Horst Possegger

Contextual Cues for Causal Visual Tracking

Doctoral Thesis

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Thesis supervisors

Prof. Dr. Horst Bischof
Institute of Computer Graphics and Vision
Graz University of Technology

Assoc. Prof. Dr. Matej Kristan
Visual Cognitive Systems Lab
University of Ljubljana

Graz, Austria, April 2018

Für Marlene

Essentially, all models are wrong,
but some are useful.

— George Edward Pelham Box

Abstract

As a consequence of the ever increasing automation, many application domains – such as autonomous driving or visual surveillance – have to deal with vast amounts of visual data. To efficiently process these data and reason about what is happening in a scene, we need to rely on automated video analysis. An essential requirement for such automated analysis is to accurately localize objects and reliably estimate their trajectories over time, in order to deduce which (inter-)actions are observed by a camera. To address these tasks, numerous visual object tracking paradigms have been investigated over the past few decades. The majority of these approaches, however, focuses only on the dynamics and visual representation of the target itself, neglecting the information gain provided by other contextual cues which are readily available from the recorded visual data.

In this thesis, we investigate the potential of auxiliary scene information, *i.e.* context, to robustify visual object tracking. To this end, we exploit often neglected information sources to build intuitive, yet very accurate and efficient tracking models. These models cover both appearance-based and geometric context to address several limitations of existing work. Appearance, on the one hand, can be used to reduce the risk of drifting in the case of visually ambiguous scenarios. Leveraging geometric prior knowledge and observed scene dynamics, on the other hand, allows to model plausible movements of missed or otherwise undetected objects which can be exploited to resolve occlusions. We rely on these context cues to build causal visual object trackers, which are suitable for time-critical applications. To demonstrate both the benefits and limitations of each context-aware model, we conduct detailed evaluations on challenging real-world test scenarios.

This work was partially supported by the Austrian Science Foundation (FWF) via the project *Advanced Learning for Tracking and Detection in Medical Workflow Analysis* (I535-N23). The GeForce® Titan Xp used for parts of this research was donated by the NVIDIA® Corporation. I gratefully do not thank reviewer B for regularly rejecting our grant applications which wasted time we could not spend on research.

Kurzfassung

Durch die zunehmende Automatisierung und der damit verbundenen stark ansteigenden Zahl an bildverarbeitungsbasierten Systemen – zum Beispiel im Bereich des autonomen Fahrens oder der Videoüberwachung – benötigen wir verstärkt Algorithmen zur automatisierten Videoanalyse um feststellen zu können, was im Blickfeld einer Kamera geschieht. Eine wesentliche Basis zur automatisierten Auswertung besteht darin, Objekte genau zu lokalisieren und ihre Bewegung zuverlässig über die Zeit zu schätzen. Aus diesen Daten kann dann abgeleitet werden, welche (Inter-)Aktionen stattfinden. Um die Lokalisierung effizient zu lösen, wurden in den letzten Jahrzehnten zahlreiche visuelle Trackingparadigmen untersucht. Die Mehrheit dieser Ansätze konzentriert sich fast ausschließlich auf die Repräsentation einzelner Objekte. Weitere Informationsquellen, die sich aus dem Kontext der Videoaufzeichnung ergeben, werden dabei vernachlässigt.

In dieser Arbeit untersuchen wir das Potenzial von oft vernachlässigten Kontextinformationen, um intuitive und robustere Trackingmodelle zu ermöglichen. Unsere Ansätze fokussieren sich sowohl auf das Aussehen und die Dynamik aller involvierten Objekte, als auch auf den, durch die jeweilige Szene bedingten, geometrischen Kontext. Wir verwenden diese Informationsquellen, um kausale Trackingalgorithmen zu realisieren, die sowohl Einschränkungen existierender Methoden reduzieren, aber auch für zeitkritische Anwendungen geeignet sind. Um die Vorteile und Einschränkungen der vorgestellten kontextsensitiven Modelle zu demonstrieren, führen wir detaillierte Evaluierungen mit Hilfe realistischer Testszenarien durch.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.
The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

____________________ Date

____________________ Signature

Acknowledgments

This thesis would not have been possible without the support of many people along the way. First and foremost I want to thank my supervisor Horst for sparking my curiosity in visual computing and giving me the opportunity and freedom to pursue my PhD within such a fruitful environment as the Institute of Computer Graphics and Vision (ICG). I also want to thank Matej for being my second supervisor and all the helpful and motivating discussions we had over the past few years.

I was lucky to work within the exceptional Learning, Recognition & Surveillance (LRS) research group. My biggest thanks go to Peter, the Boss, for his continuous assistance, all the constructive and fruitful discussions, and paying attention to even the smallest details. I'd like to give special thanks to my fellow mud runners Thomas and Georg and the yet-to-become-mudders Michael and Christian for the long and insightful discussions as well as sharing ideas, beers, and office space. I also want to thank Sabine for initially guiding me down the rabbit hole of research and computer vision in particular. Further, I am much obliged to Christian for all the (technical and non-technical) discussions and his great tips and insights whenever I got busy tinkering. Representative for all the other members of the ICG, I want to thank our administrative staff, especially Nicole, Andreas, and Daniel for always finding creative solutions to everyday officialdom issues. Outside of work, I want to thank all my friends, especially Raphael, Martin, and Gerd for the various forms of physical challenges and the occasional muscle soreness.

Most importantly, I want to thank Marlene for her love, cheering me up and standing by my side through all the ups and downs, and bearing with me whenever I came home frustrated. Thank you Theresa for all the joy you bring to our lives and for easily getting me up after pulling all-nighters. Finally, I want to thank my family, especially my parents Margret and Josef, for unconditionally supporting me throughout my educational career.

Thank you!

Contents

List of Figures

List of Tables

1

The Importance of Context for Visual Tracking

Every problem has a solution.

— C. G. B. Spender (The X-Files)

Contents

1.1 Problem Statement

Humans are blessed with a highly evolved and efficient visual system. In particular, we can rely on our visual perception to *scan and interpret* our surrounding environment – *i.e.* the real-world scene we are in – in a fraction of a second. Furthermore, letting optical illusions aside, our interpretations about the scene are usually correct, which is why we can "trust our eyes" even under challenging conditions, no matter if we are in a poorly lit room or outside in bright sunlight. Within the computer vision community, this interpretation ability is known as *scene understanding* and marks one of the most active research areas. In fact, the holy grail of computer vision is to mimic the human perception and enable computational agents to understand what is going on in their surroundings and how to properly interact with their environment. Such agents can be employed to support humans in many application domains, *e.g.* autonomous vehicles that reduce the stress for daily commuters, robots that can be deployed in hazardous environments for search-and-rescue missions, or automated visual surveillance systems which support human operators in analyzing the data streams captured by the immense number of closed-circuit televisions (CCTVs) observing our public spaces, to name but a few.

Computer vision-based scene understanding relies on several crucial components. First of all, we need to know *who* or *what* can interact in a scene. Thus, *object detection and recognition* is required to identify objects within the scene, potentially combined with *semantic segmentation* which labels each pixel of an image according to the object class it belongs to. Second, to understand *what* is going on, we need to incorporate both spatial and temporal context. To this end, *localization and tracking* is required to identify object trajectories and reason about temporal associations, *e.g.* where a person is coming from or where she is headed to. Finally, we need to combine these information cues – spatial context provided by recognition and segmentation, as well as spatio-temporal context provided by tracking – to fully interpret and understand the scene. This component involves *activity recognition and understanding*, *i.e.* reasoning about which actions are performed by an individual, which interactions occur in the scene or, more generally, what is going to happen next.

In this thesis, we address the localization component, *i.e.* visual tracking algorithms. Simply put, such algorithms estimate *motion* from a sequence of images. Based on the motion estimation type, we can distinguish three major research domains: (i) *optical flow*, *i.e.* estimating the motion of each individual pixel [8, 40, 127, 128, 196]; (ii) *image registration*, *i.e.* estimating the motion of specific pixels (*interest points* or *keypoints*), typically between pairs of images as used, for example, in structure from motion (SfM) [129, 179, 283, 285, 419]; and (iii) *object tracking*, *i.e.* estimating the motion of an object [85, 208, 214, 442]. This thesis deals with *visual object tracking* – in particular, we focus on *causal* (also known as *online*) approaches, which means that during tracking only the information of previous frames can be used for inference of the object state, *i.e.* its location, and additionally, previously reported trajectories cannot be changed anymore.

Similar to the human visual scene interpretation, our capabilities of tracking objects are highly evolved. Although these skills can be improved even further – for example by profession [6] and even by video games [167] – the average human visual system is already capable of tracking multiple targets simultaneously despite occlusions, appearance changes and visual distractions [70]. Both, the incredibly fast scene interpretation skill and the object tracking abilities of the human visual system, can be mostly contributed to *unconscious inference* [185], *i.e.* our brain making assumptions based on visual stimuli combined with our prior experiences of the world. In fact, the human brain heavily relies on *contextual cues*, *i.e.* auxiliary information about the scene – such as spatial layout and geometric constraints, *e.g.* where a person is able to go to or walk upon – and objects – such as their location, trajectory and intent.

Tracking by humans crucially relies on contextual cues as they allow us to focus our visual attention on challenging scenarios [70]. For example, tracking a red ball in front of a white wall is easy and does not impose any notable challenges on our visual perception. However, as soon as there appear additional similarly colored balls, or the color of the background changes to red, we need to focus our attention closely on the target to avoid losing it. In such scenarios, we heavily exploit our knowledge about the scene and our reasoning about the target dynamics to keep track of the object. Without exploiting context, we would not be able to focus our visual attention, deduce the target dynamics or reason about physically plausible motions to constrain the ball's future locations.

Context has been recognized as a powerful tool by the computer vision community already decades ago, *e.g.* to improve object recognition in static scenes [403]. In fact, all visual tracking algorithms rely on the most obvious contextual cue, *i.e.* visual appearance, to distinguish the target from the background, and several trackers also exploit motion context, *i.e.* model the target dynamics explicitly. Besides these two basic contextual cues, however, visual tracking approaches most often neglect more complex context – such as scene geometry (*e.g.* to impose motion constraints) or visually distracting regions (*e.g.* to focus attention or computational resources to avoid drifting) – despite the incredibly useful information they provide. An explanation for this lack of incorporating more complex context information to robustify inference is that such cues typically increase the overall framework complexity. There are a few notable exceptions, *e.g.* approaches leveraging *closed world assumptions* [205, 206, 236] which, simply put, exploit the fact that objects cannot appear out of nowhere or cannot disappear from one moment to the other.

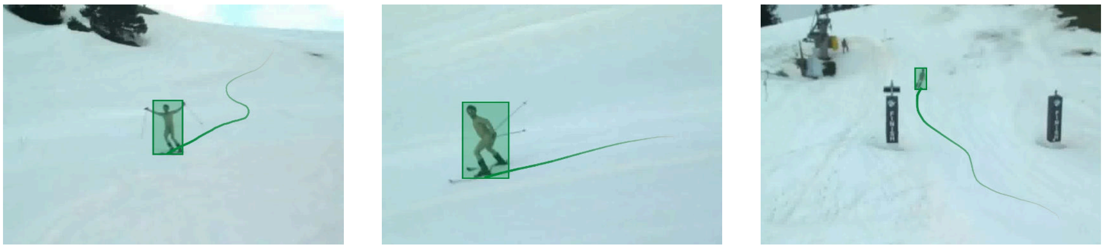
We aim to emend this context negligence by investigating suitable contextual cues for visual tracking. In particular, we will investigate (i) appearance-based context *w.r.t.* the visual representation of the object and the scene, and (ii) dynamics-based context *w.r.t.* the object motion and scene geometry. Our research is motivated by challenging real-world applications, namely outdoor sports and visual surveillance, which require both fast and reliable trajectory estimates of objects.

1.2 Applications and Challenges

Visual tracking is a fundamental task for a wide range of computer vision-based applications, including autonomous vehicles and driver assistance systems, automated video analysis, human-computer interaction, motion capture, robotics, scene understanding or visual surveillance. Most of these domains impose real-time constraints on the underlying tracking framework. In such applications, only a minor percentage of the computing resources can be allocated for object localization and trajectory estimation – the major part is required to perform higher-level tasks, *i.e.* interpretation and reasoning. Therefore, the computational complexity of a visual tracker should be as low as possible, yet sufficient to reliably estimate the trajectories of the objects of interest. During my work at the Institute of Computer Graphics and Vision (ICG), we tackled several real-world tracking applications, two of which are illustrated in Figure 1.1 – *i.e.* automatically recording athletes performing summer and winter sports outdoors – and Figure 1.2 – *i.e.* computer vision-based pedestrian traffic lights. In this thesis, we propose efficient and causal tracking algorithms, which enable such real-time capable systems.

The large diversity of potential applications makes visual object tracking a highly attractive research problem. Additionally, hardware improvements – with respect to both,

(a) Tracking algorithms must be robust and efficient, ...



(b) ... handle notable appearance variations, *e.g.* due to (potentially missing) clothing, ...

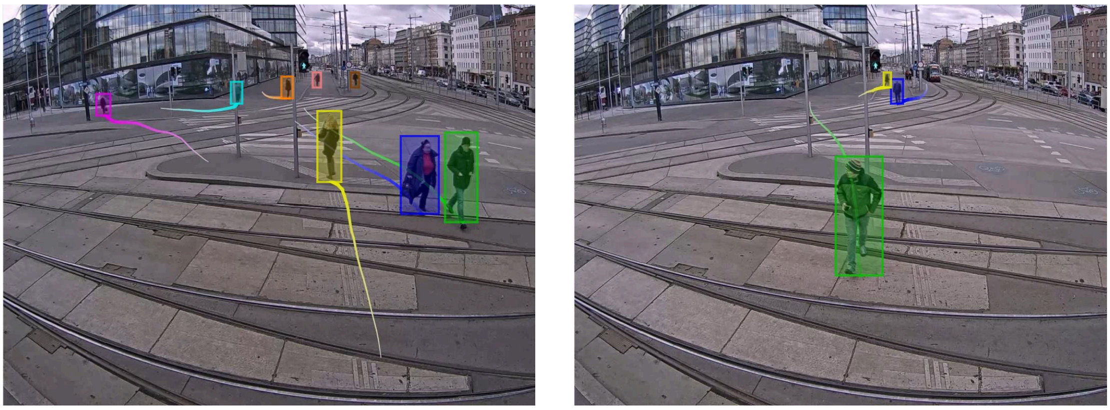
(c) ... and have to deal with unforeseen target dynamics and deformations.

Figure 1.1: Single object tracking (SOT) to automatically record athletes as they bike or ski down a slope. Localization must only take a fraction of the constrained computing time, as the remaining resources are required to adjust the pan-tilt-zoom (PTZ) camera to capture smooth videos. Tracking results – *i.e.* the current object location (blue and green rectangles, respectively) and the previous trajectory – are superimposed for better visualization.

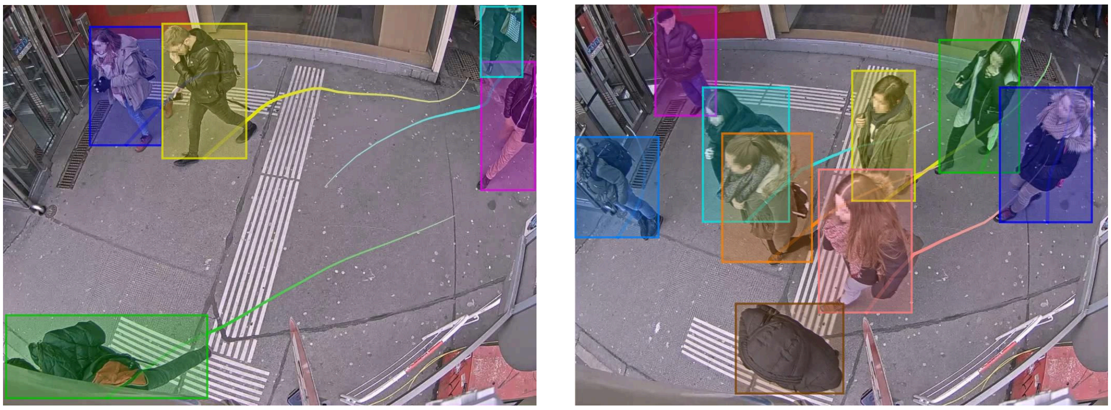
computing power and optical sensors – and the ubiquitous availability of computing devices contributed to the significant interest our research field received over the past decade. This interest is also reflected by the consistently large number of published tracking papers at major computer vision conferences alone, such as CVPR, ICCV and ECCV, with approximately 30–40 approaches annually.

Despite being a long-standing and widely studied research topic, visual tracking is far from being solved. The reason why we still have no Swiss Army knife for tracking is because tracking algorithms have to deal with considerable challenges, as illustrated in Figure 1.3. The key challenges can be summarized as follows:

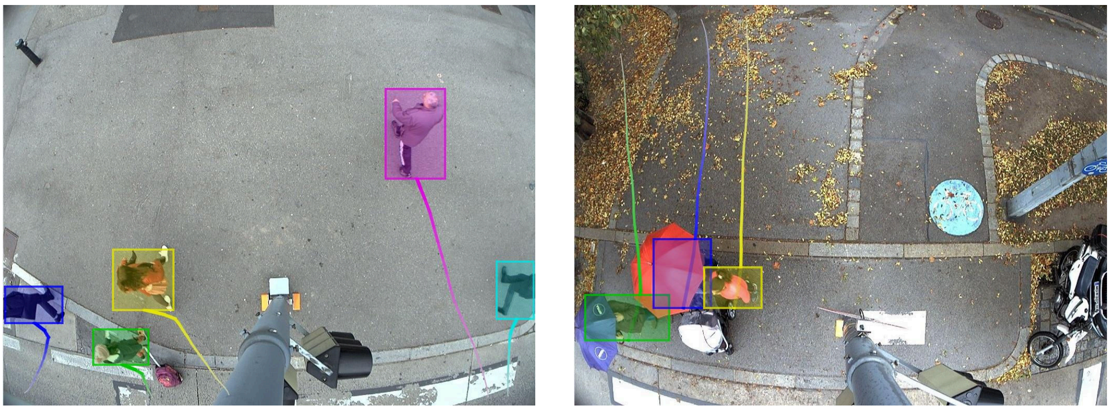
Appearance variations are caused by multiple factors, such as (rigid and non-rigid) object deformations, scale changes or illumination. On the one hand, a tracker must be robust against such kinds of varying object appearance, while on the other hand,



(a) Tracking from a typical surveillance viewpoint.

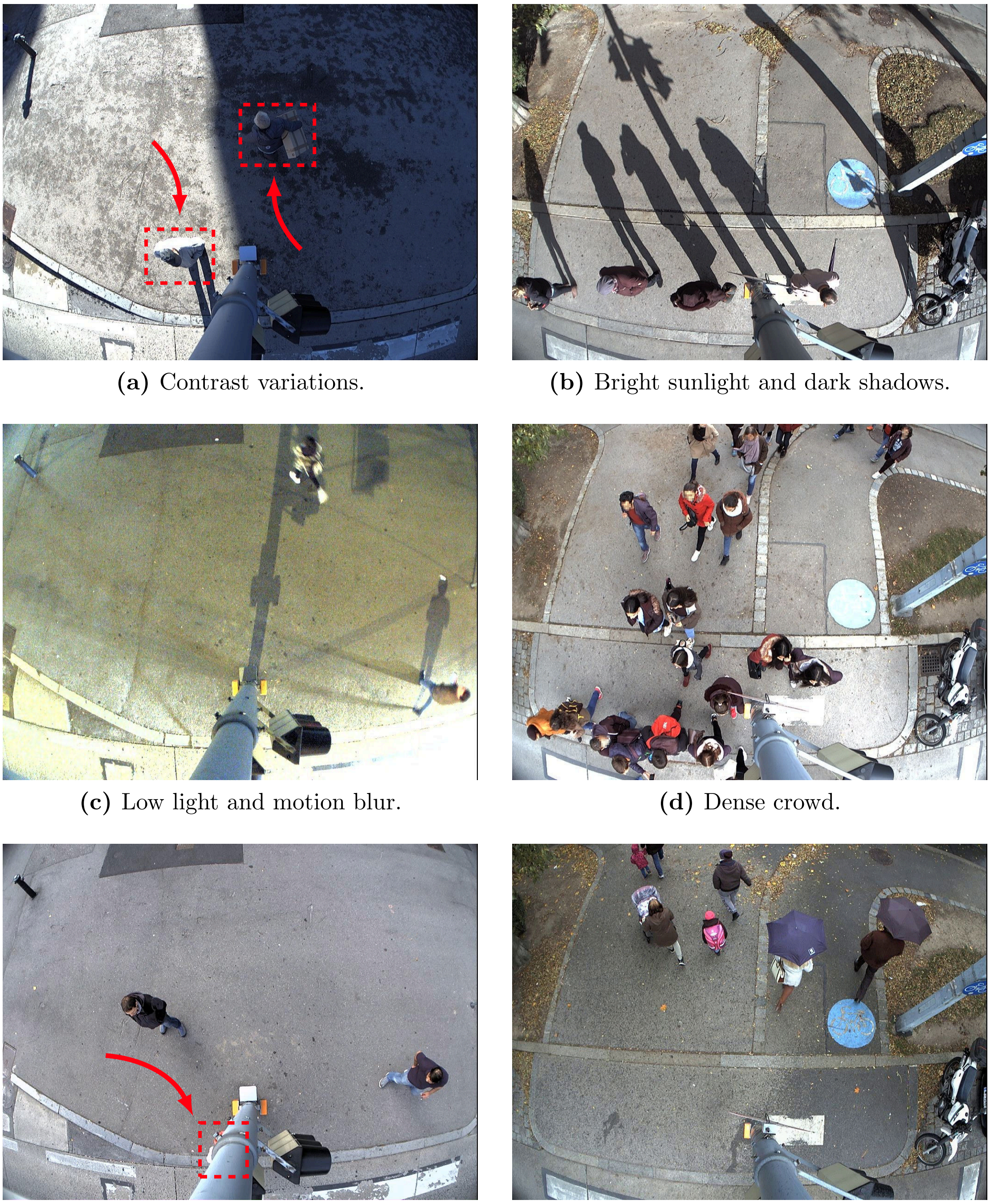


(b) Tracking from a slightly elevated viewpoint.



(c) Tracking from a notably elevated viewpoint.

Figure 1.2: Multiple object tracking (MOT) for intelligent pedestrian traffic lights from varying viewpoint elevations. The goal is to optimize the traffic flow by automatically triggering the traffic light for pedestrians who want to cross the road. This requires predictions of the pedestrians' intent and heavily relies on their dynamics and observed behavior. Additionally, note the significant appearance variations (scale and aspect ratio) due to the given viewpoints, which impedes both pedestrian detection and localization, *i.e.* reasoning about the object locations *w.r.t.* the (metric) ground plane. The superimposed, colored tracking results (rectangles and trajectories) correspond to the different object identities.



(a) Contrast variations. **(b)** Bright sunlight and dark shadows.
(c) Low light and motion blur. **(d)** Dense crowd.
(e) Occlusions. **(f)** Varying weather conditions.

Figure 1.3: Examples of difficult visual tracking scenarios highlighting two major challenges: *appearance variations* – due to (a), (b) interplay between sunlight and shadows or (c) degraded image quality at night – as well as *occlusions* – due to (d) high density crowds, (e) static obstacles in the field of view or (f) umbrellas.

it should be able to reliably distinguish multiple visually similar objects and identify failures once the tracker drifts away from the target.

Dynamics of the target, scene and camera. Depending on the velocity and continuity, motion can lead to blurry recordings or abrupt changes *w.r.t.* the predicted motion direction, thus impeding localization.

Illumination conditions play a crucial role for any computer vision-based system. While it is rather easy to record low quality images – *e.g.* via overexposure, underexposure, not paying attention to reflections of the light source or lens glare – capturing a scene at a sufficient quality level for robust automated analysis is a nontrivial task which requires a considerable amount of precaution and prior knowledge about the application domain and the intended environment.

Occlusions are either caused by objects and obstacles within the field of view (FOV), or the target (partially) occluding itself due to non-rigid deformations. The frequency, amount (*i.e.* full or partial occlusion) and duration of occlusions is heavily application and viewpoint dependent.

1.3 Contributions and Outline

A visual object tracker consists of the following two major components [87]:

Object representation and localization deals with modeling an object's appearance and generating hypotheses for its location. This is usually a *bottom-up* process, exploiting the observed (low-level) visual cues to infer hypotheses about the object state. The most important task of this component is to robustly cope with appearance variations.

Data association and filtering deals with the dynamics of the tracked object and incorporates contextual cues and prior knowledge. This is mostly a *top-down* process, evaluating and verifying the generated hypotheses to estimate the object trajectories.

Tracking approaches differ widely in the way these two components are combined and weighted, which is mostly driven by the particular application domain. This combination has a crucial effect on both the robustness and efficiency of the tracking approach. For example, tracking athletes from a PTZ camera, recall Fig. 1.1, relies more on object representation than motion, whereas tracking pedestrians and predicting their motion intent for automated traffic lights, recall Fig. 1.2 and 1.3, relies heavily on object dynamics.

Figure 1.4 illustrates the overall *visual tracking loop* and the interplay of these two major components. In this thesis, we advance the state-of-the-art by addressing both components. In particular, we make the following contributions to the processes of the visual tracking loop:

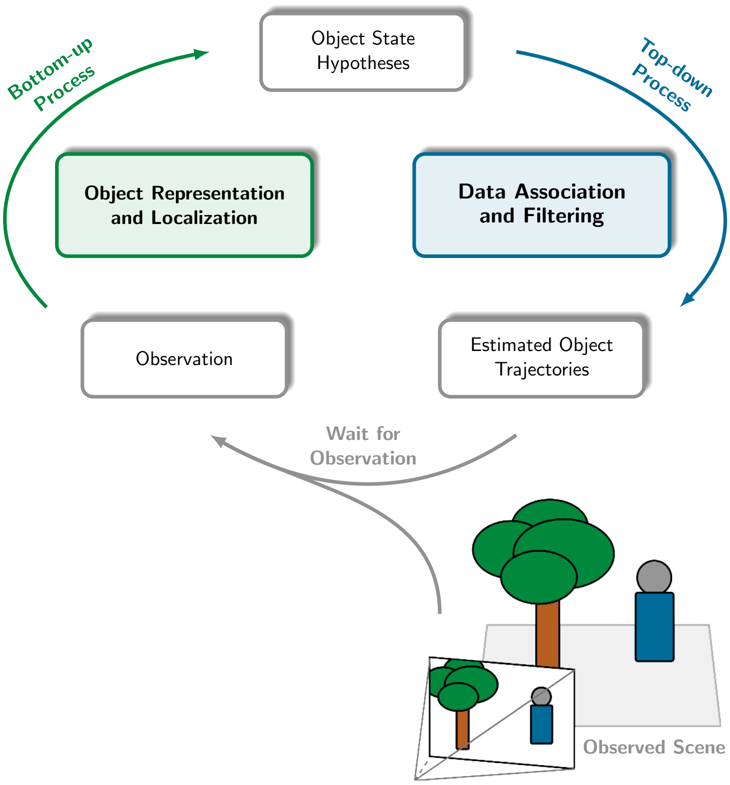


Figure 1.4: Overview of the visual tracking loop. Each tracker consists of two major components (green and blue boxes), namely (i) object representation and (ii) data association. This thesis contributes to both components by investigating (i) robust appearance-based models for object representation and (ii) physically plausible constraints for data association. Gray boxes indicate observed images (*i.e.* low-level input), intermediate state hypotheses (internal to the tracker), and trajectories (*i.e.* output of the tracker).

Bottom-up – we propose a context-aware object model which allows us to identify distracting regions in advance and suppress such regions during localization, leading to improved robustness. This significantly improves standard color-based trackers which otherwise would drift away from the object of interest towards such visually distracting regions.

Top-down – we propose a data association schema which exploits occlusion knowledge, physical plausibility and closed world assumptions. This enables robust linking of object hypotheses into trajectories.

The remainder of this thesis is structured as follows. First, we provide an overview on visual tracking approaches in Chapter 2. Second, we introduce our distractor-aware object model in Chapter 3. Third, we present occlusion geodesics for robust data association in Chapter 4. Finally, we provide detailed evaluations of the benefits and limitations of our contributions in Chapter 5 and conclude in Chapter 6.

2 Visual Object Tracking

Facts do not cease to exist because they are ignored.

— Aldous Leonard Huxley (Proper Studies)

Contents

2.1 Overview

This chapter introduces the notation used throughout this thesis and discusses prior work on visual object tracking. The following literature review should give the reader a brief overview over related methods and the state-of-the-art *w.r.t.* (generic) single object and (specialized) multiple object tracking. A more in-depth review of the most closely related methods is given in the following technical chapters, which present our contributions.

Due to the abundance of available visual tracking approaches, an exhaustive literature review is infeasible. This is also reflected by the multitude of survey papers published in recent years, *e.g.* [141, 149, 198, 242, 269, 313, 314, 347, 464, 469, 488]. Thus, we focus the following discussion on major research directions and approaches relevant to this thesis and refer the interested reader to the respective surveys for a broader overview.

2.2 Notation and Conventions

Throughout this thesis, we apply widely used mathematical conventions, as are also found in several books on pattern recognition [49], mathematical image processing [58], computer vision [179, 394, 408] and statistics [180, 336, 424]. In the following, we summarize the most important mathematical notations, as also listed in Table 2.1.

Scalar values are depicted in italic fonts, *e.g.* α or c_i. Matrices and vectors are represented in bold font, *e.g.* $\mathbf{M}$ or $\mathbf{v}$. Additionally, we use lowercase letters to denote 2D vectors, *e.g.* $\mathbf{v}$, and uppercase letters to denote 3D vectors, *e.g.* $\mathbf{X}$. Vector spaces are depicted in uppercase blackboard bold letters, *e.g.* $\mathbb{R}^2$. Functions, *i.e.* mappings between different vector spaces, are represented by uppercase italic letters, *e.g.* $H : \mathbb{R}^2 \rightarrow \mathbb{R}$. Probability measures are depicted by a lowercase italic $p(\cdot)$, *e.g.* to denote priors $p(X)$, joint probabilities $p(X, Y)$ or conditional probabilities $p(X \mid Y)$.

Although (discrete) images can be stored and processed as matrices, we apply the more formal convention that an image is a function. Thus, they are represented by uppercase italic letters and denote a mapping from a carrier set Ω to a color space $\mathcal{C}$, *i.e.* $I : \Omega \rightarrow \mathcal{C}$. The most common image representation in this thesis are discrete 2D images, *i.e.* $\Omega = \{1, \ldots, w_I\} \times \{1, \ldots, h_I\}$, where w_I and h_I denote the width and height of the image, respectively. As we most often deal with color images, the corresponding color space is usually 3D and either continuous, *i.e.* $\mathcal{C} = \mathbb{R}^3$, or discrete as in the case of 8-bit quantized images, *i.e.* $\mathcal{C} = \{0, \ldots, 2^8 - 1\}^3$.

We define image regions formally using set-builder notation. For example, an axis-aligned rectangle of size $w \times h$ centered at $\mathbf{c} = (c_x, c_y)^\top$ is defined as the set of pixels

Table 2.1: List of notations used in this thesis.

Entity	Notation
Scalar	$\alpha,\ c_i$
Vector in $\mathbb{R}^2$	$\mathbf{v} = (x, y)^\top$
Vector in $\mathbb{R}^3$	$\mathbf{X} = (x, y, z)^\top$
Matrix	$\mathbf{M} = \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix}$
Vector Space	$\mathbb{R}^3,\ \mathbb{Q}$
Function	$F : \mathbb{R}^3 \rightarrow \mathbb{R}^2$
Image	I, M
Pixel	$I(\mathbf{x}), I(x, y)$
Tuple	$R = (\mathbf{x}, w, h)^\top$
Probability measures	$p(X)\,,\ p(X, Y)\,,\ p(X \mid Y)$

$R = \left\{ \mathbf{x} = (x, y)^\top \mid |c_x - x| \leq {}^{w}\!/\!{}_{2} \wedge |c_y - y| \leq {}^{h}\!/\!{}_{2} \right\}$. To simplify and avoid cluttering the notation, we will also depict such regions by tuples, *e.g.* $R = (\mathbf{c},\ w,\ h)^\top$.

2.3 Single Object Tracking

We focus this overview of single object tracking (SOT) approaches on *generic* visual tracking using a single camera, *i.e.* *causal* trackers that do not apply pre-learned models or task-specific prior knowledge. In contrast to highly specialized tracking frameworks, *e.g.* as used to track the human eye [175, 270, 318, 470], generic approaches can be immediately applied to localize arbitrary objects without any adjustments. Due to this *genericity* property, such algorithms are particularly interesting for a large application domain. By not relying on pre-trained models, such trackers are also often referred to as *model-free* trackers[1]. Additionally, *causal* (also often denoted as *online*) trackers do not use any information from future frames, *i.e.* only previously observed frames can be exploited to infer the object location in the current frame. Thus, such trackers cause almost no delay between observation and state estimation. This property allows such approaches to be employed in time-critical applications, *e.g.* robotics or surveillance.

One of the most important components of each visual tracking approach is a sophisticated object model. From a probabilistic perspective, the goal of such a model is to correctly predict the class label y given some input features x, *i.e.* the problem is to find the conditional distribution $p(y \mid x)$. In visual tracking, we usually deal with a binary classification problem, *i.e.* $y \in \{0, 1\}$, where we want to distinguish image regions containing the object, *i.e.* $y = 1$, from the background, *i.e.* $y = 0$. The input features x we deal with are derived from the object representation, such as raw image intensities, more complex hand-crafted features – *e.g.* HOG [91], SIFT [284] or SURF [28] – or using features learned from data, *e.g.* via dictionary learning, feature embeddings, subspace representations or neural networks. After learning a suitable object model, the tracker evaluates the conditional probability to get a representative score – usually denoted *confidence*, *similarity*, *likelihood* or, loosely speaking, *probability* – which can subsequently be used to localize the object of interest throughout an image sequence.

There are two fundamentally different ways to establish a statistical model of the object of interest:

Generative methods learn a model of the joint probability $p(x, y)$, *i.e.* they model the *distribution* of the individual classes y. Predictions can then be obtained by exploiting the chain rule

$$p(x, y) = p(x \mid y)\ p(y)\,, \tag{2.1}$$

[1] Throughout this thesis, we try to avoid the term *model-free* whenever possible, as it may falsely convey that such a tracker does not employ a model at all.

and applying Bayes rule to compute the conditional probability

$$p(y \mid x) = \frac{p(x \mid y)\ p(y)}{p(x)} \tag{2.2}$$

$$\propto p(x \mid y)\ p(y)\,. \tag{2.3}$$

Discriminative methods model the conditional probability $p(y \mid x)$ directly by learning a mapping from the inputs x to the classes y, *i.e.* they learn the *boundary* between the classes.

Usually, discriminative approaches are considered to be superior to generative approaches. An intuitive reason for this belief is that discriminative approaches try to solve a simpler task by learning a direct mapping from x to y, whereas generative approaches make a detour by modeling the class distributions. Thus, generative approaches ignore the main principle of effective inference (at least from a small sample size), which – as stated by Vapnik [424] – is "*to solve the problem directly and never solve a more general problem*" [424, p. 12]. Furthermore, classifiers based on discriminative models usually have a lower asymptotic error compared to generative models. However, as shown by Ng and Jordan [328], generative classifiers (such as naïve Bayes) may converge to their (higher) asymptotic error much faster. This finding is especially important for generic visual tracking, where we have to learn a model from a very limited amount of training data, *i.e.* usually only a single annotated frame. Thus, trackers typically operate within the non-asymptotic case, where generative models may actually result in the better performance.

Since both discriminative and generative models have their advantages and disadvantages, several works try to combine the merits of both, *e.g.* [50, 252, 416], and also apply such hybrid models for visual tracking, *e.g.* [126, 445]. For more detailed discussions on the capabilities of generative and discriminative methods, we refer the interested reader to [50, 328, 478] or the excellent books on (statistical) learning [49, 180, 424].

In the following, we first categorize popular tracking algorithms by their prevailing tracking paradigm in Section 2.3.1. Afterwards, we review the state-of-the-art in generic single object tracking in Section 2.3.2.

2.3.1 Tracking Paradigms

Due to the vast research interest, a complete list of all proposed tracking paradigms or approaches is out of scope of this thesis – instead, we focus on seminal works and recent major approaches and categorize them by the underlying tracking paradigms. Note that most trackers can actually be assigned to multiple paradigms – for example, correlation filters which employ part-based models, *e.g.* DPCF [287], convolutional neural network-based approaches which learn correlation filters, *e.g.* CREST [393], Siamese network-based approaches which apply policy learning, *e.g.* EAST [203], part-based approaches which rely on color models, *e.g.* BHT [325, 326], or segmentation-based approaches which also rely on the generalized Hough transform [26] and use a part-based model, *e.g.* HoughTrack [156, 157]. To avoid a highly redundant listing, we only categorize trackers by their prevailing paradigm. Figure 2.1 summarizes the underlying paradigms of top-performing trackers on recent benchmark evaluations. Note that we provide a more detailed summary of color-based and context-aware approaches in Chapter 3, where we present our distractor-aware tracker.

Correlation Filter-based Approaches. Introduced in the seminal work on synthetic discriminant functions (SDF) by Hester and Casasent [191], correlation techniques are widely used within the pattern recognition and computer vision community [246, 294]. Initially, correlation filters were mostly used for low-level vision tasks - especially for feature point tracking (*i.e.* estimating the motion between images) and matching (*i.e.* image registration), *e.g.* [8, 25, 285, 383, 415] – as well as object tracking, *e.g.* [51, 172, 214]. The main principle is to learn a filter – usually in the frequency domain – that generates a desired response when correlated with an input signal. For visual tracking, the desired response is usually a peak at the object center, typically modeled by a 2D Gaussian function.

Recently, the interest in correlation filters increased significantly due to the notable work by Bolme *et al.* [54, 55] which addressed previous drawbacks and demonstrated robustness to challenging illumination conditions and partial occlusions at impressive frame rates. Another notable extension is the combination with circulant matrices by Henriques *et al.* [188], which enabled efficient learning via kernel ridge regression in the Fourier domain. These initial approaches have consecutively been improved by the tracking community, *e.g.* by incorporating more complex multi-channel features [93, 144, 189] or global context [316], nonlinear kernels [189], long-term memory components [291], sophisticated learning models [48, 96–98], improving scale adaptation [92, 94, 99, 271], handling non-rigid deformations [42], including part-based representations [279, 287, 405], and introducing regularization techniques to mitigate boundary effects [94, 145, 286].

Deep Learning-based Approaches. Recently, features learned with convolutional neural networks (CNNs) have shown excellent performance in large-scale object recognition benchmarks, *e.g.* [244]. Furthermore, these deep learning approaches have significantly improved the state-of-the-art in many computer vision research fields, such as object detection and recognition [154, 357–359, 362]. Motivated by their success, several approaches explore the benefits of deep learning for visual tracking, which yielded impressive results but also significantly increased the computational requirements. Several works rely on the highly discriminative deep features, *e.g.* [194, 393, 430, 431], which can be pre-trained in an offline stage and adopted to the target's object class at runtime, *e.g.* [319]. More recently, recurrent neural networks (RNN) and Siamese networks – which are basically unrolled RNNs – have been widely adopted, *e.g.* [43, 89, 124, 125, 159, 184,

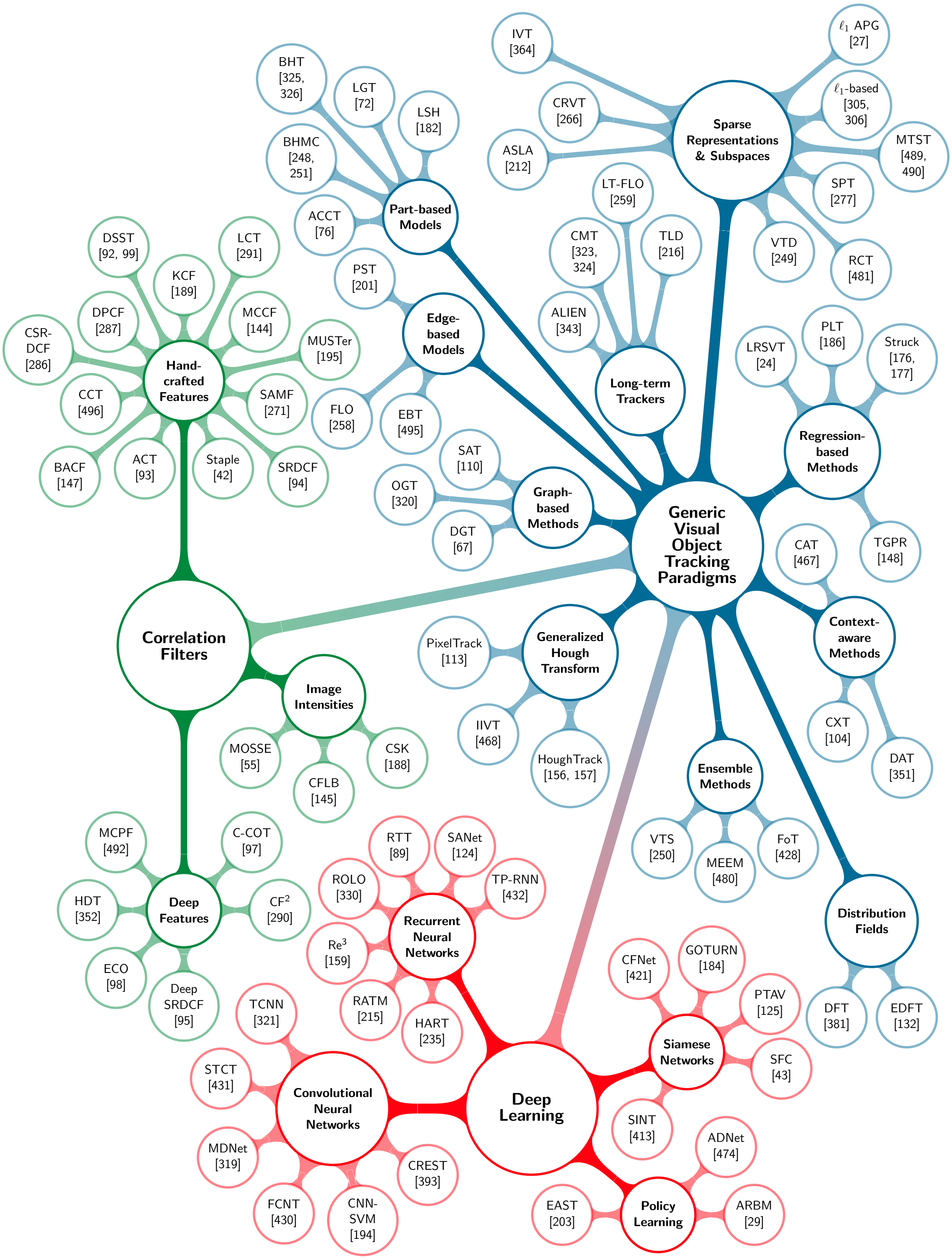


Figure 2.1: Overview of recent trends and research directions for visual tracking of a single object. We cluster trackers which are among the top-performers of recent benchmarks [133, 134, 146, 238–241, 243, 265, 274, 315, 386, 392, 448, 449] based on their prevailing tracking paradigm. To highlight the significant recent research interest in correlation filters and neural network-based approaches, these are visually separated from other paradigms.

215, 235, 330, 413, 421, 432]. Another interesting line of research is to apply ideas of re-reinforcement learning to train neural networks that regress transformations, such as translation or shifting of the object hypothesis, *e.g.* [29, 203, 474].

Distribution Field-based Approaches. To overcome the limitations of traditional kernel-based (*i.e.* histogram-based) tracking approaches – namely that applying the kernel leads to loss of spatial structure – distribution fields have been employed, *e.g.* [132, 381]. In essence, these approaches compare multi-channel object representations, which are local histograms smoothed at different scales. Distribution fields can be used to mitigate the effect of partial occlusions and misalignment during tracking. Prior to these approaches, distribution fields have mostly been used in the context of background subtraction to detect moving objects in image sequences, *e.g.* [115, 397, 398].

Edge-based Approaches. Besides color information, edges are a powerful visual cue to locate an object of interest. Most often, trackers employ hand-crafted edge-based features, such as HOG [91], *e.g.* [42, 92, 189, 271, 286]. Edge cues have been shown to be superior to plain color-based representations especially when tracking texture-less or feature-less objects under challenging illumination conditions, *e.g.* [258, 259]. Some approaches additionally rely on edge cues to generate object proposals, *e.g.* [201, 495].

Efficient Representation-based Approaches. Several tracking approaches draw their inspiration from the human visual system to enable efficient models. Sparse, reduced or compressed object representations are well suited for such biologically inspired appearance modeling tasks [488]. These representations can be efficiently compared and stored – due to their sparsity – and are beneficial when dealing with significant appearance variations, *e.g.* caused by changing illumination – due to the robust basis functions. Sparse representations can be obtained by leveraging sparse coding techniques, *e.g.* [27, 212, 277, 305, 306, 489, 490], compressed sensing, *e.g.* [266, 481], or subspace learning methods, *e.g.* [51, 172, 249, 281, 301, 364].

Ensemble Methods. Combinations of features, trackers, and machine learning techniques have been widely explored for visual tracking. The goal of all these works is to improve generalizability and robustness by fusing the output of multiple estimators over a single estimator. To this end, machine learning ensembles have been successfully applied both with averaging methods[2], such as *random forests* and *decision trees*, *e.g.* [262, 455], as well as boosting-based methods, *e.g.* [17, 18, 162, 163, 165]. Another line of research combines either multiple feature cues – to rely on the most discriminative cue for the given

[2]Machine learning ensembles can be divided into two classes: (i) *averaging methods* independently train multiple estimators and then average their predictions; and (ii) *boosting methods* train several estimators sequentially with the goal to reduce the bias of the combined estimator.

sequence challenges, *e.g.* [84, 111] – or multiple trackers – to rely on the most confident or most reliable tracker, *e.g.* [250, 371, 428, 480].

Part-based Approaches. To obtain more robust object representations, several approaches employ part-based models which notably mitigate the challenges caused by partial occlusions or non-rigid object deformations. Typically, such trackers work on image patches, *e.g.* [1, 7, 71, 72, 75, 76, 113, 114, 156, 157, 182, 211, 248, 251, 272, 320, 325, 326, 484]. Instead of using regular image patches to denote the parts, some approaches either rely on segmented superpixels, *e.g.* [67, 110], or interest points, *e.g.* [165, 299, 323, 324, 343].

Regression-based Approaches. Among the top-performers of recent tracking benchmarks is a consistently large group of regression-based approaches. Such trackers formulate tracking as the regression of image displacements from image intensities or other features. For example, this has successfully been addressed via structured output SVMs [176, 177, 186, 495], ranking SVMs [24], relevance vector machines (RVMs) [442], logistic regression [433] or Gaussian process regression [148]. Note that correlation filters, such as [92, 93, 188, 189], also formulate tracking – in particular learning of the discriminative filter – as a ridge regression problem.

2.3.2 State-of-the-Art

Most recent approaches focused on (i) improving correlation filters – by incorporating more complex and discriminative feature cues or better regularization and drift prevention – and (ii) exploring deep learning methods for visual tracking. These two tracking paradigms significantly advanced the state-of-the-art over the past few years and are consistently among the top 3 contestants of recent tracking benchmarks, such as the Visual Object Tracking (VOT) challenges [133, 134, 238–243].

Over the past four years, the top ranks of the VOT challenges were dominated by (i) correlation filters, *i.e.* CSR-DCF [286], DSST [92], KCF [189], SAMF [271] and Staple [42]; (ii) deep learning-based approaches, *i.e.* MDNet [319] and TCNN [321]; and (iii) combinations of both, *i.e.* using convolutional features within the correlation filter framework: C-COT [97], CFCF [170] and DeepSRDCF [94, 95]. There are, however, a few notable exceptions – namely (iv) trackers based on structured output SVMs, *i.e.* EBT [495] and PLT [186]; (v) an ensemble of trackers, *i.e.* FoT [428]; and (vi) a tracker relying on distribution fields, *i.e.* EDFT [132].

Note that we only discussed approaches for generic object tracking from standard RGB color sequences so far, which is the most common image modality we have to deal with. On the contrary, visual tracking from different image modalities, such as thermal infrared (TIR), has received significantly less attention. However, such non-typical image modalities are especially useful for visual surveillance, autonomous vehicles or robot vision applications, due to their robustness to illumination changes, the ability to see in total darkness and reduced privacy invasion. There have been several TIR-based tracking challenges recently, *i.e.* VOT-TIR [133, 134, 243] and PETS [267, 339, 340]. The top-performing methods on these datasets are mostly based on structured output SVMs and rely on edge proposals, *i.e.* EBT [495], DSLT [473] and PST [201]. Only few correlation filter and deep learning-based methods have been adapted for the thermal infrared imagery so far. However, two of them already are amongst the top contestants, namely SRDCF [94] and TCNN [321].

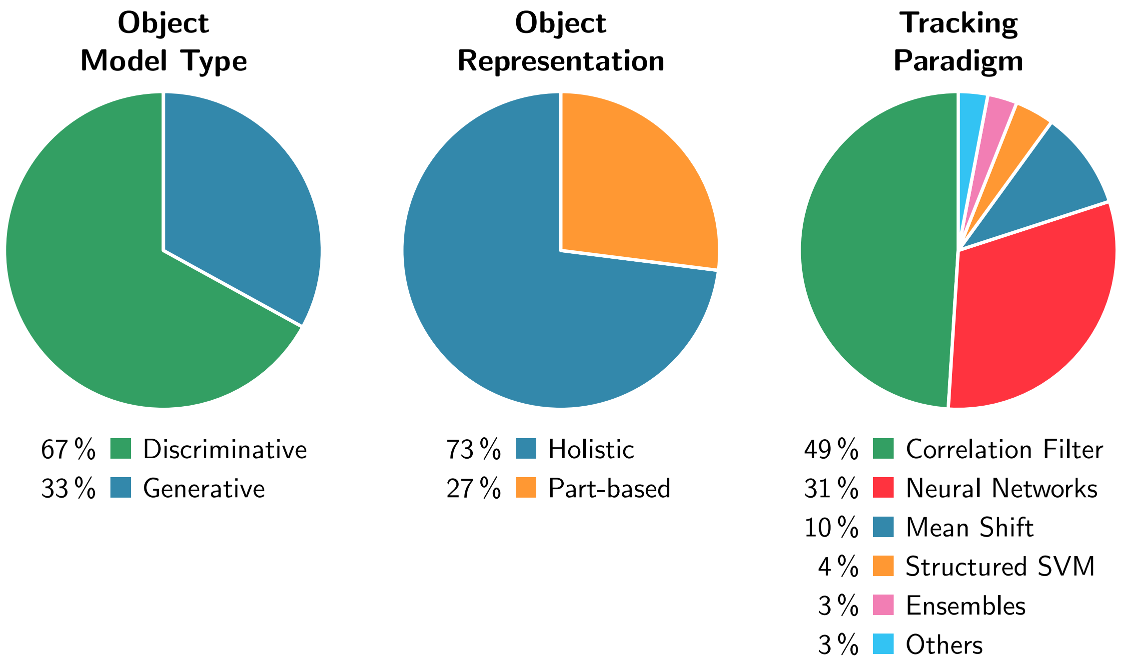
Considering the tracking benchmarks over the past three years, we can observe interesting paradigm changes. Figure 2.2 analyzes approaches which participated in the VOT challenges in 2014 [239] and 2017 [243]. The model-related comparison (leftmost and middle charts) shows a notable shift from generative to discriminative models, as well as an increase of holistic representations. These two trends can easily be explained by the rise of both correlation filters and deep learning-based trackers (see rightmost charts), as these are discriminative approaches where the majority relies on holistic representations instead of explicitly modeling parts of an object. In fact, while 2014 half of all trackers tested at VOT relied on diverse techniques (depicted as *others* within Fig. 2.2), *i.e.* boosting, generalized Hough transform, graph-based models, interest point matching or particle filter frameworks, these account for less than 3 % of all tested trackers in 2017. In contrast to this development, mean shift-based trackers seem to be the most attractive and reliable "traditional" tracking paradigm, with a constant share of approximately $^1/_{10}$ of all tested trackers over the past few years.

Although state-of-the-art approaches achieve remarkable accuracy and robustness, their additional model complexity, however, comes at the price of a highly increased demand of computational resources. Thus, most top-performing approaches are not suitable for time-critical systems. However, a recent study [146] showed that earlier correlation filter-based approaches, such as [42, 189], easily outperform more complex approaches – both deep learning-based methods and complex correlation filters – if the video sequences are recorded at a higher frame rate. Although not too surprising, this finding has practical importance: when implementing a real-world tracking system, special attention should be paid to improve the inputs, *i.e.* ensuring sufficient image quality and capturing rate, instead of prematurely inventing more complex approaches to cope with issues arising from an over-hastily chosen capturing system[3].

