, we also evaluate adaptive KF and non-adaptive CAKF variants. Table C.3 reports the average JS divergence for each dataset, aggregated over covariance components belonging to $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{P}$. Lower values indicate closer alignment with the reference distributions.

Table C.3: Average Jensen–Shannon divergence between reference and estimated covariance distributions. Lower is better. Best results are shown in bold.

Dataset	Matrix	Baseline	CAKF	Adapt. KF	Adapt. CAKF
MOT20	$\mathbf{Q}$	0.67	0.67	0.51	0.41
	$\mathbf{R}$	0.68	0.60	0.73	0.56
	$\mathbf{P}$	0.67	0.65	0.52	0.41
MOT17	$\mathbf{Q}$	0.56	0.56	0.40	0.45
	$\mathbf{R}$	0.58	0.52	0.73	0.60
	$\mathbf{P}$	0.54	0.53	0.53	0.45
DanceTrack	$\mathbf{Q}$	0.52	0.52	0.51	0.47
	$\mathbf{R}$	0.51	0.38	0.75	0.62
	$\mathbf{P}$	0.45	0.42	0.52	0.47
SportsMOT	$\mathbf{Q}$	0.59	0.60	0.53	0.34
	$\mathbf{R}$	0.70	0.36	0.81	0.41
	$\mathbf{P}$	0.54	0.38	0.53	0.34

Overall, adaptive CAKF achieves the lowest divergence in the majority of cases, particularly on MOT20 and SportsMOT. On MOT17 and DanceTrack, the results are more mixed, especially for $\mathbf{R}$, where non-adaptive CAKF achieves lower divergence. Nevertheless, adaptive CAKF provides the best or competitive covariance alignment while also achieving the strongest tracking performance in the ablation studies.

This observation is consistent with the results in Table C.2 and Table 5.8 (see Section 5.4.6) and indicates that combining adaptive covariance estimation with crowd-aware scaling improves the practical usefulness of the estimated noise models.

C.4 Additional EIoU Experiments

In Figure C.2 and Table C.4, we present additional statistics and results analogous to those reported for MOT20 and SportsMOT in Section 5.4.6. Here, the same analysis is extended to the MOT17 and DanceTrack validation sets.

Similarly to the previously analyzed datasets, the scaling factor s_w is consistently larger than s_h on both MOT17 and DanceTrack and under both Kalman filter settings. On