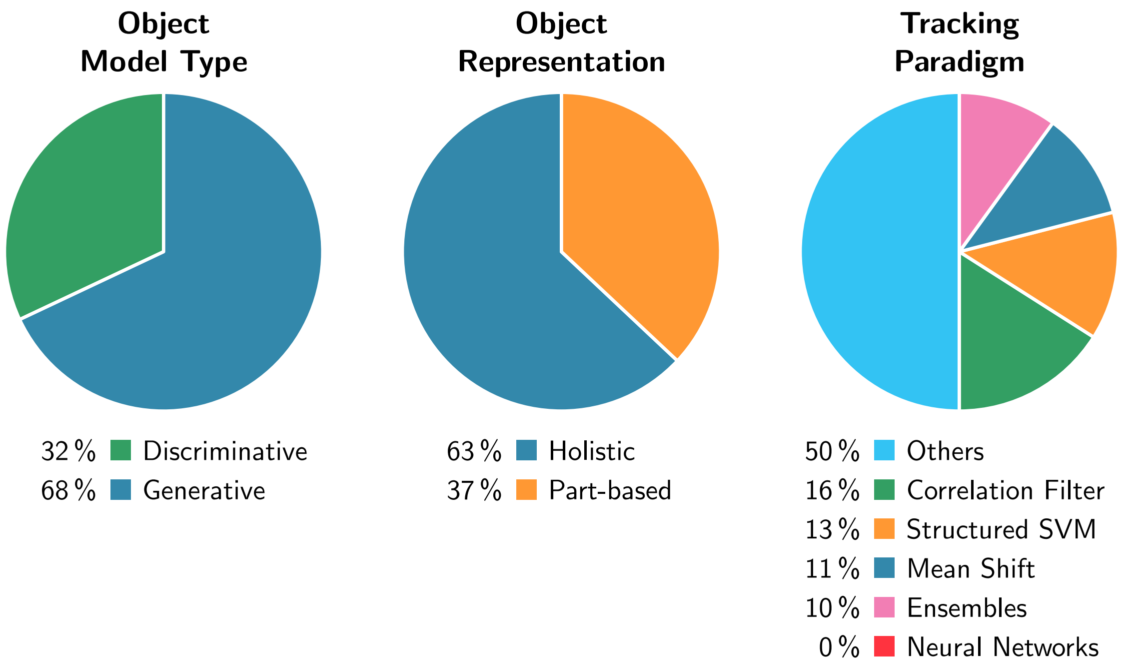
2.4 Multiple Object Tracking

The second major research field in visual tracking is multiple object tracking (MOT), also often referred to as multiple target tracking (MTT). As its name implies, the task is to locate multiple objects throughout an image sequence, maintain their identities despite

[3] Actually, every engineer should know the computer vision mantra by heart: *Garbage in, garbage out.*



(a) Characteristics of 51 trackers tested at the VOT'17 challenge [243].



(b) Characteristics of 38 trackers tested at the VOT'14 challenge [239].

Figure 2.2: Characterization of recent trends in visual tracking. A comparison of trackers tested at (a) the VOT'17 challenge and (b) the VOT'14 challenge reveals interesting regime changes over the past three years. In particular, note the significant changes *w.r.t.* the underlying model type (charts on the left) and distribution of prevailing tracking paradigms (charts on the right).

varying numbers of objects and report each individual trajectory for analysis. Typically, MOT deals with a single object class of interest, such as animals [45, 228, 288, 303, 450], cells or subcellular structures [79, 158, 385, 417], vehicles [44, 234, 354, 361] or, predominantly, humans – for which the major application domain is usually visual surveillance [9, 36, 39, 60, 66, 118, 187, 218, 226, 261, 308, 376, 378, 447, 461, 472] or in the context of sports and motion analysis [31, 32, 205, 236, 302, 329, 349, 425]. In fact, according to a recent study [289], more than 70 % of the MOT research effort is focused on pedestrian tracking alone. Some MOT approaches can also be adapted for single object tracking, *e.g.* by simultaneously tracking all (sub-)parts of an object [112, 288, 484, 485]. However, the vast majority focuses on tracking multiple individuals of the same object class, which we will address in the following review.

Our MOT contributions are also motivated from typical pedestrian tracking applications. In particular, we focus on analyzing pedestrian motion because of two major reasons. First, visual surveillance scenarios provide a challenging testbed for MOT algorithms: (i) humans are (mostly) non-rigid objects resulting in considerable shape deformations, usually of their extremities; (ii) typical surveillance setups, *i.e.* outdoor scenarios captured at long-range fields of view (FOVs), result in rather low resolution image data which impedes appearance modeling to distinguish pedestrians; additionally, (iii) pedestrians tend to wear similarly colored clothing, preferably shades of dark, which in combination with (iv) interactions between people makes it rather difficult to maintain the correct trajectory identities; and finally, (v) surveillance scenarios typically capture rather crowded scenes which lead to frequent occlusions. Second, tracking humans is a crucial component of many computer vision-based real-world applications, with a broad range of application domains, such as action recognition [2], human behavior analysis [69, 199], crowd analysis and intelligent environments [477] or visual surveillance [435].

In the following, we first provide a categorization of multiple object tracking approaches in Section 2.4.1. Then, we review the state-of-the-art according to recent benchmark evaluations [256, 309] in Section 2.4.2.

2.4.1 Categorization

Similar to SOT, there are multiple ways to categorize MOT approaches. We focus on three key aspects to group the vast literature into more easily digestible parts, as also illustrated in Figure 2.3.

Classification by Tracker Initialization. Most MOT approaches rely on the *tracking-by-detection* paradigm and apply a detector to generate object hypotheses which are then linked into consistent trajectories. Therefore, this group is also known as *detection-based* trackers and can be further divided into two sub-groups, namely (i) approaches that rely on *motion detection*, *i.e.* background modeling and (moving) foreground estimation, and (ii) approaches that apply pre-trained *object detectors*. Earlier approaches mostly relied

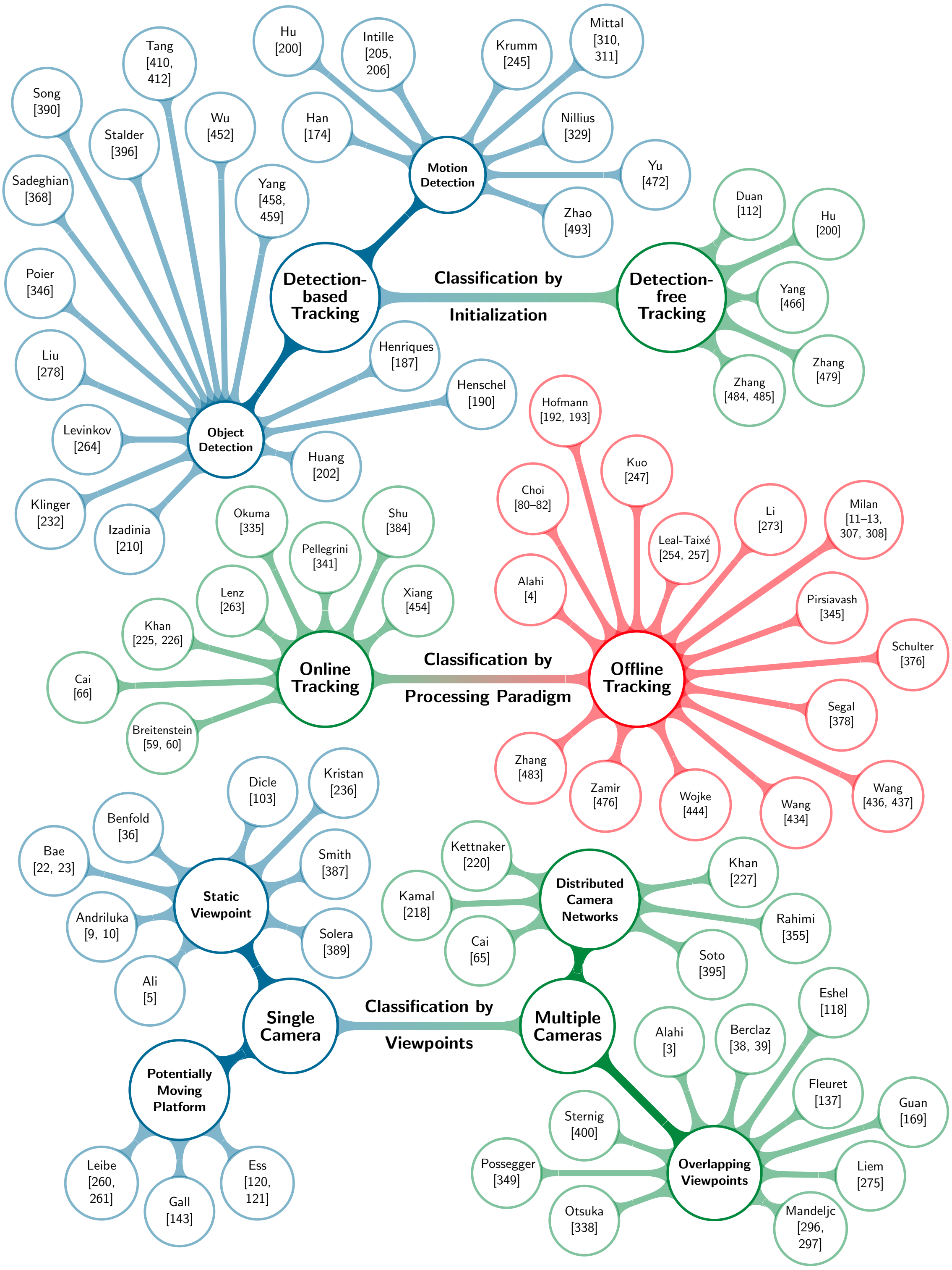


Figure 2.3: Overview of the past decade of MOT research. We focus on three categorization schemes based on the key aspects of multiple object tracking, *i.e.* initialization (top), processing paradigm (middle) and capture setup (bottom). Due to the abundance of MOT algorithms, we list only a few representative approaches for each category.

on motion detection, *i.e.* segmenting moving objects via background subtraction or frame differencing to yield object hypotheses, *e.g.* [174, 178, 205, 206, 245, 310, 311, 493]. Due to the typically static camera setup for pedestrian tracking applications, these techniques are still widely used today. For example, fitting pedestrian shapes to the segmented moving regions allows to derive probabilistic occupancy measures [3, 137] which indicate likely object locations. These occupancy measures are widely used, *e.g.* to derive edge weights for graph-based MOT approaches, such as [31, 32, 39]. On the other hand, geometrically fusing the estimated moving foreground regions across multiple (calibrated) viewpoints enables accurate 3D localization of multiple objects, *e.g.* [169, 275, 349].

The majority of detection-based MOT approaches, however, relies on object detectors. This strategy has first been explored for SOT, *e.g.* [15, 16, 161, 162], and adopted for MOT shortly after, *e.g.* [9, 260, 261, 335, 447]. As the tracking performance heavily depends on the quality of the detector, several works leverage the synergy between tracking and detection by including object priors for the detection step, derived from the object dynamics within the tracking step, *e.g.* [9, 260, 261, 396, 452]. The prevailing strategy, however, is the black box approach, *i.e.* using an off-the-shelf detector to generate object hypotheses which are then linked together into consistent target trajectories, *e.g.* [187, 190, 202, 210, 264, 278, 346, 390, 410]. To avoid discarding hypotheses prematurely in the detection step – which usually happens during non-maxima suppression or by applying a threshold on the detection confidence – some tracking approaches directly exploit the detection confidence, densely sampled across the input image, *e.g.* [59, 60].

In contrast to detection-based approaches, so-called *detection-free* trackers require manual initialization, *e.g.* by a human operator. This group contains rather few approaches, *e.g.* [112, 200, 466, 479, 484, 485]. Typically, these approaches assume that all objects are already visible within the first frame of an image sequence and that their number stays fixed. Due to the manual initialization, these approaches are seldom used for real-world applications where automation and usability is a key factor.

Classification by Processing Paradigm. MOT frameworks can also be grouped into *online*, *i.e. causal*, and *offline* approaches. Online methods, *e.g.* [60, 66, 225, 226, 263, 335, 341, 384, 454], infer the object states solely based on observations up to the current frame. Offline methods, on the other hand, *e.g.* [4, 11, 80, 192, 247, 254, 257, 273, 308, 345, 376, 378, 434, 437, 444, 476, 483], either process the whole image sequence at once, or optimize trajectory assignments over sliding temporal windows, *i.e.* process a batch of frames at once. By jointly analyzing all observations collected from a larger frame batch, offline approaches typically yield more robust tracking results but also cause a delay in reporting these results, which constrains their use for time-critical applications. Note that in contrast to most of the MOT literature, we consider a stricter definition of *online*, namely that already reported trajectories – *i.e.* all estimated object locations up to the current frame – cannot be changed anymore. Thus, we classify all batch-based trackers as offline approaches, even if they only optimize trajectory assignments over the past few seconds, such as [36, 137].

The distinction between online and offline approaches tightly correlates with the underlying inference paradigm, *i.e.* whether MOT is approached from a *probabilistic* perspective or a *deterministic optimization* perspective. Since online trackers have to deal with a significantly higher uncertainty, these approaches typically rely on probabilistic inference and represent the states of objects as a probability distribution. This allows to model the inherent uncertainty of causal MOT – *i.e.* without waiting for observations from future frames, it is virtually impossible to decide whether a previously tracked object is currently occluded, actually disappeared or the detector failed for another reason. Such approaches usually rely on sequential filtering techniques, *e.g.* multiple hypotheses tracking (MHT) [360], joint probabilistic data association filters (JPDAF) [138, 139], Kalman filters [217] or, most commonly, Monte Carlo sampling-based models which became increasingly popular for visual tracking with the introduction of particle filter frameworks, independently developed by Isard and Blake [52, 207, 208], Gordon and Salmond [160] and Kitagawa [230].

Offline approaches, on the other hand, cast tracking as a deterministic optimization problem, trying to find the optimal trajectory assignment for all object detections within the corresponding frame batch. To this end, the assignment problem is usually formulated as a graph which allows a variety of suitable solutions. A commonly used representation is that object detections (or already identified, shorter trajectories) define the nodes of the graph, whereas both temporal and appearance cues are leveraged to derive edge connections and the corresponding weights. The most popular techniques to optimize for the final trajectories are: (i) *bipartite graph matching* – either relying on greedy assignments, *e.g.* [447], or the optimal assignment via the Hungarian algorithm [317], *e.g.* [202, 456]; (ii) *dynamic programming*-based approaches – which try to find the K-shortest paths, *e.g.* [39], rely on quadratic Boolean programming, *e.g.* [101, 260], solve the combinatorial set cover problem, *e.g.* [451], or apply subgraph multicuts, *e.g.* [411, 412]; (iii) approaches which solve for the *minimum cost network flow* within a directed graph, *e.g.* [32, 64, 82, 102, 263, 345, 451, 483]; (iv) apply a *conditional random field* (CRF) model, *e.g.* [307, 460, 462]; or (v) solve for the *maximum-weight independent set* of an attributed graph, *e.g.* [61, 382].

Although both, the correlation between offline and optimization-based approaches, as well as online and probabilistic inference-based approaches holds true for the majority of MOT algorithms, there are notable exceptions. For example, causal trackers which rely on bipartite graph matching – *i.e.* assigning current object detections to previously observed trajectories, *e.g.* [59, 60, 384] – or offline trackers which employ Monte Carlo sampling to efficiently reduce the solution space of the optimization problem, *e.g.* [353, 472].

Classification by Viewpoints. Based on the employed camera setup, we can distinguish *monocular* and *multi-camera* MOT approaches. Most of the research effort has been spent on monocular setups as the majority of publicly available datasets is captured from a single camera. This is mainly due to the notable efforts required to properly record a scene from multiple viewpoints simultaneously, namely synchronizing the video streams and calibrating the cameras, both *w.r.t.* their intrinsics and extrinsics. Additionally, multi-camera datasets require manual ground truth annotations for (at least a selection of) all viewpoints, which is a tedious task. However, if carefully calibrated – *e.g.* as in the APIDIS [77], ICG Lab6 [349] or MVL Lab5 [297] datasets – the multiple viewpoints can be used to accurately track objects by leveraging 3D structural information, *e.g.* [169, 275, 349, 379], or homography constraints, *e.g.* [225, 226, 346, 400]. However, the most widely used multi-camera datasets, *i.e.* PETS'09 [136] and the EPFL sequences [39, 137], do not contain fully calibrated cameras, *i.e.* the EPFL sequences only deliver homographies between the image plane and the ground plane, whereas the PETS'09 calibrations are too inaccurate to leverage multi-view 3D structure. Nevertheless, object hypotheses can still be fused across these views to robustly track pedestrians in 2D, either on the ground plane or in image coordinates, *e.g.* [31, 32, 38, 39, 137, 193, 437]

Another line of research focuses on tracking within distributed camera networks, *i.e.* leveraging multiple but non-overlapping (or at least only partially overlapping) FOVs, *e.g.* [65, 218, 220, 227, 344, 355, 395]. Such approaches need to explicitly hand over object identities between neighboring camera sensors in wide area surveillance applications. This is a particularly challenging task for non-overlapping viewpoints, due to potentially different illumination conditions or different viewing angles.

The majority of MOT approaches relies on a single camera setup. This group can further be subdivided whether they require a static camera, *e.g.* [5, 22, 23, 103, 236, 387, 389], or are able to track from a moving platform, *e.g.* [120, 121, 143, 260, 261, 312]. Widely used static camera datasets are the TownCentre [35, 36], PETS'09 [136] (by using only a single viewpoint) and the TUD sequences [9, 10], whereas most evaluations for tracking on moving camera platforms are conducted either on the ETH sequences [119, 120] or the KITTI dataset [150].

2.4.2 State-of-the-Art

Over the past few years, MOT research focused mostly on offline or batch-processing methods due to their robustness and simplicity. Causal tracking, although required for real-world applications, has received significantly less attention from the visual tracking community. Interestingly, though, the top performing methods on the MOT'15 benchmark [256] – a benchmark initiative which aims at evaluating MOT approaches on publicly available datasets, including ETH, PETS, TownCentre and TUD – are causal trackers. In particular, online deep learning-based approaches, *i.e.* [282, 368], lead the rankings on the 2D subset (*i.e.* single camera sequences where tracking results are reported in image coordinates). These approaches either combine a state-of-the-art object detector (*i.e.* RCNN [154]) with deep appearance models and thus, leverage the powerful CNN features, *i.e.* [282], or pose tracking as a re-identification problem, leveraging deep metric learning and RNNs, *i.e.* [368]. The 3D subset (*i.e.* multi-camera sequences where tracking results are reported in 3D world coordinates), on the other hand, is led by a dynamic Bayesian network (DBN)-based approach, *i.e.* [232], which employs instance-specific online random forests [369]. These notable exceptions on MOT'15 are tightly followed by offline graph-based approaches which rely either on multicuts, *i.e.* [221], or network flow formulations, *i.e.* [444].

Considering the current rankings of the larger follow-up benchmarks, *i.e.* MOT'16 and MOT'17 [309], however, we can see a clear domination of offline approaches over online approaches. In particular, the top performing methods on both benchmarks uniformly cast MOT as a graph problem. The solution is then obtained by either seeking optimal multicuts on the trajectory-detections graph, *i.e.* [221, 412], solving an approximation of the weighted graph labeling problem, *i.e.* [190], jointly decomposing the graph and labeling its nodes, *i.e.* [264], or casting the trajectory optimization problem in a classical multiple hypotheses tracking framework, *i.e.* [229].

The reason why, in contrast to SOT, there are only few deep learning-based approaches for MOT up to now, is that tackling the key problems of MOT – *i.e.* locating an unknown (and even worse: varying) number of objects and maintaining their identities – is considerably difficult to model using fixed neural network architectures. Additionally, MOT problems require a stronger focus on target dynamics due to the usually less discriminative appearance cues (as seemingly all pedestrians tend to wear dark clothing). Robustly modeling dynamics, however, is a challenging task for recurrent neural networks as the gradients can easily explode or vanish when learning dependencies over long time windows [37]. Thus, recent approaches use rather short memory horizons of approximately 5–8 steps, *e.g.* [4, 368], which in typical surveillance camera footage corresponds to less than half a second and consequently impedes both handling of long-term occlusions as well as deducing long-term predictions, such as a pedestrian's intent, *i.e.* to which point in the scene she is headed towards.

In contrast to both optimization-based approaches and trackers which learn the object dynamics over time, we present a robust association schema for online MOT. In particular, we show how to exploit occlusion reasoning in combination with simple scene priors to guide data association in a bipartite graph matching formulation. For more details, please refer to Chapter 4.

3 Distractor-Awareness for Appearance-Based Tracking

Sooner or later, everything old is new again.

— Stephen Edwin King (The Colorado Kid)

Contents

3.1 Motivation

This chapter investigates contextual cues related to the object appearance itself. In particular, we show how to exploit appearance-based models to robustify visual tracking in the presence of distracting visual cues. We will focus on generic single object tracking approaches which are employed for scenarios where neither object class-specific prior knowledge, nor pre-learned object models are available. Although some application domains allow us to incorporate strong assumptions about the target – for example, tracking pedestrians in surveillance scenarios [60, 349, 350, 384] – it is often desirable to build a generic tracker which can readily be used for arbitrary object classes. Instead of applying pre-learned object models, such a generic tracker must learn a representative object model given a single input frame with a (possibly noisy) initial object annotation, *e.g.* an axis-aligned bounding box. Despite significant progress in recent years, creating such a generic object tracker is still a rather challenging task due to real-world phenomena, such as illumination changes, background clutter, blur caused by fast object or camera motion, abrupt motion changes, non-rigid object deformations and occlusions.

Throughout the early stages of visual tracking, color histograms, *e.g.* [87, 331, 332, 342], were a common method for appearance description. However, over the last decade, such models have widely been replaced by more complex and well engineered features, such as HOG [91], *e.g.* [92, 98, 99, 189], or more complex color representations, such as color attributes [423], *e.g.* [93, 98, 423]. Moreover, the recent research focus has shifted to trackers which learn robust data-driven models, either via correlation filters, *e.g.* [42, 55, 92, 99, 188, 405] or convolutional neural networks (CNNs), *e.g.* [97, 98, 171, 184, 194, 203, 290, 319, 321, 393, 414, 474]. Such trackers have been shown to achieve excellent performance on recent benchmark evaluations, whereas trackers based on standard color models yield inferior performance.

In particular, considering the results of recent benchmark evaluations – such as VOT'13 [238], VOT'14 [239] or ALOV++ [386] – color-based trackers often tend to drift towards regions which exhibit a similar appearance as the currently tracked target. Consequently, the state-of-the-art has focused on more complex models, trading computational efficiency for more accurate results and thus, most often sacrifice real-time capability. In contrast to this development, we argue that trackers based on simpler, yet very efficient, standard color representations can still achieve state-of-the-art performance if they properly address two key requirements for robust visual tracking:

- The underlying object model must be able to distinguish the object of interest from its immediate surroundings, both efficiently and effectively.
- A robust tracking algorithm should identify potentially distracting regions in advance and counteract appropriately to prevent drifting, once such distracting regions come close to the object of interest.

To address these key requirements, we exploit the observation that color-based trackers tend to drift towards nearby regions with similar visual appearance. By relying on an efficient color-based object representation, we can identify potentially distracting regions in advance – several frames before a standard color-based tracker would drift away – and counteract in time by adapting the object representation such that the model response is suppressed for these distractors. Using such an adaptive color model, we can significantly reduce the drifting problem, which yields robust and reliable tracking results, as illustrated in Figure 3.1. Due to the favorable simplicity of our representation, it is also well suited for time-critical applications such as surveillance and robotics.

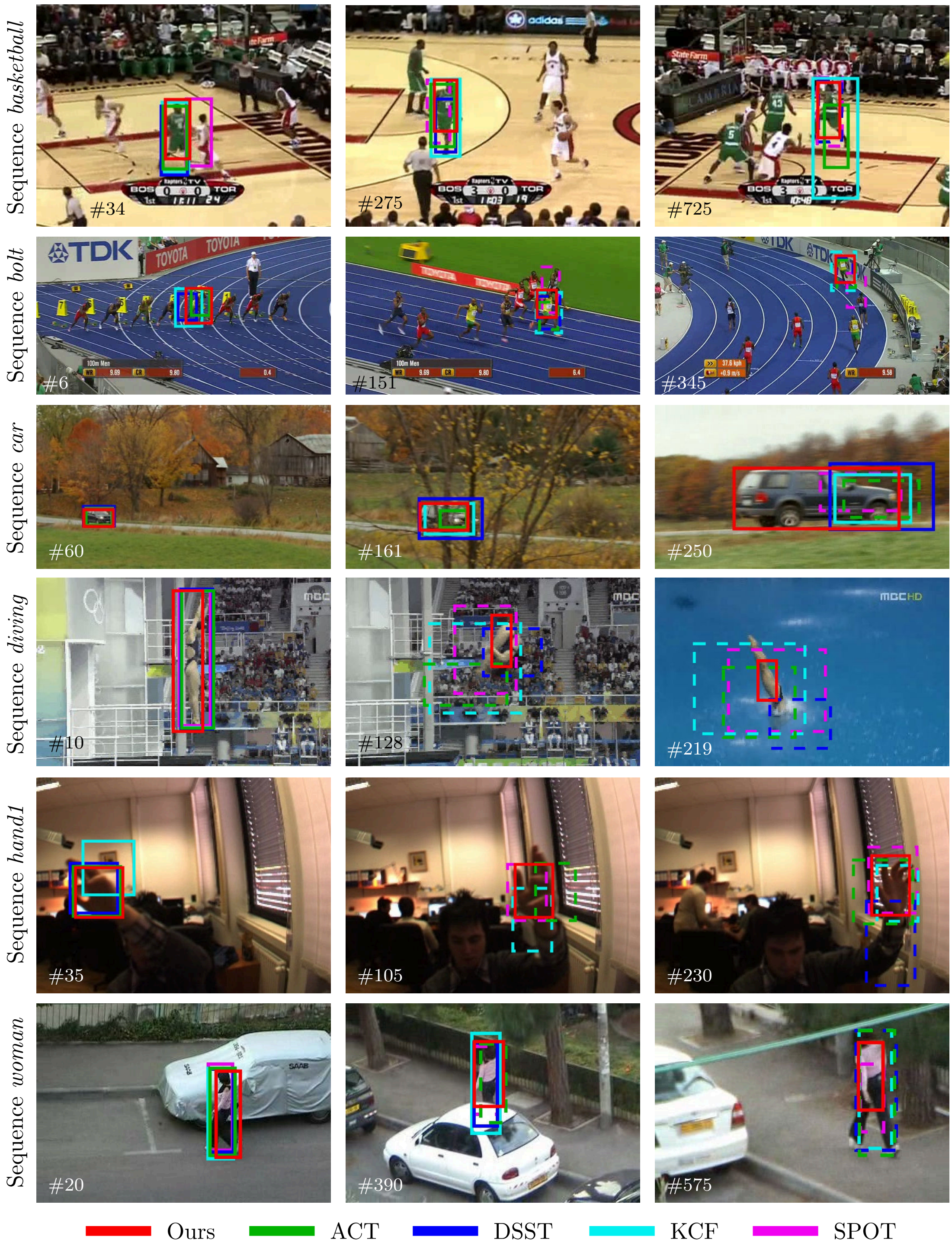


Figure 3.1: A visual comparison of our tracker to state-of-the-art approaches ACT [93], DSST [92], KCF [189] and SPOT [485] on several VOT'14 [239] sequences demonstrates the benefits of distractor-awareness. Dashed bounding boxes indicate that the corresponding tracker has been re-initialized after drifting previously. Images are slightly cropped and frame numbers are superimposed only for visualization.

This chapter is partly based on our publication on distractor-aware color-based tracking [351]. In the following, we briefly review related approaches in Section 3.2 before introducing our tracking approach in Section 3.3. In particular, we will derive our distractor-aware model starting from a standard color representation and show how to exploit this model for robust localization and efficient scale adaptation. Finally, we will summarize the key aspects of our approach in Section 3.4.

3.2 Related Generic Tracking Approaches

Due to the abundance of visual tracking approaches, in the sequel, we summarize only single object trackers which are closely related to our color-based distractor-aware approach. For a broader overview on generic object tracking, please refer to Chapter 2.

Color-based Tracking Approaches. With the increased computing power, color-based approaches became popular within the visual tracking community in the early 2000's, *e.g.* [78, 85–87, 173, 209, 304, 331–333, 342, 356]. A notable early work is the mean shift tracker by Comaniciu *et al.* [85, 86], which introduces a metric derived from the Bhattacharyya coefficient [46] to reason about the similarity of image regions based on color histogram matching. Their framework has been widely extended, *e.g.* by spatially regularizing the histogram representations with isotropic kernels [87], viewpoint-insensitive histograms [116], integrating scale adaptation [83] or replacing mean shift by a scale-adaptive, EM-like algorithm [499]. Similarly, color histograms have been used to estimate the likelihood for each sampled particle in the particle filter frameworks independently proposed by Nummiaro *et al.* [331, 332] and Pérez *et al.* [342]. Such particle filter-based approaches have been widely adopted in the visual tracking community, *e.g.* [248, 251]. Besides color histograms, tracking approaches usually employed Gaussian mixture models, *e.g.* [209, 304, 356]. A notable early work is the Bayesian filtering approach by Isard and MacCormick [209], which leverages Bayesian correlation [404] with Gaussian filter banks on the color channels and extends the particle filtering framework to handle a varying number of objects.

Due to the expressiveness and efficiency of most color-based object representations, color cues have been included in many tracking frameworks, *e.g.* [19–21, 233, 268, 293, 337, 348]. A detailed analysis of color features has been conducted by Collins *et al.* [84], who propose an online framework which automatically selects the most discriminative color feature for tracking *w.r.t.* the current sequence conditions. Several approaches extend this idea by either fusing multiple feature cues (including color), *e.g.* [111], or using an ensemble of trackers which operate on different color features, *e.g.* [249, 480].

Color information is also widely used for segmentation-based tracking approaches, *e.g.* active contour methods [47, 140], graph cut-based methods [30, 156, 157], image matting-based methods [126], probabilistic soft segmentation approaches [71, 72, 75, 113, 114], or to reason about the reliability of correlation filter responses [42, 286, 287]. In particular, the narrow band level set framework of Bibby and Reid [47] is notably similar in spirit to our work as it leverages color-based, pixel-wise posterior probabilities. However, they additionally exploit the object shape but do not incorporate any supplementary context, such as potentially distracting regions, which can easily degrade their level set segmentation if visually similar regions are close-by or even overlap with the object.

Another line of research is focused on deriving improved color descriptors, such as *color attention* [223], *color attributes* [222], *discriminative color descriptors* [224], *color names* [423] or *opponent derivative* and *hue descriptors* [422]. Recently, simple histogram- and raw pixel color-based tracking models have been replaced by such more complex color representations. In particular, color names [423] are widely used in state-of-the-art correlation filter frameworks, *e.g.* [93, 286], and have also been used to complement deep feature representations more recently, *e.g.* [97, 98, 203]. In contrast to these approaches, we show that simple histogram-based representations suffice to achieve both accurate and robust tracking results, competitive to the state-of-the-art.

Context-aware Tracking Approaches. There are two widely used contextual cues in visual object tracking, namely (i) the immediate background which must be considered when building a useful object model, *e.g.* [42, 55, 92, 97, 189, 286, 491]; and (ii) spatio-temporal context given by the previously observed object states, *e.g.* [292, 438, 482]. However, besides these essential contextual cues, exploiting additional context information – such as identifying distracting regions to focus the visual attention or leverage constraints induced from scene geometry – has received significantly less interest from the tracking community. This can be contributed to the fact that incorporating such cues leads to more complex models – in particular, context must be identified, modeled and learned on-the-fly without any prior knowledge.

There are, however, a few notable exceptions, such as [104, 164, 467, 484, 485, 491, 497]. These approaches distinguish between context provided by either *supporting* or *distracting* regions. Supporting regions, as used by [104, 164, 491, 497], exhibit different appearance than the object of interest but co-occur with it, providing valuable cues to overcome occlusions. Distractors, on the other hand, exhibit similar appearance and may therefore be confused with the object. Typically, context-aware trackers such as [467, 484, 485] assume that distractors are of the same object class (*e.g.* pedestrians) and need to track these distractors in addition to the target to prevent drifting. In contrast to these approaches, we impose no assumptions on the object class of distractors. Moreover, we adapt the object representation such that potentially distracting regions are suppressed in advance and thus, no explicit tracking of distractors is required.

3.3 Online Distractor-Aware Object Tracking

In the following, we introduce our distractor-aware visual object tracking approach, DAT. First, we derive the basic color model in Section 3.3.1 and explain its distractor-aware

extension in Section 3.3.2. Next, we discuss how to localize the object of interest based on this efficient object model in Section 3.3.3. Finally, we show how this representation can also be used to efficiently adapt to changing object scales without the need of exhaustive scale space search in Section 3.3.4.

3.3.1 Object-versus-Surroundings Model

Color is a powerful visual cue to distinguish object pixels from surrounding background pixels. To efficiently represent the joint color distribution over an image region, we employ N_C-dimensional histograms, where N_C denotes the number of color channels. To this end, let $H^I_\Omega(b)$ denote the b-th bin of the non-normalized histogram H computed over the region $\Omega \subseteq I$, where I is the input image. Then, let $b_\mathbf{x}$ denote the histogram bin b assigned to the color components of pixel $I(\mathbf{x}) \in \mathbb{R}^{N_C}$ at location $\mathbf{x} = (x, y)^\top$. For example, $I(\mathbf{x}) = (\text{red}, \text{green}, \text{blue})^\top$ using the standard RGB color space. To compute the object likelihood at the pixel location $\mathbf{x}$, we apply Bayes' theorem to get the conditional probability

$$p(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x}) = \frac{p(b_\mathbf{x} \mid \mathbf{x} \in \mathcal{O})\, p(\mathbf{x} \in \mathcal{O})}{p(b_\mathbf{x})} \tag{3.1}$$

where $\mathbf{x} \in \mathcal{O}$ denotes that the pixel at location $\mathbf{x}$ belongs to the object. Since a pixel at location $\mathbf{x}$ either belongs to the object or not, the events $\mathbf{x} \in \mathcal{O}$ and $\mathbf{x} \notin \mathcal{O}$ are obviously mutually exclusive. Thus, we can apply the law of total probability to compute the marginal probability $p(b_\mathbf{x})$ and get

$$p(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x}) = \frac{p(b_\mathbf{x} \mid \mathbf{x} \in \mathcal{O})\, p(\mathbf{x} \in \mathcal{O})}{p(b_\mathbf{x} \mid \mathbf{x} \in \mathcal{O})\, p(\mathbf{x} \in \mathcal{O}) + p(b_\mathbf{x} \mid \mathbf{x} \notin \mathcal{O})\, p(\mathbf{x} \notin \mathcal{O})}. \tag{3.2}$$

This formal definition of the conditional probability, however, relies on an accurate pixel-wise segmentation to compute the likelihood and prior terms as we need to know whether a pixel belongs to the object, *i.e.* $\mathbf{x} \in \mathcal{O}$, or not.

Such accurate annotations are usually not available to initialize a tracking algorithm as they are computationally too expensive to obtain. Instead, tracking approaches have to rely on much coarser initialization regions, typically provided as an annotated bounding box or a polygon. From a more practical point of view, these coarse initializations are both easy and fast to annotate, which allows us to start tracking (almost) immediately. Given such an annotated region O which contains the object of interest and the corresponding surrounding region S, we can estimate the missing terms in Eq. (3.2) and relax the posterior probability to

$$p(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x}) \approx \frac{p(b_\mathbf{x} \mid \mathbf{x} \in O)\, p(\mathbf{x} \in O)}{\sum\limits_{\Omega \in \{O, S\}} p(b_\mathbf{x} \mid \mathbf{x} \in \Omega)\, p(\mathbf{x} \in \Omega)}. \tag{3.3}$$

In practice, we choose S such that it covers a sufficiently large portion of the immediate surroundings of the object region O. More formally, the object region is the set of pixels

$$O = \left\{ \mathbf{x} = (x, y)^\top \;\middle|\; |c_x - x| \leq \frac{w_O}{2} \,\wedge\, |c_y - y| \leq \frac{h_O}{2} \right\}, \tag{3.4}$$

where w_O and h_O denote the width and height of the rectangular object region, respectively, and $\mathbf{c} = (c_x, c_y)^\top$ denotes its center. For a more compact notation, we denote the object region by the tuple

$$O = (\mathbf{c}, w_O, h_O)^\top \tag{3.5}$$

in the following. Then, we can define the surrounding region S to be

$$S = \left\{ \mathbf{x} \;\middle|\; |c_x - x| \leq \frac{\lambda_S w_O}{2} \,\wedge\, |c_y - y| \leq \frac{\lambda_S h_O}{2} \right\} \setminus \left\{ O \right\}, \tag{3.6}$$

where $\lambda_S > 1$ is a predefined scaling factor. Note that the regions O and S are disjoint, as also illustrated in Figure 3.2a.

Using the color distributions over these disjoint regions O and S, we can compute the likelihood terms directly from normalized color histograms as

$$p(b_\mathbf{x} \mid \mathbf{x} \in O) = \frac{H_O^I(b_\mathbf{x})}{|O|}, \tag{3.7}$$

and

$$p(b_\mathbf{x} \mid \mathbf{x} \in S) = \frac{H_S^I(b_\mathbf{x})}{|S|}, \tag{3.8}$$

where $|\cdot|$ denotes the cardinality. Similarly, the prior probabilities can be computed from the annotated regions as

$$p(\mathbf{x} \in O) = \frac{|O|}{|O| + |S|}, \tag{3.9}$$

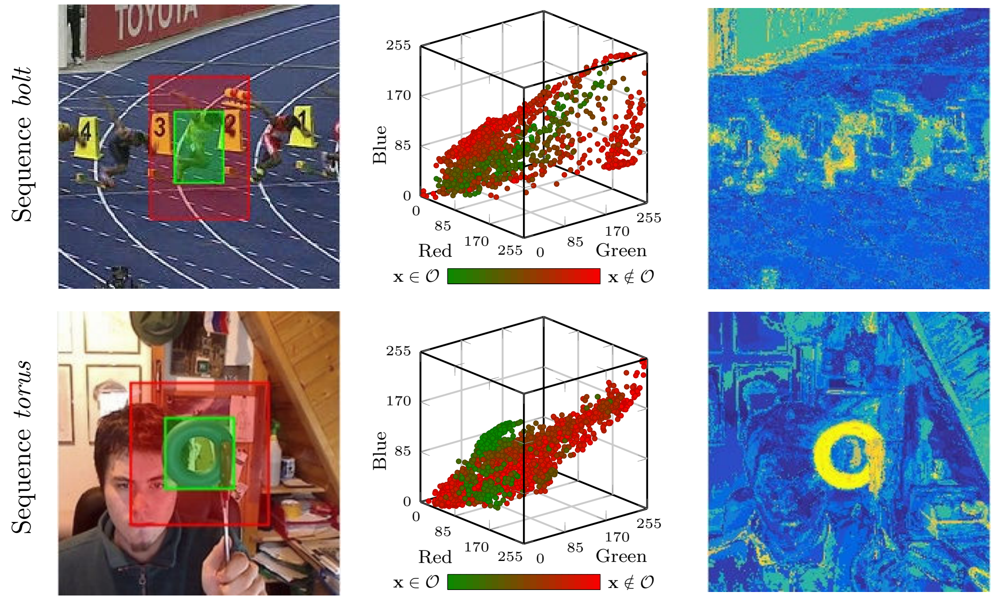
and

$$p(\mathbf{x} \in S) = \frac{|S|}{|O| + |S|}. \tag{3.10}$$

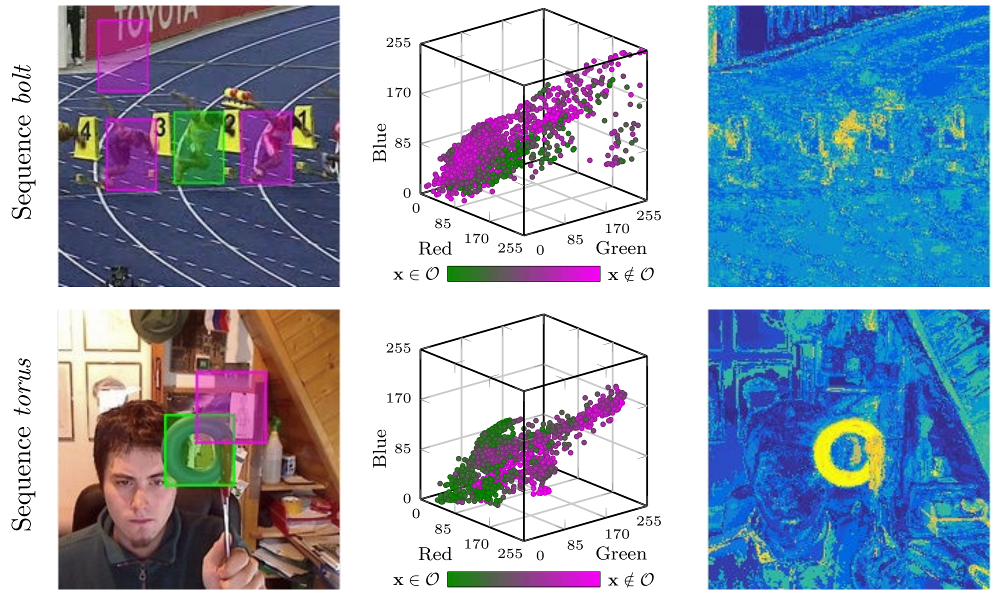
Plugging these terms into Eq. (3.3) and simplifying yields

$$p(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x}) = \frac{\frac{H_O^I(b_\mathbf{x})}{\cancel{|O|}} \frac{\cancel{|O|}}{|O| + |S|}}{\frac{H_O^I(b_\mathbf{x})}{\cancel{|O|}} \frac{\cancel{|O|}}{|O| + |S|} + \frac{H_S^I(b_\mathbf{x})}{\cancel{|S|}} \frac{\cancel{|S|}}{|O| + |S|}} \tag{3.11}$$

$$= \frac{\frac{H_O^I(b_\mathbf{x})}{\cancel{|O| + |S|}}}{\frac{H_O^I(b_\mathbf{x})}{\cancel{|O| + |S|}} + \frac{H_S^I(b_\mathbf{x})}{\cancel{|O| + |S|}}} \tag{3.12}$$



(a) Object-versus-surroundings model $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ computed from the annotated object region O (highlighted in green) and its surrounding region S (highlighted in red).



(b) Object-versus-distractors model $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ based on the object region O (highlighted in green) and the set D of distracting regions (highlighted in magenta).

Figure 3.2: Exemplary object likelihood maps for (a) the object-versus-surroundings model and (b) the object-versus-distractors model on sequences *bolt* and *torus* of the VOT'14 [239] dataset. For each model, we show the regions of interest superimposed on the input image (left) along with the joint color distribution (middle) and the corresponding object likelihood maps (right), obtained by applying the model for every pixel of the input image. Warmer colors indicate higher object likelihood scores. Note that the high object likelihoods at the banner and the close-by athletes for *bolt* in (a) are significantly reduced in (b) by the distractor-aware model, which focuses on the visual cue that distinguishes Bolt from the other athletes, *i.e.* his jersey. Similarly, there is a small blueish region right above the *torus* identified as a potential distractor.

$$= \frac{H_O^I(b_{\mathbf{x}})}{H_O^I(b_{\mathbf{x}}) + H_S^I(b_{\mathbf{x}})}. \tag{3.13}$$

This model is based on all observed pixels within the region $O \cup S$. However, it does not allow to reason about the object probabilities for colors which are not present in these regions. Thus, we initially assign the maximum entropy prior of $1/2$ to pixels which color is not contained within $O \cup S$. This expresses the corresponding uncertainty and furthermore, prevents a division by zero. Now we can define the basic object-versus-surroundings model computed for the current input image I at time t as

$$p_{O,S}^t(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = \begin{cases} \dfrac{H_O^I(b_{\mathbf{x}})}{H_O^I(b_{\mathbf{x}}) + H_S^I(b_{\mathbf{x}})} & \text{if } I(\mathbf{x}) \in I(O \cup S) \\ 1/2 & \text{otherwise,} \end{cases} \tag{3.14}$$

where we use the subscript notation $p_{O,S}^t(\cdot)$ to indicate that the conditional probability is computed from the pixels observed within the regions O and S. This model can be implemented efficiently using lookup-tables, which enables real-time capable online tracking. Note that in practice, the distinction of cases in Eq. (3.14) is not necessary as we can instead apply Laplace smoothing (also known as *additive* or *add-one smoothing*) of the probabilities [298, Chap. 13] which results in the more compact definition

$$p_{O,S}^t(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = \frac{H_O^I(b_{\mathbf{x}}) + 1}{H_O^I(b_{\mathbf{x}}) + H_S^I(b_{\mathbf{x}}) + 2}. \tag{3.15}$$

Subsequent model updates properly adjust the maximum entropy prior of previously unobserved colors according to whether such pixels belong to the object region or its surroundings. In particular, we update our model regularly to handle changing object appearance and illumination variations. More formally, we define the full object-versus-surroundings model as

$$p_{O,S}^{1:t}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = \eta_S p_{O,S}^t(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) + (1 - \eta_S)\, p_{O,S}^{1:t-1}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})\,, \tag{3.16}$$

where initially, $p_{O,S}^{1:1}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = p_{O,S}^1(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ at time step $t = 1$, and $\eta_S \in [0, 1]$ is the learning rate.

3.3.2 Object-versus-Distractors Model

By distinguishing object pixels from background pixels, the object-versus-surroundings model already provides a strong cue for localizing an object, as illustrated in Figure 3.2a. However, one of the most common problems of color-based tracking models remains – namely, that such models cannot distinguish the object from nearby regions which exhibit a similar visual appearance compared to the object of interest and thus, the tracker may drift. To overcome this limitation, we explicitly extend the object model to suppress such distracting regions. Due to the efficient realization of the object-versus-surroundings model via lookup-tables, we can easily afford to compute the posterior probability $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ over a large search region at a very low computational cost. As will be discussed in Section 3.3.3, this efficiency allows us to identify potentially distracting regions in advance and properly robustify our tracker as follows.

For now, let us assume we are given the current object region O and a set D of potentially distracting regions, *i.e.* regions that are visually similar to the object. Such exemplary distractors are illustrated in Figure 3.2b. We exploit this information to build a representation capable of distinguishing object and distracting pixels. To this end, we again employ Bayes' theorem as in Eq. (3.3), where we replace the surrounding region S by the set of distracting regions D. Similar to Eq. (3.8) and (3.10), we compute the likelihood and prior terms from color histograms as

$$p(b_{\mathbf{x}} \mid \mathbf{x} \in D) = \frac{H^I_D(b_{\mathbf{x}})}{|D|}, \tag{3.17}$$

and

$$p(\mathbf{x} \in D) = \frac{|D|}{|O| + |D|}. \tag{3.18}$$

Plugging these terms into the relaxed posterior, simplifying and applying Laplace smoothing, as in Eq. (3.15), then yields the basic object-versus-distractors model computed for the current input image I at time t as

$$p^{t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = \frac{H^I_O(b_{\mathbf{x}}) + 1}{H^I_O(b_{\mathbf{x}}) + H^I_D(b_{\mathbf{x}}) + 2}, \tag{3.19}$$

where again, pixel colors not observed within $O \cup D$ are assigned the maximum entropy prior of $1/2$. To obtain the full object-versus-distractors model, we update this model whenever visually distracting regions D are identified according to

$$p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = \eta_D p^{t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) + (1 - \eta_D)\, p^{1:t-1}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})\,, \tag{3.20}$$

where $\eta_D \in [0, 1]$ is the learning rate. If there are no distractors at $t = 1$, we initialize the object-versus-distractors model as $p^{1:1}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = p^{1}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$, which is the same as considering the surrounding region to be distracting, *i.e.* $D = S$. Otherwise, if there are distractors at $t = 1$, we use the initialization $p^{1:1}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}}) = p^{1}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$.

Note that if there are no distractors at a later time step throughout the sequence, *i.e.* $D = \{\varnothing\}$, there are two options regarding the model update. On the one hand, we can decay the distractor suppression by letting $D = S$ and performing the update as in Eq. (3.20). On the other hand, we can simply refrain from updating the object-versus-surroundings model if $D = \{\varnothing\}$. In our evaluations, both options led to the exactly same tracking performance. We observed that in general, distracting regions appear rather frequently and thus, there are very few frames where $D = \{\varnothing\}$. For these reasons, we rely on the latter, *i.e.* perform no update if there are no distracting regions.

The object-versus-distractors representation focuses on colors that distinguish the object from visually similar distractors, as illustrated in Figure 3.2b. Applying both, $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ and $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$, for each pixel of an image region, we obtain likelihood maps which can be used to robustly localize an object throughout a video sequence.

3.3.3 Target Localization

Considering the previous example in Figure 3.2, a straightforward way to localize the target would be to linearly combine the two object models and find the most likely region within the weighted likelihood map. Especially for the *bolt* sequence, it is easy to find a color bin distinguishing Bolt from the surrounding background and visually similar regions, due to the distinct color of his jersey. Thus, a combined model is sufficient to robustly track the athlete in this sequence. In general, however, applying a combined model does not always yield the most robust results and often severely degrades the likelihood maps. For example, consider the additional sequences in Figure 3.3. There, distracting regions are not as visually distinct as in the *bolt* sequence and suppressing these color cues would significantly degrade a combined model. Such a degraded model would either lead to drift or limited scale adaptation capabilities. Thus, we propose the following localization scheme which exploits both available object models in a late fusion manner.

Given a new frame at time t, we seek the image region which – according to our object representations – most likely contains the object of interest. Similar to *tracking-by-detection*-based approaches, we constrain the search region based on the previous object hypothesis. In particular, we extract a rectangular search window W^t proportional to the previous object region

$$O^{t-1} = (\mathbf{c}^{t-1},\, w_O^{t-1},\, h_O^{t-1})^\top, \tag{3.21}$$

where $\mathbf{c}^{t-1} = (c_x^{t-1},\, c_y^{t-1})^\top$ denotes the center of the rectangular object region O^{t-1} as of time $t-1$, and w_O^{t-1} and h_O^{t-1} denote its width and height, respectively. More formally, we employ the search window

$$W^t = \left(\mathbf{c}^{t-1},\, \lambda_W w_O^{t-1},\, \lambda_W h_O^{t-1}\right)^\top, \tag{3.22}$$

where $\lambda_W > \lambda_S$ is a predefined scaling factor. Within this search region, we densely sample a set of object hypotheses

$$O^t_{i,j} = \left(\mathbf{c}^t_{i,j},\, w_O^{t-1},\, h_O^{t-1}\right)^\top, \tag{3.23}$$

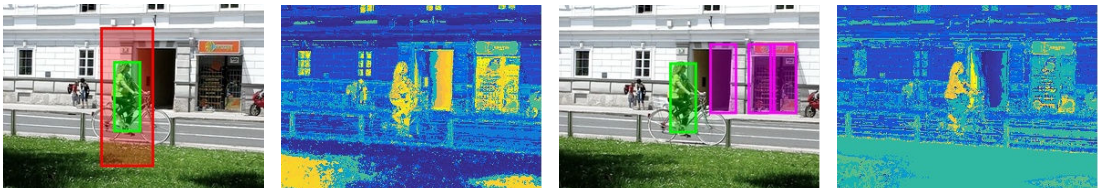
where

$$\mathbf{c}^t_{i,j} = \begin{pmatrix} \underbrace{c_x^{t-1} - \frac{\lambda_W w_O^{t-1}}{2}} + \underbrace{(i-1)(1-o_\nu)w_O^{t-1} + \frac{w_O^{t-1}}{2}} \\ \underbrace{c_y^{t-1} - \frac{\lambda_W h_O^{t-1}}{2}}_{\text{Top left corner of } W^t} + \underbrace{(j-1)(1-o_\nu)h_O^{t-1} + \frac{h_O^{t-1}}{2}}_{\text{Offset to center of } O^t_{i,j}} \end{pmatrix}, \tag{3.24}$$

Inputs I, O and S | $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ | Inputs I, O and D | $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$

(a) Sequence *hand2*. The object-versus-distractors model $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ (two rightmost images) successfully suppresses the visually similar regions on the face. Nevertheless, the overall high object likelihood scores on the palm of the hand – from the object-versus-surroundings model $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ (two leftmost images) – are also reduced due to the similar skin tone, which must be addressed for robust localization.

Inputs I, O and S | $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ | Inputs I, O and D | $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$



(b) Sequence *bicycle*. $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ (right) significantly suppresses the dark regions at the doorways which are visually similar to the dark trousers of the bicyclist. Consequently, $p^{1:t}_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ provides a valuable cue for localization, whereas $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ (left) should be preferred for scale adaptation to prevent cropping the cyclist's feet.