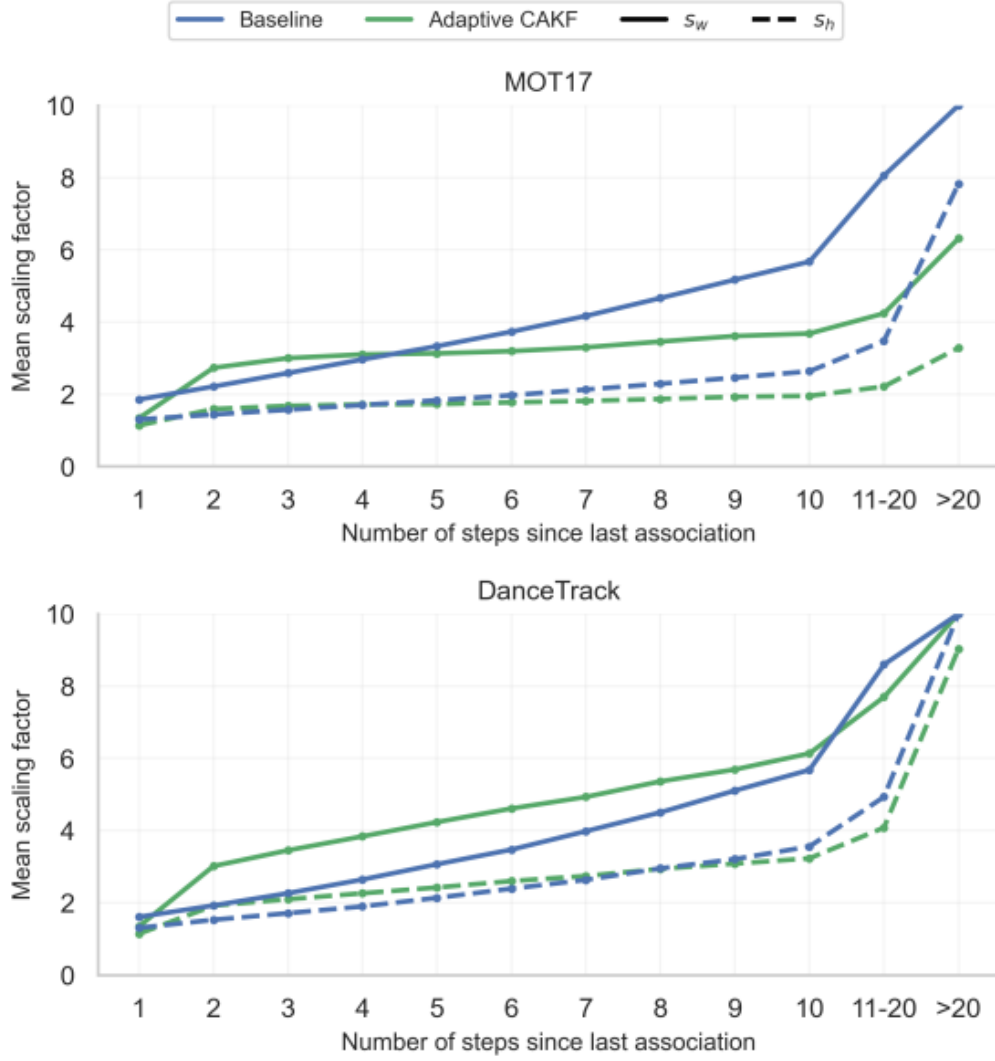


Figure C.2: Mean values of the scaling factors s_w and s_h over time on the MOT17 and DanceTrack validation sets. Solid lines correspond to s_w , while dashed lines correspond to s_h . The scaling factors are calculated using either the baseline KF setting, shown in blue, or Adapt. CAKF, shown in green. Values above 10 are clipped for visualization purposes.

the more dynamic DanceTrack dataset, adaptive CAKF produces larger s_w values than the baseline KF, reflecting higher estimated horizontal uncertainty.

However, as shown in Table C.4, incorporating EIoU without the proposed association workflow degrades tracking performance on both MOT17 and DanceTrack. This is consistent with the results from the main chapter: uncertainty-aware bounding-box expansion is not sufficient on its own, but becomes beneficial when combined with the improved association workflow.

Table C.4: Influence of EIoU under the baseline KF and Adapt. CAKF settings on the MOT17 and DanceTrack validation sets. Best results for each dataset are shown in bold.

Dataset	Adapt. CAKF	EIoU	HOTA	IDF1	MOTA	IDSW
MOT17	$\times$	$\times$	62.70	72.32	69.84	456
	$\times$	$\checkmark$	60.80	69.30	68.14	529
	$\checkmark$	$\times$	65.73	77.19	70.47	305
	$\checkmark$	$\checkmark$	64.20	74.56	68.79	355
DanceTrack	$\times$	$\times$	48.41	45.20	87.08	3164
	$\times$	$\checkmark$	47.33	44.00	86.56	3360
	$\checkmark$	$\times$	53.81	53.54	87.22	2463
	$\checkmark$	$\checkmark$	52.99	53.39	86.24	2628

C.5 Additional Association Workflow Analysis

In this section, we further examine the behavior of the proposed association and tracklet management components on the MOT17 and DanceTrack validation sets.

C.5.1 KF-Based Tracklet Confidence

Similarly to Table 5.11, in Table C.5 we show the results of adding KFTC to the baseline that uses EIoU on the MOT17 and DanceTrack validation sets. The results are consistent with those obtained on MOT20 and SportsMOT, analyzed in Table 5.11. Adding KFTC to EIoU improves association and, consequently, overall tracking performance, especially when used together with adaptive CAKF.

Table C.5: Influence of KF-based tracklet confidence (KFTC) under the baseline KF and Adapt. CAKF settings on the MOT17 and DanceTrack validation sets. Best results are shown in bold.

Dataset	Adapt. CAKF	KFTC	HOTA	IDF1	MOTA	IDSW
MOT17	$\times$	$\times$	60.80	69.30	68.14	529
	$\times$	$\checkmark$	61.39	70.11	68.23	510
	$\checkmark$	$\times$	64.20	74.56	68.79	355
	$\checkmark$	$\checkmark$	64.88	75.88	68.70	344
DanceTrack	$\times$	$\times$	47.33	44.00	86.56	3360
	$\times$	$\checkmark$	47.60	44.65	86.66	3277
	$\checkmark$	$\times$	52.99	53.39	86.24	2628
	$\checkmark$	$\checkmark$	54.43	56.02	86.49	2348