Figure 3.3: Typical challenges for localization and scale adaptation on the VOT'14 [239] benchmark. These potential issues have to be addressed to achieve a robust tracking performance.

with

$$i = 1, 2, \ldots, \left\lfloor \frac{\lambda_W - 1}{1 - o_\nu} \right\rfloor, \tag{3.25}$$

$$j = 1, 2, \ldots, \left\lfloor \frac{\lambda_W - 1}{1 - o_\nu} \right\rfloor. \tag{3.26}$$

Here, the predefined factor $o_\nu \in [0, 1)$ specifies the overlap between neighboring hypotheses, and $\lfloor \cdot \rfloor$ denotes the floor function. Then, we obtain the current object location as

$$O^t_\star = \underset{O^t_{i,j}}{\arg\max} \left\{ \underbrace{\left(\rho_S\left(O^t_{i,j}\right) + \rho_D\left(O^t_{i,j}\right) \right)}_{\text{Appearance term}} \underbrace{\exp\left(-\frac{\left\| \mathbf{c}^{t-1} - \mathbf{c}^t_{i,j} \right\|_2^2}{2\sigma^2} \right)}_{\text{Motion term}} \right\}, \tag{3.27}$$

where

$$\rho_S\left(O_{i,j}^t\right) = \frac{1}{2}\left(\frac{1}{\left|O_{i,j}^t\right|}\sum_{\mathbf{x}\in O_{i,j}^t} p_{O,S}^{1:t-1}(\mathbf{x}\in\mathcal{O}\mid b_{\mathbf{x}}) + \frac{1}{\left|\overline{O_{i,j}^t}\right|}\sum_{\mathbf{x}\in \overline{O_{i,j}^t}} p_{O,S}^{1:t-1}(\mathbf{x}\in\mathcal{O}\mid b_{\mathbf{x}})\right), \quad (3.28)$$

$$\overline{O_{i,j}^t} = \left(\mathbf{c}_{i,j}^t, \frac{w_O^{t-1}}{2}, \frac{h_O^{t-1}}{2}\right)^\top, \quad (3.29)$$

and

$$\rho_D\left(O_{i,j}^t\right) = \frac{1}{|O_{i,j}^t|}\sum_{\mathbf{x}\in O_{i,j}^t} p_{O,D}^{1:t-1}(\mathbf{x}\in\mathcal{O}\mid b_{\mathbf{x}}), \quad (3.30)$$

are the similarity scores estimated from the object-versus-surroundings and object-versus-distractors model, respectively. Note that these similarity scores can be computed efficiently via integral images [426], also known as summed area tables [88] in computer graphics. The motion term in Eq. (3.27), with

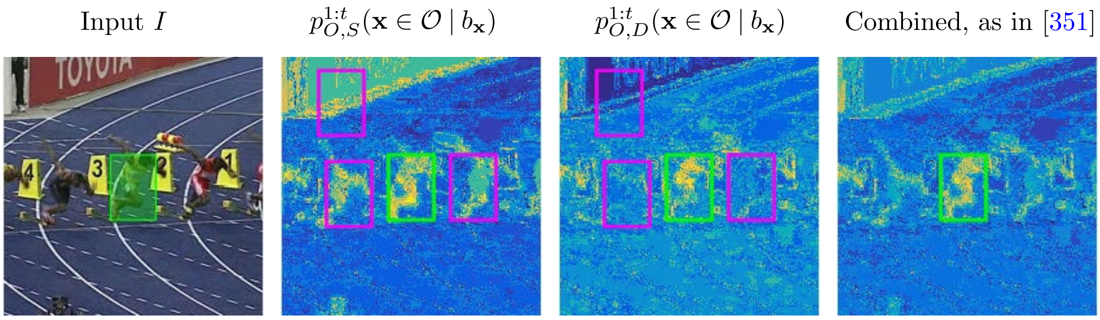
$$\sigma = \sqrt{\left(w_O^{t-1}\right)^2 + \left(h_O^{t-1}\right)^2}, \quad (3.31)$$

penalizes large inter-frame movements, similar to the Gaussian and cosine kernels used by correlation-based trackers, such as MOSSE [55], DSST [92] or KCF [189].

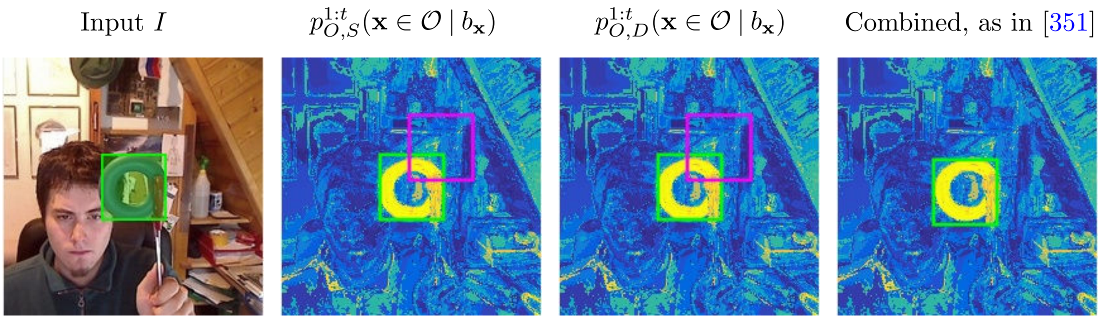
Empirically, we found that including the additional term for the inner region $\overline{O_{i,j}^t}$ in Eq. (3.28) leads to smoother localization results. On average, this increased the overlap between the estimated object locations and the ground truth by 1–2 %. Note that throughout our experiments, the same improvement could also be achieved by employing a Kalman filter [217] instead of adding this inner region term. However, employing an additional filtering step would be slightly less efficient *w.r.t.* the overall runtime[4]. Although this improvement is rather marginal, we still include this term, as it comes at a negligible computational cost due to the use of integral images.

Thus, instead of maintaining a combined model, as we did previously in [351] – which quickly degrades if there are no distinct colors to separate the object from distracting regions – we perform a late fusion of the two separate models solely during localization. This yields improved robustness for scenarios where many visually similar distractors occur. In such cases, the object-versus-distractors model focuses only on more discriminative regions, which may include parts of the local background – recall *hand2* in Figure 3.3a – or focus on smaller regions of the object – recall *bicycle* (the rider's torso) in Figure 3.3b or *bolt* (the runner's jersey) in Figure 3.2. Keeping two separate models allows both, robust localization and simplified scale adaptation, as will be discussed in the following section.

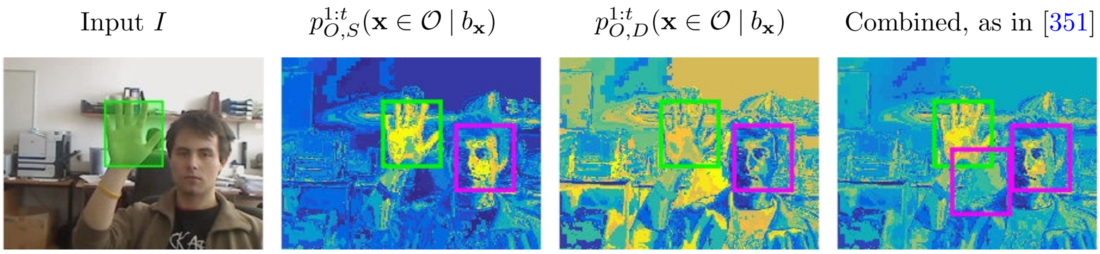
[4]On a standard desktop Intel® Core™ i7 CPU, a straightforward implementation of a Kalman filter takes about 3 ms longer per frame.



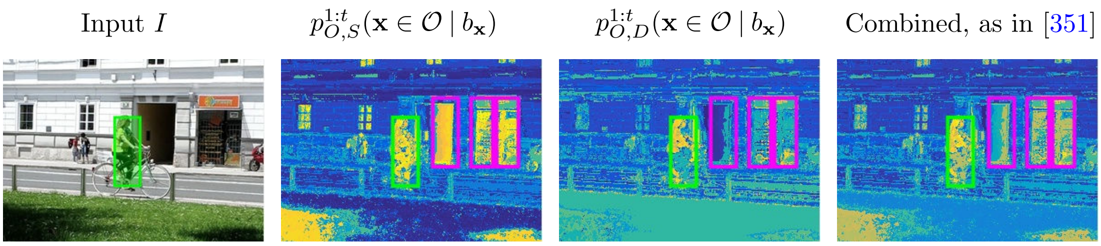
(a) Sequence *bolt*.



(b) Sequence *torus*.



(c) Sequence *hand2*.



(d) Sequence *bicycle*.

Figure 3.4: Localization via object likelihood maps. Green rectangles indicate the object location $O^t_\star$, whereas magenta rectangles illustrate hypotheses $O^t_{i,j}$ which are assigned to the distractor set D. By relying on separate models, we can robustly localize the target and still exploit $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$ for scale adaptation. Early fusion of these models (rightmost column) often leads to a deteriorated result, especially if there are almost no color differences between the object and distractors. For example, compare the probabilities within the distracting regions of the two rightmost columns in (c) (similar skin tone) and (d) (dark doorways versus trousers).

Note that we can easily extend our localization step to identify distracting regions in advance. In particular, visually similar distractors will yield a high similarity score $\rho_S(O^t_{i,j})$. Thus, we get the set of distractors at the current time stamp as

$$D = \left\{ O^t_{i,j} \mid \rho_S(O^t_{i,j}) \geq \tau_\nu \, \rho_S(O^t_\star) \right\} \setminus \left\{ O^t_\star \right\}, \tag{3.32}$$

where the predefined factor $\tau_\nu \in (0, 1)$ controls the amount of distractors to suppress by our object-versus-distractors model. To prevent selecting ambiguous distractors, *e.g.* located on the object itself if the object scale increased between two frames, we follow an iterative non-maximum suppression (NMS) strategy. In particular, after selecting a candidate – either $O^t_\star$ or a distractor – hypotheses $O^t_{i,j}$ which overlap more than 10 % are discarded to avoid including them in the set of distractors. Figure 3.4 illustrates the localization step and shows the advantages of maintaining two separate models.

3.3.4 Scale Estimation

Recently, pre-training a scale estimator on huge datasets, such as PASCAL VOC [122] or ImageNet [367], has widely been used in many state-of-the-art approaches, *e.g.* [124, 319, 321, 413]. However, our color-based object-versus-surroundings model already enables efficient scale estimation without requiring computationally expensive pre-training on such datasets. This also respects the generic nature of the tracker as we can easily adapt to objects that are not captured within these datasets. In the following, we introduce three techniques – namely, (i) *segmentation via connected components*, (ii) *likelihood map sum reduction* and (iii) *instance-specific regression* – which can subsequently be applied after localizing the target in the current frame.

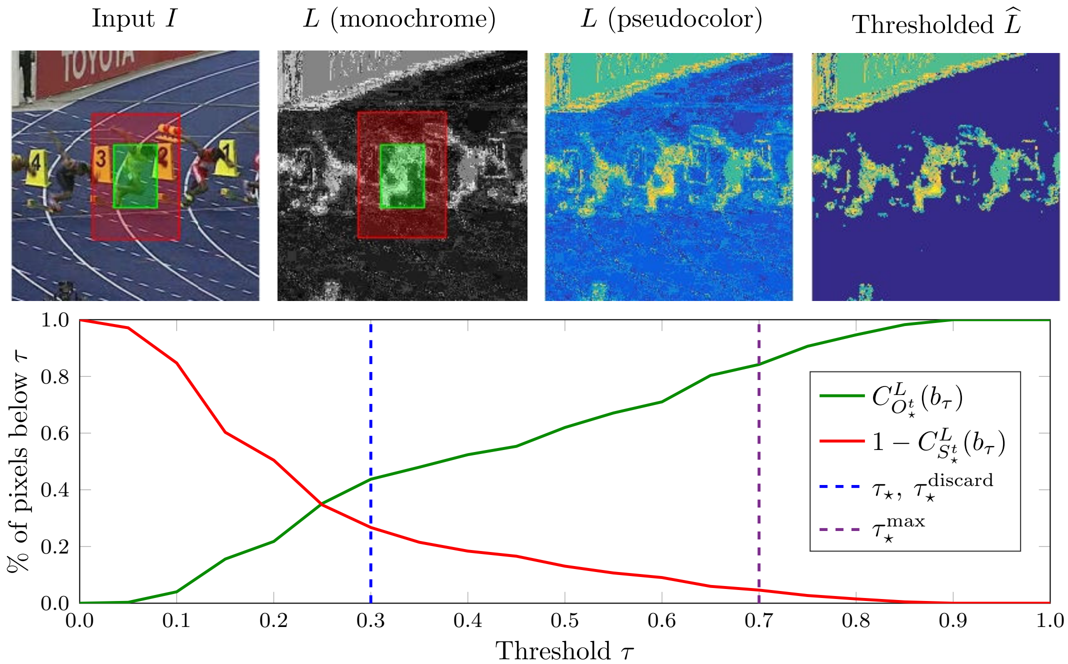
As a pre-processing step to all of these techniques, we employ a coarse pre-segmentation by thresholding the likelihood map obtained from the object-versus-surroundings model. Since choosing a predefined threshold may impede the scale adaptation due to background clutter or fast illumination changes, we employ an adaptive threshold. To this end, let L denote the object likelihood map obtained by evaluating $p^{1:t}_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$ at every location $\mathbf{x}$ of the search region, as shown in Figure 3.5. Then, we compute the cumulative likelihood histograms

$$C^L_{O^t_\star}(b) = \frac{1}{|O^t_\star|} \sum_{i=1}^{b} H^L_{O^t_\star}(i), \tag{3.33}$$

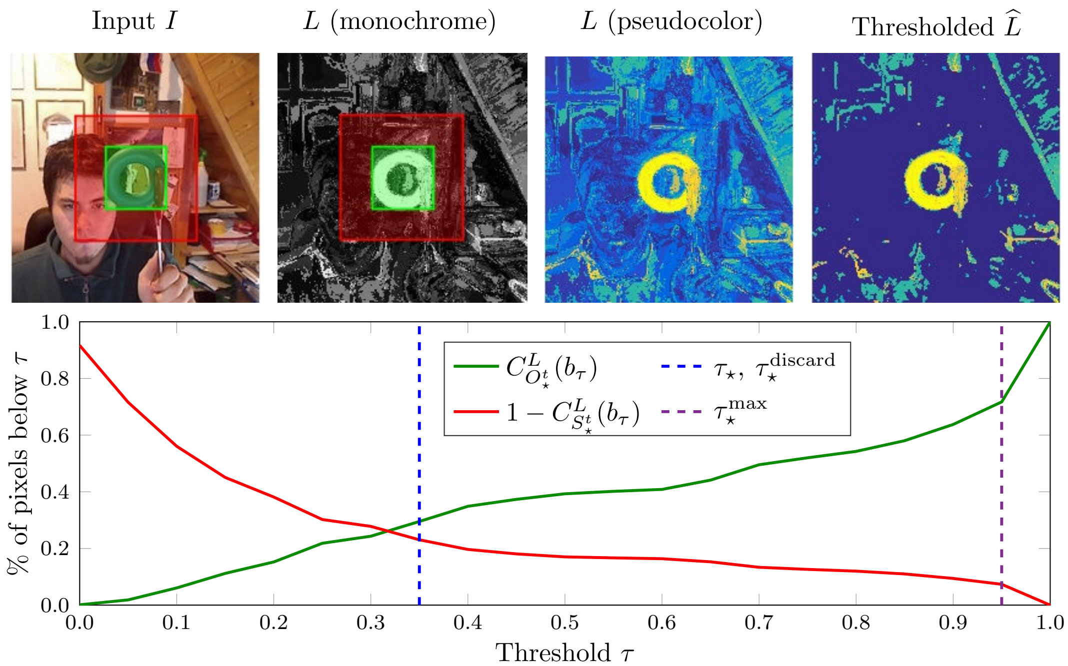
and

$$C^L_{S^t_\star}(b) = \frac{1}{|S^t_\star|} \sum_{i=1}^{b} H^L_{S^t_\star}(i), \tag{3.34}$$

where $S^t_\star$ denotes the region surrounding the estimated object hypothesis $O^t_\star$, recall Eq. (3.6). Note that $H^L_\Omega(\cdot)$ are one-dimensional histograms computed from the object likelihood maps.



(a) Sequence *bolt*.



(b) Sequence *torus*.

Figure 3.5: Computing the adaptive pre-segmentation threshold from likelihood maps. We superimpose the regions of interest for the threshold computation, *i.e.* $O^t_\star$ (green) and $S^t_\star$ (red), on the input image and the corresponding (monochrome) likelihood map. The two rightmost visualizations show the original and thresholded likelihood maps, respectively.

Our thresholding objective is to keep confident object pixels (*i.e.* foreground), while discarding as many background pixels as possible. More formally, by exploiting the cumulative histograms, we compute the adaptive pre-segmentation threshold

$$\tau_\star = \min\left\{\tau_\star^{\text{discard}},\ \tau_\star^{\max}\right\}, \tag{3.35}$$

where

$$\begin{aligned} \tau_\star^{\text{discard}} &= \arg\min_\tau \left\{ C^L_{O^t_\star}(b_\tau) - \left(1 - C^L_{S^t_\star}(b_\tau + 1)\right)\right\} \\ \text{subject to} &\quad C^L_{O^t_\star}(b_\tau) + C^L_{S^t_\star}(b_\tau + 1) \geq 1, \end{aligned} \tag{3.36}$$

seeks a threshold that keeps more object pixels than background pixels. Here, b_τ denotes the bin b which is assigned to the threshold $\tau \in [0, 1]$. The bin offset in Eq. (3.36), *i.e.* comparing $C^L_{O^t_\star}(b_\tau)$ to $C^L_{S^t_\star}(b_\tau + 1)$, ensures that the chosen threshold $\tau_\star^{\text{discard}}$ is above the crossing point of the cumulated object histogram and the (inverted) surrounding histogram, see Figure 3.5. To guarantee that at least a minimum amount of foreground pixels is kept after thresholding, we impose the hard limit

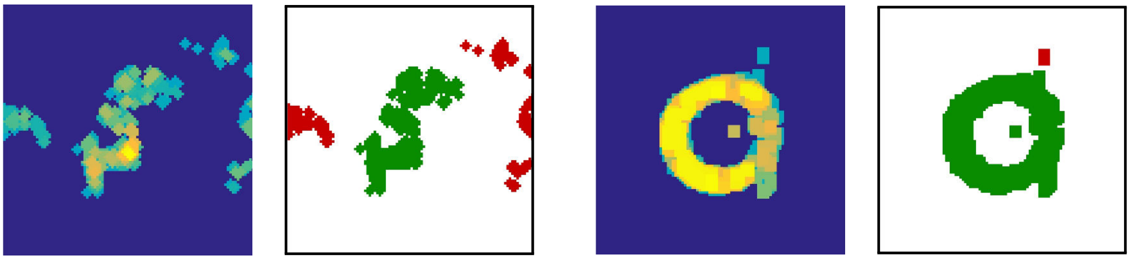
$$\begin{aligned} \tau_\star^{\max} &= \arg\max_\tau \left\{ C^L_{O^t_\star}(b_\tau) - (1 - c_\tau)\right\} \\ \text{subject to} &\quad C^L_{O^t_\star}(b_\tau) + c_\tau \leq 1, \end{aligned} \tag{3.37}$$

where c_τ controls the amount of guaranteed foreground pixels. We choose a fixed $c_\tau = 0.1$ throughout all experiments, which ensures that at least 10 % of the foreground pixels are kept after thresholding, even for extremely challenging capturing scenarios, such as suddenly under- or over-exposed images. As each pixel value of the likelihood map L lies within the range $[0, 1]$, we use a predefined bin width of 0.05 to compute the cumulative histograms $C^L_\Omega(\cdot)$ which are required to compute the pre-segmentation threshold $\tau_\star$. The thresholded likelihood map $\widehat{L}$ can then be computed as

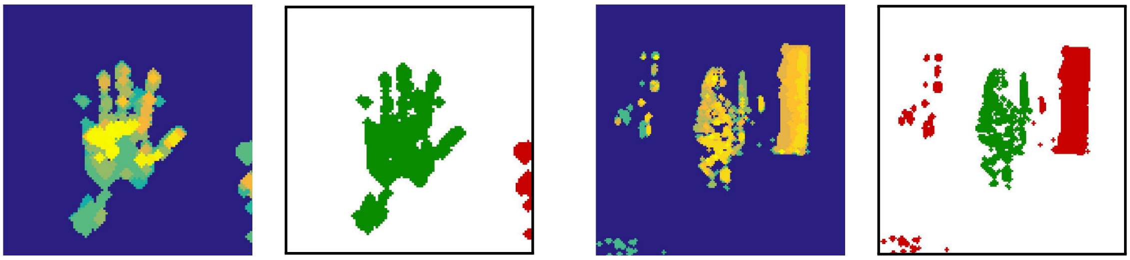
$$\widehat{L}(\mathbf{x}) = \begin{cases} L(\mathbf{x}) & \text{if } L(\mathbf{x}) \geq \tau_\star \\ 0 & \text{otherwise.} \end{cases} \tag{3.38}$$

3.3.4.1 Segmentation via Connected Components

Ideally, we could employ a sophisticated image segmentation approach, *e.g.* by relying on Total Variation-based methods [372, 420] or GrabCut [366], to properly adjust the scale of the estimated object hypothesis, as has been done in several previous tracking frameworks, *e.g.* PaFiSS [30] or HoughTrack [156, 157]. However, such approaches are usually prohibitively expensive in terms of runtime. Furthermore, we typically deal with rather low resolution imagery and low contrast, especially in visual surveillance settings. Given the commonly encountered small object size in combination with low contrast settings, it



(a) Sequence *bolt*. **(b)** Sequence *torus*.



(c) Sequence *hand2*. **(d)** Sequence *torus*.

Figure 3.6: Scale adaptation by leveraging connected components. For each sequence, we show the cropped and morphologically opened likelihood maps $\widetilde{L}$ (left) and the corresponding segmentation result (right), where green blobs are assigned to the object and red blobs are discarded.

is almost infeasible to extract a robust segmentation using sophisticated approaches in an adequate time frame. Thus, in [351] we proposed a more efficient heuristic segmentation approach to handle object scale changes based on analyzing the connected components of the thresholded likelihood map $\widehat{L}$.

To this end, we first leverage morphological opening on $\widehat{L}$ to remove small structures, which mostly correspond to noise. More formally, we compute

$$\widetilde{L} = \left(\widehat{L} \ominus E\right) \oplus E, \tag{3.39}$$

where $\ominus$ and $\oplus$ denote erosion and dilation, respectively, and E is a disk-shaped structuring element with diameter $\min(w_O^{t-1},\ h_O^{t-1})/10$. Then, we crop $\widetilde{L}$ to the square region

$$\widehat{R} = \left(\mathbf{c}_\star^t,\ \ \lambda_S \max\left(w_O^{t-1}, h_O^{t-1}\right),\ \ \lambda_S \max\left(w_O^{t-1}, h_O^{t-1}\right)\right)^\top, \tag{3.40}$$

where $\mathbf{c}_\star^t$ denotes the center of the current object hypothesis $O_\star^t$. Within this crop of the likelihood map $\widetilde{L}$, we find connected components relying on an 8-connected neighborhood. To reason about which connected component, *i.e.* blob, belongs to the object, we consider the inclusion region

$$R_{\text{inc}} = \left(\mathbf{c}_\star^t,\ \ \lambda_{\text{inc}} w_O^{t-1},\ \ \lambda_{\text{inc}} h_O^{t-1}\right)^\top, \tag{3.41}$$

which we also use to ensure a minimum hypothesis size after scale adaptation. Since we expect the object scale change between two subsequent frames to be at most 20 %, we fix the scaling parameter $\lambda_{\mathrm{inc}} = 0.8$ throughout all our experiments. Now, we can assign each blob B to the object if at least half of its area lies within the inclusion region. More formally, we compute

$$O_{\mathrm{cc}} = \left\lceil R_{\mathrm{inc}} \cup \left\{ B \;\middle|\; \frac{|B \cap R_{\mathrm{inc}}|}{|B|} \geq \frac{1}{2} \right\} \right\rfloor_{\mathrm{BB}}, \tag{3.42}$$

where $\lceil \cdot \rfloor_{\mathrm{BB}}$ returns the smallest axis-aligned rectangle containing its argument, *i.e.* the union of the inclusion region and foreground blobs. Exemplary results of this blob-based object segmentation are illustrated in Figure 3.6. Now, we can adjust the scale to get the final object hypothesis at time t as

$$O^t = \lambda_s \, O_{\mathrm{cc}} + (1 - \lambda_s) \, O^t_{\star}, \tag{3.43}$$

where λ_s is a predefined update rate. Empirically, we observed that a fixed scale update rate of $\lambda_s = 0.2$ consistently yielded the best results. Note that in contrast to recent scale-adaptive approaches, such as [92, 189], our scale estimation scheme is not limited to a fixed aspect ratio. As this scale adaptation technique uses connected components, we denote the corresponding scale-adaptive distractor-aware tracker DAT+c.

3.3.4.2 Sum Reduction of Likelihood Maps

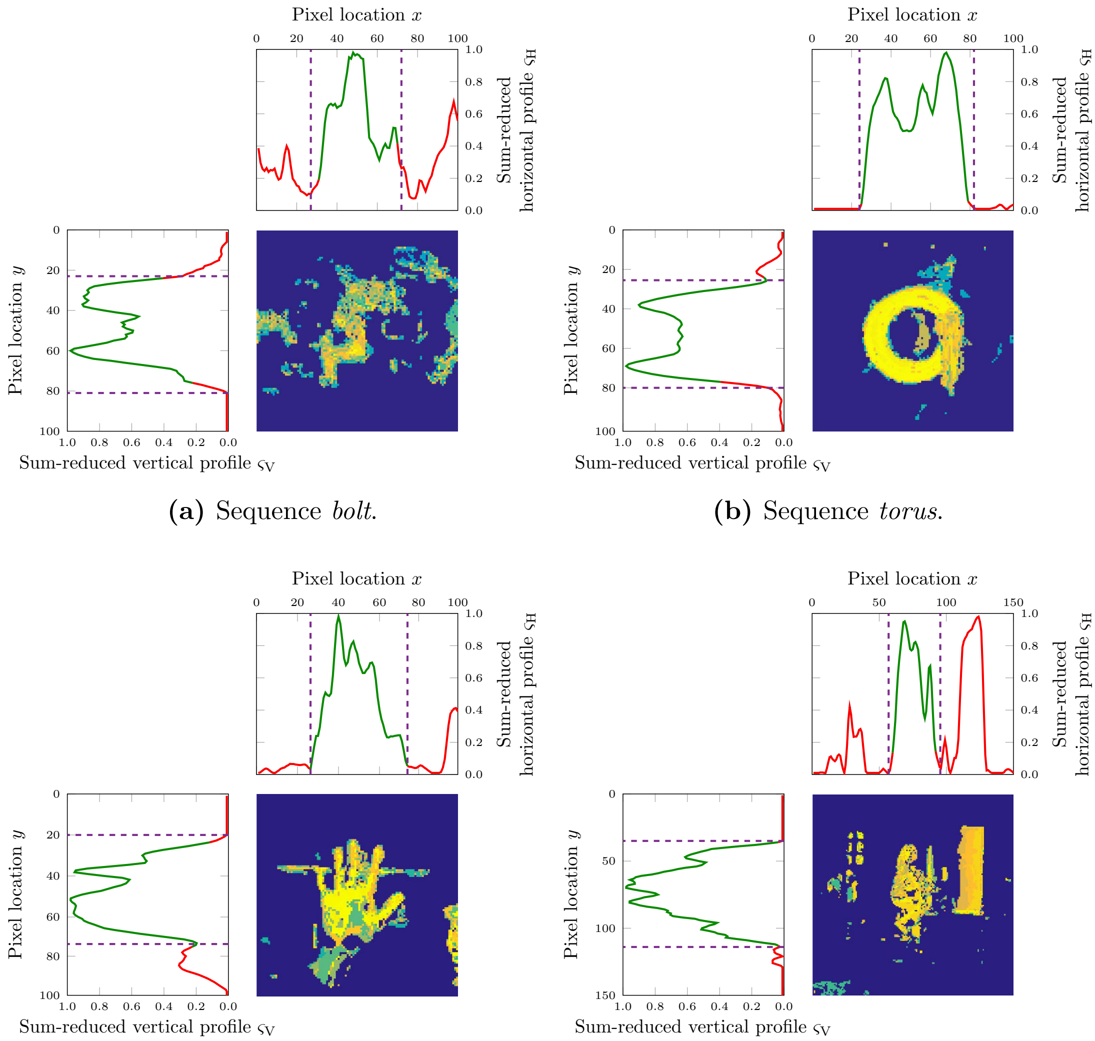
Although intuitive, identifying connected components is a non-trivial and computationally demanding task. A significantly more efficient scale adaptation technique is to separate the 2D segmentation problem into two 1D problems. To this end, we crop the thresholded likelihood map $\widehat{L}$ to the enlarged surrounding region $\widehat{R}$ – recall Eq. (3.40) – and apply sum reduction to get the horizontal likelihood profile

$$\varsigma_{\mathrm{H}}(x) = \sum_y \widehat{L}\Big((x, y)^\top\Big), \tag{3.44}$$

which we normalize such that $\max\limits_x \Big(\varsigma_{\mathrm{H}}(x)\Big) = 1$. Similarly, we compute the vertical likelihood profile

$$\varsigma_{\mathrm{V}}(y) = \sum_x \widehat{L}\Big((x, y)^\top\Big), \tag{3.45}$$

and normalize it as above. As can be seen in Figure 3.7, these likelihood profiles provide a useful cue to reason about object scale changes.



(a) Sequence *bolt*.

(b) Sequence *torus*.

(c) Sequence *hand2*.

(d) Sequence *bicycle*.

Figure 3.7: Sum reduced profiles ς_{H} and ς_{V} of the cropped and thresholded likelihood maps $\widehat{L}$ for scale adaptation. Red parts of the sum reduced profiles indicate the background region according to the ground truth annotation, whereas green parts indicate the object region. Magenta lines highlight the computed local minima used to refine the object scale for our DAT+s variant. Note that the local minima may also be located on plateaus of the profile, *e.g.* see ς_{V} on *bolt* at $y = 22$. Additionally, we can see slightly ambiguous ground truth annotations, *e.g.* consider ς_{V} on *hand2*, where the top of the middle finger is cropped from the ground truth (which defines the top edge of the bounding box at $y \approx 26$) but is included in our segmentation result (which locates the top edge at $y = 20$).

Given the center $\mathbf{c}_\star^t = \left(c_x^t, c_y^t\right)^\top$ of the object hypothesis $O_\star^t$ after localization, we compute the corner points of the hypothesis as

$$c_{\text{left}} = c_x^t - \frac{w_O^{t-1}}{2}, \qquad c_{\text{right}} = c_x^t + \frac{w_O^{t-1}}{2}, \tag{3.46}$$

and

$$c_{\text{top}} = c_y^t - \frac{h_O^{t-1}}{2}, \qquad c_{\text{bottom}} = c_y^t + \frac{h_O^{t-1}}{2}, \tag{3.47}$$

where w_O^{t-1} and h_O^{t-1} denote the width and height of $O_\star^t$, respectively. Then, we search for the local minima of the likelihood profiles in the vicinity of the corresponding corner points. In particular, we seek (i) the local minimum closest to c_{left} and (ii) the local minimum closest to c_{right} of the horizontal profile ς_{H}, as well as (iii) the local minimum closest to c_{top} and (iv) the local minimum closest to c_{bottom} of the vertical profile ς_{V}. These four extremal points then define the extent of the segmentation result O_{sr}, as illustrated by the magenta lines in Figure 3.7. Note that due to ambiguous ground truth annotations, the sum reduced segmentation result may not always yield a perfect alignment with the ground truth. However, this scale adaptation technique is highly generic, *i.e.* it can be applied to all object classes without requiring any prior knowledge, and runs at a fraction of the time required for more complex segmentation approaches.

To get the final, scaled object hypothesis at time t, we again employ the update schema

$$O^t = \lambda_s \, O_{\text{sr}} + (1 - \lambda_s) \, O_\star^t, \tag{3.48}$$

where the update rate is fixed as $\lambda_s = 0.2$, similar to DAT+c in Eq. (3.43). Since we rely on sum reduction for scale adaptation, we denote this tracker variant DAT+s.

3.3.4.3 Instance-specific Bounding Box Regression

Complementary to the efficient likelihood-based segmentation techniques DAT+c and DAT+s, we also experimented with bounding box regression, as it has successfully been applied within several state-of-the-art tracking approaches recently, *e.g.* SANet [124], MDNet [319], TCNN [321] or SINT [413]. Bounding box regression is widely used in object detection – it became popular with DPM [135] and has been extended to neural network-based detectors with R-CNN [154]. In fact, most CNN-based tracking approaches use exactly the same regression technique as proposed for R-CNN. More details on this bounding box regression can be found in the technical report on R-CNN [155].

Due to the significant improvements for both object detection and CNN-based trackers, we also tested several regression-based variants of DAT. As our goal is to provide an efficient and generic tracking approach, we extract the input features for the regression from the object likelihood maps L and $\widehat{L}$, respectively. To keep the beneficial runtime performance of our baseline DAT, we do not invoke an additional, computationally expensive CNN-based feature extraction, but only rely on the shape information captured within these likelihood maps. In particular, we experimented with raw likelihood scores, gradients of the likelihood maps and sum reduced likelihood profiles extracted from both the plain and thresholded likelihood maps, respectively.

In contrast to CNN-based features, these simpler features, however, are insufficient to pre-train class-specific linear regressors on huge datasets. One major issue we observed are the rather ambiguous ground truth annotations. For example, when asked to annotate a face tracking sequence, some human operators prefer bounding boxes which include the neck and hair of a person, whereas others only annotate boxes spanning from the forehead to the chin. Nevertheless, we can learn instance-specific bounding box regressors using only information provided during initialization.

To this end, we augment the provided initialization data by standard geometric transformations, *i.e.* translation, rotation and scaling. We also tried two different regression targets, namely (i) regression of refinement transformations as in R-CNN [154, 155], which is also similar to policy learning approaches and recent action/decision networks, *e.g.* [203, 474], and (ii) regression of the plain bounding box corners. Overall, we achieved the best results by leveraging a regressor to predict the bounding box corners based on our likelihood profiles, which we denote DAT+r throughout our evaluations. For this variant, we follow the same methodology as in [154, 155], except that (i) our input features are the concatenated likelihood profiles and (ii) our regression targets are the actual corners. Similar to DAT+c and DAT+s, we employ the update schema as in Eq. (3.43) or (3.48) to obtain the final, scaled object hypothesis. DAT+r works reasonably well for sequences which exhibit high contrast between the object and its surroundings, which we will discuss in more detail within Chapter 5.1. However, both DAT+c and DAT+s consistently outperform this regression-based variant. Additionally, the sum reduction-based DAT+s offers the additional advantage of negligible runtime cost.

3.4 Summary

We presented a generic single object tracking approach based on very efficient color models. By leveraging the color model to identify and suppress visually distracting regions in advance, our tracker achieves a significant improvement *w.r.t.* the tracking robustness. We can even handle rather noisy initializations by exploiting distinctive color features captured in our object representation. Additionally, we proposed efficient scale estimation schemes based on our object representation which allow us to obtain accurate tracking results for arbitrary object classes. Our extensive evaluation in Chapter 5.1 will show both the beneficial robustness and favorable efficiency of our distractor-aware tracker compared to state-of-the-art approaches. Overall, the proposed approach allows for an efficient implementation to enable online object tracking in real-time.

4 Occlusion Geodesics for Association-based Tracking

We demand rigidly defined areas of doubt and uncertainty!

— Douglas Noël Adams (The Hitchhiker's Guide to the Galaxy)

Contents

4.1 Motivation

This chapter investigates contextual cues to robustly handle fully occluded objects. In contrast to the object appearance-based approach in Chapter 3, here we seek cues that can be exploited whenever the object is not visible. To this end, we consider the task of causal multiple object tracking (MOT) and particularly, focus on pedestrian tracking. MOT is an ideal testbed to study occlusion-related problems, since occlusions are much more frequent – due to the larger number of objects simultaneously captured from the same viewpoint – than in single object tracking scenarios.

Due to the rapid progress in object detection, *e.g.* Poselets [57], HOG [91], ACF [109], DPM [135] or F-RCNN [362], recent research in object tracking has focused on the *tracking-by-detection* principle. Thus, multiple object tracking becomes a *data association* problem where detection responses need to be reliably linked to form target trajectories. However, this is still a difficult and only partially solved problem – mostly because state-of-the-art object detectors still often miss objects or are prone to false positive detections due to dynamic backgrounds or challenging illumination conditions.

Several recent tracking algorithms address the association problem offline, *i.e.* by optimizing detection assignments over large batches of frames (temporal windows), *e.g.* via K-shortest paths [39], Hungarian algorithm [187], and hypergraphs [193]. By exploiting information from future time steps, these approaches overcome detection failures, such as missed detections, over long occlusion periods. However, processing video sequences in large frame batches (*e.g.* via dynamic programming [137]) or even optimizing over whole sequences (*e.g.* via continuous energy minimization [308]) leads to a significant temporal delay between object observation and reporting its location. Thus, such offline approaches are not well suited for time-critical video analysis applications, where object locations must be estimated in real-time, *e.g.* for autonomous vehicles or traffic safety systems.

Instead, such applications require online tracking methods which only consider observations up to the current frame and provide robust location estimates without significant temporal delay. To model the uncertainty which arises from dealing with occluded objects and missed detections, such causal trackers often rely on probabilistic frameworks, *e.g.* Sequential Monte Carlo (SMC) methods as in [60, 349]. However, online approaches often tend to drift if objects are occluded for longer periods of time and may consequently fail to reliably re-assign these missed objects due to simplified motion models.

We aim to overcome these limitations of existing online MOT approaches and reduce re-assignment failures by leveraging contextual information, while achieving high quality tracking performance competitive to offline approaches. A key observation to identify suitable contextual information is that off-the-shelf object detectors primarily fail if the objects are significantly occluded, whereas the detection recall and precision for isolated individuals are sufficiently high to enable tracking with simple techniques. Thus, we introduce a novel confidence measure to predict the location of missed objects, solely based on geometric cues such as occlusion states, detector reliability, and motion prediction. By introducing *occlusion geodesics*, *i.e.* shortest paths – from the location an object first was lost up to its re-detection – *w.r.t.* these instance-specific confidence measures, we can reliably re-assign detections of re-appearing objects to their corresponding trajectories, *e.g.* the blue target in Figure 4.1. Additionally, inspired by the low-level tracklet generation of offline approaches, such as [202, 247], we use a conservative association scheme which links matching detections to trajectories of both isolated and visible objects, *e.g.* the red and green targets in Figure 4.1. By combining these two association strategies, we can introduce an efficient, causal MOT framework which is able to handle complex real-world scenarios, especially for typical video surveillance tasks.

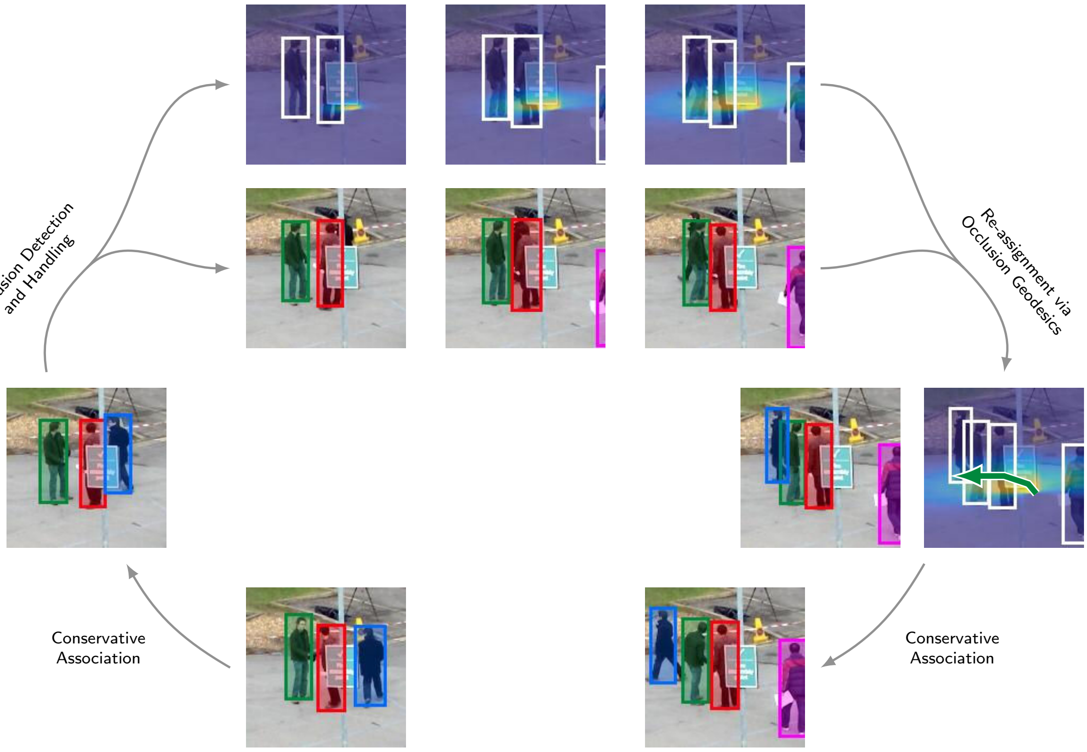


Figure 4.1: Overview of our association-based MOT approach. From bottom-left: while detections are reliable and rather isolated, we rely on a conservative linking scheme to match detections to trajectories. To overcome occlusions, we exploit a novel confidence measure (heatmap overlay, top row) which indicates likely locations for a specific occluded object (here, the blue identity). As soon as re-detection candidates are available, we leverage our knowledge about physically plausible paths through previously occluded regions based on our confidence scores (rightmost frame, denoted by the overlaid green path) and re-assign suitable detection hypotheses to the respective trajectories.

This chapter is partly based on our publication on occlusion geodesics for online multi-object tracking [350]. In the following, we briefly review related approaches and summarize the basic geometric preliminaries in Section 4.2. Next, we introduce our online MOT approach which leverages geometric constraints to robustly handle long-term occlusions in Section 4.3. Finally, we will summarize the key aspects of our approach in Section 4.4.

4.2 Related Work & Preliminaries

In the following, we first summarize approaches related to multi-object tracking (Section 4.2.1) and object detection (Section 4.2.2). As we exploit the scene geometry for our tracking approach, we also recapitulate the image formation process (Section 4.2.3).

4.2.1 Multiple Object Tracking

The most crucial component in tracking-by-detection approaches for MOT is data association, *i.e.* addressing the question, how to correctly assign potentially noisy detections to object trajectories. Traditionally, this problem has primarily been addressed by online methods incorporating Joint Probabilistic Data Association Filters [139], Multi-Hypothesis Tracking [360] or sampling-based approaches – such as Markov chain Monte Carlo methods, *e.g.* [36, 334], and sequential Monte Carlo methods (*i.e.* particle filters), *e.g.* [335, 425]. Such methods maintain multiple hypotheses until enough observations are available to resolve ambiguities. A major drawback of such methods, however, is that they suffer from exponentially increasing complexity due to the combinatorial hypotheses space.

Alternative tracking approaches rely on directly linking available detections to trajectories without keeping multiple hypotheses, *e.g.* [60, 66, 447]. For example, Breitenstein *et al.* [60] use a greedy association scheme in combination with particle filtering based on a constant velocity model. They leverage continuous confidence density maps obtained from the detector to generate object likelihood maps and rely on online learned instance-specific classifiers to resolve occlusion scenarios. In contrast to this work, we consider the object detector to be a black box and instead focus on the robust re-assignment of detections after occlusion scenarios. Our approach is motivated by the observation that object detectors typically re-detect previously occluded objects soon after they move away from the occluder. Thus, we allow missed targets to move along physically plausible paths, which are defined by combining motion prediction, our belief in the detector, and geometric knowledge of occluded regions.

In contrast to such causal tracking approaches, a major line of research focuses on optimizing trajectories over whole sequences, *e.g.* [307, 483], or large batches of frames, *e.g.* [137, 247], to find globally consistent trajectory assignments. Such offline approaches often discretize the space of target locations to simplify the underlying optimization problem, *e.g.* [32, 39, 193]. For example, Berclaz *et al.* [39] propose a graph flow model on a 2D discretization of the ground plane, where detection results are efficiently linked to trajectories using the K-shortest paths algorithm. However, as their method operates offline on a graph built over large frame batches, it cannot handle arbitrarily dense discretizations due to memory limitations. Therefore, other approaches estimate the final object locations by leveraging continuous fitting problems to obtain parametric trajectories which lead to smoother results, *e.g.* [12, 13, 308]. Several offline approaches, *e.g.* [187, 202, 247], additionally follow a hierarchical association schema, where in a first low-level step, subsequent detections are linked together to form so-called tracklets, *i.e.* short but reliable trajectories. Then, the key issue becomes to correctly link such tracklets into longer object trajectories, *e.g.* by combining motion and appearance models [192] or by learning tracklet associations from training data [273].

A major drawback of both offline and batch-processing approaches, however, is that they require detections for future frames in order to obtain robustly linked trajectories. Thus, such approaches are not suitable for time-critical real-world applications, such as visual surveillance or robotics, which we aim for with our causal MOT approach. In particular, we want to show that efficient plausibility reasoning can result in state-of-the-art results, without requiring complex modeling of group dynamics or social interactions, such as [4, 112, 341, 377, 457].

4.2.2 Object Detection

The performance of tracking-by-detection approaches substantially relies on the object detector employed to generate the location hypotheses. Generic object detection approaches traditionally either use (i) holistic, *e.g.* [91, 374, 375, 443], (ii) part-based, *e.g.* [135] or (iii) bag-of-feature models, *e.g.* [418]. The majority of holistic detectors relies on linear models, *e.g.* [91], or ensembles of trees, *e.g.* [374, 375, 443, 486], focusing on highly accurate detection of a single object class, such as faces, pedestrians or cars, suitable for time-critical applications. Extensions for multi-class detection usually train several separate holistic models, *e.g.* [300]. Part-based approaches, on the other hand, divide a model into several discriminative sub-parts, *e.g.* [135], for improved handling of (partial) occlusions and non-rigid deformations. To detect multiple object classes or handle viewpoint changes, several part-based models are combined using mixture models, *e.g.* [135, 391]. Bag-of-feature approaches extract local feature descriptors inside object regions and store these within dictionaries, *e.g.* [418]. A supervised learning framework on top of such a dictionary encoding then classifies object proposals as either object or background. These approaches can handle multi-class detection and typically share a common codebook across several classes.

More recent object detection approaches leverage convolutional neural networks (CNNs) which are pre-trained on large object classification datasets, *e.g.* [244]. Such methods are either applied fully-convolutional, *e.g.* [380], or use region proposals, *e.g.* [418, 498], to extract potential object regions from an image and classify them with a fine-tuned CNN, *e.g.* [152]. These approaches have been heavily extended by either improving speed [151] or computing region proposals using a CNN [362].

In contrast to such generic object detection tasks, pedestrian detection received notably less attention from the vision community recently. Even though there are several deep learning-based approaches specialized on pedestrian detection, such as [14, 68, 117], considering typical surveillance scenarios, these perform mostly on par with traditional pedestrian detectors. This can be attributed to several facts, namely (i) visual surveillance scenarios exhibit rather small scale pedestrians due to the large field of view, whereas the object of interest is typically captured rather prominently for standard detection and classification tasks, *e.g.* within ImageNet [367] or PASCAL VOC [122]; (ii) surveillance footage often suffers from low resolution and image quality; and additionally, (iii) there is a lack of suitable training datasets specialized on visual surveillance which would be required to leverage the capabilities of deep learning-based approaches.

More traditional pedestrian detection approaches, however, were tailored for such environments. They typically rely on intensity features, *e.g.* Haar features [426, 427], image gradients, *e.g.* histograms of oriented gradients (HOG) [91], or combinations thereof, *e.g.* Aggregated Channel Features (ACF) [109]. Highly accurate pedestrian detectors can then be realized by combining these features with boosted classifiers, *e.g.* [33, 105–107, 322, 365, 399, 401], or random forests, *e.g.* [142, 374, 375]. For a more detailed review of pedestrian detection approaches, we refer the interested reader to [34, 108, 487].

Our tracking-by-detection approach imposes no assumptions on the used detector and thus, we can easily replace this black box by any off-the-shelf detector. Therefore, we will conduct a detailed performance evaluation in Chapter 5.2 to demonstrate the effect of different state-of-the-art pedestrian detectors on our MOT approach.

4.2.3 Camera Geometry

As we exploit the scene geometry for occlusion reasoning, we will briefly recapitulate the image formation process and camera model used throughout our MOT approach. In particular, we rely on the *finite projective camera model*, *i.e.* the *pinhole camera*, which assumes that no lenses are used and thus, the camera aperture is a single point (the pinhole). This model follows the principle of collinearity, *i.e.* each world point is projected onto the image plane by a straight line through the pinhole, *i.e.* the projection center. This projection can conveniently be described by the matrix

$$\mathbf{P} = \mathbf{K}\,[\,\mathbf{R}\,|\,-\mathbf{R}\mathbf{C}\,]\,, \tag{4.1}$$

where

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

encodes the intrinsic camera parameters, *i.e.* the focal length $\mathbf{f} = (f_x, f_y)^\top$ in pixels, the principal point offset $\mathbf{p} = (p_x, p_y)^\top$ in pixels, and the skew γ in case of nonsquare sensor pixels. The position and orientation of the pinhole camera *w.r.t.* a world coordinate system is specified by the translation vector $\mathbf{C} = (c_x, c_y, c_z)^\top$ and the rotation matrix $\mathbf{R} \in SO(3)$. Leveraging homogeneous coordinates [53, 179] allows to transform a 3D world point $\mathbf{X}_{\text{world}} = (x, y, z)^\top$ to the camera coordinate system via the matrix multiplication

$$\mathbf{X}_{\text{camera}} = \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ z_{\text{cam}} \end{pmatrix} = \begin{bmatrix} \mathbf{R} \,|\, -\mathbf{R}\mathbf{C} \end{bmatrix} \begin{pmatrix} \mathbf{X}_{\text{world}} \\ 1 \end{pmatrix}. \tag{4.3}$$

Then, the projected point $\mathbf{x}_{\text{image}} = (x, y)^\top$ on the image plane can be obtained by

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{x}_{\text{norm}} \\ 1 \end{pmatrix}, \tag{4.4}$$

where

$$\mathbf{x}_{\text{norm}} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x_{\text{cam}}/z_{\text{cam}} \\ y_{\text{cam}}/z_{\text{cam}} \end{pmatrix} \tag{4.5}$$

is the point coordinate after the *normalized pinhole projection*.

In practice, however, this linear projection model is not an accurate representation of the actual camera since standard lenses usually suffer from distortion, either radial distortion – which usually increases with smaller focal lengths – or tangential distortion – which is mostly due to imperfect lens design or manufacturing, resulting in not strictly collinear centers of the lens elements [441]. To allow for accurate camera-based measurements, we use an extended projection model based on [62, 183], which mitigates the distortion effects to obtain the corrected image coordinate $\mathbf{x}_{\text{image}} = (u, v)^\top$ as

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{x}_{\text{corr}} \\ 1 \end{pmatrix}, \tag{4.6}$$

where

$$\mathbf{x}_{\text{corr}} = \begin{pmatrix} \hat{x} + \hat{x}\left(\kappa_1 r^2 + \kappa_2 r^4\right) + 2\rho_1 \hat{x}\hat{y} + \rho_2\left(r^2 + 2\hat{x}^2\right) \\ \hat{y} + \underbrace{\hat{y}\left(\kappa_1 r^2 + \kappa_2 r^4\right)}_{\text{Radial distortion}} + \underbrace{\rho_1\left(r^2 + 2\hat{y}^2\right) + 2\rho_2 \hat{x}\hat{y}}_{\text{Tangential distortion}} \end{pmatrix} \tag{4.7}$$

is the corrected normalized point coordinate, *i.e.* after including the lens distortion, and

$$r^2 = \hat{x}^2 + \hat{y}^2. \tag{4.8}$$

This distortion model relies on the radial distortion coefficients κ_1, κ_2 and the tangential distortion coefficients ρ_1, ρ_2. In practice, we rectify the camera images in a pre-processing step and use the undistorted images as inputs. This allows us to use the simple matrix notation

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \mathbf{P} \begin{pmatrix} \mathbf{X}_{\text{world}} \\ 1 \end{pmatrix}, \tag{4.9}$$

to get the projected image point as

$$\mathbf{x}_{\text{image}} = \begin{pmatrix} x/w \\ y/w \end{pmatrix}. \tag{4.10}$$

4.3 Tracking by Occlusion Geodesics

We propose to solve the data association problem for causal multi-object tracking-by-detection by two complementary steps. First, we compute reliable associations using a conservative linking strategy, as discussed in Section 4.3.1. This step assigns detections to isolated and visible objects, *i.e.* handles unambiguous associations, such as the red and green objects in Figure 4.2. Second, we introduce instance-specific cost functions which model physically plausible paths through occluded regions to handle missed detections, as detailed in Section 4.3.2. Using occlusion geodesics – *i.e.* paths with minimal instance-specific costs – future detections can be reliably re-assigned to previously missed objects, such as the blue target in Figure 4.2.

The proposed occlusion geodesics build on the observation that object detectors fail primarily whenever objects are severely occluded, either dynamically by other objects or by static scene occluders, such as benches, statues or trees. Thus, we assume that missed detections are more likely to be caused by occluders rather than detector failures. Then, in order to re-assign a candidate detection to a previously lost object there must be a physically plausible path through occluded regions, as illustrated in Figure 4.2. To properly weight such a path, we propose a novel confidence measure which combines geometric knowledge of occlusion regions, target motion prediction, and object detector belief, as detailed in Section 4.3.3.