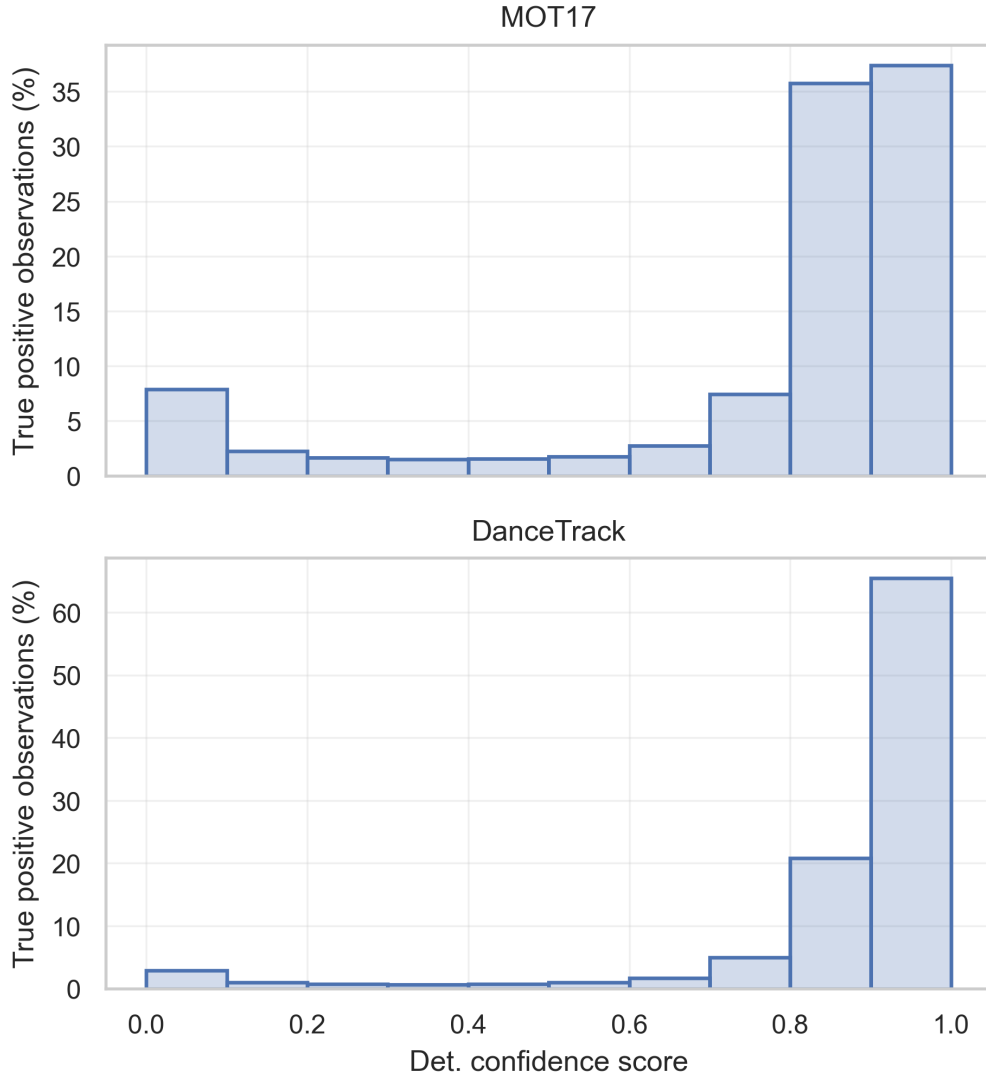


Figure C.3: Histogram of detection confidence scores for true-positive detections on the MOT17 and DanceTrack validation sets. Each bin represents a confidence interval of width 0.1.

C.5.2 Association and Trajectory Initialization Criteria

Analogously to Figure 5.12 from Section 5.4.6, in Figure C.3 we show the confidence scores of true-positive detections on MOT17 and DanceTrack. On DanceTrack, 90% of true-positive detections have confidence scores greater than 0.74, while only 2.94% have confidence scores below 0.1. However, on MOT17, 7.88% of true-positive detected bounding boxes have confidence scores less than 0.1.

In Figure 5.13 from Section 5.4.6, we analyzed the influence of Crowd-Aware Tracklet Creation (CATC) and the proposed Association Validity Criterion (AVC). We performed a grid search under multiple baseline settings and reported heatmaps of HOTA scores. In Figure C.4, we present heatmaps of HOTA, IDF1, MOTA, and IDSW for the setting

that uses adaptive CAKF, EIoU, KFTC, and $\tau_{\text{det}} = 0.01$, for all evaluated datasets.