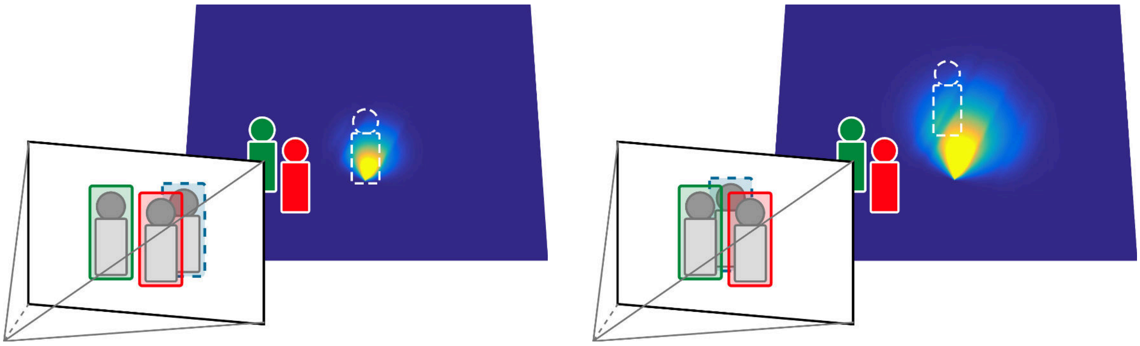
Finally, any causal MOT approach requires trajectory management capabilities to initialize and terminate target trajectories. This allows to automatically handle an unknown number of simultaneously visible targets and prevents reporting invalid trajectories, *e.g.* caused by false positive detections. We will discuss our trajectory management in Section 4.3.4.

4.3.1 Conservative Data Association

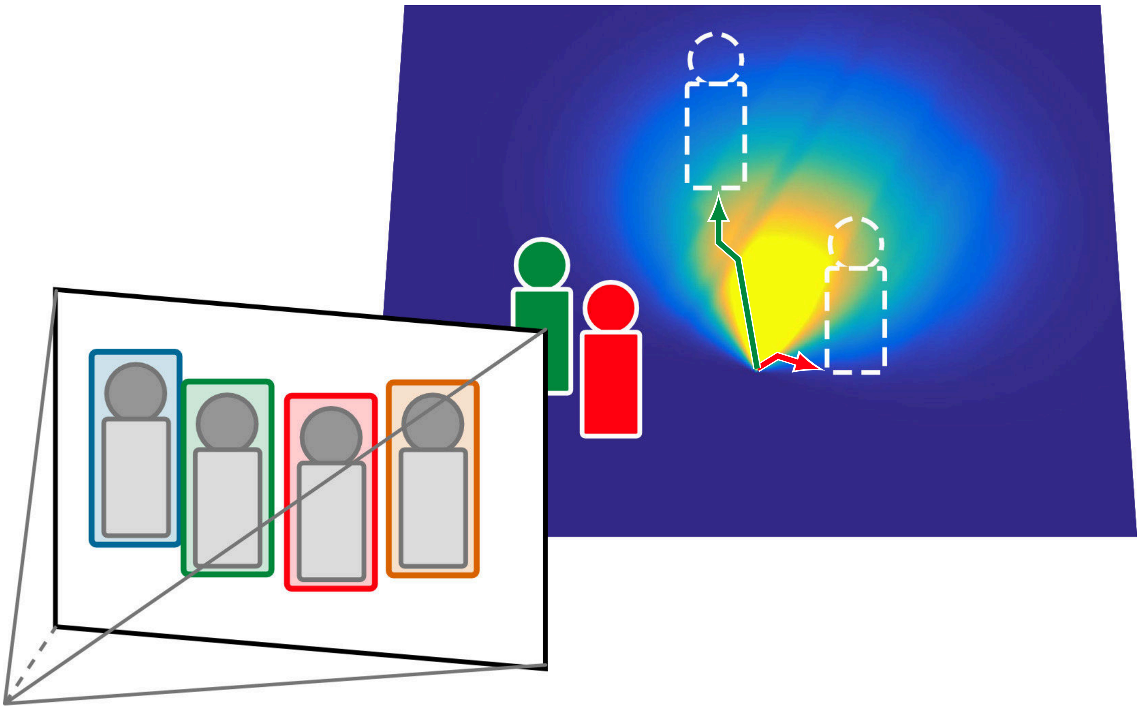
To reliably track multiple objects within a typical visual surveillance scenario, we leverage scene geometry. Therefore, similar to recent MOT approaches, such as [12, 193], we perform tracking in real-world ground plane coordinates. To this end, let

$$\mathcal{D}^{(t)} = \left\{ D_i^{(t)} \right\}_{i=1}^{N_\mathcal{D}^{(t)}}, \qquad \text{with} \qquad D_i^{(t)} = \left(\mathbf{c}_i^{(t)},\, w_i^{(t)},\, h_i^{(t)} \right)^\top, \tag{4.11}$$

denote the set of $N_\mathcal{D}^{(t)}$ object detections at time t. For notational simplicity, we assume that the detections are axis-aligned bounding boxes, represented by the tuples $D_i^{(t)}$, where $\mathbf{c}_i^{(t)}$ denotes the center in image coordinates and $w_i^{(t)}, h_i^{(t)}$ denote the width and height, respectively. Note that we slightly change the notation and denote temporal indices by parenthesized superscripts to avoid confusion with exponents in the following. Then, we project the bottom center point of a detection onto the 2D ground plane – *i.e.* the plane at world coordinate $z = 0$ – to obtain its representative position $\mathbf{x}_i^{(t)}$ *w.r.t.* the world



(a) After the first missed detection. **(b)** Halfway through the occlusion.



(c) Reassignment after occlusion. The green arrow denotes a path with minimal costs (*i.e.* maximum likelihood) *w.r.t.* our occlusion geodesics.

Figure 4.2: Evolution of the object likelihood scores on the ground plane (a),(b) for the occluded person (denoted by the dashed lines; blue identity in schematic camera view). The object likelihood maps are visualized as ground plane overlay where warm colors indicate a high likelihood score. By exploiting contextual knowledge about physically plausible movements and the spatio-temporal evolution of the occluded regions, we can assign the correct detection in (c) as we search for a shortest path *w.r.t.* the object likelihood scores. Solely relying on Euclidean distances instead, the brown detection would have been chosen, as it is closer to the last observed object position. Note that our approach explicitly assigns higher likelihood scores within occluded regions.

coordinate system as

$$\mathbf{x}_i^{(t)} = \begin{pmatrix} x_i/w_i \\ y_i/w_i \end{pmatrix}, \qquad \text{with} \qquad \begin{pmatrix} x_i \\ y_i \\ w_i \end{pmatrix} = \mathbf{H}^{-1} \left(\mathbf{c}_i^{(t)} + \begin{pmatrix} 0 \\ h_i^{(t)}/2 \end{pmatrix} \atop 1 \right), \tag{4.12}$$

where $\mathbf{H}$ is the homography matrix which maps ground plane points onto the image plane. Assuming a calibrated camera – which can be easily done for typical visual surveillance scenarios, especially when recorded from a static view point – this homography can be extracted from the camera's projection matrix $\mathbf{P}$ as

$$\mathbf{H} = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_4], \tag{4.13}$$

where $\mathbf{p}_i$ denotes the i-th column of $\mathbf{P}$ [179].

Then, we assign detections to isolated and visible objects based on spatial proximity. More formally, we represent the tracked, *i.e.* known, objects at time $t-1$ by the set

$$\mathcal{O}^{(t-1)} = \left\{ \mathcal{T}_i^{(t-1)} \right\}_{i=1}^{N_\mathcal{O}^{(t-1)}}, \tag{4.14}$$

where each object is represented by its previously observed trajectory

$$\mathcal{T}_i^{(t-1)} = \left\{ \mathbf{x}_i^u \right\}_{u=t_i^{(1)}}^{t-1}, \tag{4.15}$$

with $t_i^{(1)}$ denoting the frame at which i-th trajectory was initialized. Then, we define the cost $\psi_{i,j}^{(t)}$ of assigning detection $D_j^{(t)}$ to the i-th object as the Euclidean distance to its previously observed ground plane location, *i.e.* $\mathbf{x}_i^{(t-1)}$, as

$$\psi_{i,j}^{(t)} = \begin{cases} \|\mathbf{x}_j^{(t)} - \mathbf{x}_i^{(t-1)}\|_2 & \text{if } \|\mathbf{x}_j^{(t)} - \mathbf{x}_i^{(t-1)}\|_2 < \tau_c \\ \infty & \text{otherwise,} \end{cases} \tag{4.16}$$

where τ_c is a conservative distance threshold, and $\psi_{ij}^{(t)} = \infty$ denotes impossible assignments. To obtain the optimal assignment of reliable matches at time t, we use the Hungarian algorithm [317] for computing the binary assignment matrix $\mathbf{A}^* = \left[a_{i,j}^{(t)} \right]$, $a_{i,j}^{(t)} \in \{0, 1\}$, which minimizes the total association cost as

$$\mathbf{A}^* = \operatorname*{arg\,min}_{\mathbf{A}} \sum_{i=1}^{N_\mathcal{O}^{(t-1)}} \sum_{j=1}^{N_\mathcal{D}^{(t)}} \psi_{i,j}^{(t)}\, a_{i,j}^{(t)}, \tag{4.17}$$

$$\text{s.t.} \sum\nolimits_{i=1}^{N_\mathcal{O}^{(t-1)}} a_{i,j}^{(t)} = 1, \quad \forall j \in \left\{1, \ldots, N_\mathcal{D}^{(t)}\right\},$$

$$\sum\nolimits_{j=1}^{N_\mathcal{D}^{(t)}} a_{i,j}^{(t)} = 1, \quad \forall i \in \left\{1, \ldots, N_\mathcal{O}^{(t-1)}\right\}.$$

Since the original Hungarian algorithm assumes that $N_{\mathcal{D}}^{(t)} = N_{\mathcal{O}}^{(t-1)}$, we use an extended version [63] which can handle rectangular assignment matrices.

Any objects which could not be assigned by this conservative association step are considered to be missed by the detector. Such false negative detections are either caused by static and dynamic occluders or detection failures. Thus, future detections must be re-assigned to the corresponding trajectories whenever missed objects are re-detected, *e.g.* after they exit occluded regions. In the following, we introduce occlusion geodesics to solve this association problem efficiently.

4.3.2 Occlusion Geodesics for Data Association

To overcome missed detections, we introduce a novel confidence measure predicting the location of a missed object *w.r.t.* occlusion information, detector reliability, and motion prediction. This allows for computing weighted, physically plausible paths from the location a target was first missed up to its re-detection. Then, we leverage our confidence measure to define a path's cost and use this information to resolve an occluded trajectory, whenever a physically plausible, shortest path connects a candidate detection with the previously missed object. Since we compute a path's cost by a temporally evolving cost function – due to dynamically changing inter-object occlusions – we refer to the shortest path as *occlusion geodesic.*

More formally, let δ_i denote the occlusion length of the i-th object, *i.e.* for how long the object has been missed by the detector. Moreover, let $c_{o,i}^{(\delta_i)}$ be the *occlusion-based confidence* which accounts for occluded regions and potential detection failures, $c_{p,i}^{(\delta_i)}$ the *plausible motion confidence* which constrains physically feasible object movement, and $c_{d,i}^{(\delta_i)}$ the *directional motion confidence* based on the object's inertia model. Then, we define

$$\varphi_i^{(\delta_i)}(\mathbf{x}) = c_{o,i}^{(\delta_i)}(\mathbf{x})\, c_{p,i}^{(\delta_i)}(\mathbf{x})\, c_{d,i}^{(\delta_i)}(\mathbf{x}) \tag{4.18}$$

to indicate the likelihood that the i-th object is present at the ground plane location $\mathbf{x}$, after being missed by the detector for δ_i frames. The corresponding confidence terms will be defined in the following section.

Note that the object presence likelihood changes over time, due to changing inter-object occlusions – *e.g.* whenever occluders move – and the motion uncertainty of the occluded object. Thus, we have to explicitly address the spatio-temporal evolution of these likelihood scores in order to reliably re-assign detections to a previously missed object. In particular, we assume that an occluded object moves with an average velocity v_{avg} between subsequent frames. Then, we can weight physically plausible paths by the recursive cost function

$$\Psi_i^{(\delta_i)}(\mathbf{x}) = 1 - \varphi_i^{(\delta_i)}(\mathbf{x}) + \inf_{\mathbf{z}} \Psi_i^{(\delta_i - 1)}(\mathbf{x} + \mathbf{z})\,. \tag{4.19}$$

Accumulating the infima within the spatial neighborhood $\mathbf{x}+\mathbf{z}$, $\|\mathbf{z}\| \leq v_{\text{avg}}$ over time ensures that $\Psi_i^{(\delta_i)}(\mathbf{x})$ always contains the minimum cost of all feasible paths which lead from the i-th object's last known position up to location $\mathbf{x}$. The initial re-assignment cost for the recursive computation is set to $\Psi_i^{(0)} = 0$.

An alternative formulation would be to create a 3D cost volume, where at each time step δ_i the corresponding cost for all points on the ground plane would be stored as a separate slice of the volume – thus forming one temporal, *i.e.* δ_i, and two spatial, *i.e.* $\mathbf{x}$, dimensions. Then, we could search for the shortest path through the cost volume for every re-assignment candidate. This solution, however, would be computationally inefficient. On the one hand, it requires storing a separate 3D cost volume for each occluded object, and on the other hand, we do not need the exact shortest path for re-assignment. Instead, we only need to decide, whether a re-detection candidate is (physically) feasible and has minimal cost *w.r.t.* to our object likelihoods. Thus, we accumulate the minimum re-assignment cost recursively as in Eq. (4.19), since (i) this requires storing only an up-to-date 2D cost map $\Psi_i^{(\delta_i)}$ per occluded object and (ii) only takes a single lookup into this map to obtain the (minimum) cost of the best path leading to the corresponding location.

Similar to the conservative association step, we use the Hungarian algorithm – recall Eq. (4.17) – to obtain the optimal assignment between previously missed objects and candidate re-detections at time t. In particular, given the ground plane location $\mathbf{x}_j^{(t)}$ which corresponds to the re-detection candidate $D_j^{(t)}$, we set the assignment costs to $\psi_{i,j}^{(t)} = \Psi_i^{(\delta_i)}\left(\mathbf{x}_j^{(t)}\right)$.

4.3.3 Contextual Cues for Confidence Scores

In the following, we define the confidence terms used to compute the object likelihood measure $\varphi_i^{(\delta_i)}$ from Eq. (4.18). To this end, we will combine occlusion knowledge, detector belief and object motion reasoning.

Occlusion-based Confidence. State-of-the-art object detectors, *e.g.* [109, 135, 362], typically yield highly accurate detection results, even for partially occluded objects. Thus, we expect the object detector to primarily miss an object only if (i) it is mostly occluded or (ii) environmental conditions cause detection failures,*e.g.* due to illumination changes. Therefore, we define the occlusion-based confidence term $c_{o,i}^{(\delta_i)}$ as

$$c_{o,i}^{(\delta_i)}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathcal{P}_{\text{stat}} \cup \mathcal{P}_{\text{dyn}}^{(t)} \\ 1 - \beta^{\delta_i} & \text{otherwise,} \end{cases} \tag{4.20}$$

where $\mathcal{P}_{\text{stat}}$ and $\mathcal{P}_{\text{dyn}}^{(t)}$ denote the occluded regions caused by static occluders and dynamic occluders at time t, respectively. Our trust in the object detector is reflected by the

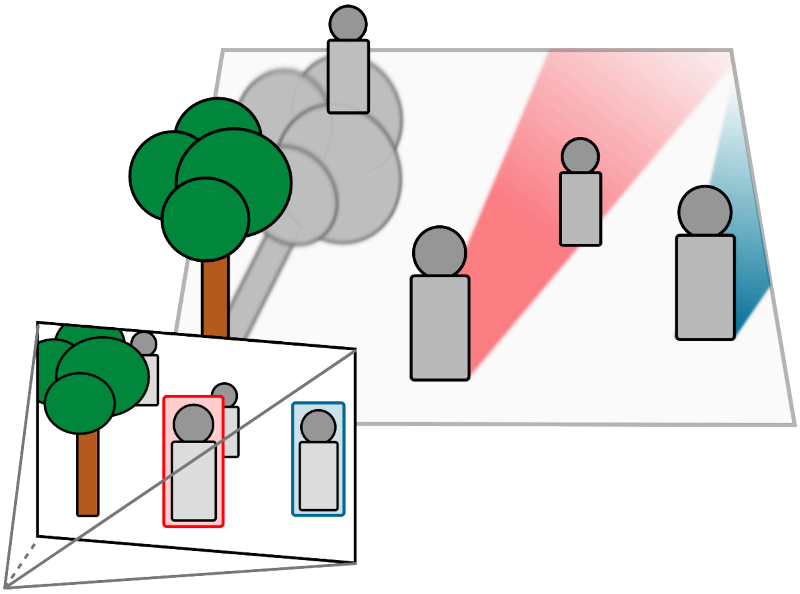


Figure 4.3: Exemplary scenario with corresponding occluded regions on the ground plane. Depending on the camera view point, the projected occlusions can take up large portions of the tracking area. The gray shadow denotes a static occlusion region – *i.e.* $\mathcal{P}_{\text{stat}}$, caused by the tree in the foreground – whereas inter-object occlusions may change over time and thus, are considered dynamic occlusion regions – *i.e.* $\mathcal{P}_{\text{dyn}}^{(t)}$, indicated by the red and blue shadows.

reliability factor $\beta \in [0, 1]$, which can be set close to one in the (theoretical) case that we expect the detector to make almost no failure at all.

Depending on the camera geometry, large parts of the tracking area may be occluded, as illustrated in Figure 4.3. Occlusion regions $\mathcal{P}_{\text{stat}}$ caused by static obstacles or scene structures can easily be provided as a predefined mask since they don't change over time. To obtain the dynamic occlusion regions $\mathcal{P}_{\text{dyn}}^{(t)}$, on the other hand, we exploit the geometric knowledge of the currently visible objects. In standard pedestrian surveillance settings, we can rely on the given person detections as the vast majority of objects in such scenarios are persons. For more generic applications with several object classes, we would require multi-class detectors trained for all relevant classes, *e.g.* people and cars. A more viable solution, however, is to either leverage semantic segmentation approaches to estimate which objects are currently visible, or to use motion detection techniques, such as background subtraction within static camera setups. Then, we can exploit the bounding rectangles of either segmented or moving object regions as potential (dynamic) occluders. Given such detection hypotheses of currently visible objects, we then project the corner points of each detection $D_i^{(t)}$ onto the ground plane – recall Eq. (4.12) – and consider the corresponding polygon to be occluded, *i.e.* objects within these regions will most likely be missed by the detector. Thus, paths through occluded regions should be favored when deciding about which candidate detection should be used for re-assignment.

Plausible Motion Confidence. In order to restrict the re-assignment candidates to detections which can be reached via physically plausible motion of the target, we define the plausibility term

$$c_{p,i}^{(\delta_i)}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\hat{\mathbf{x}}_i\|_2^2}{2\,\sigma_p^2\,\delta_i^2\,\max\left(\|\hat{\mathbf{d}}_i\|_2, v_{\text{avg}}\right)^2}\right), \tag{4.21}$$

where σ_p^2 denotes the motion variance, $\hat{\mathbf{x}}_i$ is the last known position of the i-th object at $\delta_i = 0$, and $\hat{\mathbf{d}}_i$ is its previously observed movement direction. To estimate $\hat{\mathbf{d}}_i$, we consider the previously observed target motion between subsequent frames and compute the interquartile mean to robustly handle outliers, *e.g.* which may arise due to inaccurate localization by the detector or camera calibration errors.

We also use this term to enforce the hard constraint that the distance between the last known target position $\hat{\mathbf{x}}_i$ and the ground plane location $\mathbf{x}$ must lie within physically feasible limits. To this end, we employ the predefined cut-off threshold τ_p and set $c_{p,i}^{(\delta_i)} = -\infty$, if $c_{p,i}^{(\delta_i)} < \tau_p$. Out of implementation considerations, we normalize the distances by the maximum feasible object movement at every occluded time step δ_i – thus, we threshold the plausible motion confidence $c_{p,i}^{(\delta_i)}$ directly and can employ a fixed cut-off threshold. Alternatively, we could apply a threshold on the distance $\|\mathbf{x}-\hat{\mathbf{x}}_i\|_2$, but this would require a temporally adaptive threshold.

Directional Motion Confidence. Finally, we also consider the object's inertia and penalize drastic changes of the object movement direction during occlusions. To this end, we exploit the available previous observations of its trajectory – in particular, its motion direction $\hat{\mathbf{d}}_i$ – and define

$$c_{d,i}^{(\delta_i)}(\mathbf{x}) = \exp\left(-\frac{\left(\left\langle \hat{\mathbf{d}}_i, \mathbf{d}_j \right\rangle - \|\hat{\mathbf{d}}_i\|\|\mathbf{d}_j\|\right)^2}{2\,\sigma_d^2\,\|\hat{\mathbf{d}}_i\|^2\,\|\mathbf{d}_j\|^2}\right), \tag{4.22}$$

where $\mathbf{d}_j = \mathbf{x}-\hat{\mathbf{x}}_i$ is the vector from the last known object position $\hat{\mathbf{x}}_i$ to the ground plane location $\mathbf{x}$. The directional variance σ_d^2 can be used to penalize significant changes of the motion direction. Choosing a small directional variance can be beneficial in scenarios where the object direction can easily be predicted or constrained by the scene layout, *e.g.* when observing pedestrians on a narrow sidewalk.

Exemplary Spatio-temporal Confidence Evolution. Combining these confidence measures as in Eq. (4.18) yields a time-dependent object likelihood measure. By recursively accumulating these confidence scores as in Eq. (4.19), we obtain a spatio-temporally evolving cost function, which we rely on to re-assign detections to previously occluded objects. This is illustrated for a real-world sequence in Figure 4.4. Here, we visualize the

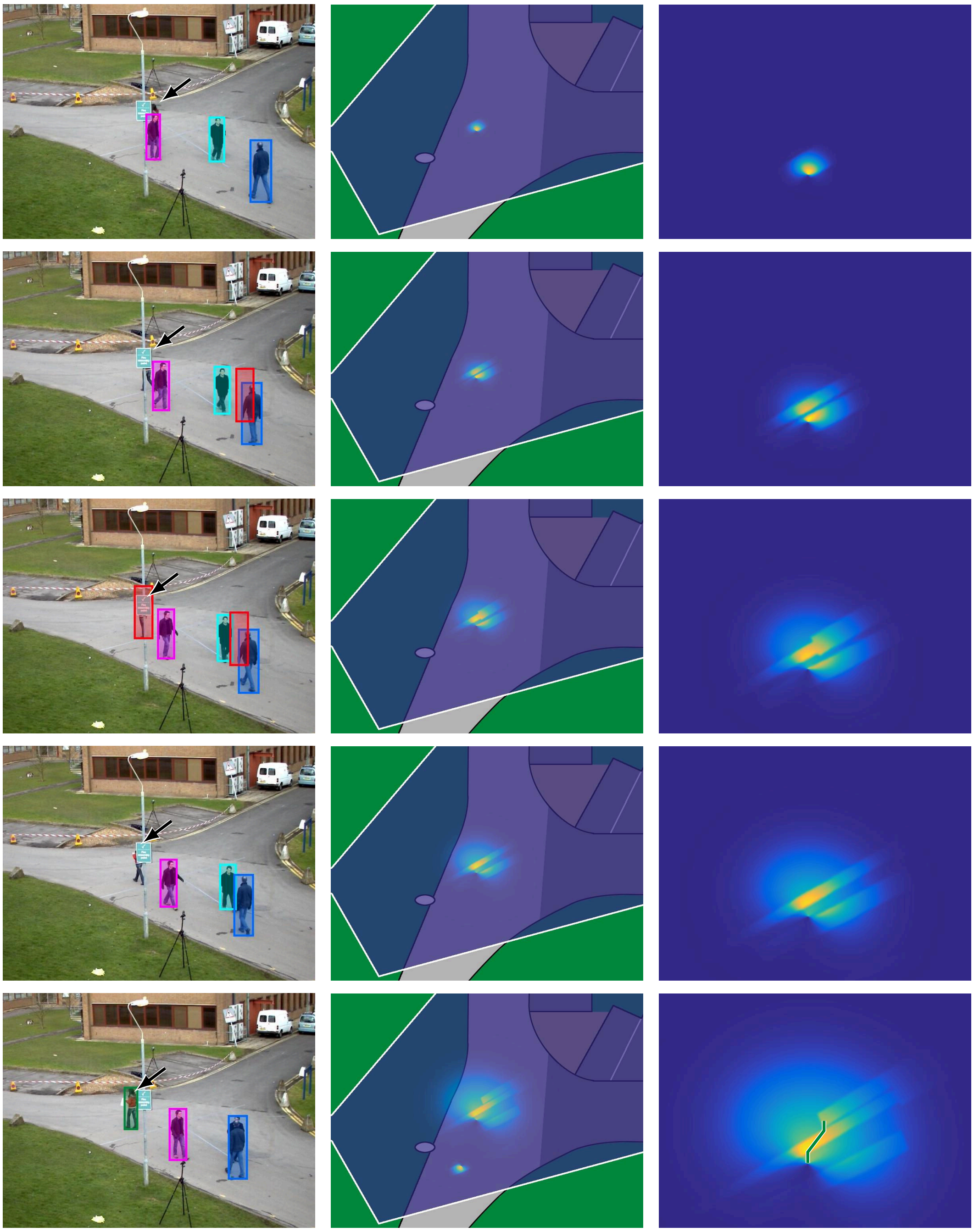


Figure 4.4: Re-assignment example on PETS'09 S2L1 [136] for the occluded woman, indicated by the arrow (left column). From top to bottom, each row corresponds to a specific time step and shows: (left) The camera view with superimposed detections – magenta, cyan, blue and green boxes show the corresponding object identity, whereas red boxes illustrate spurious, unassigned detections; (middle) Object likelihood maps and camera frustum (white border) overlaid on a schematic ground plane; (right) Close-ups of the likelihood maps for the occluded woman with overlaid re-assignment path (bottom row). The last ground plane overlay (bottom row, middle) also shows the object likelihood map for another occlusion (cyan identity) – note, however, that this is only for visualization as we compute separate object likelihood maps for each object to avoid identity switches in more crowded scenarios.

inverted cost function, *i.e.* $1 - \Psi_i^{(\delta_i)}$, and thus, regions with warm colors indicate high object likelihood scores. Note how these likelihood scores indicate that it is more likely for the missed object to move within occluded regions – *i.e.* regions where we know that the detector cannot see the object – and thus, such regions have significantly higher likelihood scores and consequently, result in paths with low re-assignment costs.

4.3.4 Trajectory Management

Despite robust associations of detections to objects, trajectory management is another crucial component in any MOT framework. This component needs to deal with track initialization, termination, as well as filtering out invalid tracks, *e.g.* caused by false positive detections. For these tasks, offline trackers have a clear advantage over causal approaches. In particular, by optimizing over all detection-trajectory assignments within a batch of frames, both new and exiting objects can be identified more easily. Additionally, spurious false positive detections can also be filtered more effectively considering the observations over a larger temporal window. Causal trackers on the other hand, must decide almost immediately whether to report detections as reliable trajectories or not.

Similar to several recent MOT approaches, such as [60, 137, 193], we employ a simplistic trajectory management strategy by explicitly defining entrance and exit regions near the image borders. Whenever we observe a stable trajectory within the entrance regions – *i.e.* subsequent close-by detections with sufficient detector confidence over a time span of approximately 1/2 second – we initialize a new trajectory and start reporting it. Similarly, we terminate existing trajectories if the corresponding objects move outside the field of view or get lost within the exit region.

4.4 Summary

We presented a causal multiple object tracking-by-detection approach which relies on occlusion geodesics – *i.e.* shortest paths *w.r.t.* novel object likelihood confidence scores – to resolve ambiguous tracking scenarios. To account for detection failures, we exploit geometric context, particularly the spatio-temporal evolution of occlusion regions, target motion prediction, and our trust in the used object detector. Using these cues to model physically plausible paths of missed objects, we can reliably re-assign detections to re-appearing objects. In combination with a conservative association strategy for visible objects, multiple objects can robustly be tracked, even in crowded scenarios. Note that in contrast to state-of-the-art approaches, such as [60, 187, 476], which rely on appearance information to resolve occluded trajectories, we only exploit the available geometric information to highlight the favorable performance and simplicity of the proposed occlusion geodesics. In Chapter 5.2, we will present extensive evaluations on several challenging real-world visual surveillance scenarios to demonstrate the benefits of our MOT approach, compared to both causal and offline state-of-the-art trackers.

5 Empirical Evidence

Now these points of data make a beautiful line.

— GLaDOS (Portal)

Contents

5.1 Distractor-Awareness to the Test

We now investigate the performance of our appearance-based, distractor-aware visual tracking approach. In particular, we focus on monocular single-target tracking scenarios. We will briefly review the relevant datasets and evaluation protocols in Sections 5.1.1 and 5.1.2, respectively. We then perform a parameter ablation study in Section 5.1.3 and compare our approach against the state-of-the-art on the Visual Object Tracking (VOT) benchmarks in Section 5.1.4 and the Online Tracking Benchmark (OTB) in Section 5.1.5. Finally, we provide runtime and implementation details in Section 5.1.6 and conclude the single object tracking evaluation in Section 5.1.7.

5.1.1 Datasets

Up until a few years ago, performance of tracking approaches has usually been demonstrated only on a handful of selected video sequences, *e.g.* refer to the evaluations of state-of-the-art approaches published at major conferences, such as MIL [17], PaFiSS [30], MOSSE [55], HoughTrack [156], Struck [176], CSK [188], TLD [216] or IVT [364]. This practice, however, made it prohibitively difficult to reason about the generalization capabilities of a tracker or its performance on slightly different scenarios. To overcome this lack of standardized datasets and evaluation protocols, several initiatives aimed at providing diverse datasets which cover realistic and challenging test sequences, *e.g.* the Amsterdam Library of Ordinary Videos (ALOV++) for tracking [386], the benchmark for isolated Apparent Motion Patterns (AMP) [475], the Need for Speed (NfS) [146] benchmark, the NUS People and Rigid Objects (NUS-PRO) dataset [265], the Online Tracking Benchmarks (OTB) [448, 449], the Princeton Tracking Benchmark (PTB) [392], the Temple Color (TColor) [274] dataset, the Visual Object Tracking (VOT) challenges [238–243] and others. Out of these publicly available datasets, we select two widely used benchmarks for our evaluations, namely VOT and OTB.

The VOT benchmarks provide a standardized evaluation framework with carefully selected sequences covering major tracking challenges, such as severe illumination changes, object deformations and appearance changes, abrupt motion changes, significant scale variations, camera motion and occlusions. Considering the number of submitted tracking approaches, the VOT challenges are the largest single object tracking benchmarks to date. The sequences contained in the VOT datasets have been collected from a large video pool, covering recent tracking evaluations, *e.g.* ALOV++ [386] and OTB-50 [448], as well as sequences published alongside major approaches, including FragTrack [1], HoughTrack [156], ABHMC [248, 251], VTD [249], and IVT [364]. In particular, the VOT committee proposed a sequence selection methodology to compile datasets which cover various real-life visual phenomena while keeping the number of sequences reasonably low. There are detailed per-frame labels of different visual attributes for each sequence which allows a less biased performance analysis. Additionally, the evaluation protocol explicitly addresses the

Table 5.1: Overview of the sequences and experiments provided by each benchmark. For each dataset, we list the number of videos, total number of frames, minimum and maximum length of its videos as well as the mean length and standard deviation. We also report whether a benchmark experiment detects tracking failures and re-initializes the tracker (*Supervised*) or only invokes the tracker once without resetting after drifting (*Unsupervised*), and if an experiment allows to initialize the tracker with perturbed annotations (*Perturbed*). VOT'15 and VOT'16 share the same set of sequences (with refined annotations for VOT'16). All OTB-50 sequences are also contained in OTB-100. Note that OTB-50 has 50 tracking sequences but only 49 distinct videos, as one video has two annotated targets. Similarly, OTB-100 has 100 tracking sequences with 98 distinct videos. These videos, however, are only considered once to obtain the frame statistic. Similarly, we only count the number of annotated frames in OTB, in contrast to the statistic provided by [449].

Benchmark	**Num. Videos**	**Number of Frames**				**Experiments**		
		Total	**Min**	**Mean**	**Max**	**Sup.**	**Unsup.**	**Pert.**
VOT'13 [238]	16	5681	172	355 ± 158	770	✓		✓
VOT'14 [239]	25	10213	164	409 ± 248	1210	✓		✓
VOT'15 [240]	60	21455	41	358 ± 266	1500	✓		
VOT'16 [241]	60	21455	41	358 ± 266	1500	✓	✓	
OTB-50 [448]	49	26499	71	541 ± 433	1918		✓	✓
OTB-100 [449]	98	58260	71	595 ± 603	3872		✓	✓

statistical significance of the results and allows to reason about the equivalence of trackers. Trackers are run multiple times on each sequence to obtain a better statistic on their performance and most VOT experiments are supervised, *i.e.* the evaluation framework detects tracking failures and re-initializes the tracker accordingly. This supervision allows minimum-variance and unbiased estimates of its performance in contrast to unsupervised experiments, where the tracker is not re-initialized after drifting away from the target, as shown by Kristan *et al.* [242]. The VOT challenges are organized annually and constantly refine the evaluation framework as well as the benchmark dataset to contain challenging and still unsolved sequences.

Complementary, we also evaluate on the OTBs as they contain additional sequences published at major literature in recent years. In contrast to VOT, these benchmarks focus on unsupervised evaluation, *i.e.* a tracker is initialized only once per sequence. Thus, trackers which can detect failures – for example, losing the target due to occlusions or whenever the target moves outside the field-of-view – and recover, *i.e.* re-detect the target afterwards, achieve notably better performance scores. The OTBs provide per-sequence attributes to identify challenging test videos, *e.g.* caused by illumination variations, occlusions or non-rigid deformation.

Table 5.1 provides a general overview of the benchmarks and their sequences. Figure 5.1 illustrates the sequence characteristics more detailed. As shown in the box plots, more recent benchmark versions introduce significantly more challenging sequences which exhibit larger object and camera motion, larger and faster scale changes and more diverse object sizes. Overall, most tracking scenes capture objects at a scale that its bounding box diagonal is approximately 100 pixels long. Assuming a square annotation for simplicity, this corresponds to an average object size of 70×70 pixels. The change plots in Figure 5.1 (b)–(d) allow to identify datasets with very challenging sequences. In particular, sudden and significant scale or motion changes will often cause immediate (or at least subsequent) tracking failures – independent from the tracker's underlying feature representation – as many trackers assume temporal consistency *w.r.t.* object or camera motion. Nevertheless, such an assumption is valid for the majority of the frames contained in all datasets, as indicated by the shown interquartile ranges.

Note that VOT'15 and VOT'16 contain the same sequences. Thus, we skip evaluations on VOT'15 and instead report our results on VOT'16, which provides refined ground truth annotations. Similarly, we will skip evaluations on OTB-50 since its sequences are a subset of the larger OTB-100 sequence pool. Exemplary frames of all datasets are shown in Figure 5.2 along with illustrative tracking results. For more details about the datasets we refer the interested reader to the respective publications.

5.1.2 Evaluation Metrics and Protocols

Evaluation metrics analyze how well a tracker's estimated object trajectory $\mathcal{T}_\mathrm{T}$ agrees with the annotated ground truth trajectory $\mathcal{T}_\mathrm{G}$, where we define the object state description throughout a video sequence of length N as

$$\mathcal{T}_\mathrm{T} = \left\{O_\mathrm{T}^t\right\}_{t=1}^N, \quad O_\mathrm{T}^t = \left(\mathbf{x}_\mathrm{T}^t,\, w_\mathrm{T}^t,\, h_\mathrm{T}^t\right)^\top, \tag{5.1}$$

and

$$\mathcal{T}_\mathrm{G} = \left\{O_\mathrm{G}^t\right\}_{t=1}^N, \quad O_\mathrm{G}^t = \left(\mathbf{x}_\mathrm{G}^t,\, w_\mathrm{G}^t,\, h_\mathrm{G}^t\right)^\top. \tag{5.2}$$

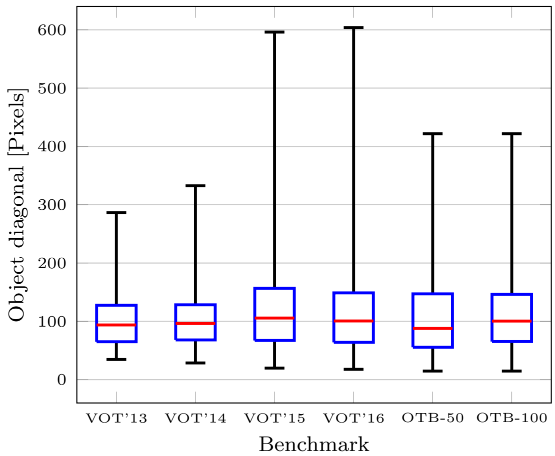
For notational simplicity, we represent the object state at frame t by the tuple O_T^t and O_G^t, respectively, *i.e.* axis-aligned bounding boxes centered at location $\mathbf{x}_\Omega^t$, $\Omega \in \{\mathrm{T}, \mathrm{G}\}$, with width w_Ω^t and height h_Ω^t. Note however, that these metrics can be easily extended to more complex or more general object state representations.

Due to the previous lack of standardized and widely accepted benchmark datasets, several performance metrics have been established to analyze tracking approaches over the years. The most commonly used and relevant metrics are:

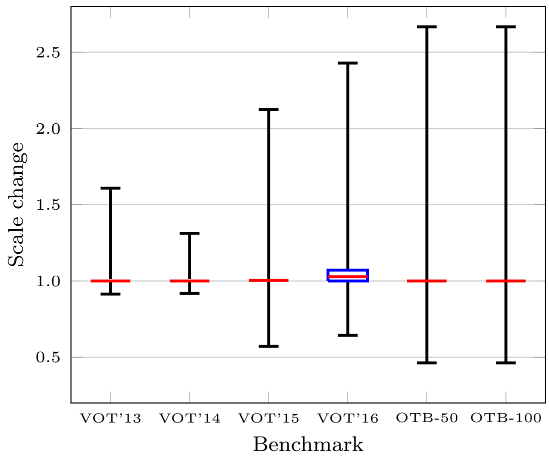
- *Center distance error* – one of the simplest and widely used performance measures, *e.g.* [1, 18, 249, 364, 371, 429, 448, 463]. This metric reports the per-frame distance between the estimated object center and the ground truth center as

$$\Delta\left(\mathcal{T}_\mathrm{T}, \mathcal{T}_\mathrm{G}\right) = \left\{\delta^t\right\}_{t=1}^N, \quad \text{where} \quad \delta^t = \left\|\mathbf{x}_\mathrm{T}^t - \mathbf{x}_\mathrm{G}^t\right\|_2. \tag{5.3}$$

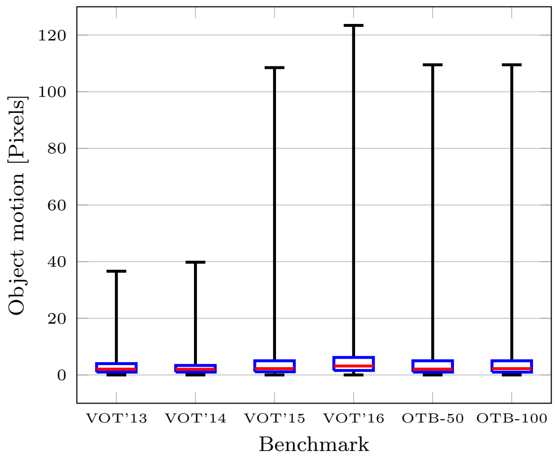
  This metric requires the least annotation effort, *i.e.* only the object center must be annotated per frame, but is also very sensitive to subjective annotation and ignores



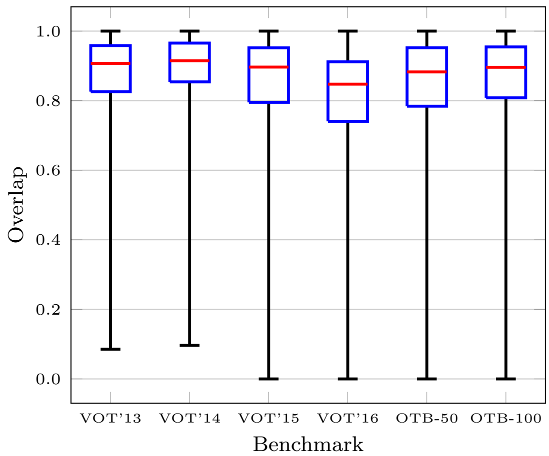
(a) Object size (bounding box diagonal).



(b) Frame-to-frame scale change.



(c) Frame-to-frame object motion.



(d) Frame-to-frame ground truth overlap.

Figure 5.1: Dataset characteristics showing the distribution of (a) object sizes, (b) relative scale changes between subsequent frames, (c) object motion between subsequent frames and (d) overlap of ground truth annotations between subsequent frames. Each box plot shows the median, first and third quartiles as well as the minimum and maximum data values. For this analysis, we removed invalid ground truth annotations, *e.g.* at sequence *car1* of VOT'16 or *Board* of OTB-50 and OTB-100. For visualization purposes, interquartile ranges in (b) are omitted if they are too close to the median. A relative scale change of 1 indicates that the object size did not change between subsequent frames. Significant frame-to-frame scale changes are caused by object deformations – *e.g.* an athlete performing a somersault captured at the *gymnastics* videos in VOT and OTB – or video cuts which abruptly change the field-of-view – *e.g.* the *DragonBaby* sequence in OTB. Zero overlap in (d) is caused by large object or camera motion and video cuts.

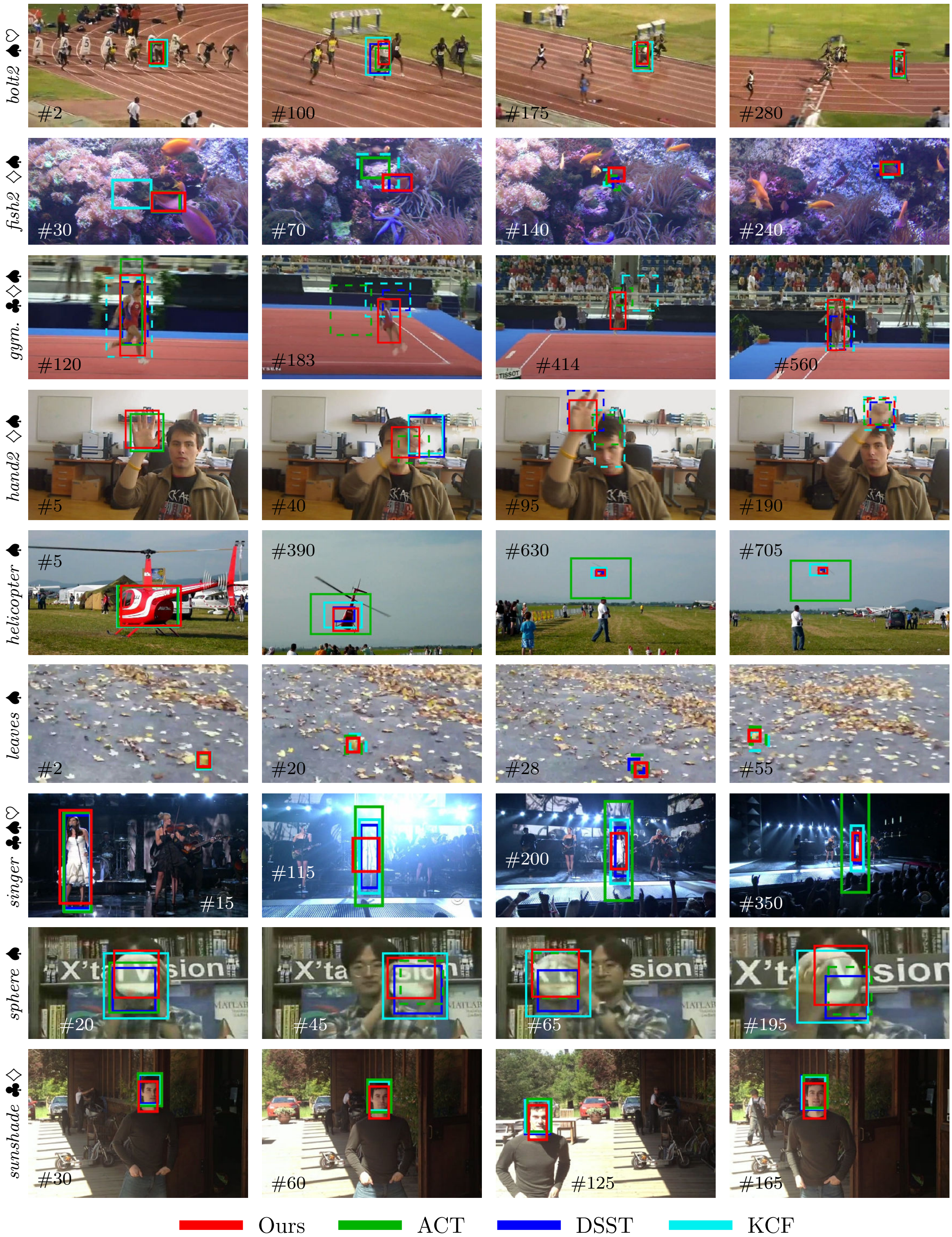


Figure 5.2: Qualitative results for our distractor-aware tracker on sequences of the ♣ VOT'13 [238], ◇ VOT'14 [239], ♠ VOT'16 [241] and ♡ OTB-100 [449] datasets. Results for ACT [93], DSST [92] and KCF [189] are also shown. Dashed bounding boxes indicate that the corresponding tracker has been re-initialized after losing the target previously. Images are slightly cropped and frame numbers are superimposed for visualization only.

the object size. Some evaluations address these limitations via object size dependent normalization factors, *e.g.* [27, 386].

- *Region overlap* – inspired by object detection and classification benchmarks, such as the PASCAL Visual Object Classes (VOC) challenge [122], several authors adopted the region overlap measure, *e.g.* [156, 251, 266, 386, 481]. This metric reports the per-frame intersection over union (IOU, also known as *Jaccard index* or *Jaccard similarity coefficient*) between the tracker's hypothesis and the ground truth region as
$$\Phi(\mathcal{T}_\mathrm{T}, \mathcal{T}_\mathrm{G}) = \left\{\phi^t\right\}_{t=1}^{N}, \quad \text{where} \quad \phi^t = \frac{O_\mathrm{T}^t \cap O_\mathrm{G}^t}{O_\mathrm{T}^t \cap O_\mathrm{G}^t}. \tag{5.4}$$
This metric allows to reason about both the distance precision and the scale adaptation capabilities of a tracker.

- *Tracking length* – measures the number of successfully tracked frames from initialization to the first tracking failure [248]. To this end, usually a threshold is applied on the center distance or overlap measure. Although this metric explicitly addresses tracking failures, it may bias the evaluation if accidentally the beginning of a video captures a very challenging tracking scenario where almost all trackers fail, *e.g.* a video cut or a sudden illumination change, such as a whiteout caused by a flash light.

- *Failure rate* – as used in [71, 72, 228, 236–241, 243] requires a supervised evaluation framework, in which a tracker is re-initialized once it fails. This measure reports the number of tracking failures and reflects real-world scenarios where a human operator supervises the tracker and manually corrects its errors.

- *Performance plots* – visualize the performance of a tracker based on a specific evaluation metric. The most widely used plot is the *center error versus frame number* plot, *e.g.* [1, 18, 27, 481]. Another important visualization technique are *measure-threshold* plots which allow intuitive visual comparison and can be computed similar to *receiver operating characteristic* (ROC) curves [130]. These measure-threshold plots are widely used within OTB [448, 449], where center error and region overlap are used as measure, respectively. A notable limitation of such evaluation curves is that including too many competing approaches clutters the plots significantly. To avoid this, the maximum number of included trackers should be limited.

Performance metrics are usually averaged over all sequences of the dataset to obtain a single score per tracker. As shown by Čehovin *et al.* [73, 74] and Smeulders *et al.* [386], several tracking metrics are highly correlated, which should be considered when defining an evaluation protocol for a novel dataset.

The VOT benchmarks explicitly aim at evaluating monocular, online single-target tracking approaches on short-term sequences. In such a short-term setting, trackers are not supposed to perform re-detection as the target usually never (fully) leaves the field-of-view. Thus, the VOT benchmarks provide a supervised evaluation framework, which re-initializes a tracker once it fails – in particular, as soon as the overlap between the tracker's hypothesis and the ground truth is zero, *i.e.* $\phi^t = 0$. To avoid introducing a bias, several frames after each failure are skipped prior to re-initialization, as the subsequent frames very likely also capture the same difficult situation which caused the failure in the first place. Trackers are run multiple times on each sequence to obtain a better statistic on their performance. Tracking performance is evaluated primarily based on *accuracy*[5] (*i.e.* region overlap) and *robustness*[6] (*i.e.* failure rate). These raw scores are used to rank trackers based on the statistical significance of their performance differences. Additionally, the VOT'15 challenge introduced the *expected average overlap* (EAO) metric for a clearer practical interpretation compared to the previously used combination of accuracy and robustness rankings. This measure is an estimator of the average region overlap a tracker is expected to achieve on short-term sequences with the same visual properties as the tested benchmark. Each VOT benchmark provides multiple *experiments* which define (i) whether the tracker is initialized using the ground truth annotation (*i.e. baseline* experiment) or via randomly perturbed bounding boxes (*i.e. region noise* experiment) and (ii) whether the tracker is re-initialized after each failure (*i.e. supervised*) or initialized only once and operates unattended throughout the sequence (*i.e. unsupervised*).

To compare tracking speed across different platforms, the VOT initiative introduced the *equivalent filter operations* (EFO) measure in VOT'14. This speed unit aims to remove the hardware bias which arises when comparing plain frames per second (FPS) speed measurements. To this end, the VOT framework benchmarks the hardware by measuring the time required to perform a maximum pixel filter on a single-channel image of size 600×600 pixels with a sliding window of 30×30 pixels. Dividing the measured tracking time by the time required for the filtering operation then gives the EFO speed unit.

In contrast to the VOT evaluation protocol, OTB focuses on unsupervised experiments. The most common evaluation protocol in OTB is the so-called *one-pass evaluation* (OPE), where a tracker is initialized with the ground truth annotation in the first frame and runs unattended throughout the rest of the sequence. Two additional evaluations analyze the tracking performance by perturbing the tracker initialization either temporally, *i.e.* starting at different frames, or spatially, *i.e.* by shifting and scaling the initial annotation by a predefined amount. These evaluations are called *temporal robustness evaluation* (TRE) and *spatial robustness evaluation* (SRE), respectively. The OTB-100 benchmark additionally introduced supervised experiments, namely *one-pass evaluation with restart* (OPER) and *spatial robustness evaluation with restart* (SRER). Failures in these supervised experiments are detected whenever the region overlap drops below a predefined threshold. In

[5]Accuracy scores are in the range $[0, 1]$, where higher scores correspond to better performance. We denote this by the symbol $\uparrow$ throughout our evaluations.

[6]Robustness scores are non-negative real numbers, $r \in \mathbb{R}_0^+ = \{s \in \mathbb{R} \mid s \geq 0\}$, where lower scores correspond to better performance (denoted by $\downarrow$ throughout our evaluations).

contrast to the supervised VOT experiments, trackers are re-initialized immediately after each failure, instead of skipping the next few frames. Thus, supervised results on OTB might be biased because challenging scenarios typically last longer than a single frame.

Tracking performance in OTB is primarily evaluated via *success* plots[7], *i.e.* the measure-threshold plot based on the region overlap metric, and *precision* plots[7], *i.e.* the measure-threshold plot based on the center distance error. To allow ranking trackers, two metrics are used to summarize these plots. The first measure is the *area under curve* (AUC) of the overlap success plot – which actually corresponds to the average region overlap over all sequences as shown by Čehovin *et al.* [73]. Distance precision plots are summarized by the percentage of frames with center distance error below 20 pixels, *i.e.* $\delta^t < 20$, as suggested by Babenko *et al.* [18]. Considering the median object diagonal of approximately 100 pixels throughout the sequences, this distance threshold roughly corresponds to a region overlap between the tracker hypothesis and ground truth of $1/2$ and thus, mimics standard object detection evaluations based on the PASCAL overlap criterion. In the following, we refer to this score as *representative distance precision* (RDP). The OTB framework measures a tracker's speed in frames per second (FPS) without the time required to load the images, but ignoring the potential hardware bias.

We focus our evaluations on the VOT benchmarks, where we conduct all defined experiments following the official evaluation protocol. The analysis uses the measured accuracy and robustness scores, the tracker ranking based on these scores as well as the expected average overlap (EAO) metric. Complementary experiments are performed on OTB via the most commonly used one-pass-evaluation (OPE) and analyzed using overlap success plots and distance precision curves. For more details about the evaluation protocols and metrics we again refer the interested reader to the corresponding benchmark documentations and recent surveys [73, 74, 242, 386].

5.1.3 Ablation Study

We begin our evaluation with a parameter ablation study to show the sensitivity of our distractor-aware tracker (DAT) *w.r.t.* its parameter settings. In particular, we analyze (i) suitable color spaces, (ii) color histogram representations, (iii) learning rates, (iv) window sizes, (v) parameters related to non-maxima suppression, and (vi) different scale adaptation techniques. This evaluation is divided into two sections, where we analyze parameters related to the object model first , *i.e.* (i)-(iii), and second, parameters related to localization and scaling, *i.e.* (iv)-(vi).

We will vary one parameter of DAT while keeping all others fixed to allow reasoning about the effect of each parameter. In particular, we use the default parameter settings as

[7] To clearly denote which measures are used for the OTB measure-threshold plots, we refer to these explicitly as *overlap success* plot (*i.e.* region overlap) and *distance precision* plot (*i.e.* center distance error) throughout our evaluations.