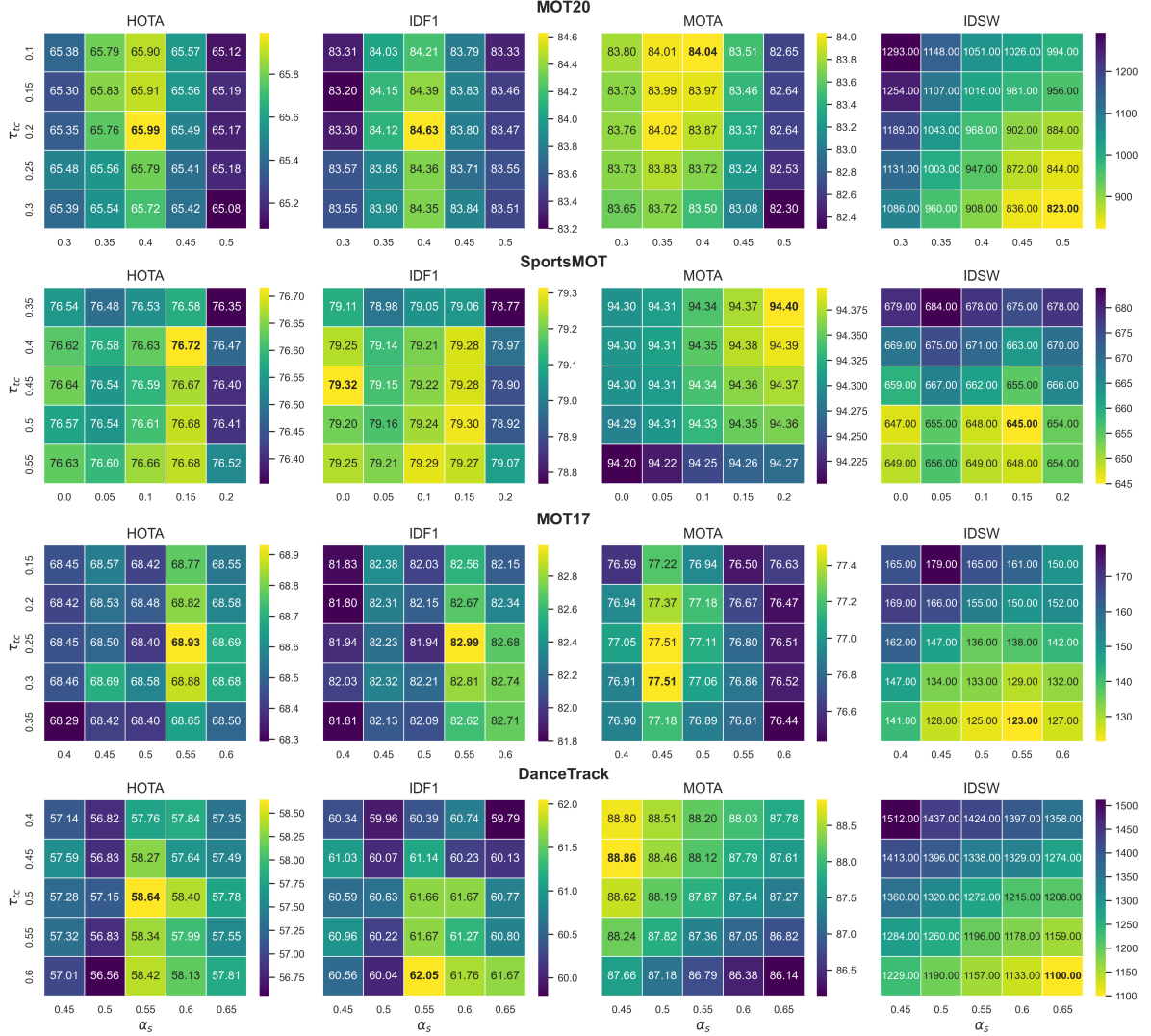


Figure C.4: Grid-search results for the CATC threshold τ_{tc} and the admissible match hyperparameter α_s on the MOT20, SportsMOT, MOT17, and DanceTrack validation sets. Each cell represents the tracking performance obtained for a specific parameter combination, measured by HOTA, IDF1, MOTA, and IDSW. The best value in each heatmap is shown in bold.

C.5.3 Trajectory Update Criterion

In Table C.6, we present the results of the trajectory update criterion (TUC) analysis on the MOT17 and DanceTrack validation sets. The experimental setup is analogous to the one used in Table 5.13, where TUC was analyzed on MOT20 and SportsMOT.