Table 5.2: Default parameter settings for the distractor-aware tracker variants (DAT). Unless stated otherwise, these parameters have been kept fixed throughout all experiments.

Parameter		Value
Color space		RGB
Histogram bins		$16 \times 16 \times 16$
Learning rate for $p^t_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$	$\eta_S \in [0, 1]$	0.05
Learning rate for $p^t_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_{\mathbf{x}})$	$\eta_D \in [0, 1]$	0.20
Scaling factor for surrounding region S	$\lambda_S \in (1, \lambda_W)$	2.00
Scaling factor for search window W	$\lambda_W \in (\lambda_S, \infty)$	4.00
NMS patch overlap	$o_\nu \in [0, 1)$	0.90
NMS reporting threshold	$\tau_\nu \in (0, 1)$	0.50

listed in Table 5.2. All ablation experiments are conducted on the VOT'13 [238] dataset. We select this dataset on purpose as it allows to demonstrate the performance difference using raw accuracy (*i.e.* average overlap per sequence) and robustness (*i.e.* average number of failures per sequence) scores. On larger benchmark datasets, such as VOT'16 [241] or OTB-100 [449], subtle performance differences (caused by minor parameter changes) might not be as obvious due to averaging over a larger number of sequences.

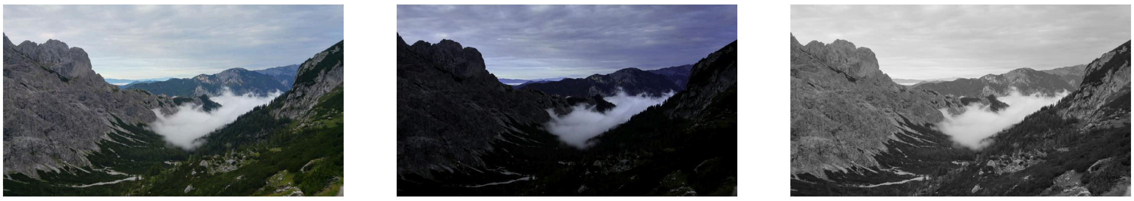
In addition to the tracking performance (*i.e.* accuracy and robustness), we also report the runtime of all parameter variations to indicate the performance versus speed tradeoff. To ensure consistent runtime measurements, all experiments have been conducted on a dedicated computer, in particular an Intel® NUC *Skull Canyon* with a 6$^{\text{th}}$ generation Core™ i7 processor, on which only the MATLAB® framework was running along with the default set of operating system processes of a clean Ubuntu 16.04.1 installation.

5.1.3.1 Object Model Parameters

Color Spaces. For this evaluation, we consider the following commonly used color spaces, which are also illustrated in Figure 5.3. We coarsely summarize these color spaces as a detailed derivation and discussion is out of scope of this thesis. For more details, we refer the interested reader to the book on color appearance models by Fairchild [123].

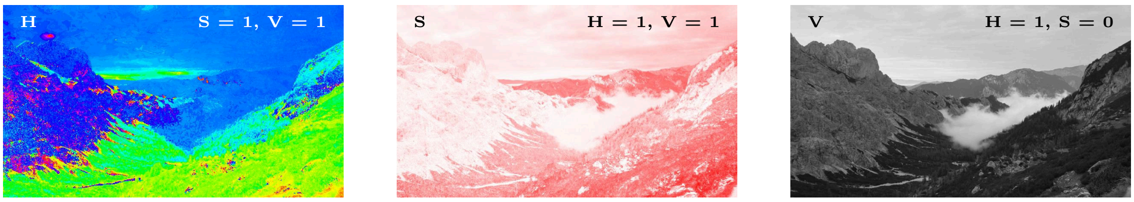
RGB RGB is the default color space we deal with in digital image processing. Each pixel is identified by a 3-dimensional vector, where each component indicates the intensity of the corresponding primary color, *i.e.* red, green and blue. Technically, RGB is not a *color space* but a *color model* – several RGB color spaces are derived from the RGB model, such as *sRGB*, the *standard RGB* color space. In the following, we use RGB to denote the color model and color space interchangeably, unless an explicit distinction is required.

We also evaluate our model using the *rg chromaticity* space – denoted *rg chroma* – which is derived from normalized RGB values. There, a color is

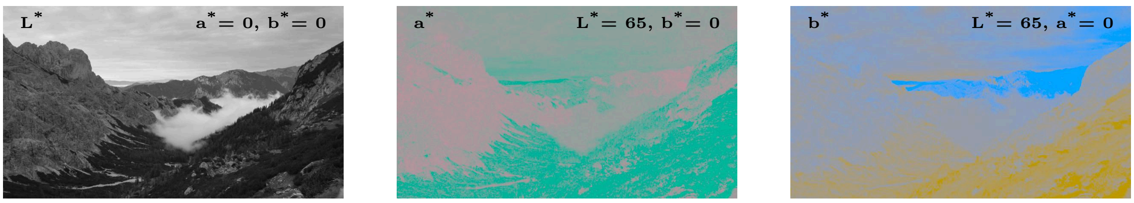


(a) RGB. (b) XYZ. (c) Grayscale.

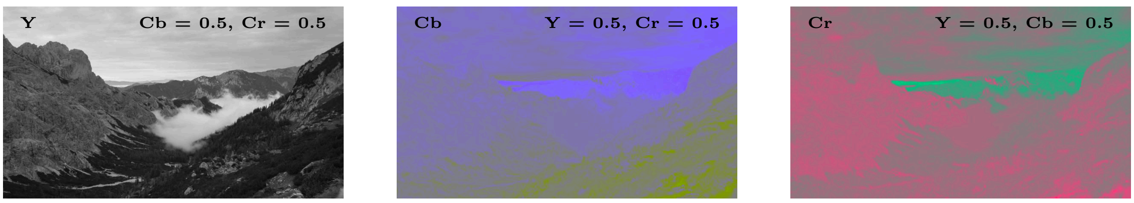
(d) Normalized RGB. (e) rg chromaticity – proportions of red (left) and green (right).



(f) HSV – hue (left), saturation (middle), value (right).



(g) $L^*a^*b^*$ – lightness L^* (left), chromaticity a^* (middle), chromaticity b^* (right).



(h) YCbCr – luminance Y (left), chrominance Cb (middle), chrominance Cr (right).

Figure 5.3: Color space representations. (a) standard RGB. (b) XYZ using the CIE standard illuminant D65 reference white point. (c) Grayscale. (d) normalized RGB – note the effect on foliage and scrubs (front and valley) as well as shadows (mountain range in the back). Red and green proportions of normalized RGB form the (e) rg chromaticity space. (f) HSV – note the distinctive appearance of fog, sky and shadows in the hue and saturation components. (g) $L^*a^*b^*$. (h) YCbCr. For better visualization, we gamma corrected normalized RGB with $\gamma = 0.4$ and stretched the contrast of the chromaticity and chrominance components. We use standard pseudo-coloring schemes to visualize separate color space components (visualization parameters are superimposed).

represented by its proportion of the primary colors instead of their intensities as in standard RGB. Since the proportions of all three primary colors for each pixel sum to one, the third dimension can be discarded and thus, rg chroma is a 2-dimensional representation, denoting the red and green proportions, respectively.

HSV HSV is a cylindrical-coordinate representation of points in the Cartesian cube spanned by the RGB color space. A color is represented by its hue (*i.e.* the H channel; angle around the central vertical axis of the cylindrical coordinate system), saturation (*i.e.* the S channel; distance from the central vertical axis), and value (*i.e.* brightness; the V channel; distance along the central vertical axis).

Additionally, we evaluate a two-channel variant which only uses the hue and saturation, denoted *HS*. By ignoring the brightness component, this variant represents only the color purity.

L*a*b* CIE[8] L*a*b* is a perceptually uniform color space, *i.e.* perceptually similar colors yield lower Euclidean distances of their respective L*a*b* vectors. A color is represented by its lightness (*i.e.* L*channel) and the two chromaticity components: (a*) its position between red and green, and (b*) its position between yellow and blue. The relations between these channels are nonlinear, mimicking the nonlinear response of the human eye.

YCbCr YCbCr represents a color by its luminance (brightness, *i.e.* Y channel, also denoted luma) and chrominance (color information, *i.e.* Cb and Cr channels, also denoted chroma) components. The Cb and Cr components denote an image's blue difference chrominance and red difference chrominance, respectively. Since the human eye is most sensitive to luminance (achromatic) changes, this color space representation allows for subsampling the chrominance components and thus, is often used for efficient storage and transmission of video data.

XYZ CIE XYZ was one of the first mathematically defined color spaces (introduced in 1931) and is a device invariant color representation. To this end, the CIE defined the tristimulus values X, Y and Z to avoid negative numbers which arise in additive trichromatic color spaces based on real (physical) primary colors (which can be created by a spectral distribution of wavelengths), such as RGB. A color is then represented as a mixture of these tristimulus values.

Gray To highlight the importance of using color, we also demonstrate our approach using only grayscale imagery. The grayscale value of a color pixel can be easily

[8] *Commission Internationale de l'Éclairage* (CIE, French name of the *International Commission on Illumination*) is the international authority on light, illumination, color and color spaces.

Table 5.3: Performance of our distractor-aware tracker (DAT) and its distractor-agnostic baseline (noDAT) *w.r.t.* different color spaces. The columns denoted *Acc.* and *Rob.* show the raw accuracy (*i.e.* overlap) and robustness (*i.e.* number of failures) scores on the two experiments *baseline* and *region noise*, respectively. These scores are averaged over all sequences of the VOT'13 dataset. The three rightmost columns show the expected average overlap (EAO) combined over both experiments and the tracking speed in frames per second (FPS). Best, second best and third best results have been highlighted in each column. The model size is 16 bins per channel, *i.e.* we use histograms of size $16 \times 16 \times 16$ for trichromatic color spaces, 16×16 for bichromatic (*i.e.* HS and rg chroma) and 16 for monochromatic color spaces (*i.e.* grayscale). The discrepancies between our frame rate measurements ($\mathrm{FPS_{Ours}}$) and the VOT toolkit ($\mathrm{FPS_{VOT}}$) are discussed in Section 5.1.3.1 (page 77). The symbol $\uparrow$ indicates that higher scores correspond to better tracking performance, whereas $\downarrow$ indicates that lower scores are better.

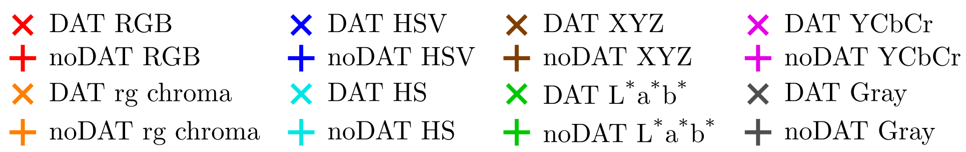
Tracker	Color Space	Experiment *baseline* Acc.$^\uparrow$	Experiment *baseline* Rob.$^\downarrow$	Experiment *region noise* Acc.$^\uparrow$	Experiment *region noise* Rob.$^\downarrow$	Overall EAO$^\uparrow$	Overall $\mathrm{FPS}^\uparrow_{\mathrm{Ours}}$	Overall $\mathrm{FPS}^\uparrow_{\mathrm{VOT}}$
DAT	RGB	0.60	0.08	0.59	0.12	0.55	113.0	64.5
noDAT	RGB	0.60	0.19	0.59	0.21	0.51	160.5	77.4
DAT	HSV	0.61	0.34	0.60	0.28	0.46	71.9	46.0
noDAT	HSV	0.61	0.42	0.60	0.35	0.43	89.1	52.5
DAT	L*a*b*	0.59	0.19	0.58	0.22	0.46	36.5	27.7
noDAT	L*a*b*	0.59	0.32	0.58	0.30	0.42	39.3	29.7
DAT	YCbCr	0.58	0.23	0.57	0.18	0.45	83.3	50.8
noDAT	YCbCr	0.58	0.15	0.57	0.22	0.46	106.2	59.5
DAT	XYZ	0.53	1.38	0.53	1.26	0.25	31.9	23.2
noDAT	XYZ	0.54	2.76	0.54	2.30	0.18	34.5	24.5
DAT	HS	0.59	0.48	0.57	0.43	0.39	79.0	47.7
noDAT	HS	0.58	0.63	0.57	0.61	0.37	95.9	53.6
DAT	rg chroma	0.57	1.39	0.56	1.29	0.20	115.0	58.2
noDAT	rg chroma	0.57	1.83	0.56	1.75	0.16	143.3	66.1
DAT	Gray	0.53	3.70	0.52	3.39	0.14	169.1	61.3
noDAT	Gray	0.52	4.51	0.53	4.66	0.11	217.4	64.3

computed as the weighted sum of its RGB components. In particular, we use the standard grayscale conversion[9], *i.e.* $\hat{G} = 0.2989R + 0.5870G + 0.1140B$.

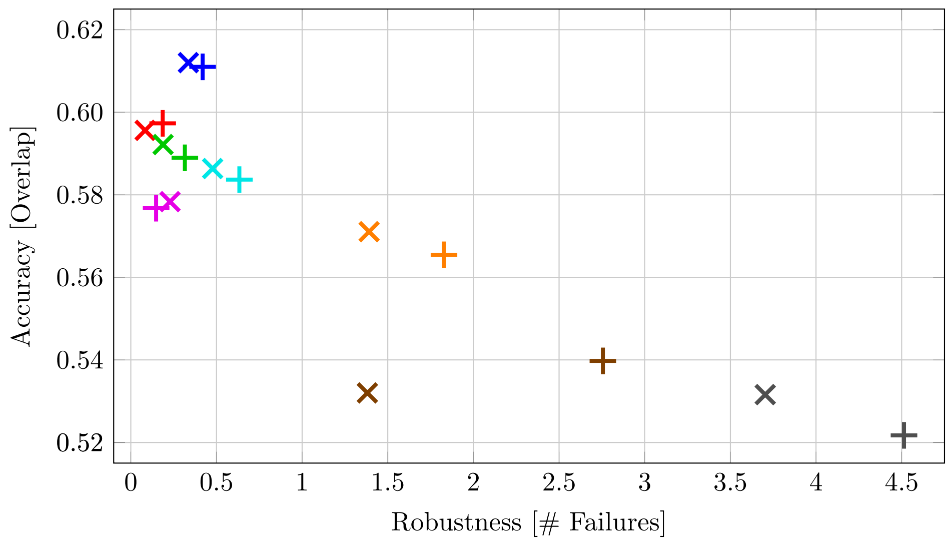
We compare our distractor-aware color models to their respective distractor-agnostic baselines, *i.e.* a tracker which only uses the object-versus-surroundings model $p_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$ but can neither identify visually distracting regions, nor suppress them. Consequently, these trackers localize the target solely relying on $p_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$ and not on the combination of $p_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$ and $p_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$ as is the case for the DAT variants. These baseline models are denoted *noDAT* throughout our evaluations.

Detailed results for the color space evaluation are listed in Table 5.3 and illustrated in Figure 5.4. Overall, our distractor-aware approaches consistently outperform the

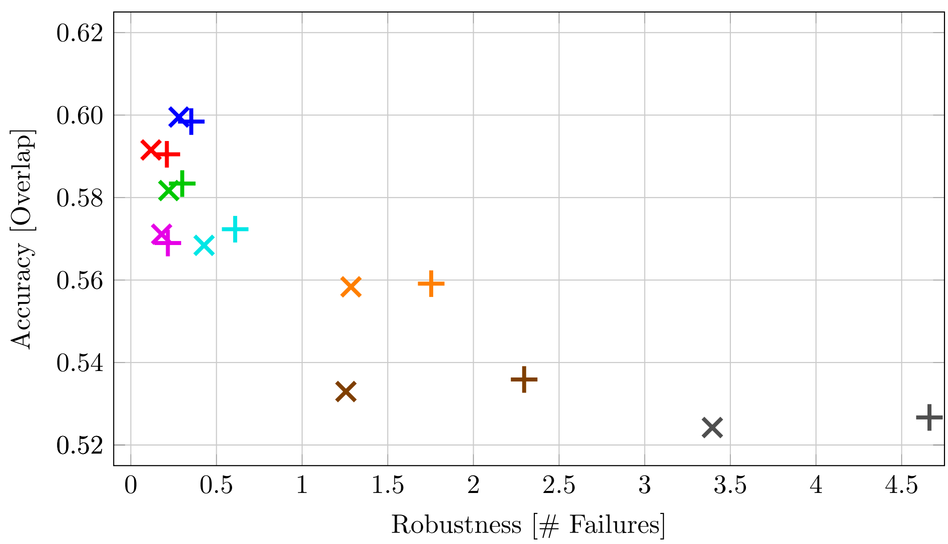
[9] These conversion weights are commonly used in digital television – for example, refer to the construction of luminance from RGB values in the recommendation ITU-R BT.601-5 of the *International Telecommunication Union* (ITU).



(a) Legend.



(b) Experiment *baseline*.



(c) Experiment *region noise*.

Figure 5.4: Accuracy-robustness plots for our distractor-aware tracker (DAT) and its distractor-agnostic baseline (noDAT) *w.r.t.* different color spaces on the VOT'13 dataset. Top-performing trackers should achieve a high overlap and low number of failures, thus be located at the top left.

distractor-agnostic baselines on all color spaces. Interestingly, using standard RGB overall results in the best combined accuracy and robustness scores. Similarly, also using the HSV color space yields very accurate but slightly less robust results. Overall, these results demonstrate the importance of modeling appearance as a joint color distribution. Trichromatic inputs consistently outperform models which rely on bichromatic or monochromatic representations, especially considering the average robustness. This is true for all trichromatic representations except for CIE XYZ, which only results in slightly better performance than using pure grayscale imagery. We hypothesize this is due to fixing the reference white point required for CIE XYZ, which may lead to low contrast imagery in many sequences.

Discarding the intensity information as in HS or rg chroma leads to frequent target loss if the object is partially transparent (*e.g.* as in the *bag* sequence of VOT'16) or due to similar colored backgrounds where intensity would be required to distinguish the object from its surroundings (as in the *hand2* sequences of VOT'14). Ignoring available color information at all yields minor speed benefits but significantly decreases the overall tracking performance, as indicated by the grayscale results. Even though one might assume that perceptually uniform color spaces, such as $L^*a^*b^*$, or chromaticity spaces, such as rg chroma (which has the benefit of illumination invariance), might be beneficial for visual tracking, our results demonstrate that standard RGB achieves the best overall performance. While there are some specific application domains which require special color representations, the task of tracking arbitrary objects using color models is best tackled by standard RGB models.

Note that noisy initializations do not negatively affect our DAT variants. This can be seen from the results for the *region noise* experiment, where both the accuracy and robustness scores do not significantly change compared to the ground truth initialization in the *baseline* experiment. This can be mostly contributed to the underlying color cue which allows to snap towards the actual object, despite randomly perturbed initializations.

The runtime performance measured in frames per second (FPS) is also listed for all color models in Table 5.3. We report two frame rate measurements, namely FPS_{Ours}, where we directly measure the processing time within the MATLAB® framework (ignoring the image loading time) and FPS_{VOT}, which is measured by the official VOT toolkit. Note that the VOT toolkit reports significantly lower frame rates with a much higher variability, although it also measures only the tracker's processing time without loading the images. This is due to the fact that a tracker runs as a separate process within the VOT framework and these timings are notably skewed by inter-process communication and process start-up time. For the remainder of this ablation study, we will only report our more accurate and consistent runtime measurements, *i.e.* FPS_{Ours}.

The runtime evaluation shows the favorable efficiency of our tracker. Despite computing two object models with 16 bins per input channel, we can easily process videos at more than 100 FPS if we rely on raw RGB input or simple color transformations, such as rg chroma or grayscale. Note that the processing bottleneck is neither the object model

Table 5.4: Performance of our distractor-aware tracker DAT with varying histogram sizes. Best, second best and third best results have been highlighted in each column.

Model Size	Experiment *baseline* Acc.$^{\uparrow}$	Rob.$^{\downarrow}$	Experiment *region noise* Acc.$^{\uparrow}$	Rob.$^{\downarrow}$	Overall EAO$^{\uparrow}$	FPS$^{\uparrow}$
$8 \times 8 \times 8$	0.58	0.19	0.57	0.21	0.47	117.9
$10 \times 10 \times 10$	0.59	0.23	0.58	0.24	0.44	124.1
$16 \times 16 \times 16$	0.60	0.08	0.59	0.16	0.53	112.1
$32 \times 32 \times 32$	0.60	0.38	0.59	0.38	0.45	105.9
$64 \times 64 \times 64$	0.59	1.17	0.57	1.05	0.29	91.2

nor the localization step, but the more complex color transformations. The most computationally demanding color space transformation is CIE XYZ, although we are still able to process at least 30 FPS, which is sufficient to realize time-critical applications. Overall, the best color space choice would be RGB, as these models yield the best combined accuracy and robustness scores at very high frame rates.

Histogram Size. The next important model component in our analysis is the histogram size, *i.e.* the number of bins per channel. We use uniform binning to model the joint color distribution. Table 5.4 summarizes the results for different model sizes, using $8, 10, 16, 32$ and 64 bins per channel, respectively.

Overall, the RGB model with $16 \times 16 \times 16$ bins achieves the best results, where we group $^{256}/_{16} = 16$ intensity values per channel into a single bin. The performance gracefully degrades when using more or less bins. Not surprisingly, the most runtime efficient representations use 8 and 10 bins per channel, which achieve about 120 FPS. Note that the $10 \times 10 \times 10$ model is slightly faster than the $8 \times 8 \times 8$ variant, which can be contributed to internal memory layout and memory access performance.

Learning Rates. The final object model related parameters are the model learning rates, which influence the tracker's adaptation capability to changing object appearance, *e.g.* caused by illumination variations. Finding suitable model learning rates is a non-trivial task, as we have to trade off the ability of being adaptive to such appearance changes while avoiding drifting due to incorrect updates, *e.g.* during partial occlusions. This problem is well-known as the *stability-plasticity dilemma* [168] or the *template update problem* [301].

As shown in Table 5.5, it is beneficial to use lower learning rates for the object-versus-surroundings model and larger learning rates for the object-versus-distractors model. These results confirm our intention that the tracker should quickly adapt its distractor model to suppress visually similar regions, while the general discriminative background model should be updated more conservatively. Our observation of improved robustness

Table 5.5: Performance of our distractor-aware tracker with varying learning rates η_S for the object-versus-surroundings model $p^t_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$ and η_D for the object-versus-distractors model $p^t_{O,D}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$. Best, second best and third best results have been highlighted in each column.

Learning Rate		Experiment *baseline*		Experiment *region noise*		Overall	
η_S	η_D	Acc.↑	Rob.↓	Acc.↑	Rob.↓	EAO↑	FPS↑
0.01	0.20	0.60	0.26	0.59	0.29	0.49	111.5
0.05	0.20	0.60	0.08	0.59	0.15	0.54	113.8
0.10	0.20	0.59	0.14	0.59	0.11	0.53	113.6
0.15	0.20	0.60	0.49	0.58	0.30	0.43	113.0
0.20	0.20	0.57	0.69	0.58	0.42	0.42	113.2
0.25	0.20	0.58	0.73	0.57	0.58	0.40	115.4
0.05	0.01	0.60	0.15	0.59	0.16	0.52	111.2
0.05	0.05	0.60	0.08	0.59	0.16	0.53	112.9
0.05	0.10	0.60	0.08	0.59	0.14	0.54	110.8
0.05	0.15	0.60	0.08	0.59	0.12	0.54	108.7
0.05	0.20	0.60	0.08	0.59	0.15	0.53	111.9
0.05	0.25	0.60	0.12	0.59	0.15	0.51	114.1

at lower learning rates for the object-versus-surroundings model is also in line with the experimental findings of other approaches, such as MOSSE [55] or KCF [189].

5.1.3.2 Localization and Scaling

Window Sizes. The scaling parameters λ_W and λ_S constrain the size of the search region W – used to localize the target – and the size of the surrounding region S – used to update the object-versus-surroundings model $p^t_{O,S}(\mathbf{x} \in \mathcal{O} \mid b_\mathbf{x})$. From the results in Table 5.6 we see that a search region four times the size of the tracked object gives the best performance versus speed tradeoff. Note that a search window scaling factor of $\lambda_W = 8$ requires a significant amount of image padding, which leads to unnecessary processing of these padded regions and explains the lower runtime performance. On the other hand, setting the search region too small decreases the robustness, since the tracker fails more frequently for sequences which exhibit large object or camera motion because the object leaves the search region.

The best tracking accuracy is achieved with a surrounding region twice the size of the object, and slowly degrades with both higher and lower values. The robustness scores, on the other hand, depend stronger on a proper choice of the surrounding region size. In particular, a scaling factor of $\lambda_S = 2$ yields significantly more robust results on average.

Non-Maximum Suppression. The two NMS parameters o_ν and τ_ν control the overlap and distractor-reporting threshold while densely sampling hypotheses within the search region W. The overlap controls how accurate the localization will be, *i.e.* densely overlap-

Table 5.6: Effects of varying window sizes on the tracking performance of our distractor-aware tracker. Best, second best and third best results have been highlighted in each column.

Window Scale Parameter λ_W	λ_S	Experiment *baseline* Acc.↑	Rob.↓	Experiment *region noise* Acc.↑	Rob.↓	Overall EAO↑	FPS↑
2.0	1.5	0.56	0.94	0.56	0.87	0.36	212.9
4.0	2.0	0.60	0.15	0.58	0.22	0.51	114.1
4.0	3.0	0.58	0.43	0.57	0.37	0.44	99.5
8.0	2.0	0.59	0.08	0.58	0.24	0.51	46.6
8.0	3.0	0.57	0.53	0.57	0.37	0.45	44.7
8.0	4.0	0.57	0.39	0.56	0.52	0.44	39.8
8.0	5.0	0.57	0.39	0.56	0.50	0.42	37.0
8.0	6.0	0.57	0.43	0.56	0.66	0.39	33.8
8.0	7.0	0.57	0.67	0.55	0.70	0.33	32.3

Table 5.7: Tracking performance for varying overlap parameters o_ν and reporting thresholds τ_ν of the non-maximum suppression step. Best, second best and third best results have been highlighted in each column.

NMS Parameter o_ν	τ_ν	Experiment *baseline* Acc.↑	Rob.↓	Experiment *region noise* Acc.↑	Rob.↓	Overall EAO↑	FPS↑
0.95	0.50	0.60	0.19	0.59	0.16	0.51	104.8
0.90	0.50	0.60	0.08	0.59	0.16	0.53	113.2
0.85	0.50	0.59	0.08	0.59	0.16	0.52	109.7
0.75	0.50	0.58	0.29	0.57	0.19	0.47	147.1
0.50	0.50	0.50	0.10	0.50	0.13	0.45	115.8
0.90	0.75	0.60	0.15	0.60	0.15	0.53	118.8
0.90	0.50	0.60	0.08	0.59	0.16	0.53	113.2
0.90	0.25	0.60	0.08	0.59	0.14	0.54	105.5

ping hypotheses lead to more accurate results as it is more likely to sample a hypothesis directly on the target. The reporting threshold, on the other hand, controls how many regions are considered distracting and thus, influences the robustness and adaptability *w.r.t.* to visually similar regions. This is also reflected by the tracking results in Table 5.7.

Scale Adaptation. Table 5.8 lists the results for the different scale adaptation techniques we apply on top of DAT. Overall, scaling via sum reduction of the likelihood maps (DAT+s) yields the most stable performance at a very minor speed tradeoff, easily exceeding 100 FPS. Performing a connected component analysis (DAT+c) takes significantly longer and, as we observed experimentally, its results are quite sensitive *w.r.t.* choosing the inclusion and exclusion regions for segmented blobs. We also experimented with

Table 5.8: Performance of the different scale adaptation approaches compared to the scale-agnostic DAT baseline. Best, second best and third best results have been highlighted in each column.

Tracker	Experiment *baseline*		Experiment *region noise*		Overall	
	Acc.↑	Rob.↓	Acc.↑	Rob.↓	EAO↑	FPS↑
DAT	0.60	0.08	0.59	0.12	0.55	113.0
DAT+s	0.57	0.00	0.56	0.07	0.56	108.8
DAT+c	0.58	0.09	0.57	0.12	0.51	56.7
DAT+r	0.51	0.45	0.49	0.56	0.44	90.7

different segmentation approaches, in particular graph cut-based [366] and total variation-based [370, 420] approaches, where we exploited our object-versus-surroundings model to provide seed regions for the segmentation. These approaches, however, only performed on par with DAT+c and required significantly more computing resources, which prohibits their use in time-critical applications. Additionally, segmentation-based approaches fail to robustly segment the object in low resolution or low contrast imagery, which often occurs in typical tracking sequences. Thus, because of its favorable results, simplicity and efficiency, the sum reduction approach should be preferred over the remaining scale adaptation techniques.

Instance-specific scale regression (DAT+r) only works reasonably well for a few sequences and never outperformed the sum reduction technique in our experiments. The significantly lower overall scores are slightly misleading as some very challenging initializations will negatively influence the scores, *e.g.* consider the *david* sequence, which starts in a dark room with extremely low contrast imagery. Although state-of-the-art approaches also use scale regression, *e.g.* [319, 321], they usually train object class-specific regressors on large training sets and additionally, exploit more complex (deep) features. We also experimented with pre-training object class-specific regressors based on our probability maps. This, however, yields quite unsatisfying and unstable results because these features are too simplistic (compared to CNN features) to learn a robust regression for ambiguous ground truth annotations. For example, consider face tracking – some annotators prefer bounding boxes which include the neck and hair of a person, whereas others only annotate boxes spanning from the forehead to the chin. Such contradicting ground truth annotations are the reason why we experimented with an instance-specific approach, trying to slightly overfit to the object of interest by perturbing the initialization region. Overall, however, our sum reduction-based approach is able to robustly deal with a significantly larger amount of tracking challenges out-of-the-box.

5.1.4 Comparison to the State-of-the-Art on VOT

To ensure a fair and unbiased comparison, we use the official tracking results verified by the VOT committee for our evaluations on VOT'13 [238], VOT'14 [239] and VOT'16 [241]. On each benchmark, we compare our DAT variants to the three top-performing trackers and several state-of-the-art approaches published at major conferences and journals. Additionally, we include a simple template-based tracker, in particular a *normalized cross correlation* (NCC) filter, which was used by the VOT committee as a reference baseline which had to be outperformed by each challenge contestant. According to the official evaluation protocols, we use the combined accuracy and robustness rank to sort competing trackers on VOT'13 and VOT'14, whereas VOT'16 results are ranked according to their expected average overlap. We report per-benchmark results combined over all sequences. Detailed per-sequence results can be found in Appendix C.1. Slightly different (raw) accuracy or robustness scores compared to the original challenge reports are caused by the updated VOT evaluation methodology[10]. Note that the rankings depend on the number of compared trackers and thus, the accuracy and robustness ranks cannot be compared to the original challenge reports.

VOT'13. The top-performing approaches on VOT'13 were PLT [186], FoT [428] and EDFT [132]. PLT employs an online sparse structural support vector machine (SVM) similar to [176], based on color, grayscale and gradient information where color histograms are used to weight features during SVM training. FoT combines the target displacements estimated from multiple local tracker covering the object. EDFT extends the distribution field tracker (DFT) [381] by using more efficient *channel representations* (CRs) [166] to approximate kernel density estimates.

Results for the VOT'13 experiments *baseline* and *region noise* are summarized in Table 5.9. DAT performs on par with the challenge winner PLT, and achieves the better ranking *w.r.t.* the Wilcoxon signed-rank test which is used within the VOT toolkit to test statistical significance of performance differences between the trackers.

VOT'14. The top-performing approaches on VOT'14 were DSST [92], SAMF [271] and KCF [189], all of which are correlation filters extending the MOSSE tracker [55]. DSST learns separate discriminative correlation filters for translation and scale estimation and is based on image intensities and HOG [91] features. KCF is a scale-adaptive extension of CSK [188] based on kernel ridge regression, which is efficiently trained from thousands of sample patches by exploiting the Fourier transform. SAMF is an extension of KCF which additionally introduces color names [423] as a separate feature cue to complement the raw image intensities and HOG features.

[10] We use the latest VOT toolkit for all our evaluations, *i.e.* commit `6b4447f1` to the official git repository `https://github.com/votchallenge/vot-toolkit`, from 6 July 2017.

Table 5.9: Results on the VOT'13 benchmark. Best, second best, and third best results have been highlighted. The top 3 challenge contestants are sorted according to their official challenge rank, where the first row shows the results for the winner (*i.e.* PLT). State-of-the-art trackers from major literature are sorted according to their expected average overlap score (EAO).

(a) Experiment *baseline*.

	Tracker	**EAO**$^\uparrow$	**Combined Rank**$^\downarrow$	**Accuracy Score**$^\uparrow$	**Accuracy Rank**$^\downarrow$	**Robustness Score**$^\downarrow$	**Robustness Rank**$^\downarrow$
Ours	DAT	0.56	3.78	0.59	6.00	0.08	1.56
	DAT+s	0.60	4.60	0.56	8.19	0.00	1.00
	DAT+c	0.52	4.91	0.58	8.19	0.09	1.63
	DAT+r	0.49	6.50	0.50	9.44	0.45	3.56
	DAT'15 [351]	0.51	4.85	0.61	6.00	0.26	3.69
	noDAT	0.53	4.16	0.59	5.75	0.19	2.56
Top 3	PLT [186]	0.66	3.85	0.61	6.69	0.00	1.00
	FoT [428]	0.29	6.10	0.64	5.44	1.54	6.75
	EDFT [132]	0.35	5.47	0.60	5.38	0.79	5.56
Major literature	LGT [72]	0.44	5.78	0.54	8.06	0.26	3.50
	MIL [18]	0.24	8.72	0.52	9.94	1.41	7.50
	IVT [364]	0.22	6.66	0.60	5.88	1.62	7.44
	CT [481]	0.18	10.63	0.47	12.50	1.76	8.75
	Struck'11 [176]	0.10	7.28	0.53	8.75	3.58	5.81
	HoughTrack [157]	0.08	9.41	0.49	11.56	4.25	7.25
	TLD [216]	0.07	9.06	0.60	6.56	6.60	11.56
	NCC	0.09	9.63	0.62	5.13	6.14	14.13

(b) Experiment *region noise*.

	Tracker	**EAO**$^\uparrow$	**Combined Rank**$^\downarrow$	**Accuracy Score**$^\uparrow$	**Accuracy Rank**$^\downarrow$	**Robustness Score**$^\downarrow$	**Robustness Rank**$^\downarrow$
Ours	DAT	0.53	3.66	0.59	5.25	0.12	2.06
	DAT+s	0.53	3.88	0.55	6.56	0.07	1.19
	DAT+c	0.50	4.60	0.56	7.81	0.12	1.38
	DAT+r	0.40	7.65	0.48	10.00	0.56	5.31
	noDAT	0.50	4.16	0.59	5.75	0.21	2.56
Top 3	PLT [186]	0.60	3.81	0.59	6.31	0.06	1.31
	FoT [428]	0.24	6.47	0.60	6.13	1.66	6.81
	EDFT [132]	0.29	6.41	0.57	6.44	1.09	6.38
Major literature	LGT [72]	0.45	5.22	0.53	7.69	0.20	2.75
	MIL [18]	0.21	7.97	0.50	9.44	1.51	6.50
	IVT [364]	0.19	7.16	0.55	7.44	1.91	6.88
	CT [481]	0.16	9.72	0.47	11.94	2.01	7.50
	Struck'11 [176]	0.08	7.66	0.50	8.94	3.91	6.38
	TLD [216]	0.07	8.72	0.57	6.50	6.71	10.94
	HoughTrack [157]	0.07	8.60	0.49	10.25	4.87	6.94
	NCC	0.08	9.85	0.57	5.94	6.76	13.75

Table 5.10: Results on the VOT'14 benchmark, sorted similar to the VOT'13 results (Table 5.9).

(a) Experiment *baseline*.

	Tracker	EAO↑	Combined Rank↓	Accuracy Score↑	Accuracy Rank↓	Robustness Score↓	Robustness Rank↓
Ours	DAT	0.28	4.56	0.53	5.12	0.90	4.00
	DAT+s	0.26	4.76	0.50	5.92	1.00	3.60
	DAT+c	0.29	5.04	0.51	6.44	0.84	3.64
	DAT+r	0.26	6.74	0.42	9.20	1.17	4.28
	DAT'15 [351]	0.27	4.50	0.55	4.72	0.97	4.28
	noDAT	0.24	5.00	0.54	5.04	1.21	4.96
Top 3	DSST [92]	0.30	3.88	0.63	3.08	0.84	4.68
	SAMF [271]	0.27	3.96	0.63	3.00	0.92	4.92
	KCF [189]	0.27	3.66	0.64	2.60	0.99	4.72
Major literature	LGT [72]	0.33	6.62	0.46	8.04	0.62	5.20
	ACT [93]	0.23	5.94	0.53	5.92	1.09	5.96
	PixelTrack [113]	0.22	8.68	0.44	10.92	1.31	6.44
	Struck'11 [176]	0.19	7.36	0.51	6.20	1.73	8.52
	FoT [428]	0.19	8.22	0.50	6.76	1.97	9.68
	CMT [323]	0.16	8.24	0.48	7.84	2.40	8.64
	MIL [18]	0.16	10.38	0.41	12.36	1.94	8.40
	OGT [320]	0.15	7.98	0.55	6.20	3.26	9.76
	IVT [364]	0.15	9.62	0.47	8.36	2.27	10.88
	NCC	0.08	11.42	0.53	7.32	7.87	15.52

(b) Experiment *region noise*.

	Tracker	EAO↑	Combined Rank↓	Accuracy Score↑	Accuracy Rank↓	Robustness Score↓	Robustness Rank↓
Ours	DAT	0.26	3.92	0.53	4.04	0.98	3.80
	DAT+s	0.28	3.96	0.51	4.76	0.83	3.16
	DAT+c	0.26	4.22	0.51	5.12	1.02	3.32
	DAT+r	0.25	6.74	0.45	8.32	1.23	5.16
	DAT'15 [351]	0.28	3.50	0.55	3.20	1.06	3.80
	noDAT	0.24	4.14	0.53	4.04	1.22	4.24
Top 3	DSST [92]	0.26	3.72	0.59	2.96	0.97	4.48
	SAMF [271]	0.23	3.92	0.59	3.04	0.99	4.80
	KCF [189]	0.23	4.30	0.59	3.40	1.14	5.20
Major literature	LGT [72]	0.32	6.00	0.45	7.72	0.57	4.28
	PixelTrack [113]	0.20	7.72	0.44	9.12	1.26	6.32
	ACT [93]	0.19	6.20	0.49	6.04	1.35	6.36
	Struck'11 [176]	0.17	7.20	0.48	6.72	1.79	7.68
	FoT [428]	0.16	9.86	0.47	8.00	2.52	11.72
	CMT [323]	0.15	9.04	0.44	9.16	2.33	8.92
	IVT [364]	0.15	10.08	0.44	10.00	2.47	10.16
	OGT [320]	0.13	8.02	0.51	6.24	3.09	9.80
	MIL [18]	0.11	11.84	0.35	14.68	2.15	9.00
	NCC	0.07	11.40	0.48	7.24	7.48	15.56

Table 5.11: Results on the VOT'16 benchmark. Best, second best, and third best results have been highlighted. All state-of-the-art trackers are sorted according to their expected average overlap score, as this was used to obtain the official challenge rankings. The last column shows the result for the *unsupervised* experiment, which is evaluated using only the average overlap (AO) measure.

	Tracker	Experiment *Supervised* EAO↑	Comb. Rank↓	Accuracy Score↑	Accuracy Rank↓	Robustness Score↓	Robustness Rank↓	Exp. *Unsup.* AO↑
Ours	DAT	0.21	6.28	0.47	5.45	1.99	7.10	0.28
	DAT+s	0.23	6.11	0.45	6.33	1.67	5.88	0.33
	DAT+c	0.24	5.81	0.46	5.37	1.70	6.25	0.29
	DAT+r	0.21	6.22	0.42	6.42	1.92	6.02	0.29
	DAT'15 [351]	0.22	5.99	0.47	5.25	1.99	6.73	0.31
	noDAT	0.19	6.82	0.47	5.12	2.21	8.53	0.27
Top 3	C-COT [97]	0.33	3.47	0.54	3.18	0.89	3.75	0.47
	TCNN [321]	0.32	3.58	0.55	2.33	0.83	4.83	0.49
	SSAT [241]	0.32	3.15	0.58	1.77	1.05	4.53	0.51
Major literature	Staple [42]	0.29	4.32	0.54	2.90	1.42	5.73	0.39
	EBT [495]	0.29	4.64	0.46	6.12	1.05	3.17	0.37
	MDNet [319]	0.26	3.68	0.54	2.40	0.91	4.95	0.46
	KCF [189]	0.19	6.68	0.48	5.45	1.95	7.90	0.30
	SAMF [271]	0.19	5.84	0.50	4.40	1.91	7.28	0.35
	DSST [92]	0.18	6.70	0.52	4.43	2.38	8.97	0.33
	ACT [93]	0.17	7.96	0.44	7.23	2.34	8.68	0.28
	FoT [428]	0.14	10.04	0.37	9.27	3.36	10.80	0.17
	Struck'16 [177]	0.14	9.65	0.45	7.53	3.40	11.78	0.24
	CMT [323]	0.08	13.59	0.38	11.07	6.75	16.10	0.15
	NCC	0.08	11.53	0.47	6.45	10.31	16.60	0.17

VOT'14 uses the same experiments as its predecessor benchmark, *i.e. baseline* and *region noise*. The results are listed in Table 5.10. Despite the simple color model, our DAT variants perform on par with many state-of-the-art trackers but achieve favorable robustness on both experiments. Interestingly, our probabilistic approach using raw pixel colors significantly outperforms sophisticated color representations, such as used by ACT [93].

VOT'16. The top-performing approaches on VOT'16 were C-COT [97], TCNN [321] and SSAT[11]. C-COT learns a discriminative continuous convolution operator in the continuous spatial domain to efficiently fuse multi-resolution feature maps from a pre-trained convolutional neural network (CNN). TCNN employs multiple CNNs which collaborate in a tree structure to represent the target appearance. SSAT is also a CNN-based tracker and extends MDNet [319], the winner of the VOT'15 challenge [240].

[11]The *scale-and-state aware tracker* (SSAT) is an extension of MDNet [319] and has only been published as appendix to the official VOT'16 challenge report [241].

Table 5.11 summarizes the results for the *supervised* and *unsupervised* experiments. Although DAT does not achieve top 3 performance on this benchmark, we perform on par in terms of robustness and accuracy with recent state-of-the-art approaches, such as MDNet [319]. Moreover, we can easily outperform top-performing trackers from previous VOT benchmarks, such as KCF [189], SAMF [271], DSST [92] or ACT [93], despite the significantly more challenging sequences of VOT'16.

Performance *w.r.t.* Specific Tracking Challenges. The VOT benchmarks provide a rich per-frame annotation of common challenges, namely (i) *occlusion*, (ii) *illumination change*, (iii) *motion change*, (iv) *size change* and (v) *camera motion*. Frames which correspond to none of these attributes are denoted as (vi) *unassigned*. We use the annotations of the VOT'16 benchmark to evaluate the performance of our distractor-aware tracker *w.r.t.* these attributes, as the large number of frames within this benchmark allows for a meaningful conclusion.

These attribute-based evaluations are summarized in Table 5.12 and Figure 5.5. Note that the robustness scores show the total number of failures and are not averaged over the annotated frames, in contrast to the previous evaluations. We can see that DAT performs on par with the VOT'16 challenge leaders for the attributes *illumination change* and *occlusion*. The most challenging attributes for our approach are *camera motion*, *motion change* and *size change*.

5.1.5 Comparison to the State-of-the-Art on OTB

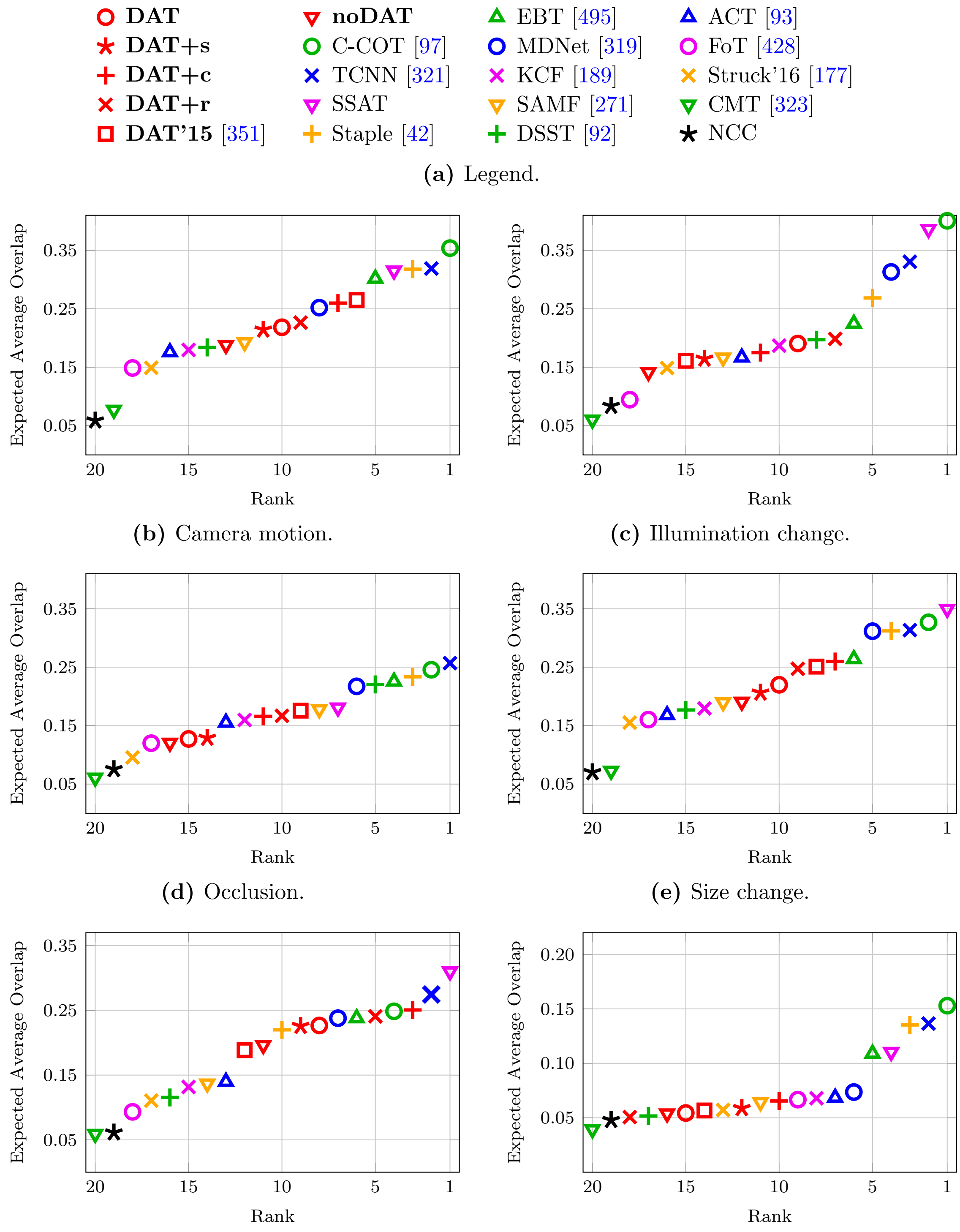
Complementary to VOT, we additionally evaluate our approach on the sequence collection provided by the OTB. For a fair comparison, we use the official benchmark results distributed for OTB-100 [449]. Since DAT is color-based, we evaluate on the 76 color sequences of OTB-100 and skip the monochrome videos. OTB defines 11 visual attributes to classify tracking challenges, *i.e.* (i) *background clutter*, (ii) *fast motion*, (iii) *illumination variation*, (iv) *in-plane rotation* and (v) *out-of-plane rotation* of the target, (vi) *low resolution*, (vii) *motion blur*, (viii) *non-rigid deformation*, (ix) *occlusion*, (x) if the target moves *out-of-view*, and (xi) *scale variation*. A detailed list of sequences along with their attribute annotations can be found in [449]. Note that in contrast to the VOT benchmarks, the OTB only provides per-sequence attributes.

We compare against the OTB-100 top-performing Struck [176], SCM [494], ASLA [212] and CSK [188] trackers. Additionally, we include results for the context-aware CXT [104] and the color-based VTD [249] and VTS [250]. Struck is an adaptive tracking-by-detection approach which employs an online structured output support vector machine (SVM). SCM uses a sparse collaborative appearance model based on a discriminative object-versus-background classifier and a sparse generative histogram model. ASLA leverages sparse coding and uses a structural local sparse appearance model in combination with incremental subspace learning. CSK efficiently learns a correlation filter via kernel ridge regression

Table 5.12: Performance on VOT'16 *w.r.t.* to the six annotated visual attributes. State-of-the-art trackers are sorted according to their official overall VOT'16 rank.

	Tracker	*Camera Motion* **Acc.**↑	*Camera Motion* **Rob.**↓	*Illum. Change* **Acc.**↑	*Illum. Change* **Rob.**↓	*Occlusion* **Acc.**↑	*Occlusion* **Rob.**↓
Ours	DAT	0.47	45.00	0.51	18.00	0.47	6.00
	DAT+s	0.46	36.00	0.49	11.00	0.33	3.00
	DAT+c	0.45	39.00	0.50	21.00	0.36	5.00
	DAT+r	0.40	37.00	0.46	19.00	0.32	3.00
	DAT'15 [351]	0.49	55.00	0.54	24.00	0.69	8.00
	noDAT	0.46	55.00	0.53	18.00	0.43	11.00
Top 3	C-COT [97]	0.56	24.00	0.58	11.00	0.65	2.00
	TCNN [321]	0.55	27.93	0.58	8.47	0.64	3.13
	SSAT	0.57	30.07	0.61	8.87	0.67	2.27
Major literature	Staple [42]	0.55	34.00	0.58	13.00	0.71	7.00
	MDNet [319]	0.49	20.00	0.52	11.00	0.41	3.00
	EBT [495]	0.55	33.00	0.56	18.47	0.64	3.80
	DSST [92]	0.52	53.00	0.55	30.00	0.56	6.00
	SAMF [271]	0.53	66.00	0.58	31.00	0.68	6.00
	KCF [189]	0.47	56.00	0.49	40.07	0.44	5.00
	ACT [93]	0.36	67.00	0.42	35.00	0.46	17.00
	Struck'16 [177]	0.47	81.00	0.53	46.00	0.39	7.00
	FoT [428]	0.41	166.00	0.41	61.00	0.55	28.00
	CMT [323]	0.46	36.00	0.50	25.00	0.43	6.00
	NCC	0.50	246.00	0.53	89.00	0.43	18.00

	Tracker	*Size Change* **Acc.**↑	*Size Change* **Rob.**↓	*Motion Change* **Acc.**↑	*Motion Change* **Rob.**↓	*Unassigned* **Acc.**↑	*Unassigned* **Rob.**↓
Ours	DAT	0.47	34.00	0.41	41.00	0.40	24.00
	DAT+s	0.44	34.00	0.39	43.00	0.41	25.00
	DAT+c	0.46	31.00	0.43	31.00	0.41	19.00
	DAT+r	0.39	30.00	0.36	36.00	0.36	17.00
	DAT'15 [351]	0.42	52.00	0.43	20.00	0.47	31.00
	noDAT	0.46	40.00	0.44	45.00	0.40	30.00
Top 3	C-COT [97]	0.47	20.00	0.44	14.00	0.50	13.00
	TCNN [321]	0.52	22.13	0.51	15.33	0.51	14.93
	SSAT	0.54	21.73	0.51	23.67	0.55	15.07
Major literature	Staple [42]	0.51	35.00	0.43	24.00	0.51	15.00
	MDNet [319]	0.44	19.00	0.37	17.00	0.36	11.00
	EBT [495]	0.51	21.40	0.49	12.87	0.51	12.07
	DSST [92]	0.47	44.00	0.44	25.00	0.43	30.00
	SAMF [271]	0.48	60.00	0.41	22.00	0.51	33.00
	KCF [189]	0.42	51.13	0.40	24.00	0.34	31.93
	ACT [93]	0.35	69.00	0.28	34.00	0.39	34.00
	Struck'16 [177]	0.43	63.00	0.37	48.00	0.34	36.00
	FoT [428]	0.36	125.00	0.34	74.00	0.40	92.00
	CMT [323]	0.46	30.00	0.45	31.00	0.40	22.00
	NCC	0.45	128.00	0.47	61.00	0.38	107.00



(a) Legend.

(b) Camera motion.

(c) Illumination change.

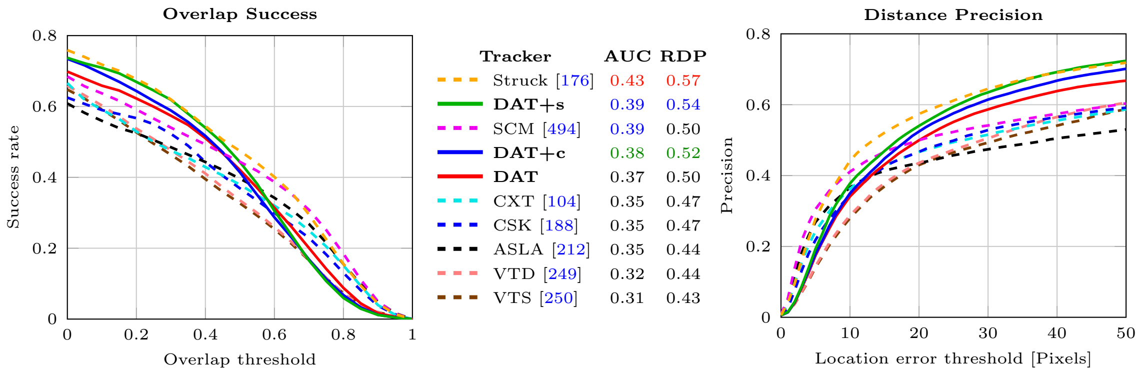
(d) Occlusion.

(e) Size change.

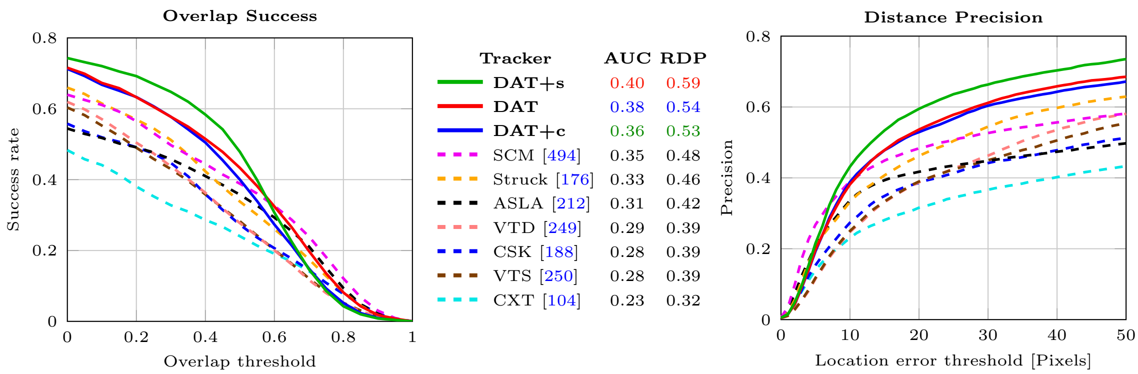
(f) Motion change.

(g) Unassigned.

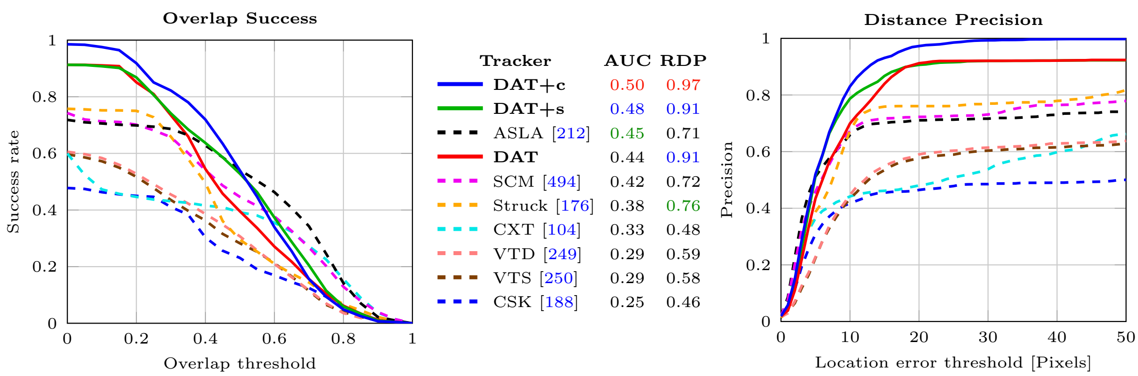
Figure 5.5: Ranking plots using the expected average overlap (EAO) metric for all 6 annotated attributes of the VOT'16 benchmark. Better trackers are located to the top right. Note that the ranking for all unassigned frames in (g) has a different y axis range due to the overall lower performance of all trackers for these frames.



(a) Results over all 76 color sequences.

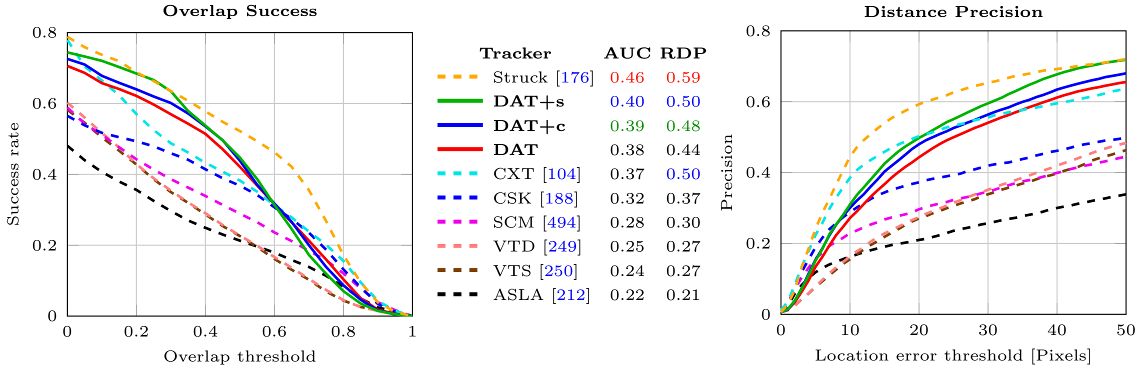


(b) Results for attribute *non-rigid deformation* (38 sequences).

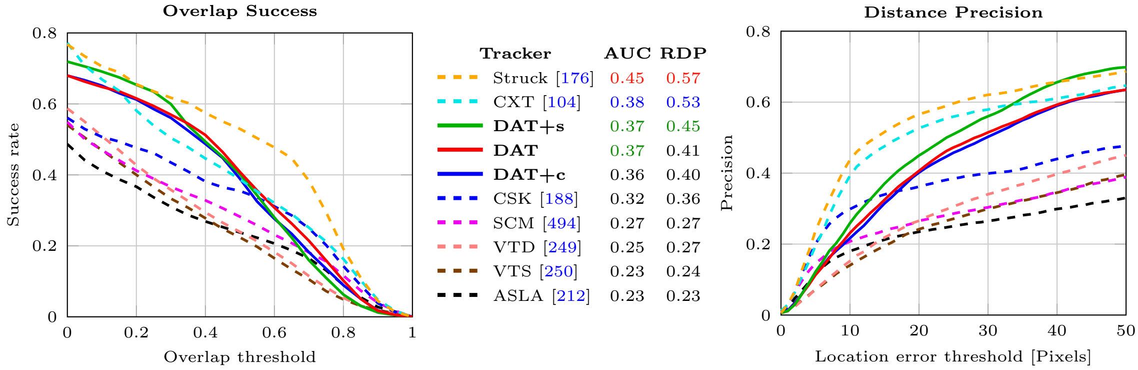


(c) Results for attribute *low resolution* (6 sequences).

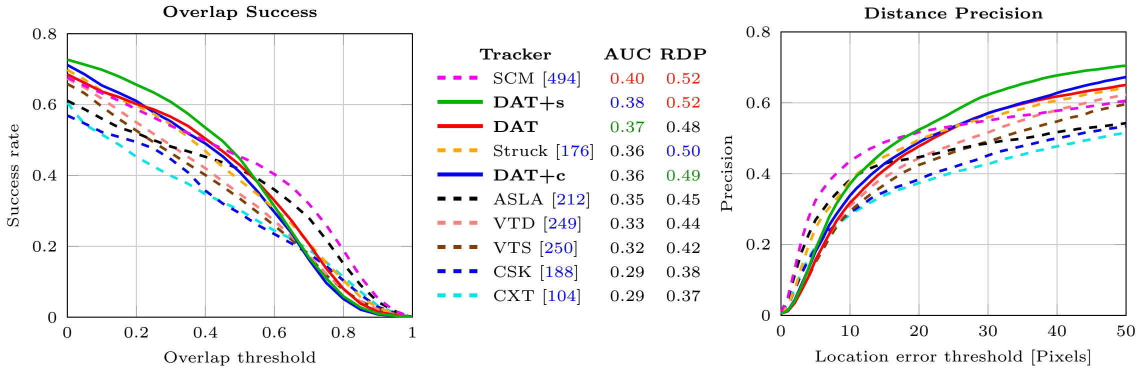
Figure 5.6: Results on the OTB-100 [449] dataset for (a) all color sequences, as well as the (b) *non-rigid deformation* and (c) *low resolution* attributes. Each experiment shows the *success plot* (left column; overlap ratio *w.r.t.* ground truth) and *precision plot* (right column; center distance) for each attribute. The legend (middle column) shows the area under the success curve (AUC) and the representative distance precision score (RDP, percentage of frames with center distance less than 20 pixels). Best, second best and third best success and precision scores have been highlighted in the legend. Legend entries are sorted according to their AUC score. Solid lines denote our DAT variants, whereas dashed lines illustrate the performance of state-of-the-art trackers.



(a) Results for attribute *fast motion* (33 sequences).

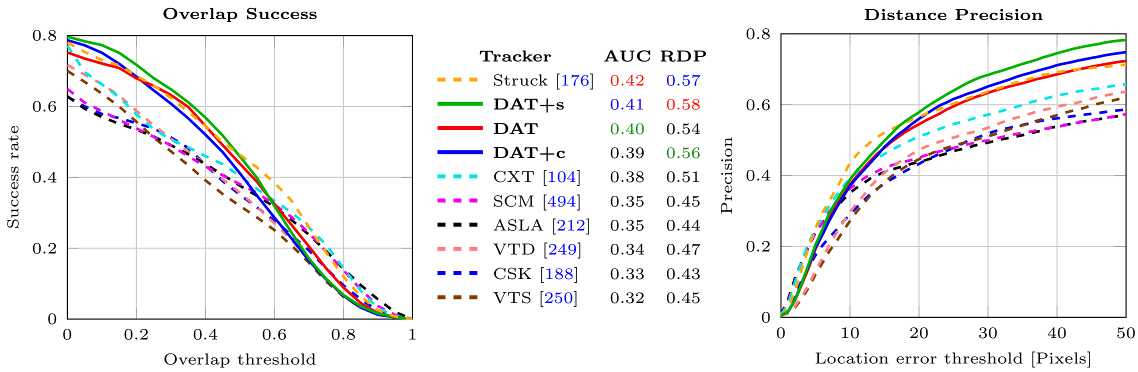


(b) Results for attribute *motion blur* (26 sequences).

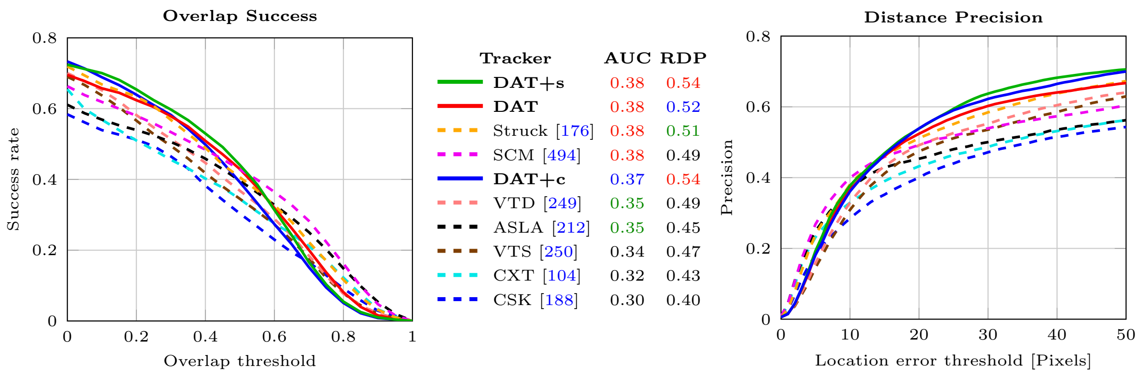


(c) Results for attribute *occlusion* (42 sequences).

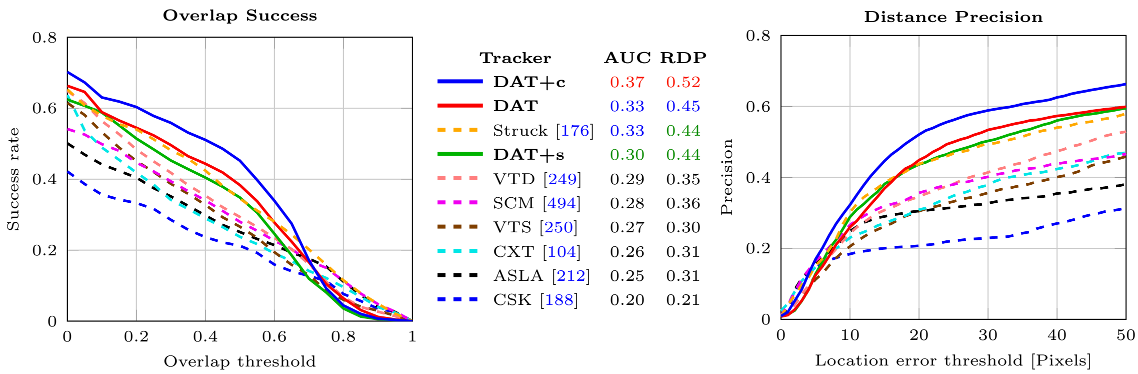
Figure 5.7: Results on the OTB-100 [449] dataset for the attributes (a) *fast motion*, (b) *motion blur* and (c) *occlusion*. Each experiment shows the *success plot* (left column; overlap ratio *w.r.t.* ground truth) and *precision plot* (right column; center distance) for each attribute. The legend (middle column) shows the area under the success curve (AUC) and the representative distance precision score (RDP, percentage of frames with center distance less than 20 pixels). Best, second best and third best success and precision scores have been highlighted in the legend. Legend entries are sorted according to their AUC score. Solid lines denote our DAT variants, whereas dashed lines illustrate the performance of state-of-the-art trackers.



(a) Results for attribute *in-plane rotation* (36 sequences).

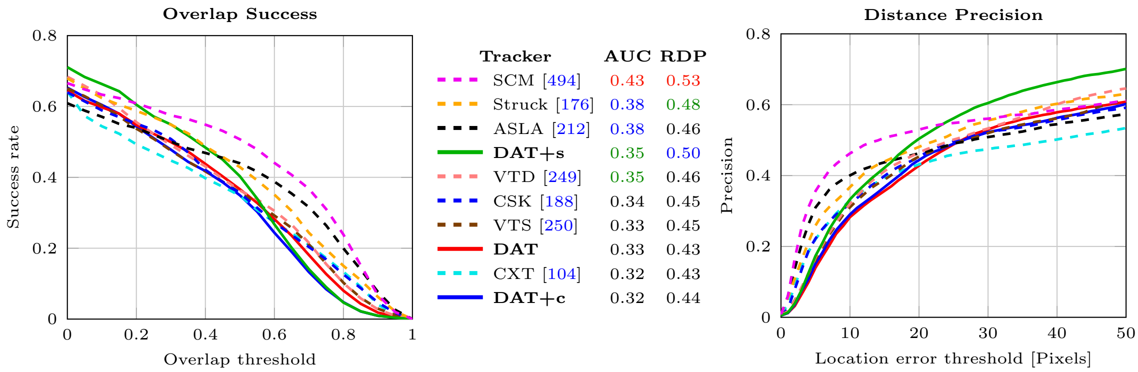


(b) Results for attribute *out-of-plane rotation* (49 sequences).

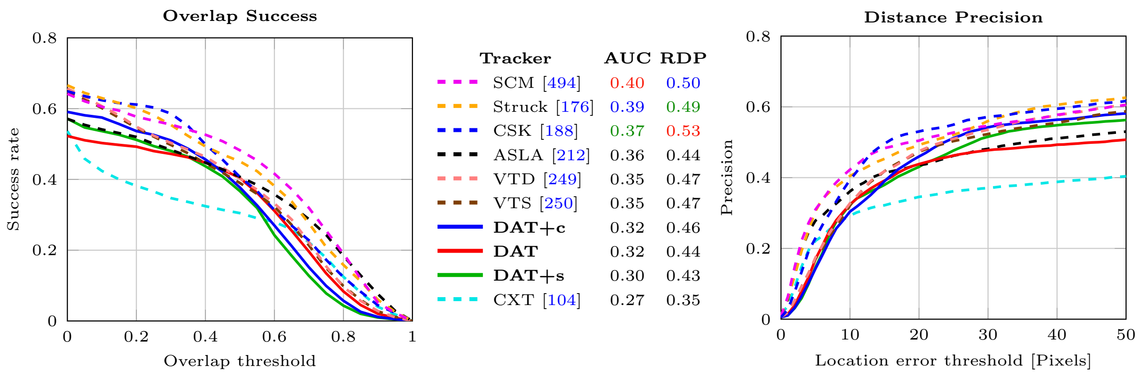


(c) Results for attribute *out-of-view* (11 sequences).

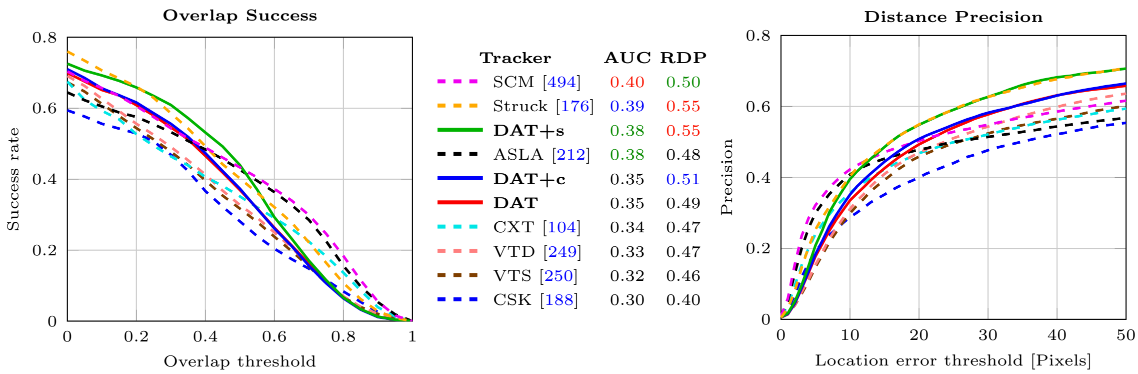
Figure 5.8: Results on the OTB-100 [449] dataset for the attributes (a) *in-plane rotation*, (b) *out-of-plane rotation* and (c) *out-of-view*. Each experiment shows the *success plot* (left column; overlap ratio *w.r.t.* ground truth) and *precision plot* (right column; center distance) for each attribute. The legend (middle column) shows the area under the success curve (AUC) and the representative distance precision score (RDP, percentage of frames with center distance less than 20 pixels). Best, second best and third best success and precision scores have been highlighted in the legend. Legend entries are sorted according to their AUC score. Solid lines denote our DAT variants, whereas dashed lines illustrate the performance of state-of-the-art trackers.



(a) Results for attribute *illumination variation* (32 sequences).



(b) Results for attribute *background clutter* (24 sequences).



(c) Results for attribute *scale variation* (50 sequences).

Figure 5.9: Results on the OTB-100 [449] dataset for the attributes (a) *illumination variation*, (b) *background clutter* and (c) *scale variation*. Each experiment shows the *success plot* (left column; overlap ratio *w.r.t.* ground truth) and *precision plot* (right column; center distance) for each attribute. The legend (middle column) shows the area under the success curve (AUC) and the representative distance precision score (RDP, percentage of frames with center distance less than 20 pixels). Best, second best and third best success and precision scores have been highlighted in the legend. Legend entries are sorted according to their AUC score. Solid lines denote our DAT variants, whereas dashed lines illustrate the performance of state-of-the-art trackers.

from grayscale imagery in the Fourier domain. CXT identifies and exploits distractors and supporters, where the latter are image regions that consistently co-occur with the target and exhibit high motion correlation. VTD decomposes the object model into multiple observation models constructed by sparse principal component analysis (SPCA) and combines the model estimates within an interactive Markov Chain Monte Carlo (IMCMC) framework.

To avoid cluttering the evaluation plots, we skip the results for noDat and DAT+r, as these are constantly outperformed by the other DAT variants. Detailed per-sequence results for all DAT variants and selected state-of-the-art approaches are provided in Appendix C.1. Figures 5.6–5.9 show the result plots, where Fig. 5.6a summarizes the performance over all sequences and the remaining plots show the performance for each annotated attribute. DAT consistently performs on par with the benchmark winner Struck, where the latter achieves a better overlap at sequences with fast motion and motion blur. All DAT variants outperform the context-aware CXT which indicates that our probabilistic distractor-aware model is beneficial compared to explicitly handling distractors and supporters. It is also interesting to see that DAT outperforms specialized trackers. For example, VTD is explicitly designed to handle drastic appearance changes, abrupt motion changes and illumination variations. However, our approach outperforms VTD on all attributes except for sequences with significant background clutter.

A drawback of OTB is that it focuses on unsupervised short-term experiments. Thus, trackers with explicit re-detection capability usually achieve better ranks on this benchmark. Additionally, OTB does not provide per-frame attribute annotations, which makes it challenging to draw valid conclusions out of its attribute evaluations. For example, Figure 5.9a indicates inferior performance of DAT for sequences with *illumination variation*. This is in stark contrast to the VOT evaluation, which showed that DAT has a favorable robustness during illumination changes, *i.e.* it does not fail immediately. Looking closely at the tracking output of the corresponding sequences, we can observe that immediate illumination changes (*e.g.* caused by a flashing light) cause DAT to partially drift but it still stays on the target (leading to a rather low average overlap score), which is also indicated by the corresponding distance precision plot in Fig. 5.9a – note DAT's high distance precision after relaxing the distance threshold.

5.1.6 Runtime Evaluation

Our final evaluation compares the tracking speed as measured by the respective benchmark frameworks. We implemented a simple measure to limit the maximum processing time to guarantee real-time capable applications of DAT. Judging from our previous dataset analysis – recall Figure 5.1a in Section 5.1.1 – the median object diagonal measures approximately 100 pixels. Thus, we limit the maximum target diagonal to $d_\tau = 100$ pixels and resize the input image I at time t by the factor

$$\lambda_I = \min\left(1, \left\lceil \frac{d_\tau}{d^{t-1}} \right\rfloor_{1/10}\right), \tag{5.5}$$

where d^{t-1} is the target diagonal at frame $t-1$ and $\lceil\cdot\rfloor_{1/10}$ denotes rounding to the closest one-tenth, *e.g.* $\lceil 0.83\rfloor_{1/10} = 0.8$. Note that we only downscale the image if the object becomes too large and would span large regions of the input image. To resample the image efficiently, we use nearest neighbor interpolation as more complex interpolation schemes did not noticeably influence the overall tracking scores. The limit of $d_\tau = 100$ pixels ensures that the size of the object region is approximately 70×70 pixels (assuming a perfectly square bounding box for simplicity), which in practice is sufficient to compute distinctive color distributions for DAT.

Table 5.13 summarizes the implementation details and runtime performance of DAT and selected state-of-the-art approaches. If there are multiple implementations available for state-of-the-art trackers, such as Struck [176, 177] or CMT [323, 324], we only report the fastest. Since VOT'14, the VOT toolkits report speed in terms of equivalent filter operations (EFO) instead of raw frames per second (FPS) to provide a platform independent runtime analysis, recall Section 5.1.2. Note however, that especially for MATLAB®-based trackers, these runtime measurements are not always accurate, as discussed in Section 5.1.3.1 (page 77). OTB, on the other hand, reports tracking speed in raw FPS without addressing the hardware bias.

Overall, all DAT prototypes rank amongst the fastest and most efficient trackers, despite being implemented in pure MATLAB®. Our scale-agnostic DAT and the sum reduction-based (scale-aware) DAT+s consistently exceed 100 FPS across all sequences. In contrast to computationally demanding CNN-based trackers, such as C-COT [97], MDNet [319] or TCNN [321], and trackers which rely on accurate segmentation, such as HoughTrack [156, 157], our approach fulfills all requirements for robust tracking in time-critical applications.

5.1.7 Discussion

Overall, our distractor-aware tracker ranks amongst the state-of-the-art trackers both with respect to accuracy and robustness. Even if provided with noisy initializations, our tracker is able to recover and stay on the target, as indicated by the results on the VOT *region noise* experiments. A key finding is that the proposed context-aware object representation significantly outperforms other color-based models, such as ACT [93] and OGT [320] (recall Tables 5.10 and 5.11), as well as trackers based on a combination of image gradients and color information, such as PixelTrack [113] (recall Table 5.10).

Typical failure cases of DAT are illustrated in Figure 5.10. Fast scale changes, especially in combination with partial occlusions, such as captured by the *graduate* sequence,

Table 5.13: Implementation details and runtime comparison for selected trackers on the (a) VOT and (b) OTB datasets. Reported runtimes – EFO on VOT and FPS on OTB – are measured using the official evaluation frameworks. Trackers from major literature are sorted alphabetically.

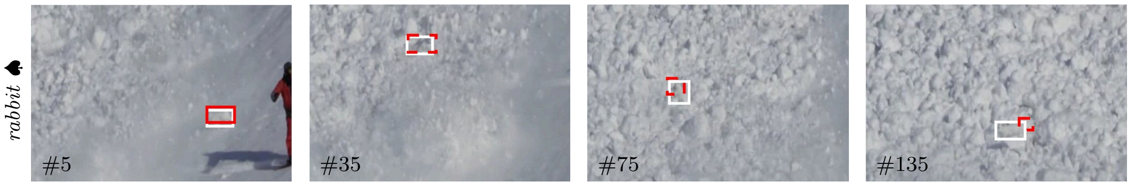
(a) VOT'13 [238], VOT'14 [239] and VOT'16 [241].

	Tracker	Publication	Implementation	GPU	EFO
Ours	DAT		MATLAB®		17.2
	DAT+s		MATLAB®		17.0
	DAT+c		MATLAB®		9.9
	DAT+r		MATLAB®		13.8
	noDAT		MATLAB®		20.1
Major literature	ACT [93]	CVPR'14	MATLAB®/MEX		18.3
	C-COT [97]	ECCV'16	MATLAB®/MEX	✓	0.5
	CT [481]	ECCV'12	C/C++		6.3
	DSST [92]	BMVC'14	MATLAB®/MEX		12.7
	EDFT [132]	VOT'13	MATLAB®		3.9
	FoT [428]	CVWW'11	C/C++		114.6
	HoughTrack [157]	CVIU'13	C/C++		0.9
	KCF [189]	TPAMI'15	MATLAB®/MEX		24.2
	LGT [72]	TPAMI'13	MATLAB®/MEX		4.1
	MDNet [319]	CVPR'16	MATLAB®/MEX	✓	0.6
	MIL [18]	TPAMI'11	C/C++		1.9
	PixelTrack [113]	ICCV'13	C/C++		49.9
	PLT [186]	VOT'13	C/C++		75.9
	SAMF [271]	VOT'14	MATLAB®/MEX		4.0
	SSAT	VOT'16	MATLAB®/MEX	✓	0.5
	Staple [42]	CVPR'16	MATLAB®/MEX		11.1
	Struck'16 [177]	TPAMI'16	C/C++		14.6
	TCNN [321]	–	MATLAB®/MEX	✓	1.0

(b) OTB-100 [449].

	Tracker	Publication	Implementation	FPS
Ours	DAT		MATLAB®	143.1
	DAT+s		MATLAB®	132.8
	DAT+c		MATLAB®	70.2
	DAT+r		MATLAB®	93.8
	noDAT		MATLAB®	180.2
Major literature	ASLA [212]	CVPR'12	MATLAB®/MEX	7.1
	CSK [188]	ECCV'12	MATLAB®/MEX	229.6
	CXT [104]	CVPR'11	C/C++	14.3
	SCM [494]	TIP'14	MATLAB®/MEX	0.4
	Struck'11 [176]	ICCV'11	C/C++	10.0
	VTD [249]	CVPR'10	MATLAB®/MEX	3.3
	VTS [250]	ICCV'11	MATLAB®/MEX	3.1

(a) Scale changes in combination with partial occlusions.



(b) Indistinguishable color distributions for object and surroundings.

(c) Immediate illumination changes, occlusions and low contrast.

Figure 5.10: Challenging sequences from the ◇ VOT'14, ♠ VOT'16 and ♡ OTB benchmarks, which cause DAT to fail. White and red bounding boxes denote the annotated ground truth and DAT tracking results, respectively. Dashed bounding boxes indicate a previous target loss. Images are slightly cropped and frame numbers are superimposed only for visualization. See text for details.

lead to DAT focusing on small sub-parts of the target, *e.g.* a person's face instead of the full body. This reduces the overall accuracy significantly and additionally, may lead to failure once the target turns around and the face is no longer visible, as can be seen from the sequence's last frame in Figure 5.10a. Obviously, by relying on a color-based model, DAT cannot track objects which are indistinguishable from their surroundings, as illustrated by the *rabbit* sequence in Figure 5.10b, where a mountain hare tries to cross an avalanche. Similarly, scenes with low contrast and drastic illumination changes will also lead to frequent failures, as shown by the *skating* sequence in Figure 5.10c.

Although color information on its own is obviously not the solution to all tracking-related problems, it is a highly efficient and powerful cue for a large variety of typical tracking scenarios. Additionally, including our distractor-aware representation comes at a reasonably low computational cost and proves to be a crucial extension to standard color-based models. In particular, all DAT variants significantly outperform the distractor-agnostic baseline, *noDAT*, especially *w.r.t.* robustness. Without suppressing visually similar regions, a standard color model as in *noDAT* is prone to drifting. Similarly, our DAT variants are consistently more robust than ACT, which uses a more complex color representation but also lacks the ability to identify and handle distractors accordingly. Finally, our proposed scale adaptation techniques are very efficient, especially the proposed sum reduction-based adaptation which both, achieves the best performance of all DAT variants and processes more than 100 FPS. Thus, the proposed DAT tracker is well suited for time-critical application domains, such as visual surveillance or robotics.

5.2 Occlusion Geodesics to the Test

In the following, we investigate our occlusion-aware multiple object tracking approach. To this end, we focus on standard monocular visual surveillance scenarios. We will briefly review relevant sequences and evaluation protocols in Sections 5.2.1 and 5.2.2, respectively. Next, we perform a detailed parameter ablation study in Section 5.2.3 and compare our approach against the state-of-the-art in Section 5.2.4. Finally, we discuss limitations and potential improvements in Section 5.2.5.

5.2.1 Datasets

In contrast to single object tracking, there are notably less publicly available datasets to evaluate multiple object tracking approaches, such as the TUD sequences [10], the EPFL multi-camera sequences [39, 137], the ETH sequences [121], the MVL multi-camera dataset Lab5 [297], the ICG multi-camera dataset Lab6 [349], the NIST TRECVid sequences [363] or the PNNL Parking Lot sequences [384]. The reduced availability of MOT datasets can be attributed to the fact that providing accurate ground truth annotations for such evaluations requires a significant manual effort. Nevertheless, there are a few initiatives which aim to standardize MOT evaluations, such as CLEAR [41, 402], PETS [136, 267, 339, 340, 471] or the MOT challenges [256, 309].

We focus our MOT evaluation on visual surveillance scenarios, since tracking pedestrians provides a challenging testbed for such algorithms[12]. Although the majority of MOT research focuses on such pedestrian scenarios, there are only very few publicly available surveillance sequences which provide a sufficiently accurate calibration *w.r.t.* both intrinsic and extrinsic camera parameters. Since we rely on world coordinates to leverage geometric context information, we select the widely used PETS'09 [136] and TownCentre [36] sequences for our evaluations.

The PETS'09 dataset [136] shows an outdoor scene with numerous pedestrians recorded from multiple cameras at 7 FPS. One of the viewpoints, *i.e. View 1*, is a standard surveillance camera mounted on a pole which enables a large field of view. We only use this viewpoint, as this monocular setup yields typical visual surveillance challenges, namely frequent occlusions – either caused dynamically by people occluding each other or static occlusions due to a traffic sign which covers large parts of the intersection. This dataset

[12] For a discussion of the key benefits of visual surveillance scenarios for MOT evaluations recall Section 2.4.

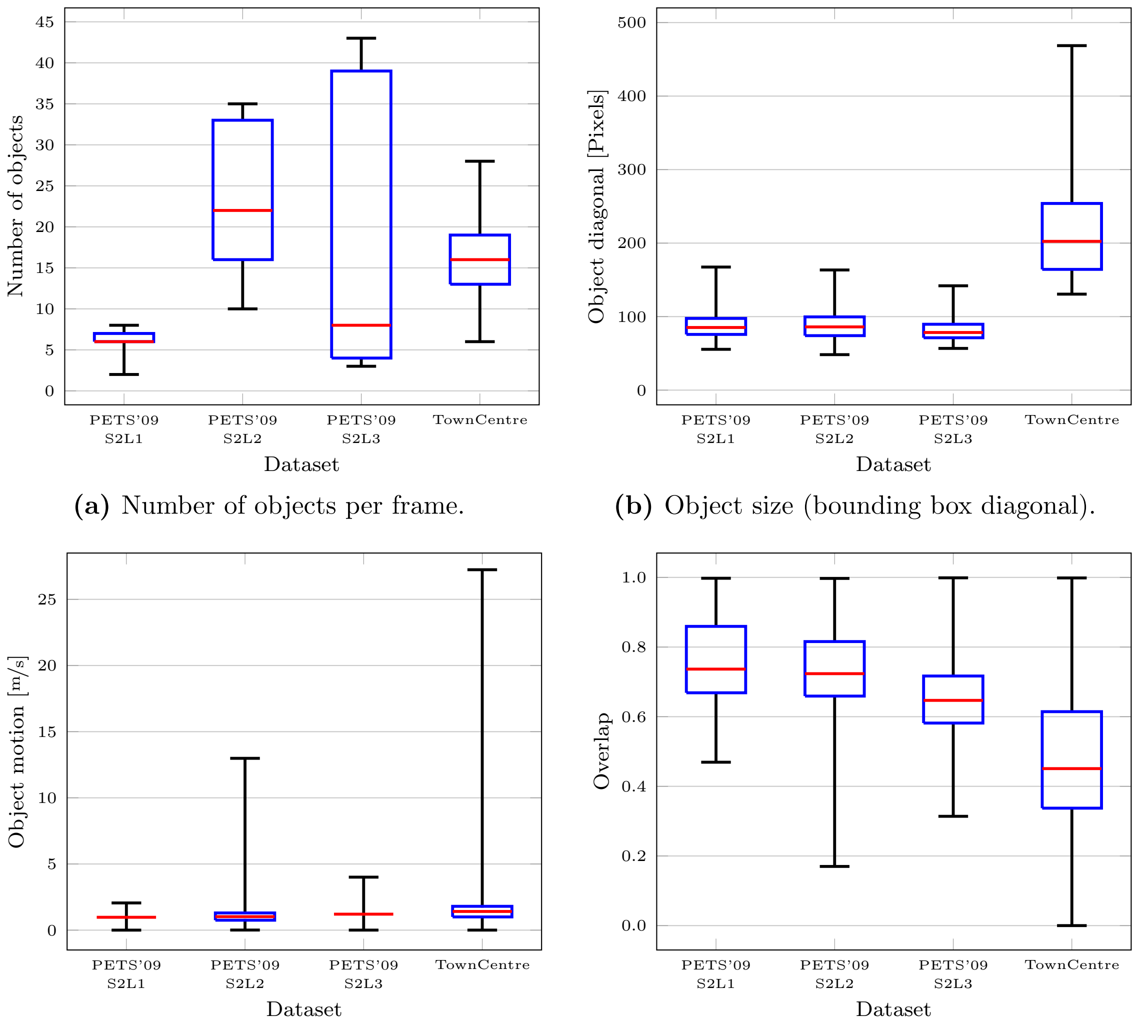
Table 5.14: Overview of the visual surveillance sequences used to benchmark our MOT approach. For each sequence, we list the capture settings, the number of annotated ground truth trajectories, as well as the corresponding rectangular tracking area (in meters).

Sequence	**Image Resolution**	**Frame Rate**	**Num. Frames**	**Num. Trajectories**	**Tracking Area**
PETS'09 S2L1 [136]	768×576	7.0	795	19	19.1×16.0
PETS'09 S2L2 [136]	768×576	7.0	436	68	19.1×16.0
PETS'09 S2L3 [136]	768×576	7.0	240	44	19.1×16.0
TownCentre [36]	1920×1080	2.5	450	227	36.0×19.0

contains three tracking sequences – *i.e.* S2L1, S2L2, and S2L3 – which capture differently crowded scenarios. As PETS'09 does not provide official ground truth annotations, we use the ground truth provided by Milan *et al.* [308].

The TownCentre sequence [36] shows a busy pedestrian precinct from a single elevated camera. On average, 16 people are visible at any time, resulting in frequent dynamic occlusions. Additionally, scene structures cause several detection failures, *e.g.* benches which partially occlude pedestrians or mannequins in shop displays which confuse the object detector. The dataset provides manually refined HOG [91] detections as ground truth annotations for every 10-th frame. Although the original sequence is recorded at 25 FPS, usually only every 10-th frame is used for tracking, *e.g.* [253, 256], which results in an actual frame rate of 2.5 FPS. This temporal undersampling of the surveillance footage allows us to demonstrate the robustness of our MOT approach at low frame rates and larger object movements between subsequent frames.

A general overview of all used sequences is provided in Table 5.14. The characteristics of each sequence are illustrated more detailed in Figure 5.11, where we analyze the ground truth annotations via box plots. In particular, we can observe the large variation *w.r.t.* the number of simultaneously visible objects. PETS'09 S2L3 is the most crowded scene – however, this is a rather short sequence where a single large group of people walks across the field of view. PETS'09 S2L2 and TownCentre, on the other hand, are typical visual surveillance scenarios where only few pedestrians interact with each other – *e.g.* people meeting on the street. On average, the walking speed throughout all scenarios is 1.2 [$^{m}/_{s}$], which very accurately resembles the pedestrian characteristics used to design public pedestrian facilities [131]. The few outliers in terms of velocity are caused by fast moving pedestrians (PETS'09 S2L2) and cyclists (TownCentre), respectively. Additionally, the box plots also highlight differences caused by the capture setups used for PETS'09 and TownCentre, namely object size – influenced by the camera sensor resolution and view point – and overlap between subsequent ground truth annotations – which depends mostly on the frame rate, due to the rather small variations of the object velocities.



(a) Number of objects per frame.

(b) Object size (bounding box diagonal).

(c) Object motion.

(d) Frame-to-frame ground truth overlap.

Figure 5.11: Sequence characteristics showing the distribution of (a) crowd densities, (b) pedestrian sizes, (c) motion (on the ground plane) of the pedestrians and (d) overlap of ground truth annotations (on the image plane) between subsequent frames. Each box plot shows the median, first and third quartiles as well as the minimum and maximum data values. For visualization purposes, interquartile ranges in (c) are omitted if they are too close to the median. Large object motion in (c), as well as zero overlap in (d) are caused by fast moving people, *e.g.* cyclists, as well as the low frame rate of the TownCentre sequence.

5.2.2 Evaluation Metrics and Protocols

Similar to single object tracking evaluations, there exists a multitude of different performance metrics to analyze MOT approaches, *e.g.* [41, 219, 273, 373, 388, 402, 446]. We follow the recent evaluation trend, *e.g.* [22, 193, 213, 308, 439], where two widely established sets of metrics are reported, namely the CLEAR MOT metrics [41, 402] in combination with a set of trajectory quality measures [446].

To compute these metrics, we first need to assign valid tracker hypotheses to ground truth trajectories. Since we track in ground plane coordinates, we use the Euclidean distance to cut off invalid assignments. In particular, a tracker's hypothesis $\mathbf{x}_{\mathrm{T}}^t$ at time t is considered to be a valid match for the ground truth annotation $\mathbf{x}_{\mathrm{G}}^t$, iff $\|\mathbf{x}_{\mathrm{T}}^t - \mathbf{x}_{\mathrm{G}}^t\|_2 \leq \tau_d$. Similar to several recent tracking evaluations, such as [13, 41, 193, 256, 308], we employ the cut-off threshold $\tau_d = 1$ [m]. To assign hypotheses to ground truth trajectories, we follow the protocol defined within the 3D MOT'15 benchmark [256], *i.e.* the optimal matching is found using the Hungarian algorithm [317] with additionally considering the temporal consistency. In particular, if at time $t-1$ the i-th ground truth object – located at $\mathbf{x}_{\mathrm{G},i}^{t-1}$ – was matched to the j-th hypothesis – located at $\mathbf{x}_{\mathrm{T},j}^{t-1}$ – *and* their distance $\|\mathbf{x}_{\mathrm{G},i}^t - \mathbf{x}_{\mathrm{T},j}^t\|_2 \leq \tau_d$ at time t, *then* i and j are matched again at frame t, even if there exists another hypothesis which is closer to the annotation $\mathbf{x}_{\mathrm{G},i}^t$. Afterwards, we can count the number of *true positives* (TP) – *i.e.* hypotheses which were matched to a ground truth annotation – and *false positives* (FP) – *i.e.* hypotheses which could not be assigned to an annotated object location. Any annotated ground truth object for which there is no matching hypothesis within a radius of τ_d is considered a *false negative* (FN), *i.e.* it is missed by the tracker.

Using the successfully matched trajectories, we can count the number of *identity switches* (IDS[13]). In particular, we follow the definitions of [256, 273] and count an identity switch iff a ground truth annotation $\mathbf{x}_{\mathrm{G},i}^t$ is matched to hypothesis $\mathbf{x}_{\mathrm{T},j}^t$, *and* its previously assigned hypothesis was $\mathbf{x}_{\mathrm{T},k}^{t-1}$, with $j \neq k$. This is a stricter definition than the original formulation of the CLEAR MOT metrics [402]. Although the overall number of identity switches should be as low as possible for good tracking approaches, this absolute measure alone is not always expressive of the actual tracking performance. For example, the IDS score could be kept rather low by only reporting a small fraction of the tracked hypotheses. Thus, instead of focusing on a single metric, it is important to consider multiple performance measures for a valid conclusion about a tracker's performance. To this end, we rely on the following evaluation metrics throughout our experiments:

- *Multiple Object Tracking Accuracy* (MOTA[14]) [41, 402] – combines three sources of errors and thus, is one of the most widely used metrics to summarize the tracking

[13] IDS is the absolute number of identity switches, *i.e.* IDS $\in \mathbb{Z}_0^+ = \{s \in \mathbb{Z} \,|\, s \geq 0\}$, where lower scores correspond to better performance. We denote this by $\downarrow$ throughout our evaluations.

[14] MOTA $\in (-\infty, 1]$, where higher scores correspond to better performance (denoted by $\uparrow$).

performance in a single score. This metric is defined as

$$\mathrm{MOTA} = 1 - \frac{\sum_{t=1}^{N} \mathrm{FN}^t + \mathrm{FP}^t + \mathrm{IDS}^t}{\sum_{t=1}^{N} \mathrm{GT}^t}, \tag{5.6}$$

where N is the number of time steps and FN^t, FP^t, IDS^t denote the number of false negatives, false positives, and identity switches at time t, respectively. Similarly, GT^t denotes the number of annotated ground truth objects at time t.

- *Multiple Object Tracking Precision* (MOTP[15]) [41, 402] – measures the localization precision of a tracker as the average location error. This metric is defined as

$$\mathrm{MOTP} = 1 - \frac{\sum_{t=1}^{N} \sum_{i=1}^{\mathrm{TP}^t} \|\mathbf{x}_{\mathrm{G},i}^t - \mathbf{x}_{\mathrm{T},m}^t\|_2}{\tau_d \sum_{t=1}^{N} \mathrm{TP}^t}, \tag{5.7}$$

where $\mathbf{x}_{\mathrm{T},m}^t$ is the location of the m-th hypothesis which has been matched with the ground truth annotation $\mathbf{x}_{\mathrm{G},i}^t$ at time t. $\mathrm{TP}^t = \mathrm{GT}^t - \mathrm{FN}^t$ denotes the number of true positive tracker hypotheses for the current time step. Note that MOTP, despite the similar name, is not related to precision (*i.e. positive predictive value* or *relevance*) in the context of evaluating classifiers and object detectors.

- *Trajectory Quality Measures* [446] – these measures are widely used to reason about the consistency of the tracking output. In particular, each ground truth trajectory can be classified as *mostly tracked* (MT), *partially tracked* (PT) or *mostly lost* (ML), depending on how much of it is covered by the tracker's hypotheses. More precisely, if a ground truth trajectory is successfully tracked – *i.e.* if there is a matching hypothesis – for at least 80 % of its total length, it is considered to be mostly tracked. If the trajectory is only covered by tracker hypotheses for less than 20 % of its total length, it is considered to be mostly lost. Otherwise, the trajectory is classified as partially tracked. Note that identity switches have no effect on these quality measures. To avoid cluttering the result listings, we will only report MT and ML as fractions of the number of ground truth trajectories[16], *i.e.* MT/GT and ML/GT, since GT – the total number of ground truth trajectories – can be recalled from Table 5.14 and PT is redundant, *i.e.* PT = GT − MT − ML.

  Additionally, these quality measures include the number of identity switches (IDS) and the number of *trajectory fragmentation* (FM[17]). The latter counts how many times a ground truth trajectory is interrupted, *i.e.* how often its status changed from being tracked to being missed by the tracker. Thus, lower FM scores indicate that the tracker is able to generate long and persistent trajectories.

[15] $\mathrm{MOTP} \in [0,1]$, where higher scores correspond to better performance (denoted by $\uparrow$).
[16] Thus, $\mathrm{MT} \in [0,1]$ and $\mathrm{ML} \in [0,1]$, where higher MT scores (denoted by $\uparrow$) and lower ML scores (denoted by $\downarrow$) correspond to better tracking performance, respectively.
[17] $\mathrm{FM} \in \mathbb{Z}_0^+$, where lower numbers correspond to better performance (denoted by $\downarrow$).

Table 5.15: Default parameter settings for the occlusion geodesics-based tracker variants (OccGeo). Unless stated otherwise, these parameters have been fixed throughout all experiments.

Parameter		**Value**
Conservative association threshold in [m/s]	$\tau_c \in (0, \infty)$	2.00
Physically feasible motion cut-off	$\tau_p \in [0, 1]$	10^{-4}
Plausible motion variance	$\sigma_p^2 \in (0, \infty)$	1.30
Directional motion variance	$\sigma_d^2 \in (0, 1]$	0.40
Detector belief factor	$\beta_d \in [0, 1]$	0.70

We use the official MOT challenge framework [256, 309] to compute all metrics. To allow initialization and termination in our causal tracking framework, we allocate a 100 [px] wide border around each camera image as the entrance and exit regions. We skip these regions during evaluation for a fair comparison between causal and offline approaches. Thus, we effectively track on the inner regions of size 568×376 for all PETS'09 sequences and 1720×880 for the TownCentre sequence, respectively. Additionally, we linearly interpolate missing object detections for the reported trajectories to prevent skewing the results.

5.2.3 Ablation Study

The following experiments provide detailed insights into the sensitivity of out MOT approach regarding (i) its parameter settings and (ii) its dependency on the used detector. For this ablation study, we report the tracking performance averaged over all sequences, *i.e.* PETS'09 S2L1, S2L2, and S2L3, as well as TownCentre. We will vary one parameter of our occlusion geodesics-based tracker – denoted as OccGeo – while keeping all others fixed. In particular, we use the default parameter settings as summarized in Table 5.15.

Additionally, we report the runtime – in frames per second (FPS) – of all experiments to indicate the performance versus speed tradeoff. Similar to our single object tracking evaluation, all experiments have been conducted on a dedicated computer – an Intel® NUC *Skull Canyon* with a 6th generation Core™ i7 processor, recall Section 5.1.3 – to ensure consistent runtime measurements. To avoid skewing these measures, we only report the tracking time, *i.e.* without the time required to obtain the input detections. A separate analysis of different object detectors will be presented in Section 5.2.3.2. Detection experiments which require a GPU have been conducted on a PC with a 2nd generation Core™ i7 processor and an NVIDIA® GeForce® Titan Xp GPU.

5.2.3.1 Trajectory Model Parameters

To track multiple objects, our occlusion geodesics-based tracking algorithm relies on several intuitive parameters, namely (i) thresholds to avoid implausible assignments, (ii) vari-

Table 5.16: Effects of varying threshold parameters τ_c and τ_p of our MOT approach. Best, second best and third best results have been highlighted for each measure.

(a) Conservative association threshold τ_c.

τ_c	MOTA↑	MOTP↑	MT/GT↑	ML/GT↓	IDS↓	FM↓	FPS↑
0.50	0.49	0.65	0.39	0.16	657	571	6.1
1.00	0.49	0.65	0.41	0.16	640	562	8.2
1.50	0.48	0.65	0.40	0.15	599	544	9.6
2.00	0.48	0.65	0.38	0.16	576	561	12.8
2.50	0.47	0.65	0.38	0.16	599	536	15.3
3.00	0.47	0.65	0.37	0.16	612	532	16.9
3.50	0.47	0.65	0.36	0.16	640	554	19.1
4.00	0.47	0.66	0.36	0.16	622	553	19.9
4.50	0.48	0.65	0.38	0.15	640	570	20.3

(b) Feasible movement threshold τ_p.

τ_p	MOTA↑	MOTP↑	MT/GT↑	ML/GT↓	IDS↓	FM↓	FPS↑
10^{-7}	0.49	0.66	0.35	0.14	688	679	8.4
10^{-6}	0.49	0.66	0.35	0.14	688	679	8.4
10^{-5}	0.46	0.66	0.35	0.16	589	589	9.5
10^{-4}	0.48	0.65	0.38	0.16	576	561	12.8
10^{-3}	0.45	0.65	0.38	0.15	577	507	12.1
10^{-2}	0.40	0.64	0.38	0.15	505	471	12.2
10^{-1}	0.30	0.63	0.33	0.19	517	418	9.4

ances to penalize significant motion deviations, and (iii) a factor to represent our degree of belief in the object detector. For all of the following experiments, we rely on object detections obtained by the Aggregated Channel Features (ACF) [109] detector.

Threshold Parameters. We start this ablation study by analyzing the effects of the two threshold parameters, summarized in Table 5.16. The threshold τ_c influences how many detections are handled within the conservative association step. As a rule of thumb, it should be set to the expected average object velocity. Thus, we use a default setting of $\tau_c = 2\,[\text{m}/\text{s}]$ which allows to handle both inaccurate 3D coordinate projections – *e.g.* caused by loose object bounding boxes – and pedestrians moving faster than the average walking speed. The threshold τ_p controls how fast we expect an occluded object to move while it is not visible. Since our implementation relies on normalized distances – to avoid a temporally dependent parameter, recall Section 4.3.3 – we use a default setting of $\tau_p = 10^{-4}$. Note that the overall tracking performance is very stable when varying either of the threshold levels.

Table 5.17: Effects of varying motion variances σ_d^2 and σ_p^2. Best, second best and third best results have been highlighted for each measure.

(a) Directional variance σ_d^2.

σ_d^2	MOTA↑	MOTP↑	MT/GT↑	ML/GT↓	IDS↓	FM↓	FPS↑
0.10	0.49	0.65	0.38	0.16	575	556	12.7
0.20	0.48	0.65	0.38	0.16	569	550	12.5
0.30	0.48	0.65	0.37	0.15	581	560	12.9
0.40	0.48	0.65	0.38	0.16	576	561	12.8
0.50	0.49	0.65	0.39	0.15	589	552	13.1
0.60	0.48	0.65	0.40	0.15	595	551	13.0
0.70	0.48	0.65	0.40	0.16	596	550	13.0
0.80	0.48	0.65	0.39	0.16	590	554	12.9
0.90	0.48	0.65	0.39	0.15	587	549	13.0
1.00	0.47	0.66	0.38	0.16	597	552	13.1

(b) Plausible motion variance σ_p^2.

σ_p^2	MOTA↑	MOTP↑	MT/GT↑	ML/GT↓	IDS↓	FM↓	FPS↑
0.10	0.23	0.62	0.26	0.24	525	331	10.4
0.30	0.29	0.63	0.33	0.20	512	410	8.4
0.50	0.36	0.64	0.37	0.16	535	463	9.4
0.70	0.41	0.64	0.39	0.15	537	495	12.4
0.90	0.46	0.65	0.39	0.15	547	505	12.4
1.10	0.47	0.66	0.40	0.14	559	502	12.8
1.30	0.48	0.65	0.38	0.16	576	561	12.8
1.50	0.47	0.65	0.35	0.15	609	576	12.6
1.70	0.48	0.66	0.35	0.15	603	613	9.7
1.90	0.48	0.65	0.36	0.14	651	645	8.4

Motion Variance. The motion-based confidence terms in our object likelihood function rely on two predefined variance parameters. More precisely, σ_p^2 influences the plausible motion term, whereas σ_d^2 influences the penalization of changing the movement direction. As can bee seen from the results in Table 5.17, our tracking approach is again rather insensitive to these parameter settings. The only notable performance degradation occurs when choosing σ_p^2 too low, as this constrains the plausible motion of the occluded object too much and thus, prevents re-assigning detections to the corresponding trajectory. This results in many lost trajectories, as can be seen by the low MT and MOTA scores for $\sigma_p^2 \leq 0.5$. As a consequence, the corresponding IDS and FM scores are also low, which shows that IDS and FM on their own are not indicative of good tracking performance, as already mentioned in Section 5.2.2. Hence, it is important to always consider several complementary metrics to analyze a MOT approach, *e.g.* MOTA in combination with MT and IDS.

Table 5.18: Effects of varying the detector belief factor β_d. Best, second best and third best results have been highlighted in each column.