Similarly to the observations in the main text, moderate values of τ_u provide the best performance. On MOT17, the best results are obtained for $\tau_u \in [0.15, 0.30]$, while on

DanceTrack the optimal range is $\tau_u \in [0.25, 0.45]$.

Using $\tau_u = 0.25$ improves performance on MOT17 by +1.02 HOTA, +0.75 IDF1, and +1.74 MOTA, with an increase of +39 IDSW. On DanceTrack, the same setting improves performance by +0.55 HOTA, +0.23 IDF1, and +1.70 MOTA, while reducing IDSW by 26, compared to the baseline setting without TUC.

Table C.6: Influence of the trajectory update hyperparameter τ_u on the MOT17 and DanceTrack validation sets. The first row corresponds to the standard update criterion based on three consecutive matches. Best results are shown in bold.

τ_u	MOT17				DanceTrack			
	HOTA	IDF1	MOTA	IDSW	HOTA	IDF1	MOTA	IDSW
/	68.93	82.99	76.80	138	58.64	61.66	87.87	1272
0	63.12	74.27	54.63	187	56.65	59.29	80.94	1162
0.05	69.34	83.03	77.17	162	58.61	61.34	87.85	1136
0.10	69.72	83.49	78.16	172	58.87	61.59	88.68	1147
0.15	69.95	83.70	78.61	177	59.03	61.74	89.14	1169
0.20	69.99	83.74	78.61	177	59.13	61.84	89.43	1214
0.25	69.95	83.74	78.54	177	59.19	61.89	89.57	1246
0.30	69.89	83.74	78.42	177	59.24	61.95	89.68	1284
0.35	69.75	83.65	78.15	169	59.23	61.95	89.63	1317
0.40	69.66	83.62	78.02	164	59.24	61.98	89.65	1316
0.45	69.52	83.54	77.73	170	59.24	62.00	89.58	1378
0.50	69.17	83.21	77.09	190	59.07	61.86	89.04	1566

In Table C.7, we compare the proposed online method with the baseline without TUC and with the baseline that applies linear interpolation as offline post-processing. The results are consistent with those in Table 5.14, where TUC was compared with offline post-processing on MOT20 and SportsMOT. On MOT17, which exhibits more regular motion, the online TUC setting is slightly worse than the post-processed baseline. In contrast, on DanceTrack, which contains more dynamic motion, TUC achieves better HOTA, IDF1, and MOTA than both baselines.

Table C.7: Comparison between the proposed trajectory update criterion (TUC) and the standard update criterion with linear interpolation as post-processing. Results are reported on the MOT17 and DanceTrack validation sets, with the best results shown in bold.

Dataset	TUC	Lin. inter.	HOTA	IDF1	MOTA	IDSW
MOT17	$\times$	$\times$	68.93	82.99	76.80	138
	$\times$	$\checkmark$	70.51	84.21	79.24	116
	$\checkmark$	$\times$	69.95	83.74	78.54	177
DanceTrack	$\times$	$\times$	58.64	61.66	87.87	1272
	$\times$	$\checkmark$	58.61	61.60	88.08	1024
	$\checkmark$	$\times$	59.19	61.89	89.57	1246

Biography

Vukašin Stanojević was born on March 9, 1995, in Pirot, Serbia. He completed “Mladost” Elementary School in Veliko Bonjince, Municipality of Babušnica, and “9. maj” Gymnasium in Niš as a Vuk Diploma holder and the top student of his generation.

In the academic year 2014/2015, he enrolled in undergraduate academic studies at the Department of Computer Science, Faculty of Sciences and Mathematics, University of Niš. He completed the studies in 2017 with a grade point average of 9.92. In the academic year 2017/2018, he enrolled in master academic studies at the same department, module Software Development. He completed the master studies in 2019 with a grade point average of 10.00. His master thesis was entitled *Reinforcement Learning Algorithms with Application to Dynamic System Control*. In the academic year 2019/2020, he enrolled in doctoral academic studies at the Department of Computer Science, Faculty of Sciences and Mathematics, University of Niš. During his doctoral studies, he passed all exams with a grade point average of 10.00.

He has been employed at the Faculty of Sciences and Mathematics, University of Niš, since 2020, first as a junior researcher and, since 2021, as a teaching assistant at the Department of Computer Science. As a junior researcher, he worked on the project *Development of Methods of Computation and Information Processing: Theory and Applications*, supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

As a teaching assistant, he has been involved in teaching activities in several courses in programming, machine learning, artificial intelligence, parallel and distributed programming, and related areas of computer science. In the academic years 2021/2022, 2022/2023, and 2023/2024, he also taught at “9. maj” Gymnasium in Niš, in the specialized class for students with talents in computer science and informatics.