β_d	MOTA$^\uparrow$	MOTP$^\uparrow$	MT/GT$^\uparrow$	ML/GT$^\downarrow$	IDS$^\downarrow$	FM$^\downarrow$	FPS$^\uparrow$
0.00	0.47	0.65	0.38	0.15	560	547	12.8
0.10	0.47	0.65	0.38	0.15	558	543	12.7
0.20	0.46	0.65	0.38	0.15	575	545	12.7
0.30	0.46	0.65	0.37	0.15	570	545	12.8
0.40	0.47	0.65	0.38	0.15	574	547	12.9
0.50	0.46	0.65	0.38	0.15	580	545	12.8
0.60	0.47	0.65	0.38	0.16	584	553	12.9
0.70	0.48	0.65	0.38	0.16	576	561	12.8
0.80	0.48	0.65	0.38	0.16	581	557	13.3
0.90	0.48	0.65	0.37	0.15	575	555	13.2
1.00	0.46	0.65	0.31	0.15	756	637	13.3

Detector Reliability. The final parameter in our tracking model represents our belief in the detector – more precisely, how well we expect the detector to perform if the object is fully visible, *i.e.* not occluded at all. If we expect the detector to never fail under such ideal conditions, then any missed object is only allowed to move within occluded regions. In practice, however, such an optimal detector is not available and thus, our model also allows missed objects to move in nonoccluded regions. Nevertheless, such cases occur only rarely as state-of-the-art detectors typically achieve high recall levels, at least for fully visible objects. To model this uncertainty, we use the detector belief factor β_d, which is evaluated in Table 5.18. As a rule of thumb, this parameter should be set to approximately the area under the detector's precision-recall curve[18] (AUC). For example, the average AUC of the used ACF detector over all sequences is 0.73 and consequently, choosing a belief factor $\beta_d \in [0.6, 0.9]$ yields the best tracking results. Note also, that the tracking performance degrades gracefully when choosing sub-optimal belief factors.

5.2.3.2 Object Detector Influence

As any tracking-by-detection approach heavily relies on the quality of the employed object detector, we analyze the effects of using various state-of-the-art detectors for our tracker.

Detector Evaluation. Before analyzing the tracking performance *w.r.t.* different object detectors, we first evaluate their detection performance. In particular, we investigate both classical approaches – based on hand-crafted features, such as Aggregated Channel Features (ACF) [109], Deformable Part-based Models (DPM) [135], HOG-based Intersection Kernel Support Vector Machines (IKSVM) [295], Locally Decorrelated Features (LDCF) [322], and Poselets [56] – as well as recent neural network-based frameworks, such as Faster R-CNN (F-RCNN) [362], Region-based Fully Convolutional Net-

[18] Detection performance will be analyzed within the next section.

works (R-FCN) [90], Single Shot Multi-Box Detector (SSD) [280], and You Only Look Once (YOLO) [359]. For all F-RCNN, R-FCN and SSD variants we use the corresponding TensorFlow [204] models[19]. For all other detectors, we used the publicly available implementations with the default parameter settings as suggested in the corresponding publications.

We conduct all detection experiments on the same four video sequences, *i.e.* PETS'09 and TownCentre. Since we focus on pedestrian detection, we employ the widely used recall (*i.e.* sensitivity) and precision (*i.e.* positive predictive value) metrics to compare these approaches via precision-recall curves (PRC). These are defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{and} \qquad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5.8}$$

where TP, FP, FN denotes the number of true positives, false positives and false negative (*i.e.* missed) detections, respectively. Note that neither recall nor precision depend on the number of true negatives (TN). For this reason, PRCs are considered more informative when evaluating on imbalanced datasets [100] in contrast to the alternative receiver operating characteristics (ROC). Thus, we rely on precision-recall curves to avoid skewing the following evaluation. To summarize the plots and rank detectors according to their performance, we use the area under the precision-recall curve (AUC[20]).

Note that comparing the detection outputs directly would result in a highly skewed and inconclusive evaluation. On the one hand, each detector depends substantially on the annotations and quality of its training dataset[21]. However, there is a large variation *w.r.t.* the annotations of publicly available pedestrian detection datasets, *e.g.* compare INRIA [91] with Caltech [108]. On the other hand, the publicly available ground truth annotations for the PETS'09 and TownCentre sequences were obtained by manually refining the output of off-the-shelf detectors. Thus, a direct comparison would favor the class of detectors used to obtain the ground truth annotations.

To avoid such a biased analysis, we apply a bounding box regression step. In contrast to the refinement step of recent object detectors, such as DPM [135] or R-CNN [154], we learn a transformation from the detector's final bounding box output to the corresponding ground truth annotations. Thus, our refinement step simulates fine-tuning each detector on the corresponding video sequence. More precisely, we uniformly sample 10 % of the frames and match the ground truth annotations with the detector's output via the Hungarian algorithm [317], where we use the bounding box intersection over union (IOU) to define the assignment cost. Additionally, let $D_i = (\mathbf{c}_{D_i},\, w_{D_i},\, h_{D_i})^\top$ denote the i-th $w_{D_i} \times h_{D_i}$ detection bounding box centered at $\mathbf{c}_{D_i} = (x_{D_i}, y_{D_i})^\top$. With a slight abuse of notation we use tuples as vectors, *i.e.* $D_i = (x_{D_i},\, y_{D_i},\, w_{D_i},\, h_{D_i})^\top$, in the following. Then, our

[19] We use the pretrained network weights from the TensorFlow detection model zoo, *i.e.* commit `f7e99c0` to the official repository https://github.com/tensorflow/models, from 18 November 2017.

[20] AUC $\in [0, 1]$, where higher scores indicate better performance (denoted by $\uparrow$).

[21] The attentive reader might recall our mantra from Section 2.3.2.

goal is to learn the transformation coefficients $\mathbf{w}_\Omega$, with $\Omega \in \{x, y, w, h\}$, to transform the elements of D_i such that we obtain the refined detection $\widehat{D}_i = \left(x_{\widehat{D}_i},\, y_{\widehat{D}_i},\, w_{\widehat{D}_i},\, h_{\widehat{D}_i}\right)^\top$. In particular, we transform the center coordinates as

$$x_{\widehat{D}_i} = w_{D_i}\, D_i^\top\, \mathbf{w}_x + x_{D_i}, \qquad \text{and} \qquad y_{\widehat{D}_i} = h_{D_i}\, D_i^\top\, \mathbf{w}_y + y_{D_i}, \tag{5.9}$$

and the bounding box dimensions as

$$w_{\widehat{D}_i} = w_{D_i}\, D_i^\top\, \mathbf{w}_w, \qquad \text{and} \qquad h_{\widehat{D}_i} = h_{D_i}\, D_i^\top\, \mathbf{w}_h. \tag{5.10}$$

We learn the coefficients $\mathbf{w}_\Omega$ by optimizing the regularized least squares objective

$$\mathbf{w}_\Omega = \arg\min_{\widehat{\mathbf{w}}_\Omega} \sum_{i=1}^{N} \| D_i^\top\, \widehat{\mathbf{w}}_\Omega - t_{\Omega,i} \|_2^2 + \lambda\, \| \widehat{\mathbf{w}}_\Omega \|_2^2, \tag{5.11}$$

where N is the number of matches, $t_{\Omega,i}$ denotes the corresponding regression target, and λ is a regularization factor. This standard ridge regression problem can be solved in closed form, *e.g.* via Cholesky factorization. We define the regression targets using the matching ground truth annotation $G_i = (x_{G_i},\, y_{G_i},\, w_{G_i},\, h_{G_i})^\top$ by the relative center offsets

$$t_{x,i} = \frac{x_{G_i} - x_{D_i}}{w_{D_i}}, \qquad \text{and} \qquad t_{y,i} = \frac{y_{G_i} - y_{D_i}}{h_{D_i}}, \tag{5.12}$$

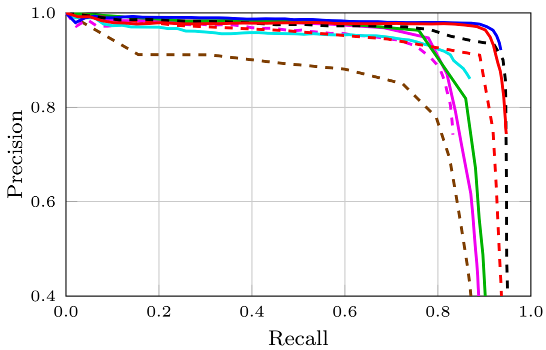
and the relative scale changes

$$t_{w,i} = \frac{w_{G_i}}{w_{D_i}}, \qquad \text{and} \qquad t_{h,i} = \frac{h_{G_i}}{h_{D_i}}. \tag{5.13}$$

Although our regression targets and inputs differ from the bounding box regression in [135, 154], we similarly found it necessary to center and decorrelate the targets – *i.e.* apply a whitening transform – before optimization and use a larger regularization factor of $\lambda = 10^3$.

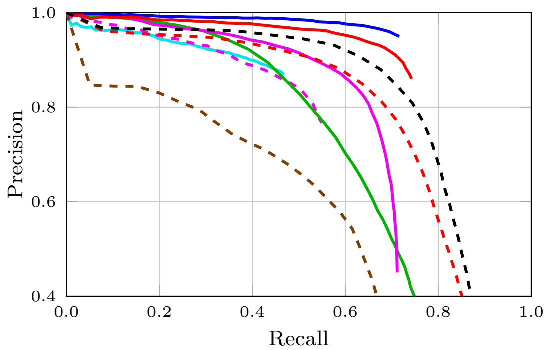
Given the refined bounding boxes, we now can fairly compare the different object detectors. The precision-recall curves for the best detector variants are shown in Figure 5.12 and summarized in Tables 5.19 and 5.20. The tables also show the improvement due to the refinement step. Note that this post-processing step is especially crucial for a fair comparison of IKSVM [295], as it originally reports very loose bounding boxes which have a low IOU with the tight ground truth annotations. Overall, the top-performing deep-learning based detectors, *i.e.* F-RCNN [362] and R-FCN [90], perform on par with the best detectors based on hand-crafted features, *i.e.* DPM [135], ACF [109] and Poselets [56]. Furthermore, these results show the advantage of region-sampling approaches – which either densely score detection hypotheses in a sliding window manner, *e.g.* [91, 135], or employ region-of-interest selection in a pre-processing step, *e.g.* [154, 362] – over the significantly faster, but less accurate SSD [280] and YOLO [359] – which classify pre-fixed sets of candi-

Detector	AUC$^{\uparrow}$
DPM$_{v4}$$^{\oplus}$ [135]	0.92
ACF$^{\oplus}$ [109]	0.92
F-RCNN NAS [362, 500]	0.92
R-FCN ResNet$_{101}$ [90, 181]	0.89
Poselets [56]	0.87
IKSVM$^{\oplus}$ [295]	0.85
LDCF [322]	0.83
YOLO$_{v2}$ [358]	0.80
SSD Inception$_{v2}$ [280, 406]	0.76



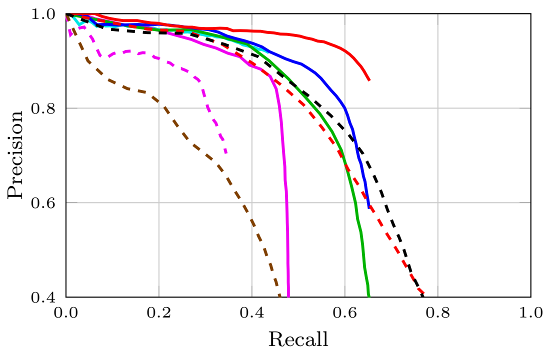
(a) PETS'09 S2L1.

Detector	AUC$^{\uparrow}$
F-RCNN ResNet$_{101}$ [181, 362]	0.79
R-FCN ResNet$_{101}$ [90, 181]	0.75
ACF$^{\oplus}$ [109]	0.72
DPM$_{v4}$$^{\oplus}$ [135]	0.71
IKSVM$^{\oplus}$ [295]	0.66
Poselets [56]	0.65
YOLO$_{v2}$ [358]	0.51
SSD Inception$_{v2}$ [280, 406]	0.50
LDCF [322]	0.44



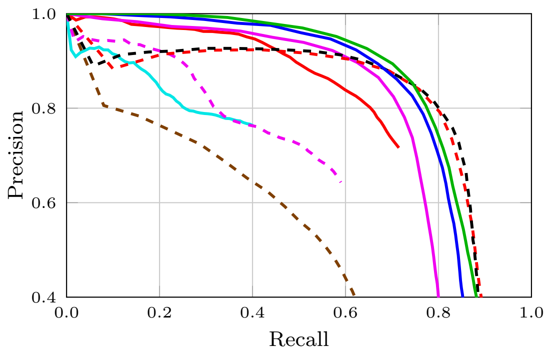
(b) PETS'09 S2L2.

Detector	AUC$^{\uparrow}$
F-RCNN ResNet$_{101}$ [181, 362]	0.68
R-FCN ResNet$_{101}$ [90, 181]	0.67
ACF$^{\oplus}$ [109]	0.63
DPM$_{v5}$$^{\oplus}$ [135]	0.60
Poselets [56]	0.60
IKSVM$^{\oplus}$ [295]	0.45
LDCF [322]	0.42
SSD Inception$_{v2}$ [280, 406]	0.35
YOLO$_{v2}$ [358]	0.31



(c) PETS'09 S2L3.

Detector	AUC$^{\uparrow}$
Poselets [56]	0.82
DPM$_{v5}$ Person Grammar [135, 153]	0.80
R-FCN ResNet$_{101}$ [90, 181]	0.78
F-RCNN Inception-ResNet$_{v2}$ [362, 407]	0.78
IKSVM [295]	0.74
ACF [109]	0.66
YOLO$_{v2}$ [358]	0.49
SSD Inception$_{v2}$ [280, 406]	0.45
LDCF [322]	0.33



(d) TownCentre.

Figure 5.12: Precision-recall plots for various state-of-the-art pedestrian detectors on the MOT sequences. Each legend is sorted by the area under the precision-recall curve (AUC). The symbols $^{\oplus}$ and $^{\ominus}$ indicate that the best detection performance was achieved by upsampling or downsampling the input image, respectively.

Table 5.19: Detection results on the PETS'09 [136] dataset, showing the best configuration of various off-the-shelf pedestrian detectors. The detectors are ranked by the area under the precision-recall curve (AUC). Numbers in parentheses show the improvement due to bounding box refinement. The symbol $\oplus$ indicates that the best detection performance was achieved by upsampling the input image. Results for DPM$_{v4}^{\oplus}$ have been kindly provided by the authors of [192, 193].

(a) PETS'09 S2L1.

Detector	**Training Data**	**AUC$^{\uparrow}$**	**GPU**	**FPS$^{\uparrow}$**
DPM$_{v4}^{\oplus}$ [135]	VOC$_{09}$ [122]	$0.92_{(+0.01)}$		—
ACF$^{\oplus}$ [109]	INRIA [91]	$0.92_{(+0.00)}$		8.11 ± 0.47
F-RCNN NAS [362, 500]	COCO [276]	$0.92_{(+0.00)}$	✓	2.60 ± 0.13
R-FCN ResNet$_{101}$ [90, 181]	COCO [276]	$0.89_{(+0.00)}$	✓	9.06 ± 0.52
Poselets [56]	H3D [56]	$0.87_{(+0.00)}$		0.07 ± 0.01
IKSVM$^{\oplus}$ [295]	INRIA [91]	$0.85_{(+0.85)}$		0.03 ± 0.00
LDCF [322]	Caltech [108]	$0.83_{(+0.02)}$		3.28 ± 0.19
YOLO$_{v2}$ [358]	COCO [276]	$0.80_{(+0.00)}$	✓	62.76 ± 2.90
SSD Inception$_{v2}$ [280, 406]	COCO [276]	$0.76_{(+0.01)}$	✓	16.06 ± 1.31

(b) PETS'09 S2L2.

Detector	**Training Data**	**AUC$^{\uparrow}$**	**GPU**	**FPS$^{\uparrow}$**
F-RCNN ResNet$_{101}$ [181, 362]	COCO [276]	$0.79_{(+0.03)}$	✓	7.23 ± 0.47
R-FCN ResNet$_{101}$ [90, 181]	COCO [276]	$0.75_{(+0.03)}$	✓	9.07 ± 0.57
ACF$^{\oplus}$ [109]	INRIA [91]	$0.72_{(+0.04)}$		8.36 ± 0.64
DPM$_{v4}^{\oplus}$ [135]	VOC$_{09}$ [122]	$0.71_{(+0.03)}$		—
IKSVM$^{\oplus}$ [295]	INRIA [91]	$0.66_{(+0.61)}$		0.02 ± 0.01
Poselets [56]	H3D [56]	$0.65_{(+0.05)}$		0.03 ± 0.01
YOLO$_{v2}$ [358]	COCO [276]	$0.51_{(+0.07)}$	✓	63.50 ± 1.88
SSD Inception$_{v2}$ [280, 406]	COCO [276]	$0.50_{(+0.08)}$	✓	15.99 ± 1.45
LDCF [322]	Caltech [108]	$0.44_{(+0.04)}$		3.36 ± 0.20

(c) PETS'09 S2L3.

Detector	**Training Data**	**AUC$^{\uparrow}$**	**GPU**	**FPS$^{\uparrow}$**
F-RCNN ResNet$_{101}$ [181, 362]	COCO [276]	$0.68_{(+0.05)}$	✓	7.30 ± 0.55
R-FCN ResNet$_{101}$ [90, 181]	COCO [276]	$0.67_{(+0.04)}$	✓	9.04 ± 0.69
ACF$^{\oplus}$ [109]	INRIA [91]	$0.63_{(+0.07)}$		9.14 ± 1.06
DPM$_{v5}^{\oplus}$ [135]	INRIA [91]	$0.60_{(+0.09)}$		0.08 ± 0.00
Poselets [56]	H3D [56]	$0.60_{(+0.09)}$		0.06 ± 0.03
IKSVM$^{\oplus}$ [295]	INRIA [91]	$0.45_{(+0.45)}$		0.03 ± 0.01
LDCF [322]	Caltech [108]	$0.42_{(+0.08)}$		3.59 ± 0.22
SSD Inception$_{v2}$ [280, 406]	COCO [276]	$0.35_{(+0.00)}$	✓	15.76 ± 1.61
YOLO$_{v2}$ [358]	COCO [276]	$0.31_{(+0.03)}$	✓	62.56 ± 2.96

Table 5.20: Pedestrian detection results on the TownCentre [36] dataset, showing the best configuration of various off-the-shelf pedestrian detectors. The detectors are ranked by the area under the precision-recall curve (AUC). Numbers in parentheses show the improvement due to bounding box refinement.

Detector	Training Data	AUC$^{\uparrow}$	GPU	FPS$^{\uparrow}$
Poselets [56]	H3D [56]	$0.82_{(+0.06)}$		0.01 ± 0.00
DPM_{v5} Person Grammar [135, 153]	VOC_{10} [122]	$0.80_{(+0.06)}$		0.04 ± 0.00
R-FCN ResNet_{101} [90, 181]	COCO [276]	$0.78_{(+0.04)}$	✓	8.52 ± 0.54
F-RCNN Inception-$\text{ResNet}_{\text{v2}}$ [362, 407]	COCO [276]	$0.78_{(+0.02)}$	✓	2.43 ± 0.13
IKSVM [295]	INRIA [91]	$0.74_{(+0.66)}$		0.02 ± 0.00
ACF [109]	INRIA [91]	$0.66_{(+0.03)}$		7.40 ± 0.35
YOLO_{v2} [358]	COCO [276]	$0.49_{(+0.05)}$	✓	65.39 ± 0.99
SSD $\text{Inception}_{\text{v2}}$ [280, 406]	COCO [276]	$0.45_{(+0.07)}$	✓	15.63 ± 1.47
LDCF [322]	Caltech [108]	$0.33_{(+0.02)}$		2.84 ± 0.09

date bounding boxes. This accuracy versus speed tradeoff can be seen particularly well for more crowded scenarios, such as PETS'09 S2L2 or S2L3. More detailed detection results – including different variants of each detector – can be found in Appendix C.2.

Despite the promising detection results, there is still room for future improvements, especially considering denser crowds of pedestrians. Although there have been some attempts on detecting highly occluded pedestrians, *e.g.* [409, 410], these mostly focus on groups of 2–3 people and still cannot handle larger crowds sufficiently well. Additionally, there is a lack of large-scale training datasets specialized on classical surveillance scenarios – which instead of capturing fronto-parallel or side views of pedestrians need to be recorded from an elevated viewpoint with a large field of view. Such datasets would particularly contribute to performance improvements of data-driven approaches, *i.e.* deep learning-based detectors, and could also be used to refine object proposals in crowded scenarios.

Detection-based Tracking Performance. We use the best variant of each detector class to analyze the MOT performance *w.r.t.* the underlying detector. Table 5.21 summarizes the tracking results, whereas detailed per-sequence results can be found in Appendix C.3. Overall, our tracking approach achieves the best performance by employing ACF [109], DPM [135], F-RCNN [362] or R-FCN [90] detections. Furthermore, considering the substantially lower scores when relying on SSD [280] and YOLO [359] detections, this analysis shows the importance of choosing a suitable object detector. In particular, for visual surveillance scenarios a detector should be able to robustly detect partially occluded pedestrians.

Table 5.21: Influence of different state-of-the-art object detectors on the tracking-by-detection performance of our OccGeo tracker. Best, second best and third best results have been highlighted in each column.

Detector	MOTA$^{\uparrow}$	MOTP$^{\uparrow}$	MT/GT$^{\uparrow}$	ML/GT$^{\downarrow}$	IDS$^{\downarrow}$	FM$^{\downarrow}$	FPS$^{\uparrow}$
DPM [135]	0.58	0.65	0.28	0.23	429	489	17.4
F-RCNN [362]	0.50	0.62	0.24	0.34	326	459	16.9
ACF [109]	0.48	0.65	0.38	0.16	576	561	12.7
R-FCN [90]	0.48	0.62	0.25	0.23	550	556	16.6
IKSVM [295]	0.46	0.61	0.18	0.30	321	410	17.5
Poselets [56]	0.46	0.64	0.24	0.20	485	549	16.0
LDCF [322]	0.35	0.64	0.15	0.34	482	479	10.1
SSD [280]	0.29	0.60	0.08	0.41	465	512	13.0
YOLO [359]	0.29	0.58	0.08	0.39	442	618	9.1

5.2.4 Comparison to the State-of-the-Art

To compare our approach to the state-of-the-art, we rely on the 3D MOT'15 [256] benchmark, which consists of the PETS'09 S2L2 and the TownCentre sequences. As the ground truth annotations used for the 3D MOT'15 benchmark are not publicly available, we rely on the widely used annotations provided by [36, 308]. For a fair comparison, we use the official 3D MOT'15 evaluation framework and the raw tracking results published for the following state-of-the-art approaches:

- GPR-DBN [232] is the leading 3D MOT'15 approach and extends a dynamic Bayesian network (DBN)-based tracker [231] with Gaussian process regression (GPR).
- K-SFM [341] combines a Kalman filtering framework with a social force model (SFM) to efficiently handle pedestrian interactions.
- LP-3D [255] is the 3D MOT'15 baseline approach and solves a global optimization problem on the 3D coordinates via linear programming.
- LP-SFM [253] also solves a global optimization problem via linear programming, but additionally uses a social force model which addresses pedestrian interactions and group behavior to obtain consistent trajectory assignments.
- S-RNN [368] leverages a structure of multiple recurrent neural networks (RNNs) to encode several contextual cues, including appearance, motion and interactions between pedestrians.
- STV [440] builds a space-time-view hypergraph which encodes higher order constraints based on both, geometric and appearance cues, and solves the trajectory assignment by searching for dense sub-hypergraphs using a sampling-based approach.

Table 5.22: Comparison with the state-of-the-art on the 3D MOT'15 [256] benchmark. The second and third column indicate if the corresponding tracker uses an instance-specific appearance model (A) and is causal (C), respectively. All trackers were evaluated with the official input detections provided by the 3D MOT'15 committee. For our occlusion geodesics-based tracker (OccGeo), we additionally report the results using standard DPM [135] detections. Runtime measurements for the officially benchmarked trackers are provided by [256]. Best, second best and third best results have been highlighted for each metric.

Tracker	A	C	MOTA↑	MOTP↑	MT↑	ML↓	IDs↓	FM↓	FPS↑
OccGeo (DPM)		✓	0.51	0.62	0.26	0.24	350	370	7.5
OccGeo (3D MOT'15)		✓	0.31	0.59	0.16	0.32	414	411	4.8
GPR-DBN [232]	✓	✓	0.48	0.62	0.33	0.21	181	270	0.1
LP-SFM [253]			0.31	0.52	0.16	0.22	396	467	8.4
STV [440]	✓		0.31	0.55	0.14	0.25	383	439	1.9
LP-3D [255]			0.30	0.52	0.24	0.14	487	542	83.5
S-RNN [368]	✓	✓	0.22	0.54	0.03	0.36	785	1053	1.2
K-SFM [341]		✓	0.21	0.52	0.07	0.14	1463	1322	30.6

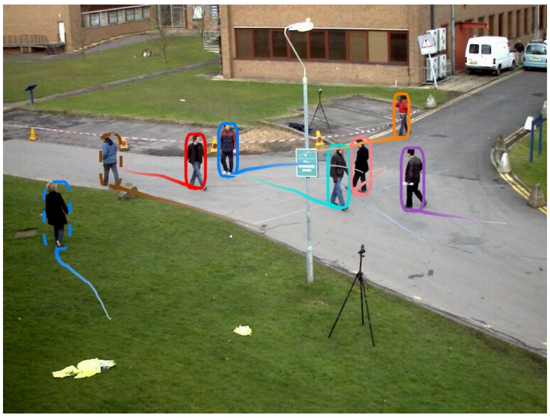
The results of this analysis are summarized in Table 5.22. The minor differences to the official benchmark results can be contributed to the different ground truth annotations and our smaller evaluation region, as we ignore the boundary regions to allow for a fairer comparison between causal and offline approaches, recall Section 5.2.2. More detailed results are listed in Appendix C.3. Considering the publicly available 3D MOT'15 input detections, our approach is only outperformed by the appearance-based GPR-DBN [232] and performs on par with the offline LP-SFM [253] and STV [440]. Furthermore, re-assignment based on our occlusion geodesics is significantly more robust than using complex social force models – as used by the causal K-SFM [341] – or incorporating multiple data-driven models, such as S-RNN [368]. The substantial performance gain when relying on DPM detections again highlights the importance of using a suitable object detector for real-world applications.

Our approach also achieves a favorable runtime performance compared to most of the other tracking approaches. Note that our MATLAB® implementation updates the re-assignment costs of all missed trajectories sequentially. Thus, for a real-world application the tracking speed could by substantially improved by leveraging parallel computation. Nevertheless, our single-threaded prototype already achieves frame rates suitable for time-critical surveillance scenarios, due to the moderate walking speed of pedestrians. Additionally, using a better object detector leads to less ambiguous situations and thus, less computational effort. This can be seen by comparing the speed of our tracker using off-the-shelf DPM detections against the ACF detections published by the 3D MOT'15 committee. The latter cause a significantly higher number of FP and FN detections and thus, require our re-assignment calculations more often.

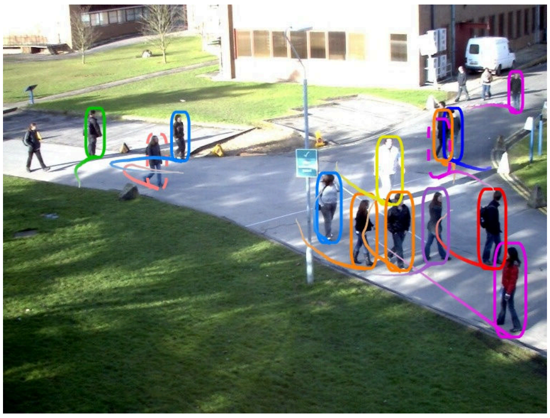
5.2.5 Discussion

Our occlusion geodesics-based tracker ranks amongst the state-of-the-art approaches both with respect to tracking performance and speed. In particular, by leveraging only geometric context information, we can build a powerful model while keeping the complexity low. Moreover, we perform on par with the best appearance-based approaches and also outperform methods which rely on explicitly modeling object behavior via sophisticated interaction models. Qualitative results of our tracker are shown in Figure 5.13.

As mentioned in Section 5.2.3.2, the object detector plays a major role in achieving good tracking-by-detection results. This is also shown by our detailed evaluations, especially when considering more crowded scenarios, such as PETS'09 S2L2 and S2L3. There, object detectors often miss the pedestrians due to the frequent inter-object occlusions, as shown in Figure 5.14. Since our model only relies on geometric cues, identity switches cannot be avoided in such dense crowds, as there are usually several missed ob-



(a) PETS'09 S2L1.

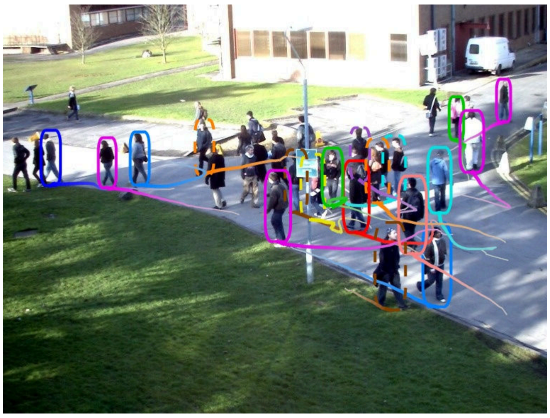


(b) PETS'09 S2L2.

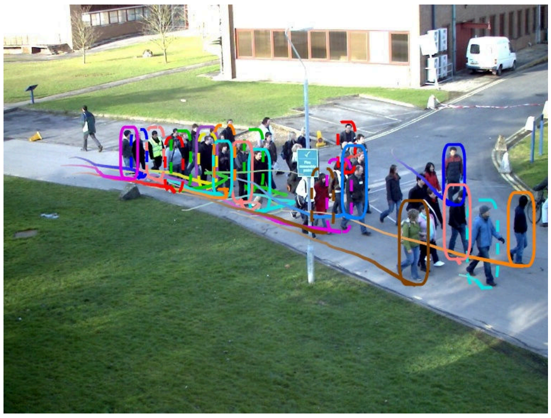
(c) TownCentre.

Figure 5.13: Qualitative results for our occlusion geodesics-based MOT approach on the PETS'09 [136] and TownCentre [36] sequences using DPM [135] detections. Dashed bounding boxes indicate that the corresponding person has been missed by the detector. The coloring of the bounding boxes and trajectories corresponds to the object identities.

jects within a single, narrow, occluded region. In such scenarios, individual appearance models can be helpful to resolve the ambiguities, as shown by [232]. To improve the detection performance in such challenging scenarios, one could either fine-tune the detector to the scene-specific challenges or leverage additional motion cues. For example, static visual surveillance setups allow us to employ background subtraction techniques to locate moving regions, which can then be used to reason about detector failures.



(a) PETS'09 S2L2.



(b) PETS'09 S2L3.

(c) TownCentre.

Figure 5.14: Difficult scenarios for our tracking-by-detection approach, where the object detector misses people too frequently due to full (top row) or partial (bottom row) occlusions. Especially for dense crowds as in PETS'09, we often fail to obtain a reliable motion estimate before a person is missed by the detector which impedes the correct re-assignment.

6 Conclusion

Full speed ahead, hard and fast!

— Pennywise (Every Single Day)

Contents

6.1 Recapitulation

The aim of this thesis was to investigate the benefits of incorporating contextual information to boost the performance of causal visual object tracking approaches. We set out to improve the two major components of the visual tracking loop, namely (i) object representation and (ii) data association – for more details, recall Chapter 1. Following the maxim *temet nosce*[22], we analyzed typical failure cases of the state-of-the-art in visual object tracking. More specifically, we focused on limitations of causal tracking approaches, since these enable time-critical real-world applications, such as autonomous vehicles or automated visual surveillance. In these application domains, tracking approaches often rely on simple models due to their favorable efficiency in order to meet the given runtime requirements. This reduced model complexity, however, often leads to tracking failures whenever the object's visual appearance becomes ambiguous – *i.e.* the tracking model gets confused by the object's surroundings, which subsequently leads to drifting – or whenever the object disappears, at least from the viewpoint of the camera, *e.g.* due to occlusions.

[22] Latin aphorism meaning *know thyself*, translated from the Delphic maxim *gnōthi seauton* (Greek).

In this thesis, we tried to address these issues by tackling the following research questions:

(i) How can we ensure a robust object model for localization in the presence of visually similar regions?

Causal color-based tracking approaches typically drift towards such visually distracting regions. To overcome this limitation, we introduced a distractor-aware object model which substantially reduces the risk of tracking failures in Chapter 3. This allowed us to exploit the favorable simplicity and efficiency of color-based models while achieving state-of-the-art robustness, as shown in Chapter 5.1.

(ii) How can we model the likelihood of an object being present at a specific location while it is occluded, to allow for a consistent trajectory re-assignment once the object is re-detected?

When dealing with scenarios in which the object of interest may be occluded frequently, causal trackers often fail to reliably re-assign detections to the corresponding object trajectory. To address this issue, we introduced a recursive cost function which weights hidden movements – *i.e.* object motion not seen from the camera viewpoint, either due to occlusions or detection failures – by their plausibility in Chapter 4. Relying on geometric context, we were able to combine the benefits of efficient association-based methods with a reliable re-assignment to increase the tracking robustness, as demonstrated in Chapter 5.2.

Although localizing objects without leveraging context information is infeasible, most tracking approaches only incorporate two rather basic cues, namely the visual appearance of a target and its motion. Other auxiliary information about the target's surroundings is mostly neglected by the research community. In this thesis, we highlighted the importance and benefits of such unattended contextual cues, in particular (i) leveraging the appearance and visual similarity of distractors in combination with the target's appearance, as well as (ii) combining geometric reasoning about target motion within occluded regions with the expected reliability of the object detector. Leveraging these contextual cues for our tracking frameworks allowed us to improve the real-world applications which motivated our research tasks initially, recall Chapter 1. These applications demonstrate the robustness and efficiency of our trackers on a daily basis.

The tracking approaches we investigated cover the two extrema of the visibility spectrum, namely (i) what to do if the object is visible – but so are distracting regions too – and (ii) what to do if the object is not visible, *i.e.* is occluded – and thus, cannot be located until it moves out of the occluded region to be detected again. Each of these cues can be leveraged on its own to make tracking approaches *see*, *i.e.* simple, efficient, and effective. Although real-world applications would definitely benefit from combining these cues, we deliberately focused on analyzing them separately in order to highlight their individual benefits and limitations, respectively.

We performed detailed experimental studies in Chapter 5 and include additional evaluations in Appendix C. To show benefits and limitations of the proposed tracking approaches, we selected suitable testbeds and tracking tasks. On the one hand, single object tracking (SOT) benchmarks cover a wide variety of typical challenges, including illumination variations, non-rigid deformations, generic object classes, as well as camera and object motion. As such sequences usually capture only short-term occlusions and focus on cluttered or distracting backgrounds instead, these provide an ideal testbed for our appearance-based, distractor-aware object model. Multiple object tracking (MOT), on the other hand, requires reasoning about hidden movements, due to the frequent inter-object occlusions. Following recent research trends, as discussed in Chapter 2, we focused on pedestrian tracking tasks to evaluate our occlusion geodesics-based tracker. Although there are significantly less publicly available benchmarks than for SOT, we could select suitable visual surveillance scenarios that exhibit typical MOT challenges, such as varying crowd densities, group interactions, as well as frequent detector failures.

6.2 Outlook

With the rapid progress of computer vision research over the past few years, more and more contextual cues will become easily available and thus, open up new potential improvements. For example, the accuracy of semantic segmentation approaches increased notably on challenging large-scale datasets, such as [276, 327]. By leveraging pixel-accurate semantic knowledge about the scene, visual tracking approaches could be substantially robustified. After all, the world around us fortunately follows well understood physical principles and thus, it should be at the very least highly unlikely to capture object movements which, for example, violate the law of gravity.

Another driving force of future tracking improvements is the steady increase of hardware capabilities. More powerful hardware consequently allows training more complex data-driven models, but even more important, also enables efficient inference required for time-critical applications. Recently, promising results have been obtained by learning pedestrian interactions with recurrent neural networks, *e.g.* [4]. With suitable training datasets and the ability to predict object trajectories in an online setting, such approaches may become a valuable component for causal object trackers.

Summarizing the findings of this thesis, we have shown that often neglected, but easily obtainable, contextual cues can substantially improve visual tracking performance. We demonstrated the benefits of visual appearance and geometric reasoning for both SOT and MOT, by leveraging these cues within rather simplistic frameworks. These models can also be integrated in more complex tracking pipelines to robustify state-of-the-art approaches. Additionally, there are still many information sources left to be explored, not to mention frameworks which jointly leverage these cues. Thus, visual object tracking remains an interesting research field which will continue to contribute to our quest for computer vision's holy grail, *i.e.* fully automated visual scene understanding.

A

List of Acronyms

We live in a world where there is more and more information, and less and less meaning.

— Jean Baudrillard (Simulacra and Simulation)

ABHMC	Adaptive Basin Hopping Monte Carlo
Acc.	Accuracy
ACCT	Adaptive Complex Cell-based Tracker
ACF	Aggregated Channel Features
ACT	Adaptive Color Attributes Tracker
ADNet	Action-Decision Network-based Tracker
ALIEN	Appearance Learning In Evidential Nuisance
ALOV++	Amsterdam Library of Ordinary Videos
AMP	Apparent Motion Patterns
AO	Average Overlap
APG	Accelerated Proximal Gradient
APIDIS	Autonomous Production of Images based on Distributed and Intelligent Sensing
ARBM	Attentional Restricted Boltzmann Machine
ASEF	Average of Synthetic Exact Filters
ASLA	Adaptive Structural Local Sparse Appearance-based Tracker
AUC	Area under the Curve
BACF	Background-aware Correlation Filter
BHMC	Basin Hopping Monte Carlo
BHT	Block Histogram-based Tracker
C-COT	Continuous Convolution Operators Tracker
Caltech	California Institute of Technology

CAT	Context-aware Tracker
CCT	Collaborative Correlation Filter
CCTV	Closed-Circuit Television
CF	Correlation Filter
CF^2	Correlation Filters with Convolutional Features
CFCF	Convolutional Features for Correlation Filters
CFLB	Correlation Filters with Limited Boundaries
CFNet	Correlation Filter Neural Network-based Tracker
CIE	Commission Internationale de l'Éclairage
CLEAR	Classification of Events, Activities and Relationships
CMT	Consensus-based Matching and Tracking
CN	Color Names
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CR	Channel Representation
CREST	Convolutional Residual Tracking
CRF	Conditional Random Field
CRVT	Compressive Sensing-based Real-time Visual Tracker
CSK	Circulant Structure Kernel
CSR-DCF	Channel and Spatial Reliability for DCFs
CVPR	Conference on Computer Vision and Pattern Recognition
CXT	Context Tracker
DAT	Distractor-Aware Tracker
DBN	Dynamic Bayesian Network
DCF	Discriminative Correlation Filter
DFT	Distribution Fields-based Tracker
DGT	Dynamic Graph-based Tracker
DPCF	Deformable Parts Correlation Filters
DPM	Deformable Part-based Model
DSST	Discriminative Scale Space Tracker
EAO	Expected Average Overlap
EAST	Early-Stopping Tracker
EBT	Edge Box Tracker
ECCV	European Conference on Computer Vision
ECO	Efficient Convolution Operators
EDFT	Enhanced Distribution Field Tracking
EFO	Equivalent Filter Operations
EM	Expectation-Maximization
EPFL	École Polytechnique Fédérale de Lausanne
Eq.	Equation
ETH	Eidgenössische Technische Hochschule
F-RCNN	Faster R-CNN
FCNT	Fully Convolutional Network-based Tracker

Fig.	Figure
FLO	Feature-less Object Tracker
FM	Fragmentation
FN	False Negative
FoT	Flock of Trackers
FOV	Field of View
FP	False Positive
FPS	Frames per Second
FRT	Fragment-based Tracker
GLaDOS	Genetic Lifeform and Disk Operating System
GMM	Gaussian Mixture Model
GOTURN	Generic Object Tracking using Regression Networks
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
H3D	Humans in 3D
HART	Hierarchical Attentive Recurrent Tracking
HDT	Hedged Deep Tracking
HOG	Histogram of Oriented Gradients
ICCV	International Conference on Computer Vision
ICG	Institute of Computer Graphics and Vision
IDS	Identity Switches
IIVT	Initialization-Insensitive Visual Tracker
IKSVM	Intersection Kernel Support Vector Machine
IMCMC	Interactive Markov Chain Monte Carlo
INRIA	Institut National de Recherche en Informatique et en Automatique
IOU	Intersection over Union
IQR	Interquartile Range
ITU	International Telecommunication Union
IVT	Incremental Learning-based Visual Tracking
JPDAF	Joint Probabilistic Data Association Filter
KCF	Kernelized Correlation Filter
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
KLT	Kanade-Lucas-Tomasi Tracker
LCT	Long-term Correlation Tracking
LDCF	Locally Decorrelated Features
LGT	Local-Global Tracker
LRS	Learning, Recognition & Surveillance
LRSVT	Laplacian Ranking Support Vector Tracker
LSH	Locality Sensitive Histogram-based Tracker
LSTM	Long Short-term Memory

LT-FLO	Long-term FLO
MCCF	Multi-Channel Correlation Filters
MCMC	Markov Chain Monte Carlo
MCPF	Multi-task Correlation Particle Filter
MDNet	Multi-Domain Convolutional Neural Network-based Tracker
MEEM	Multiple Experts Entropy Minimization Tracker
MHT	Multiple Hypotheses Tracking
MIL	Multiple Instance Learning
MILF	MIL Forests-based Tracker
ML	Mostly Lost
MOSSE	Minimum Output Sum of Squared Error
MOT	Multiple Object Tracking
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MT	Mostly Tracked
MTST	Multi-Task Sparse Learning-based Tracker
MTT	Multiple Target Tracking
MUSTer	Multi-Store Tracker
MVL	Machine Vision Laboratory
NAS	Neural Architecture Search
NCC	Normalized Cross-Correlation
NFS	Need for Speed
NIST	National Institute of Standards and Technology
NMS	Non-Maximum Suppression
noDAT	Distractor-Agnostic Tracker
NUS-PRO	National University of Singapore People and Rigid Objects Dataset
OGT	Online Graph-based Tracker
OPE	One-pass Evaluation
OPER	One-pass Evaluation with Reset
OTB	Online Tracking Benchmark
PaFiSS	Particle Filter with Sample Segmentation
PASCAL	Pattern Analysis, Statistical Modelling and Computational Learning
PETS	Performance Evaluation of Tracking and Surveillance
Pixel	Picture Element
PLT	Pixel-based Lookup-Table Tracker
PNNL	Pacific Northwest National Laboratory
PRC	Precision Recall Curve
PST	Proposal Selection Tracker
PT	Partially Tracked
PTAV	Parallel Tracking and Verification
PTB	Princeton Tracking Benchmark
PTZ	Pan-Tilt-Zoom

R-CNN	Regions with CNN Features
R-FCN	Regression-based Fully Convolutional Network
RATM	Recurrent Attentive Tracking Model
RCT	Real-time Compressive Tracking
RDP	Representative Distance Precision
Re^3	Real-Time Recurrent Regression Network-based Tracker
ResNet	Residual Network
RNN	Recurrent Neural Network
Rob.	Robustness
ROC	Receiver Operating Characteristic
ROLO	Recurrent YOLO-based Tracker
RTT	Recurrently Target-Attending Tracking
RVM	Relevance Vector Machine
SAMF	Scale Adaptive Multiple Features Tracker
SANet	Structure-aware Network-based Tracker
SAT	Structure-aware Hypergraph-based Tracker
SCM	Sparsity-based Collaborative Model for Tracking
SDF	Synthetic Discriminant Function
SFC	Siamese Fully Convolutional Network-based Tracker
SFM	Social Force Model
SfM	Structure from Motion
SINT	Siamese Instance Search Tracker
SMC	Sequential Monte Carlo
SOT	Single Object Tracking
SPOT	Structure Preserving Online Tracker
SPT	Sparse Appearance-based Tracker
SRDCF	Spatially Regularized Discriminative Correlation Filters
SRE	Spatial Robustness Evaluation
SRER	Spatial Robustness Evaluation with Reset
SSAT	Scale- and State-aware Tracker
SSD	Single Shot Multi-Box Detector
Staple	Sum of Template and Pixel-wise Learners
STCT	Sequentially Trained Convolutional Network-based Tracker
Struck	Structured Output Tracking with Kernels
SVM	Support Vector Machine
TCNN	Tree-structured Convolutional Neural Network-based Tracker
TColor	Temple Color
TGPR	Tracking with Gaussian Process Regression
TIR	Thermal Infrared
TLD	Tracking-Learning-Detection
TN	True Negative
TP	True Positive
TP-RNN	Trajectory Predictor using Recurrent Neural Networks

TRE	Temporal Robustness Evaluation
TRECVid	Text Retrieval Conference Video Retrieval Evaluation
TUD	Technische Universität Darmstadt
VOC	Visual Object Challenge
VOT	Visual Object Tracking
VTD	Visual Tracking via Decomposition
VTS	Visual Tracker Sampling
YOLO	You Only Look Once

B
List of Publications

I know kung fu.

— Neo (The Matrix)

Contents

B.1 Conference and Journal Publications

My work at the Institute of Computer Graphics and Vision led to the following peer-reviewed publications. For the sake of completeness of this thesis, all papers are listed chronologically along with their corresponding abstract.

2012

Unsupervised Calibration of Camera Networks and Virtual PTZ Cameras
Horst Possegger, Matthias Rüther, Sabine Sternig, Thomas Mauthner, Manfred Klopschitz, Peter M. Roth, and Horst Bischof
In *Proceedings of the Computer Vision Winter Workshop (CVWW)*
Mala Nedelja (Slovenia), February 2012
Accepted for oral presentation
Winner of the Best Student Paper award

Abstract: Pan-Tilt-Zoom (PTZ) cameras are widely used in video surveillance tasks. In particular, they can be used in combination with static cameras to provide high resolution imagery of interesting events in a scene on demand. Nevertheless, PTZ cameras only provide a single trajectory at a time. Hence, engineering algorithms for common computer vision tasks, such as automatic calibration or tracking, for camera networks including PTZ cameras is difficult. Therefore, we propose a *virtual PTZ* (vPTZ) camera to simplify the algorithm development for such camera networks. The vPTZ camera is built on a cylindrical panoramic view of the scene and allows to re-position its field of view arbitrarily to provide several trajectories. Further, we propose an unsupervised extrinsic self-calibration method for a network of static cameras and PTZ cameras solely based on correspondences between tracks of a walking human. Our experimental results show that we can obtain accurate estimates of the extrinsic camera parameters in both, outdoor and indoor scenarios.

2013

Robust Real-Time Tracking of Multiple Objects by Volumetric Mass Densities

Horst Possegger, Sabine Sternig, Thomas Mauthner, Peter M. Roth, and Horst Bischof

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

Portland (Oregon), June 2013

Accepted for poster presentation

Abstract: Combining foreground images from multiple views by projecting them onto a common ground-plane has been recently applied within many multi-object tracking approaches. These planar projections introduce severe artifacts and constrain most approaches to objects moving on a common 2D ground-plane. To overcome these limitations, we introduce the concept of an *occupancy volume* – exploiting the full geometry and the objects' center of mass – and develop an efficient algorithm for 3D object tracking. Individual objects are tracked using the local mass density scores within a particle filter based approach, constrained by a Voronoi partitioning between nearby trackers. Our method benefits from the geometric knowledge given by the *occupancy volume* to robustly extract features and train classifiers on-demand, when volumetric information becomes unreliable. We evaluate our approach on several challenging real-world scenarios including the public APIDIS dataset. Experimental evaluations demonstrate significant improvements compared to state-of-the-art methods, while achieving real-time performance.

2014

Occlusion Geodesics for Online Multi-Object Tracking

Horst Possegger, Thomas Mauthner, Peter M. Roth, and Horst Bischof
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
Columbus (Ohio), June 2014
Accepted for poster presentation

Abstract: Robust multi-object tracking-by-detection requires the correct assignment of noisy detection results to object trajectories. We address this problem by proposing an online approach based on the observation that object detectors primarily fail if objects are significantly occluded. In contrast to most existing work, we only rely on geometric information to efficiently overcome detection failures.

In particular, we exploit the spatio-temporal evolution of occlusion regions, detector reliability, and target motion prediction to robustly handle missed detections. In combination with a conservative association scheme for visible objects, this allows for real-time tracking of multiple objects from a single static camera, even in complex scenarios. Our evaluations on publicly available multi-object tracking benchmark datasets demonstrate favorable performance compared to the state-of-the-art in online and offline multi-object tracking.

A novel method for the analysis of sequential actions in team handball

Paul Rudelsdorfer, Norbert Schrapf, **Horst Possegger**, Thomas Mauthner, Horst Bischof, and Markus Tilp
International Journal of Computer Science in Sport (IJCSS), 13(1), pages 69–84, 2014

Abstract: Performance in team sports crucially depends on the knowledge about the own and the opponents strengths and weaknesses. Since the analysis of single actions only provides restricted information on the game process, the analysis of sequential actions is from great importance to understand team tactics. In this paper, we introduce a novel method to analyze tactical behavior in team sports based on action sequences of positional data which are subsequently analyzed with artificial neural networks.

We present custom-made software which allows annotating single actions with accurate manual position information. The process of building action sequences with the notational information of single actions in team handball is described step-by-step and the accuracy of the position determination is evaluated. The evaluation revealed a mean error of 0.16 m ($\pm$0.17 m) for field positions on a handball field. Inter- and intra-rater reliability for identical camera setups are excellent (ICC = 0.92 and 0.95, respectively). However, tests revealed that position accuracy is depending on camera setup (ICC = 0.36).

The results of the study demonstrate the applicability of the described method to gain action sequence data with accurate position information. The combination with neural networks gives an alternative approach to T-patterns for the analysis of sport games.

2015

In Defense of Color-based Model-free Tracking

Horst Possegger, Thomas Mauthner, and Horst Bischof
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
Boston (Massachusetts), June 2015
Accepted for poster presentation

Abstract: In this paper, we address the problem of model-free online object tracking based on color representations. According to the findings of recent benchmark evaluations, such trackers often tend to drift towards regions which exhibit a similar appearance compared to the object of interest. To overcome this limitation, we propose an efficient discriminative object model which allows us to identify potentially distracting regions in advance. Furthermore, we exploit this knowledge to adapt the object representation beforehand so that distractors are suppressed and the risk of drifting is significantly reduced. We evaluate our approach on recent online tracking benchmark datasets demonstrating state-of-the-art results. In particular, our approach performs favorably both in terms of accuracy and robustness compared to recent tracking algorithms. Moreover, the proposed approach allows for an efficient implementation to enable online object tracking in real-time.

Encoding based Saliency Detection for Videos and Images

Thomas Mauthner, **Horst Possegger**, Georg Waltner, and Horst Bischof
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
Boston (Massachusetts), June 2015
Accepted for poster presentation

Abstract: We present a novel video saliency detection method to support human activity recognition and weakly supervised training of activity detection algorithms. Recent research has emphasized the need for analyzing salient information in videos to minimize dataset bias or to supervise weakly labeled training of activity detectors. In contrast to previous methods we do not rely on training information given by either eye-gaze or annotation data, but propose a fully unsupervised algorithm to find salient regions within videos. In general, we enforce the Gestalt principle of figure-ground segregation for both appearance and motion cues. We introduce an encoding approach that allows for efficient computation of saliency by approximating joint feature distributions. We evaluate our approach on several datasets, including challenging scenarios with cluttered background and camera motion, as well as salient object detection in images. Overall, we demonstrate favorable performance compared to state-of-the-art methods in estimating both ground-truth eye-gaze and activity annotations.

Both authors contributed equally.

2016

Grid Loss: Detecting Occluded Faces

Michael Opitz, Georg Waltner, Georg Poier, **Horst Possegger**, and Horst Bischof
In *Proceedings of the European Conference on Computer Vision (ECCV)*
Amsterdam (Netherlands), October 2016
Accepted for poster presentation

Abstract: Detection of partially occluded objects is a challenging computer vision problem. Standard Convolutional Neural Network (CNN) detectors fail if parts of the detection window are occluded, since not every sub-part of the window is discriminative on its own. To address this issue, we propose a novel loss layer for CNNs, named grid loss, which minimizes the error rate on sub-blocks of a convolution layer independently rather than over the whole feature map. This results in parts being more discriminative on their own, enabling the detector to recover if the detection window is partially occluded. By mapping our loss layer back to a regular fully connected layer, no additional computational cost is incurred at runtime compared to standard CNNs. We demonstrate our method for face detection on several public face detection benchmarks and show that our method outperforms regular CNNs, is suitable for realtime applications and achieves state-of-the-art performance.

Efficient Model Averaging for Deep Neural Networks

Michael Opitz, **Horst Possegger**, and Horst Bischof
In *Proceedings of the Asian Conference on Computer Vision (ACCV)*
Taipei (Taiwan), November 2016
Accepted for poster presentation

Abstract: Large neural networks trained on small datasets are increasingly prone to overfitting. Traditional machine learning methods can reduce overfitting by employing bagging or boosting to train several diverse models. For large neural networks, however, this is prohibitively expensive. To address this issue, we propose a method to leverage the benefits of ensembles without explicitly training several expensive neural network models. In contrast to Dropout, to encourage diversity of our sub-networks, we propose to maximize diversity of individual networks with a loss function: DivLoss. We demonstrate the effectiveness of DivLoss on the challenging CIFAR datasets.