His research interests include computer vision, multiple object tracking, efficient implementation of machine learning algorithms, and parallel computing.

He is the author or co-author of several scientific papers published in international journals and has presented his results at scientific conferences. He is also a co-author of a teaching textbook used in the course Artificial Neural Networks.

Bibliography

Journal Articles and Manuscript

- **Vukašin Stanojević** and Branimir Todorović. *BoostTrack++: using tracklet information to detect more objects in multiple object tracking*. Filomat, 39(16), 2025.
- **Vukašin Stanojević** and Branimir Todorović. *BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking*. Machine Vision and Applications, 35(3), 2024.
- **Vukašin Stanojević**, Vyacheslav Kazakovtsev, Predrag Stanimirović, Natalia Rezova, and Galina Shkaberina. *Calculating the Moore–Penrose Generalized Inverse on Massively Parallel Systems*. Algorithms, 15, 2022.
- **Vukašin Stanojević**. *Towards Simple, Robust, and Real-Time Multiple Object Tracking with Coupled Motion Estimates*. Manuscript under review in *International Journal of Computer Vision*.

Abstract Presentations

- **Vukašin Stanojević** and Branimir Todorović. *Multiple Object Tracking with Adaptive Kalman Filter*. Artificial Intelligence Conference, Belgrade, Serbia, 2024.
- **Vukašin Stanojević**. *Humanity and the Future of AI: Utopia or Dystopia*. Artificial Intelligence Conference, Belgrade, Serbia, 2024.
- **Vukašin Stanojević**. *Beyond Simple Thresholding: Using Tracklet Information to Improve Selection of True Positive Detections for Multiple Object Tracking*. XV Serbian Mathematical Congress, Belgrade, Serbia, 2024.

Textbooks and Teaching Materials

- **Vukašin Stanojević** and Branimir Todorović. *Veštačke neuronske mreže: zbirka zadataka i problema sa rešenjima u Python-u*. University of Niš, Faculty of Sciences and Mathematics, 2023.

Thesis Documentation

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је докторска дисертација, под насловом

НАПРЕДНИ АЛГОРИТМИ ПРАЋЕЊА ВЕЋЕГ БРОЈА ОБЈЕКТА

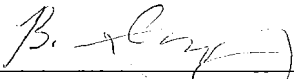
која је одбрањена на Природно-математичком факултету Универзитета у Нишу:

- резултат сопственог истраживачког рада;
- да ову дисертацију, ни у целини, нити у деловима, нисам пријављивао на другим факултетима, нити универзитетима;
- да нисам повредио ауторска права, нити злоупотребио интелектуалну својину других лица.

Дозвољавам да се објаве моји лични подаци, који су у вези са ауторством и добијањем академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада, и то у каталогу Библиотеке, Дигиталном репозиторијуму Универзитета у Нишу, као и у публикацијама Универзитета у Нишу.

У Нишу, 20. 8. 2026. године.

Потпис аутора дисертације:


Вукашин Д. Станојевић

AUTHORSHIP STATEMENT

I declare that doctoral dissertation entitled

ADVANCED MULTIPLE OBJECT TRACKING ALGORITHMS

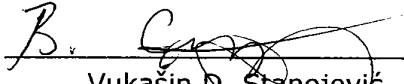
defended at Faculty of Sciences and Mathematics, University of Niš, is

- the result of original personal research;
- has not been submitted, either partially or in a whole, to any other faculty or university;
- not abusing authorship rights or intellectual property of anyone else.

I am allowing publication of my personal data related to the authorship of this dissertation, as well as the data related to gaining the academic title Doctor of Science, such as name, last name, year and place of birth, PhD defending date, to be published in the library catalogue of the University of Niš, digital repository of the University of Niš and in publication in general of the University of Niš.

In Niš, 20. 8. 2026.

Dissertation author's signature:


Vukašin D. Stanojević

**ИЗЈАВА О ИСЛОВЕТНОСТИ ЕЛЕКТРОНСКОГ И ШТАМПАНОГ ОБЛИКА
ДОКТОРСКЕ ДИСЕРТАЦИЈЕ**

Наслов дисертације:

НАПРЕДНИ АЛГОРИТМИ ПРАЋЕЊА ВЕЋЕГ БРОЈА ОБЈЕКТА

Изјављујем да је електронски облик моје докторске дисертације, коју сам предао за уношење у **Дигитални репозиторијум Универзитета у Нишу**, истоветан штампаном облику.

У Нишу, 20. 8. 2026. године.

Потпис аутора дисертације:


Вукашин Д. Станојевић

STATEMENT ABOUT IDENTITY OF PRINTED AND ELECTRONIC FORMS OF THE DISSERTATION

Dissertation title:

ADVANCED MULTIPLE OBJECT TRACKING ALGORITHMS

I declare that the electronic form of my PhD dissertation, delivered to be uploaded in the **Digital repository of University of Niš**, is identical to the printed form of the dissertation.

In Niš, 20. 8. 2026.

Dissertation author's signature:



Vukašin D. Stanojević

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Никола Тесла“ да у Дигитални репозиторијум Универзитета у Нишу унесе моју докторску дисертацију, под насловом

НАПРЕДНИ АЛГОРИТМИ ПРАЋЕЊА ВЕЋЕГ БРОЈА ОБЈЕКТА

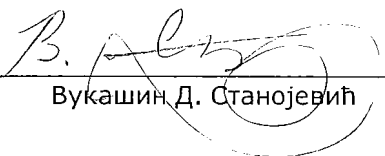
Дисертацију са свим прилозима предао сам у електронском облику, погодном за трајно архивирање.

Моју докторску дисертацију, унету у Дигитални репозиторијум Универзитета у Нишу, могу користити сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons), за коју сам се одлучио.

1. Ауторство **(CC BY)**
2. Ауторство – некомерцијално **(CC BY-NC)**
3. Ауторство – некомерцијално – без прераде **(CC BY-NC-ND)**
4. Ауторство – некомерцијално – делити под истим условима **(CC BY-NC-SA)**
5. Ауторство – без прераде **(CC BY-ND)**
6. Ауторство – делити под истим условима **(CC BY-SA)**

У Нишу, 20. 8. 2026. године.

Потпис аутора дисертације:


Вукашин Д. Станојевић

USING STATEMENT

I am authorising herewith the library "Nikola Tesla" of the University of Niš to archive my dissertation into Digital repository of the University of Niš, entitled:

ADVANCED MULTIPLE OBJECT TRACKING ALGORITHMS

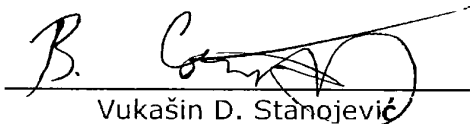
I have submitted the dissertation, along with its enclosures, in the electronic form, appropriate for permanent archiving.

My dissertation, archived in the Digital repository of the University of Niš, can be used by anyone who respects restrictions imposed by the chosen type of Creative Commons licence.

1. Authorship (CC BY)
2. Noncommercial authorship (CC BY-NC)
- 3. Noncommercial authorship-no alterations (CC BY-NC-ND)**
4. Noncommercial authorship-share under the same conditions (CC BY-NC-SA)
5. Authorship-no alterations (CC BY-ND)
6. Authorship-share under the same conditions (CC BY-SA)

In Niš, 20. 8. 2026.

Dissertation author's signature:


Vukašin D. Stanojević



**ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ**

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	монографска
Тип записа, ТЗ:	текстуални / графички
Врста рада, ВР:	докторска дисертација
Аутор, АУ:	Вукашин Станојевић
Ментор, МН:	Бранимир Тодоровић
Наслов рада, НР:	Напредни алгоритми праћења већег броја објеката
Језик публикације, ЈП:	енглески
Језик извода, ЈИ:	енглески
Земља публикавања, ЗП:	Србија
Уже географско подручје, УГП:	Србија
Година, ГО:	2026.
Издавач, ИЗ:	ауторски репринт
Место и адреса, МА:	Ниш, Вишеградска 33.
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	6/153/153/33/27/0/3
Научна област, НО:	Рачунарске науке
Научна дисциплина, НД:	Вештачка интелигенција
Предметна одредница/Кључне речи, ПО:	Праћење већег броја објеката, праћење засновано на детекцијама, анализа видеа, визуелно праћење, праћење засновано на кретању, придруживање података, Калманов филтер, праћење у реалном времену
УДК	004.932:004.8(043)
Чува се, ЧУ:	библиотека
Важна напомена, ВН:	

Извод, ИЗ:	Ова дисертација проучава онлајн праћење већег броја објеката у оквиру парадигме праћења заснованог на детекцијама, са фокусом на интеракцију између детекције, естимације кретања, придруживања података и управљања праћеним објектима. Циљ је развој алгоритама за праћење који су тачни, робусни, адаптивни, једноставни и погодни за рад у реалном времену. У дисертацији се истражује како се непоуздане детекције и неизвесне предикције кретања могу ефикасније користити. Уводе се мере поузданости које током придруживања података имплицитно фаворизују поуздане парове детекција и праћених објеката. Додатни механизми користе информације из трајекторија за избор тачно позитивних детекција ниске поузданости и побољшано преидруживање података помоћу једноставних мера сличности. У дисертацији се затим предлаже онлајн систем заснован на информацијама о кретању, који повезује адаптивну процену кретања, моделирање неизвесности у зависности од густине сцене, проширење граничних оквира на основу неизвесности, придруживање података у једној фази и управљање праћеним објектима. Уместо ослањања на визуелне карактеристике, постпроцесирање или сложене процедуре придруживања података, једноставне компоненте система деле информације о неизвесности и поузданости. Последњи метод у дисертацији постиже резултате на нивоу најбољих савремених метода у својој категорији и остаје упоредив са сложенијим онлајн методама заснованим на визуелном изгледу и офлајн методама заснованим на кретању, или их надмашује, на већини евалуираних скупова података. У целини, дисертација показује да се робусно онлајн праћење више објеката може постићи пажљивим повезивањем једноставних компоненти.
Датум прихватања теме, ДП:	29. 1. 2025.
Датум одбране, ДО:	
Чланови комисије, КО:	Председник: Члан: Члан: Члан: Члан, ментор:



**ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ**

KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	monograph
Type of record, TR :	textual / graphic
Contents code, CC :	doctoral dissertation
Author, AU :	Vukašin Stanojević
Mentor, MN :	Branimir Todorović
Title, TI :	Advanced Multiple Object Tracking Algorithms
Language of text, LT :	English
Language of abstract, LA :	English
Country of publication, CP :	Serbia
Locality of publication, LP :	Serbia
Publication year, PY :	2026
Publisher, PB :	author's reprint
Publication place, PP :	Niš, Višegradska 33.
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	6/153/153/33/27/0/3
Scientific field, SF :	Computer science
Scientific discipline, SD :	Artificial intelligence
Subject/Key words, S/KW :	Multiple object tracking, tracking-by-detection, video analysis, visual tracking, motion-based tracking, data association, Kalman filtering, real-time tracking
UC	004.932:004.8(043)
Holding data, HD :	library
Note, N :	

Abstract, AB :	The dissertation studies online multiple object tracking (MOT) in the tracking-by-detection paradigm, focusing on the interaction between detection, motion estimation, data association, and tracklet management. Its objective is to develop tracking algorithms that are accurate, robust, adaptive, simple, and suitable for real-time use. It investigates how unreliable detections and uncertain motion predictions can be used more effectively. Detection-tracklet confidence scores are introduced to implicitly favor reliable detection-tracklet pairs in one-stage association. Additional mechanisms use tracklet information to select true-positive low-confidence detections and improve association through lightweight similarity cues. The dissertation further proposes a strictly online, motion-based framework that couples adaptive motion estimation, crowd-aware uncertainty modeling, uncertainty-guided bounding-box expansion, one-stage association, and tracklet management. Instead of relying on appearance features, post-processing, or complex association pipelines, this framework propagates uncertainty and reliability information between simple tracking components. The final framework achieves state-of-the-art results within its category and remains comparable to, or outperforms, more complex online appearance-based and offline motion-based methods on most evaluated benchmarks. Overall, the dissertation shows that robust online MOT can be achieved by carefully coupling simple components.										
Accepted by the Scientific Board on, ASB :	29. 1. 2025.										
Defended on, DE :											
Defended Board, DB :	<table border="1"> <tr> <td data-bbox="399 1288 614 1332">President:</td><td data-bbox="614 1288 1466 1332"></td></tr> <tr> <td data-bbox="399 1332 614 1377">Member:</td><td data-bbox="614 1332 1466 1377"></td></tr> <tr> <td data-bbox="399 1377 614 1422">Member:</td><td data-bbox="614 1377 1466 1422"></td></tr> <tr> <td data-bbox="399 1422 614 1467">Member:</td><td data-bbox="614 1422 1466 1467"></td></tr> <tr> <td data-bbox="399 1467 614 1514">Member, Mentor:</td><td data-bbox="614 1467 1466 1514"></td></tr> </table>	President:		Member:		Member:		Member:		Member, Mentor:	
President:											
Member:											
Member:											
Member:											
Member, Mentor:											

Образац Q4.09.13 - Издање 1