2017

Pedestrian Detection in RGB-D Images from an Elevated Viewpoint

Christian Ertler, **Horst Possegger**, Michael Opitz, and Horst Bischof
In *Proceedings of the Computer Vision Winter Workshop (CVWW)*
Retz (Austria), February 2017
Accepted for oral presentation

Abstract: We propose an extension to the state-of-the-art Faster R-CNN detection model for multi-modal pedestrian detection from RGB-D images. The proposed architectures address this problem by fusing convolutional neural network (CNN) representations. We elaborate two architectures, which primarily differ in the position of the fusion inside the model, and further compare several static and parametrized fusion layers. Moreover, we show how recent advances in the area of non-maximum suppression (NMS) can improve the detection results of our models and make them more robust in applications with varying pedestrian densities. Our models are trained and evaluated on a custom dataset comprising images of crosswalk scenes taken from an elevated viewpoint. This viewpoint results in uncommon and highly variable poses of pedestrians, demanding powerful detection models.

BIER - Boosting Independent Embeddings Robustly

Michael Opitz, Georg Waltner, **Horst Possegger**, and Horst Bischof
In *Proceedings of the International Conference on Computer Vision (ICCV)*
Venice (Italy), October 2017
Accepted for oral presentation

Abstract: Learning similarity functions between image pairs with deep neural networks yields highly correlated activations of large embeddings. In this work, we show how to improve the robustness of embeddings by exploiting independence in ensembles. We divide the last embedding layer of a deep network into an embedding ensemble and formulate training this ensemble as an online gradient boosting problem. Each learner receives a reweighted training sample from the previous learners. This leverages large embedding sizes more effectively by significantly reducing correlation of the embedding and consequently increases retrieval accuracy of the embedding. Our method does not introduce any additional parameters and works with any differentiable loss function. We evaluate our metric learning method on image retrieval tasks and show that it improves over state-of-the-art methods on the CUB-200-2011, Cars-196, Stanford Online Products, In-Shop Clothes Retrieval and VehicleID datasets by a significant margin.

2018

Spatiotemporal Saliency Estimation by Spectral Foreground Detection

Çağlar Aytekin, **Horst Possegger**, Thomas Mauthner, Serkan Kiranyaz, Horst Bischof, and Moncef Gabbouj

IEEE Transactions on Multimedia (TMM), 20(1), pages 82–95, 2018

Abstract: We present a novel approach for spatiotemporal saliency detection by optimizing a unified criterion of color contrast, motion contrast, appearance and background cues. To this end, we first abstract the video by temporal superpixels. Second, we propose a novel graph structure exploiting the saliency cues to assign the edge weights. The salient segments are then extracted by applying a spectral foreground detection method, Quantum Cuts, on this graph. We evaluate our approach on several public datasets for video saliency and activity localization to demonstrate the favorable performance of the proposed *Video Quantum Cuts* (VQCUT) compared to the state-of-the-art.

B.2 Visual Object Tracking Challenges

We participated with our prototype implementations at several tracking challenges organized by the Visual Object Tracking (VOT) challenge committee. These challenges allow to compare short-term single object trackers which do not apply pre-learned appearance models, *i.e.* as our approach presented in Chapter 3. In order to be listed as a co-author of the joint result paper, the submitted approach had to outperform a baseline performance specified by the organization committee for each challenge and the results must be reproducible. All our submissions outperformed the required baseline and thus, led to the following co-authored publications, listed in chronological order.

2014

The Visual Object Tracking VOT2014 Challenge Results

Matej Kristan, Roman Pflugfelder, Aleš Leonardis, Jiří Matas, Luka Čehovin, Georg Nebehay, Tomáš Vojíř, Gustavo Fernández, Alan Lukežič, Aleksandar Dimitriev, Alfredo Petrosino, Amir Saffari, Bo Li, Bohyung Han, Cherkeng Heng, Christophe Garcia, Dominik Pangeršič, Gustav Häger, Fahad Shahbaz Khan, Franci Oven, **Horst Possegger**, Horst Bischof, Hyeonseob Nam, Jianke Zhu, JiJia Li, Jin Young Choi, Jin-Woo Choi, João F. Henriques, Joost van de Weijer, Jorge Batista, Karel Lebeda, Kristoffer Öfjäll, Kwang Moo Yi, Lei Quin, Longyin Wen, Mario Edoardo Maresca, Martin Danelljan, Michael Felsberg, Ming-Ming Cheng, Philip Torr, Quingming Huang, Richard Bowden, Sam Hare, Samantha YueYing Lim, Seunghoon Hong, Shengcai Liao, Simon Hadfield, Stan Z. Li, Stefan Duffner, Stuart Golodetz, Thomas Mauthner, Vibhav Vineet, Weiyao Lin, Yang Li, Yuankai Qui, Zhen Lei, and Zhiheng Niu

In *Proceedings of the Workshop on the Visual Object Tracking Challenge (VOT), in conjunction with the European Conference on Computer Vision (ECCV)*
September 2014, Zürich (Switzerland)
Participated with the *Appearance-Based Shape Filter (ABS)*

2015

The Visual Object Tracking VOT2015 Challenge Results
Matej Kristan, Jiří Matas, Aleš Leonardis, Michael Felsberg, Luka Čehovin, Gustavo Fernández, Tomáš Vojíř, Gustav Häger, Georg Nebehay, Roman Pflugfelder, Abhinav Gupta, Adel Bibi, Alan Lukežič, Alvaro Garcia-Martin, Alfredo Petrosino, Amir Saffari, Andrés Solís Montero, Anton Varfolomieiev, Atilla Baskurt, Baojun Zhao, Bernard Ghanem, Brais Martinez, Byeong Ju Lee, Bohyung Han, Chaohui Wang, Christophe Garcia, Chunyuan Zhang, Cordelia Schmid, Dacheng Tao, Daijin Kim, Dafei Huang, Danil Prokhorov, Dawei Du, Dit-Yan Yeung, Eraldo Ribeiro, Fahad Shahbaz Khan, Fatih Porikli, Filiz Bunyak, Gao Zhu, Guna Seetharaman, Hilke Kieritz, Hing Tuen Yau, Hongdong Li, Honggang Qi, Horst Bischof, **Horst Possegger**, Hyemin Lee, Hyeonseob Nam, Ivan Bogun, Jae-chan Jeong, Jae-il Cho, Jae-Yeong Lee, Jianke Zhu, Jianping Shi, Jiatong Li, Jiaya Jia, Jiayi Feng, Jin Gao, Jin Young Choi, Ji-Wan Kim, Jochen Lang, Jose M. Martinez, Jongwon Choi, Junliang Xing, Kai Xue, Kannappan Palaniappan, Karel Lebeda, Karteek Alahari, Ke Gao, Kimin Yun, Kin Hong Wong, Lei Luo, Liang Ma, Lipeng Ke, Longyin Wen, Luca Bertinetto, Mahdieh Pootschi, Mario Maresca, Martin Danelljan, Mei Wen, Mengdan Zhang, Michael Arens, Michel Valstar, Ming Tang, Ming-Ching Chang, Muhammad Haris Khan, Nana Fan, Naiyan Wang, Ondrej Miksik, Philip Torr, Qiang Wang, Rafael Martin-Nieto, Rengarajan Pelapur, Richard Bowden, Robert Laganière, Salma Moujtahid, Sam Hare, Simon Hadfield, Siwei Lyu, Siyi Li, Song-Chun Zhu, Stefan Becker, Stefan Duffner, Stephen L Hicks, Stuart Golodetz, Sunglok Choi, Tianfu Wu, Thomas Mauthner, Tony Pridmore, Weiming Hu, Wolfgang Hübner, Xiaomeng Wang, Xin Li, Xinchu Shi, Xu Zhao, Xue Mei, Yao Shizeng, Yang Hua, Yang Li, Yang Lu, Yuezun Li, Zhaoyun Chen, Zehua Huang, Zhe Chen, Zhe Zhang, Zhenyu He, and Zhibin Hong
In *Proceedings of the Workshop on the Visual Object Tracking Challenge (VOT), in conjunction with the International Conference on Computer Vision (ICCV)*
December 2015, Santiago de Chile (Chile)
Participated with the *Distractor Aware Tracker (DAT)*

2016

The Visual Object Tracking VOT2016 Challenge Results
Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomáš Vojiř, Gustav Häger, Alan Lukežič, Gustavo Fernández, Ab-

hinav Gupta, Alfredo Petrosino, Alireza Memarmoghadam, Alvaro Garcia-Martin, Andrés Solíss Montero, Andrea Vedaldi, Andreas Robinson, Andy J. Ma, Anton Varfolomieiev, Aydin Alatan, Aykut Erdem, Bernard Ghanem, Bin Liu, Bohyung Han, Brais Martinez, Chang-Ming Chang, Changsheng Xu, Chong Sun, Chong Sun, Daijin Kim, Dapeng Chen, Dawei Du, Dawei Du, Deepak Mishra, Dit-Yan Yeung, Erhan Gündoğdu, Erkut Erdem, Fahad Khan, Fahad Shahbaz Khan, Fatih Porikli, Fei Zhao, Filiz Bunyak, Francesco Battistone, Gao Zhu, Giorgio Roffo, Gorthi R. K. Sai Subrahmanyam, Guilherme Bastos, Guna Seetharaman, Henry Medeiros, Hongdong Li, Honggang Qi, Horst Bischof, **Horst Possegger**, Huchuan Lu, Huchuan Lu, Hyemin Lee, Hyeonseob Nam, Hyung Jin Chang, Isabela Drummond, Jack Valmadre, Jae-chan Jeong, Jae-il Cho, Jae-Yeong Lee, Jianke Zhu, Jiayi Feng, Jin Gao, Jin Young Choi, Jingjing Xiao, Ji-Wan Kim, Jiyeoup Jeong, João F. Henriques, Jochen Lang, Jongwon Choi, Jose M. Martinez, Junliang Xing, Junyu Gao, Kannappan Palaniappan, Karel Lebeda, Ke Gao, Krystian Mikolajczyk, Lei Qin, Lijun Wang, Lijun Wang, Longyin Wen, Longyin Wen, Luca Bertinetto, Madan kumar Rapuru, Mahdieh Poostchi, Mario Maresca, Martin Danelljan, Matthias Mueller, Mengdan Zhang, Michael Arens, Michel Valstar, Ming Tang, Mooyeol Baek, Muhammad Haris Khan, Naiyan Wang, Nana Fan, Noor Al-Shakarji, Ondrej Miksik, Osman Akin, Payman Moallem, Pedro Senna, Philip H. S. Torr, Pong C. Yuen, Qingming Huang, Qingming Huang, Rafael Martin-Nieto, Rengarajan Pelapur, Richard Bowden, Robert Laganière, Rustam Stolkin, Ryan Walsh, Sebastian B. Krah, Shengkun Li, Shengping Zhang, Shizeng Yao, Simon Hadfield, Simone Melzi, Siwei Lyu, Siwei Lyu, Siyi Li, Stefan Becker, Stuart Golodetz, Sumithra Kakanuru, Sunglok Choi, Tao Hu, Thomas Mauthner, Tianzhu Zhang, Tony Pridmore, Vincenzo Santopietro, Weiming Hu, Wenbo Li, Wolfgang Hübner, Xiangyuan Lan, Xiaomeng Wang, Xin Li, Yang Li, Yiannis Demiris, Yifan Wang, Yuankai Qi, Zejian Yuan, Zexiong Cai, Zhan Xu, Zhenyu He, and Zhizhen Chi
In *Proceedings of the Workshop on the Visual Object Tracking Challenge (VOT), in conjunction with the European Conference on Computer Vision (ECCV)*
October 2016, Amsterdam (Netherlands)
Participated with the *Distractor Aware Tracker (DAT)*

The Thermal Infrared Visual Object Tracking VOT-TIR2016 Challenge Results

Michael Felsberg, Matej Kristan, Jiří Matas, Aleš Leonardis, Roman Pflugfelder, Gustav Häger, Amanda Berg, Abdelrahman Eldesokey, Jörgen Ahlberg, Luka Čehovin, Tomáš Vojiř, Alan Lukežič, Gustavo Fernández, Alfredo Petrosino, Alvaro Garcia-Martin, Andrés Solís Montero, Anton Varfolomieiev, Aykut Erdem, Bohyung Han, Chang-Ming Chang, Dawei Du, Erkut Erdem, Fahad Shahbaz

Khan, Fatih Porikli, Fei Zhao, Filiz Bunyak, Francesco Battistone, Gao Zhu, Guna Seetharaman, Hongdong Li, Honggang Qi, Horst Bischof, **Horst Possegger**, Hyeonseob Nam, Jack Valmadre, Jianke Zhu, Jiayi Feng, Jochen Lang, Jose M. Martinez, Kannappan Palaniappan, Karel Lebeda, Ke Gao, Krystian Mikolajczyk, Longyin Wen, Luca Bertinetto, Mahdieh Poostchi, Mario Maresca, Martin Danelljan, Michael Arens, Ming Tang, Mooyeol Baek, Nana Fan, Noor Al-Shakarji, Ondrej Miksik, Osman Akin, Philip H. S. Torr, Qingming Huang, Rafael Martin-Nieto, Rengarajan Pelapur, Richard Bowden, Robert Laganière, Sebastian B. Krah, Shengkun Li, Shizeng Yao, Simon Hadfield, Siwei Lyu, Stefan Becker, Stuart Golodetz, Tao Hu, Thomas Mauthner, Vincenzo Santopietro, Wenbo Li, Wolfgang Hübner, Xin Li, Yang Li, Zhan Xu, and Zhenyu He
In *Proceedings of the Workshop on the Visual Object Tracking Challenge (VOT), in conjunction with the European Conference on Computer Vision (ECCV)*
October 2016, Amsterdam (Netherlands)
Participated with the *Distractor Aware Tracker (DAT)*, reduced to a monochrome model (instead of exploiting the joint color distribution)

C

Detailed Evaluation Results

...

— Gordon Freeman (Hλlf-Life)

Contents

C.1 Single Object Tracking Results

In the following, we list the detailed per-sequence results of our distractor-aware tracking approach (with and without scale, *i.e.* DAT+s and DAT) and its distractor-agnostic baseline (noDAT). On the VOT benchmarks, we additionally compare our approaches against ACT [93], a recent color-based state-of-the-art approach. On the OTB dataset, we compare against CXT [104], a context-aware tracking approach. For a detailed discussion of the tracking results, used datasets and evaluation protocols refer to Chapter 5.

Table C.1 lists the detailed results on the VOT'13 [238] benchmark for both experimental stacks, *i.e. baseline* and *region noise*. Tables C.2 and C.3 list the results on the VOT'14 [239] benchmark experiments *baseline* and *region noise*, respectively. Tables C.4 and C.5 list the results on the VOT'16 [241] benchmark experiments *baseline* and *unsupervised*, respectively. Finally, Table C.6 lists the results on all color sequences of the OTB-100 [449] dataset.

Table C.1: Per-sequence results on the VOT'13 [238] benchmark. Best, second best and third best accuracy results have been highlighted for each sequence. Robustness scores have been **boldfaced** for sequences where the tracker did not drift and thus, no re-initialization was necessary throughout this sequence. For each sequence, we additionally list its length in numbers of frames, denoted #F.

(a) Experiment *baseline*.

Sequence	#F	DAT+s Acc.↑	DAT+s Rob.↓	DAT Acc.↑	DAT Rob.↓	noDAT Acc.↑	noDAT Rob.↓	ACT [93] Acc.↑	ACT [93] Rob.↓
bicycle	271	0.40	**0.00**	0.45	**0.00**	0.45	**0.00**	0.46	1.00
bolt	350	0.62	**0.00**	0.66	**0.00**	0.66	**0.00**	0.79	1.00
car	374	0.54	**0.00**	0.46	**0.00**	0.46	**0.00**	0.43	1.00
cup	303	0.78	**0.00**	0.73	**0.00**	0.74	**0.00**	0.76	**0.00**
david	770	0.47	**0.00**	0.64	**0.00**	0.64	**0.00**	0.68	**0.00**
diving	231	0.39	**0.00**	0.34	1.00	0.35	2.00	0.41	1.00
face	415	0.54	**0.00**	0.60	**0.00**	0.60	**0.00**	0.85	**0.00**
gymnastics	207	0.61	**0.00**	0.57	**0.00**	0.56	**0.00**	0.55	2.00
hand	244	0.53	**0.00**	0.63	1.00	0.63	1.00	0.50	3.00
iceskater	500	0.49	**0.00**	0.64	**0.00**	0.64	**0.00**	0.48	1.00
juice	404	0.82	**0.00**	0.61	**0.00**	0.61	**0.00**	0.65	**0.00**
jump	228	0.32	**0.00**	0.44	**0.00**	0.44	**0.00**	0.58	**0.00**
singer	351	0.62	**0.00**	0.40	**0.00**	0.43	1.00	0.37	**0.00**
sunshade	172	0.59	**0.00**	0.60	**0.00**	0.59	**0.00**	0.64	**0.00**
torus	264	0.72	**0.00**	0.76	**0.00**	0.76	**0.00**	0.78	**0.00**
woman	597	0.55	**0.00**	0.66	**0.00**	0.66	**0.00**	0.71	3.00
Total		0.56	**0.00**	0.59	0.08	0.59	0.19	0.62	0.82

(b) Experiment *region noise*.

Sequence	#F	DAT+s Acc.↑	DAT+s Rob.↓	DAT Acc.↑	DAT Rob.↓	noDAT Acc.↑	noDAT Rob.↓	ACT [93] Acc.↑	ACT [93] Rob.↓
bicycle	271	0.43	0.07	0.43	0.13	0.44	0.33	0.46	1.00
bolt	350	0.61	**0.00**	0.62	**0.00**	0.63	**0.00**	0.64	0.80
car	374	0.54	0.07	0.49	**0.00**	0.49	**0.00**	0.43	0.87
cup	303	0.78	**0.00**	0.74	**0.00**	0.72	**0.00**	0.70	**0.00**
david	770	0.46	**0.00**	0.64	**0.00**	0.64	0.07	0.65	**0.00**
diving	231	0.38	0.33	0.32	1.20	0.33	1.13	0.33	1.93
face	415	0.54	**0.00**	0.59	**0.00**	0.60	**0.00**	0.73	0.67
gymnastics	207	0.58	**0.00**	0.58	**0.00**	0.53	**0.00**	0.42	2.33
hand	244	0.58	0.93	0.60	0.60	0.58	0.80	0.47	4.40
iceskater	500	0.49	**0.00**	0.64	**0.00**	0.64	**0.00**	0.42	0.40
juice	404	0.82	**0.00**	0.63	**0.00**	0.62	**0.00**	0.62	**0.00**
jump	228	0.33	**0.00**	0.44	**0.00**	0.43	**0.00**	0.55	**0.00**
singer	351	0.63	0.07	0.44	0.60	0.46	1.13	0.39	**0.00**
sunshade	172	0.59	**0.00**	0.59	**0.00**	0.58	**0.00**	0.67	0.93
torus	264	0.71	**0.00**	0.73	**0.00**	0.74	**0.00**	0.70	0.20
woman	597	0.48	**0.00**	0.65	**0.00**	0.65	0.33	0.65	2.13
Total		0.55	0.07	0.59	0.12	0.59	0.21	0.57	0.85

Table C.2: Per-sequence results on the VOT'14 [239] benchmark, experiment *baseline*. Best, second best and third best accuracy results have been highlighted for each sequence. Robustness scores have been **boldfaced** for sequences where the tracker did not drift and thus, no re-initialization was necessary throughout this sequence. For each sequence, we additionally list its length in numbers of frames, denoted #F.

Sequence	**#F**	**DAT+s**		**DAT**		**noDAT**		**ACT [93]**	
		Acc.↑	**Rob.↓**	**Acc.↑**	**Rob.↓**	**Acc.↑**	**Rob.↓**	**Acc.↑**	**Rob.↓**
ball	602	0.72	**0.00**	0.66	**0.00**	0.66	**0.00**	0.41	**0.00**
basketball	725	0.64	**0.00**	0.68	1.00	0.68	1.00	0.66	**0.00**
bicycle	271	0.43	**0.00**	0.48	**0.00**	0.47	**0.00**	0.45	1.00
bolt	350	0.51	**0.00**	0.47	**0.00**	0.47	**0.00**	0.54	1.00
car	252	0.60	**0.00**	0.38	**0.00**	0.42	1.00	0.52	1.00
david	770	0.41	**0.00**	0.63	**0.00**	0.63	**0.00**	0.72	**0.00**
diving	219	0.29	**0.00**	0.36	2.00	0.37	**0.00**	0.20	4.00
drunk	1210	0.47	1.00	0.46	1.00	0.44	**0.00**	0.46	**0.00**
fernando	292	0.37	3.00	0.39	2.00	0.42	4.00	0.43	1.00
fish1	436	0.35	**0.00**	0.39	**0.00**	0.38	**0.00**	0.43	**0.00**
fish2	310	0.48	1.00	0.43	1.00	0.44	2.00	0.31	5.00
gymnastics	207	0.61	**0.00**	0.61	**0.00**	0.58	**0.00**	0.51	2.00
hand1	244	0.61	**0.00**	0.62	1.00	0.62	1.00	0.40	5.00
hand2	267	0.51	3.00	0.53	2.00	0.55	1.00	0.38	8.00
jogging	307	0.67	1.00	0.72	1.00	0.73	2.00	0.70	1.00
motocross	164	0.50	3.00	0.43	4.00	0.46	3.00	0.47	3.00
polarbear	371	0.57	**0.00**	0.55	**0.00**	0.55	**0.00**	0.51	**0.00**
skating	400	0.39	10.00	0.46	9.00	0.43	13.00	0.50	**0.00**
sphere	201	0.81	**0.00**	0.72	**0.00**	0.72	**0.00**	0.72	**0.00**
sunshade	172	0.59	**0.00**	0.61	**0.00**	0.61	**0.00**	0.78	**0.00**
surfing	282	0.64	**0.00**	0.64	**0.00**	0.64	**0.00**	0.82	**0.00**
torus	264	0.73	**0.00**	0.76	**0.00**	0.76	**0.00**	0.79	**0.00**
trellis	569	0.47	**0.00**	0.52	**0.00**	0.50	**0.00**	0.58	2.00
tunnel	731	0.33	3.00	0.27	**0.00**	0.38	3.00	0.31	**0.00**
woman	597	0.41	**0.00**	0.69	1.00	0.69	1.00	0.66	3.00
Total		0.51	1.00	0.53	0.90	0.54	1.21	0.53	1.09

Table C.3: Per-sequence results on the VOT'14 [239] benchmark, experiment *region noise*. Best, second best and third best accuracy results have been highlighted for each sequence. Robustness scores have been **boldfaced** for sequences where the tracker did not drift and thus, no re-initialization was necessary throughout this sequence. For each sequence, we additionally list its length in numbers of frames, denoted #F.

Sequence	**#F**	**DAT+s**		**DAT**		**noDAT**		**ACT [93]**	
		Acc.↑	**Rob.↓**	**Acc.↑**	**Rob.↓**	**Acc.↑**	**Rob.↓**	**Acc.↑**	**Rob.↓**
ball	602	0.70	**0.00**	0.64	**0.00**	0.64	**0.00**	0.39	0.73
basketball	725	0.63	0.13	0.66	1.00	0.67	1.00	0.65	0.13
bicycle	271	0.46	**0.00**	0.46	0.13	0.45	**0.00**	0.43	0.87

Table continued on next page.

Table C.3: SOT on VOT'14, experiment *region noise* – *Continued from previous page.*

Sequence	#F	DAT+s Acc.↑	DAT+s Rob.↓	DAT Acc.↑	DAT Rob.↓	noDAT Acc.↑	noDAT Rob.↓	ACT [93] Acc.↑	ACT [93] Rob.↓
bolt	350	0.50	0.13	0.48	0.13	0.48	0.40	0.52	0.73
car	252	0.59	**0.00**	0.38	**0.00**	0.39	0.53	0.41	0.07
david	770	0.47	**0.00**	0.63	**0.00**	0.63	0.20	0.67	**0.00**
diving	219	0.30	**0.00**	0.41	1.00	0.43	1.07	0.21	4.33
drunk	1210	0.47	**0.00**	0.44	0.33	0.44	**0.00**	0.44	**0.00**
fernando	292	0.34	3.07	0.38	2.13	0.38	2.40	0.37	1.67
fish1	436	0.37	0.27	0.39	0.07	0.40	0.33	0.32	6.53
fish2	310	0.46	1.47	0.43	1.87	0.42	1.73	0.29	4.80
gymnastics	207	0.60	**0.00**	0.59	**0.00**	0.56	0.33	0.44	2.80
hand1	244	0.60	1.07	0.58	0.87	0.61	0.73	0.46	4.73
hand2	267	0.52	1.80	0.52	1.93	0.54	1.13	0.37	9.53
jogging	307	0.67	1.47	0.67	1.27	0.67	1.73	0.65	1.00
motocross	164	0.45	2.80	0.40	3.80	0.44	2.53	0.39	2.53
polarbear	371	0.57	**0.00**	0.55	**0.00**	0.53	**0.00**	0.48	**0.00**
skating	400	0.33	7.07	0.43	9.60	0.41	12.73	0.46	**0.00**
sphere	201	0.78	**0.00**	0.72	**0.00**	0.72	**0.00**	0.70	**0.00**
sunshade	172	0.59	**0.00**	0.60	**0.00**	0.59	**0.00**	0.72	0.07
surfing	282	0.65	**0.00**	0.67	**0.00**	0.67	**0.00**	0.73	**0.00**
torus	264	0.72	**0.00**	0.74	**0.00**	0.75	**0.00**	0.72	0.33
trellis	569	0.47	**0.00**	0.50	**0.00**	0.51	**0.00**	0.56	1.27
tunnel	731	0.37	3.33	0.32	2.40	0.36	3.60	0.31	**0.00**
woman	597	0.48	**0.00**	0.67	**0.00**	0.68	0.73	0.63	2.00
Total		0.51	0.83	0.53	0.98	0.53	1.22	0.49	1.35

Table C.4: Per-sequence results on the VOT'16 [241] benchmark, experiment *baseline*. Best, second best and third best accuracy results have been highlighted for each sequence. Robustness scores have been **boldfaced** for sequences where the tracker did not drift and thus, no re-initialization was necessary throughout this sequence. For each sequence, we additionally list its length in numbers of frames, denoted #F.

Sequence	#F	DAT+s Acc.↑	DAT+s Rob.↓	DAT Acc.↑	DAT Rob.↓	noDAT Acc.↑	noDAT Rob.↓	ACT [93] Acc.↑	ACT [93] Rob.↓
bag	196	0.49	**0.00**	0.48	**0.00**	0.48	**0.00**	0.40	**0.00**
ball1	105	0.73	**0.00**	0.77	**0.00**	0.78	**0.00**	0.73	1.00
ball2	41	0.50	1.00	0.50	1.00	0.52	1.00	0.01	4.00
basketball	725	0.63	**0.00**	0.65	1.00	0.64	1.00	0.54	1.00
birds1	339	0.22	2.00	0.45	6.00	0.44	7.00	0.48	3.00
birds2	539	0.37	1.00	0.43	1.00	0.43	1.00	0.22	**0.00**
blanket	225	0.66	**0.00**	0.56	**0.00**	0.55	**0.00**	0.58	2.00
bmx	76	0.29	**0.00**	0.29	**0.00**	0.29	**0.00**	0.21	**0.00**

Table continued on next page.

Table C.4: SOT on VOT'16, experiment *baseline* – *Continued from previous page.*

Sequence	**#F**	**DAT+s**		**DAT**		**noDAT**		**ACT** [93]	
		Acc.$^{\uparrow}$	**Rob.**$^{\downarrow}$	**Acc.**$^{\uparrow}$	**Rob.**$^{\downarrow}$	**Acc.**$^{\uparrow}$	**Rob.**$^{\downarrow}$	**Acc.**$^{\uparrow}$	**Rob.**$^{\downarrow}$
bolt1	350	0.44	**0.00**	0.54	1.00	0.45	2.00	0.46	**0.00**
bolt2	293	0.57	**0.00**	0.53	**0.00**	0.56	1.00	0.50	**0.00**
book	175	0.47	1.00	0.36	1.00	0.36	1.00	0.35	7.00
butterfly	151	0.46	**0.00**	0.47	**0.00**	0.50	**0.00**	0.39	1.00
car1	742	0.34	3.00	0.47	2.00	0.42	6.00	0.67	3.00
car2	393	0.32	2.00	0.28	5.00	0.26	3.00	0.73	**0.00**
crossing	131	0.46	1.00	0.44	1.00	0.44	1.00	0.44	1.00
dinosaur	326	0.45	1.00	0.53	**0.00**	0.57	**0.00**	0.47	1.07
fernando	292	0.37	2.00	0.36	2.00	0.37	3.00	0.29	1.00
fish1	366	0.46	2.00	0.45	2.00	0.45	2.00	0.32	6.07
fish2	310	0.47	1.00	0.42	2.00	0.39	3.00	0.22	7.00
fish3	519	0.46	**0.00**	0.57	**0.00**	0.58	**0.00**	0.47	**0.00**
fish4	682	0.36	2.00	0.44	1.00	0.42	1.00	0.25	1.00
girl	1500	0.66	1.00	0.64	1.00	0.64	**0.00**	0.47	2.00
glove	120	0.55	2.00	0.55	2.00	0.57	2.00	0.44	4.00
godfather	366	0.50	1.00	0.49	2.00	0.49	2.00	0.44	**0.00**
graduate	844	0.33	8.00	0.32	8.00	0.33	9.00	0.34	5.93
gymnastics1	567	0.57	**0.00**	0.40	1.00	0.54	1.00	0.40	6.07
gymnastics2	240	0.54	1.00	0.53	2.00	0.50	2.00	0.56	3.00
gymnastics3	118	0.43	3.00	0.32	1.00	0.16	3.00	0.26	2.00
gymnastics4	465	0.51	2.00	0.52	2.00	0.53	1.00	0.41	3.00
hand	267	0.55	1.00	0.55	2.00	0.54	2.00	0.44	6.00
handball1	377	0.43	2.00	0.54	2.00	0.50	2.00	0.45	3.07
handball2	402	0.40	2.00	0.45	2.00	0.44	3.00	0.45	4.93
helicopter	708	0.55	**0.00**	0.47	1.00	0.47	1.00	0.35	**0.00**
iceskater1	661	0.52	**0.00**	0.53	1.00	0.53	1.00	0.40	3.00
iceskater2	707	0.59	2.00	0.54	1.00	0.52	2.00	0.47	4.00
leaves	63	0.49	**0.00**	0.45	**0.00**	0.45	**0.00**	0.31	2.00
marching	201	0.34	4.00	0.42	4.00	0.39	5.00	0.75	**0.00**
matrix	100	0.42	1.00	0.51	1.00	0.48	1.00	0.35	3.00
motocross1	164	0.45	4.00	0.35	2.00	0.35	2.00	0.36	2.00
motocross2	61	0.29	**0.00**	0.31	1.00	0.57	1.00	0.54	**0.00**
nature	999	0.48	3.00	0.47	3.00	0.56	2.00	0.33	4.00
octopus	291	0.31	**0.00**	0.30	**0.00**	0.30	**0.00**	0.31	**0.00**
pedestrian1	140	0.60	1.00	0.58	1.00	0.58	1.00	0.70	6.00
pedestrian2	713	0.22	**0.00**	0.22	**0.00**	0.22	**0.00**	0.54	3.00
rabbit	158	0.38	4.00	0.43	5.00	0.30	6.00	0.26	5.00
racing	156	0.21	**0.00**	0.32	**0.00**	0.41	1.00	0.32	**0.00**
road	558	0.48	**0.00**	0.52	1.00	0.59	1.00	0.56	**0.00**
shaking	365	0.27	1.00	0.58	6.00	0.59	5.00	0.54	**0.00**
sheep	251	0.31	**0.00**	0.34	1.00	0.35	1.00	0.48	**0.00**

Table continued on next page.

Table C.4: SOT on VOT'16, experiment *baseline* – *Continued from previous page.*

Sequence	**#F**	**DAT+s** Acc.↑	Rob.↓	**DAT** Acc.↑	Rob.↓	**noDAT** Acc.↑	Rob.↓	**ACT [93]** Acc.↑	Rob.↓
singer1	351	0.65	**0.00**	0.48	1.00	0.48	1.00	0.32	**0.00**
singer2	366	0.14	2.00	0.38	4.00	0.37	5.00	0.53	3.00
singer3	131	0.23	2.00	0.14	2.00	0.34	2.00	0.31	1.00
soccer1	392	0.50	9.00	0.43	8.00	0.47	9.00	0.44	1.00
soccer2	129	0.60	2.00	0.61	2.00	0.58	3.00	0.00	17.00
soldier	138	0.37	1.00	0.45	**0.00**	0.45	**0.00**	0.46	2.00
sphere	201	0.75	**0.00**	0.71	**0.00**	0.70	**0.00**	0.26	4.00
tiger	365	0.50	2.00	0.54	2.00	0.47	1.00	0.66	3.00
traffic	191	0.39	2.00	0.40	2.00	0.40	2.00	0.68	**0.00**
tunnel	312	0.25	1.00	0.54	1.00	0.51	2.00	0.43	**0.00**
wiper	341	0.16	7.00	0.21	6.00	0.22	8.00	0.66	**0.00**
Total		0.45	1.67	0.47	1.99	0.47	2.21	0.44	2.34

Table C.5: Per-sequence results on the VOT'16 [241] benchmark, experiment *unsupervised*. As there is no supervision, this experimental stack is only evaluated using average overlap (AO). Best, second best and third best results have been highlighted for each sequence. For each sequence, we additionally list its length in numbers of frames, denoted #F.

Sequence	**# F**	**DAT+s**	**DAT**	**noDAT**	**ACT [93]**
bag	196	0.49	0.48	0.48	0.40
ball1	105	0.73	0.76	0.78	0.37
ball2	41	0.06	0.06	0.06	0.03
basketball	725	0.63	0.59	0.58	0.02
birds1	339	0.22	0.04	0.06	0.40
birds2	539	0.31	0.35	0.35	0.22
blanket	225	0.66	0.57	0.55	0.16
bmx	76	0.32	0.32	0.32	0.25
bolt1	350	0.44	0.11	0.21	0.47
bolt2	293	0.56	0.53	0.27	0.50
book	175	0.31	0.20	0.20	0.18
butterfly	151	0.47	0.48	0.50	0.33
car1	742	0.20	0.25	0.01	0.53
car2	393	0.04	0.04	0.04	0.73
crossing	131	0.46	0.44	0.44	0.45
dinosaur	326	0.39	0.53	0.58	0.37
fernando	292	0.25	0.27	0.27	0.23
fish1	366	0.20	0.20	0.20	0.02
fish2	310	0.42	0.15	0.15	0.03
fish3	519	0.47	0.58	0.58	0.48
fish4	682	0.05	0.05	0.25	0.21

Table continued on next page.

Table C.5: SOT on VOT'16, experiment *unsupervised*.
Continued from previous page.

Sequence	**# F**	**DAT+s**	**DAT**	**noDAT**	**ACT [93]**
girl	1500	0.54	0.36	0.64	0.07
glove	120	0.12	0.12	0.12	0.07
godfather	366	0.40	0.43	0.26	0.44
graduate	844	0.21	0.20	0.18	0.24
gymnastics1	567	0.57	0.30	0.19	0.19
gymnastics2	240	0.44	0.41	0.42	0.27
gymnastics3	118	0.12	0.12	0.12	0.12
gymnastics4	465	0.43	0.44	0.44	0.28
hand	267	0.35	0.12	0.29	0.16
handball1	377	0.06	0.31	0.45	0.26
handball2	402	0.36	0.40	0.40	0.16
helicopter	708	0.55	0.35	0.35	0.36
iceskater1	661	0.53	0.19	0.19	0.18
iceskater2	707	0.38	0.41	0.05	0.25
leaves	63	0.49	0.45	0.45	0.01
marching	201	0.03	0.02	0.02	0.75
matrix	100	0.28	0.23	0.36	0.12
motocross1	164	0.09	0.09	0.09	0.08
motocross2	61	0.31	0.27	0.08	0.54
nature	999	0.11	0.10	0.10	0.11
octopus	291	0.32	0.32	0.31	0.32
pedestrian1	140	0.36	0.35	0.36	0.04
pedestrian2	713	0.22	0.22	0.22	0.12
rabbit	158	0.08	0.09	0.09	0.05
racing	156	0.22	0.34	0.08	0.35
road	558	0.48	0.56	0.04	0.56
shaking	365	0.03	0.03	0.03	0.54
sheep	251	0.31	0.04	0.04	0.49
singer1	351	0.66	0.18	0.17	0.34
singer2	366	0.08	0.10	0.09	0.07
singer3	131	0.15	0.15	0.14	0.15
soccer1	392	0.21	0.17	0.17	0.40
soccer2	129	0.10	0.10	0.09	0.03
soldier	138	0.09	0.44	0.44	0.20
sphere	201	0.75	0.71	0.70	0.18
tiger	365	0.38	0.38	0.47	0.63
traffic	191	0.25	0.24	0.24	0.68
tunnel	312	0.14	0.10	0.18	0.44
wiper	341	0.04	0.04	0.02	0.66
Total		0.33	0.28	0.27	0.28

Table C.6: Per-sequence results on all 76 color videos of the OTB-100 [449] dataset reporting the area under the success curve (AUC, average overlap) and the representative distance precision score (RDP, percentage of frames with center distance less than 20 pixels). Videos with multiple targets are reported as separate sequences, where the target identifier is listed as a post-fix, *i.e.* *Jogging* and *Skating2*.

Sequence	**#F**	**DAT+s**		**DAT**		**noDAT**		**CXT [104]**	
		AUC↑	**RDP**↑	**AUC**↑	**RDP**↑	**AUC**↑	**RDP**↑	**AUC**↑	**RDP**↑
Basketball	725	0.67	1.00	0.75	1.00	0.75	1.00	0.02	0.04
Biker	142	0.18	0.50	0.18	0.50	0.18	0.50	0.41	0.54
Bird1	408	0.24	0.42	0.22	0.31	0.23	0.33	0.03	0.03
Bird2	99	0.67	0.99	0.75	1.00	0.74	0.99	0.25	0.19
BlurBody	334	0.57	0.89	0.43	0.38	0.49	0.46	0.72	0.95
BlurCar1	742	0.38	0.43	0.09	0.05	0.01	0.01	0.24	0.34
BlurCar2	585	0.37	0.00	0.47	0.11	0.28	0.01	0.76	0.97
BlurCar3	357	0.16	0.11	0.52	0.57	0.53	0.57	0.60	1.00
BlurCar4	380	0.71	0.76	0.80	0.97	0.80	0.98	0.75	1.00
BlurFace	493	0.48	0.05	0.48	0.05	0.49	0.06	0.82	1.00
BlurOwl	631	0.80	0.99	0.79	1.00	0.80	1.00	0.26	0.98
Board	698	0.15	0.10	0.19	0.12	0.18	0.13	0.30	0.11
Bolt	350	0.59	0.97	0.64	0.96	0.64	0.97	0.02	0.03
Bolt2	293	0.43	0.69	0.44	0.67	0.44	0.66	0.01	0.02
Box	1161	0.10	0.04	0.46	0.54	0.05	0.05	0.31	0.34
Boy	602	0.71	1.00	0.76	1.00	0.76	1.00	0.54	0.94
Car24	3059	0.33	0.52	0.24	0.55	0.23	0.55	0.77	1.00
CarDark	393	0.04	0.11	0.04	0.11	0.03	0.11	0.56	0.73
CarScale	252	0.63	0.69	0.40	0.67	0.41	0.64	0.67	0.74
Coke	291	0.45	0.47	0.54	0.62	0.36	0.43	0.42	0.65
Couple	140	0.55	0.95	0.54	0.95	0.55	0.96	0.47	0.64
Crossing	120	0.57	1.00	0.61	1.00	0.61	1.00	0.36	0.63
Crowds	347	0.69	0.97	0.70	0.95	0.70	0.94	0.09	0.13
David	471	0.45	0.64	0.44	0.69	0.44	0.69	0.64	1.00
David3	252	0.49	0.22	0.68	0.70	0.68	0.73	0.12	0.15
Deer	71	0.17	0.21	0.17	0.21	0.07	0.06	0.69	1.00
Diving	215	0.37	0.69	0.32	0.49	0.28	0.45	0.19	0.19
Dog	127	0.56	1.00	0.37	1.00	0.37	1.00	0.64	1.00
Doll	3872	0.37	0.17	0.35	0.27	0.35	0.27	0.73	0.99
DragonBaby	113	0.63	0.87	0.60	0.81	0.60	0.83	0.35	0.58
FaceOcc1	892	0.38	0.13	0.43	0.19	0.43	0.20	0.63	0.34
Football1	74	0.67	1.00	0.68	1.00	0.60	0.89	0.75	1.00
Girl	500	0.46	0.77	0.58	0.94	0.49	0.81	0.55	0.77
Girl2	1500	0.58	0.79	0.57	0.78	0.69	0.91	0.18	0.18
Gym	767	0.46	0.63	0.47	0.85	0.47	0.84	0.45	0.75
Human2	1128	0.15	0.10	0.16	0.11	0.16	0.11	0.28	0.28

Table continued on next page.

Table C.6: SOT on OTB-100 – *Continued from previous page.*

Sequence	#F	DAT+s		DAT		noDAT		CXT [104]	
		AUC$\uparrow$	RDP$\uparrow$	AUC$\uparrow$	RDP$\uparrow$	AUC$\uparrow$	RDP$\uparrow$	AUC$\uparrow$	RDP$\uparrow$
Human3	1698	0.02	0.03	0.06	0.10	0.08	0.12	0.01	0.01
Human4	667	0.56	0.98	0.32	0.50	0.38	0.59	0.06	0.11
Human5	713	0.03	0.05	0.03	0.05	0.03	0.03	0.23	0.33
Human6	792	0.20	0.31	0.24	0.32	0.19	0.33	0.15	0.17
Human7	250	0.47	1.00	0.14	0.16	0.44	0.76	0.43	0.96
Human8	128	0.33	0.48	0.33	0.59	0.45	0.91	0.11	0.19
Human9	305	0.51	0.82	0.32	0.26	0.32	0.30	0.08	0.12
Ironman	166	0.09	0.13	0.02	0.03	0.02	0.03	0.05	0.04
Jogging.1	307	0.17	0.23	0.18	0.23	0.18	0.23	0.75	0.96
Jogging.2	307	0.12	0.17	0.14	0.20	0.72	0.98	0.13	0.16
Jump	122	0.07	0.06	0.06	0.05	0.06	0.05	0.06	0.07
KiteSurf	84	0.61	1.00	0.30	0.48	0.63	1.00	0.32	0.42
Lemming	1336	0.56	0.63	0.58	0.59	0.58	0.59	0.45	0.73
Liquor	1741	0.19	0.20	0.19	0.22	0.22	0.26	0.25	0.21
Man	134	0.21	0.61	0.34	0.69	0.24	0.49	0.84	0.99
Matrix	100	0.23	0.36	0.25	0.35	0.45	0.75	0.07	0.06
MotorRolling	164	0.10	0.08	0.10	0.07	0.10	0.06	0.14	0.04
MountainBike	228	0.39	0.56	0.11	0.12	0.10	0.11	0.22	0.28
Panda	1000	0.52	0.98	0.52	0.98	0.52	0.98	0.19	0.31
RedTeam	1918	0.42	1.00	0.49	1.00	0.49	1.00	0.39	0.65
Rubik	1997	0.62	0.73	0.48	0.39	0.48	0.38	0.36	0.23
Shaking	365	0.02	0.02	0.04	0.02	0.03	0.02	0.13	0.13
Singer1	351	0.65	0.96	0.25	0.16	0.18	0.16	0.49	0.97
Singer2	366	0.02	0.01	0.02	0.01	0.02	0.01	0.07	0.06
Skater2	435	0.25	0.31	0.16	0.15	0.16	0.15	0.41	0.34
Skating1	400	0.06	0.09	0.07	0.10	0.07	0.10	0.14	0.24
Skating2.1	473	0.39	0.26	0.37	0.27	0.38	0.26	0.13	0.16
Skating2.2	473	0.02	0.01	0.02	0.01	0.02	0.01	0.06	0.04
Skiing	81	0.53	1.00	0.51	1.00	0.50	1.00	0.09	0.15
Soccer	392	0.19	0.24	0.14	0.16	0.20	0.18	0.15	0.23
Subway	175	0.54	1.00	0.53	0.73	0.68	0.97	0.17	0.26
Surfer	376	0.56	0.95	0.41	0.99	0.31	0.66	0.72	1.00
Sylvester	1345	0.31	0.75	0.53	0.72	0.53	0.72	0.59	0.85
Tiger1	354	0.33	0.15	0.45	0.29	0.45	0.30	0.21	0.12
Tiger2	365	0.47	0.62	0.44	0.49	0.45	0.51	0.36	0.34
Trans	124	0.39	0.25	0.39	0.25	0.38	0.24	0.51	0.39
Trellis	569	0.57	0.80	0.57	0.82	0.57	0.81	0.65	0.97
Walking	412	0.65	1.00	0.54	1.00	0.54	0.99	0.17	0.24
Walking2	500	0.31	0.41	0.28	0.37	0.28	0.37	0.37	0.41
Woman	597	0.63	0.91	0.70	0.91	0.70	0.91	0.20	0.37
Total		0.39	0.54	0.37	0.50	0.39	0.52	0.35	0.47

C.2 Multiple Object Detection Results

In the following, we list the detailed evaluation results for pedestrian detection on the surveillance scenes we used for detection-based multiple object tracking. The results on the PETS'09 S2L1, S2L2 and S2L3 [136] sequences are summarized in Table C.7, C.8 and C.9, respectively. Table C.10 lists the detection results on the TownCentre [36] dataset. Note that for each detector, we list both the original and the refined results, *i.e.* after bounding box regression as detailed in Section 5.2.3.2. In addition to the results of publicly available state-of-the-art detectors, we also include widely used detections for each sequence, namely the ACF$^{\oplus}$ detections (for the PETS'09 S2L1, S2L2 and TownCentre sequences) provided by the MOT'15 committee [256], the DPM$_{v4}$$^{\oplus}$ detections (for all PETS'09 sequences) kindly provided by the authors of [192, 193], and the HOG detections distributed in combination with the TownCentre [36] dataset. For the deep learning meta-architectures F-RCNN [362], R-FCN [90], and SSD [280], we report the results from using different feature extractors, as indicated in the tables.

Table C.7: Evaluation of state-of-the-art pedestrian detectors on the PETS'09 S2L1 [136] dataset. The superscript $^{\oplus}$ indicates that the input images have been upsampled (to twice the size) in order to better match the object sizes used during training the detector model. Best, second best and third best results have been highlighted for each measure.

Detector	Training Data	AUC$^{\uparrow}$	GPU	FPS$^{\uparrow}$
ACF$^{\oplus}$ [109]	Caltech [108]	$0.80_{(+0.01)}$		2.88 ± 0.13
ACF [109]	Caltech [108]	$0.84_{(+0.02)}$		10.01 ± 0.95
ACF$^{\oplus}$ [109]	INRIA [91]	$0.92_{(+0.00)}$		8.11 ± 0.47
ACF [109]	INRIA [91]	$0.65_{(+0.19)}$		32.08 ± 1.61
ACF$^{\oplus}$ [109], provided by [256]	INRIA [91]	$0.89_{(+0.01)}$		—
DPM$_{v5}$$^{\oplus}$ [135]	INRIA [91]	$0.89_{(+0.02)}$		0.08 ± 0.00
DPM$_{v5}$ [135]	INRIA [91]	$0.84_{(+0.03)}$		0.24 ± 0.05
DPM$_{v5}$$^{\oplus}$ [135]	VOC$_{07}$ [122]	$0.85_{(+0.02)}$		0.08 ± 0.02
DPM$_{v5}$ [135]	VOC$_{07}$ [122]	$0.73_{(+0.03)}$		0.17 ± 0.12
DPM$_{v5}$ Person Grammar$^{\oplus}$ [135, 153]	VOC$_{07}$ [122]	$0.81_{(+0.01)}$		0.06 ± 0.01
DPM$_{v5}$ Person Grammar [135, 153]	VOC$_{07}$ [122]	$0.79_{(+0.02)}$		0.16 ± 0.04
DPM$_{v4}$$^{\oplus}$ [135], provided by [192, 193]	VOC$_{09}$ [122]	$0.92_{(+0.01)}$		—
DPM$_{v5}$$^{\oplus}$ [135]	VOC$_{10}$ [122]	$0.84_{(+0.01)}$		0.08 ± 0.02
DPM$_{v5}$ [135]	VOC$_{10}$ [122]	$0.72_{(+0.03)}$		0.17 ± 0.12
DPM$_{v5}$ Person Grammar$^{\oplus}$ [135, 153]	VOC$_{10}$ [122]	$0.84_{(+0.02)}$		0.06 ± 0.01
DPM$_{v5}$ Person Grammar [135, 153]	VOC$_{10}$ [122]	$0.82_{(+0.02)}$		0.16 ± 0.04
F-RCNN Inception-ResNet$_{v2}$ [362, 407]	COCO [276]	$0.91_{(+0.00)}$	✓	2.57 ± 0.13

Table continued on next page.

Table C.7: Pedestrian detection on PETS'09 S2L1 – *Continued from previous page.*

Detector	**Training Data**	**AUC**$^{\uparrow}$	**GPU**	**FPS**$^{\uparrow}$
F-RCNN Inception$_{v2}$ [362, 406]	COCO [276]	$0.87_{(+0.04)}$	✓	10.99 ± 0.46
F-RCNN NAS [362, 500]	COCO [276]	$0.92_{(+0.00)}$	✓	2.60 ± 0.13
F-RCNN ResNet$_{101}$ [181, 362]	COCO [276]	$0.89_{(+0.01)}$	✓	7.30 ± 0.40
F-RCNN ResNet$_{101}$ [181, 362]	KITTI [150]	$0.65_{(+0.08)}$	✓	12.46 ± 0.54
F-RCNN ResNet$_{50}$ [181, 362]	COCO [276]	$0.88_{(+0.01)}$	✓	7.91 ± 0.37
IKSVM$^{\oplus}$ [295]	INRIA [91]	$0.85_{(+0.85)}$		0.03 ± 0.00
IKSVM [295]	INRIA [91]	$0.59_{(+0.59)}$		0.14 ± 0.01
LDCF$^{\oplus}$ [322]	Caltech [108]	$0.81_{(+0.02)}$		0.99 ± 0.05
LDCF [322]	Caltech [108]	$0.83_{(+0.02)}$		3.28 ± 0.19
Poselets [56]	H3D [56]	$0.87_{(+0.00)}$		0.07 ± 0.01
R-FCN ResNet$_{101}$ [90, 181]	COCO [276]	$0.89_{(+0.00)}$	✓	9.24 ± 0.47
SSD Inception$_{v2}$ [280, 406]	COCO [276]	$0.76_{(+0.01)}$	✓	16.74 ± 1.15
SSD MobileNet [197, 280]	COCO [276]	$0.71_{(+0.03)}$	✓	17.97 ± 1.09
YOLO$_{v2}$ [358]	COCO [276]	$0.80_{(+0.00)}$	✓	62.76 ± 2.90

Table C.8: Evaluation of state-of-the-art pedestrian detectors on the PETS'09 S2L2 [136] dataset. The superscript $^{\oplus}$ indicates that the input images have been upsampled (to twice the size) in order to better match the object sizes used during training the detector model. Best, second best and third best results have been highlighted for each measure.

Detector	**Training Data**	**AUC**$^{\uparrow}$	**GPU**	**FPS**$^{\uparrow}$
ACF$^{\oplus}$ [109]	Caltech [108]	$0.41_{(+0.04)}$		2.86 ± 0.22
ACF [109]	Caltech [108]	$0.43_{(+0.04)}$		9.48 ± 1.50
ACF$^{\oplus}$ [109]	INRIA [91]	$0.72_{(+0.04)}$		8.36 ± 0.64
ACF [109]	INRIA [91]	$0.35_{(+0.06)}$		28.99 ± 2.72
ACF$^{\oplus}$ [109], provided by [256]	INRIA [91]	$0.56_{(+0.05)}$		—
DPM$_{v5}$$^{\oplus}$ [135]	INRIA [91]	$0.67_{(+0.03)}$		0.08 ± 0.01
DPM$_{v5}$ [135]	INRIA [91]	$0.48_{(+0.02)}$		0.30 ± 0.05
DPM$_{v5}$$^{\oplus}$ [135]	VOC$_{07}$ [122]	$0.62_{(+0.02)}$		0.11 ± 0.02
DPM$_{v5}$ [135]	VOC$_{07}$ [122]	$0.42_{(+0.01)}$		0.34 ± 0.10
DPM$_{v5}$ Person Grammar$^{\oplus}$ [135, 153]	VOC$_{07}$ [122]	$0.59_{(+0.03)}$		0.06 ± 0.01
DPM$_{v5}$ Person Grammar [135, 153]	VOC$_{07}$ [122]	$0.48_{(+0.01)}$		0.21 ± 0.04

Table continued on next page

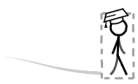


Table C.8: Pedestrian detection on PETS'09 S2L2 – *Continued from previous page.*

Detector	**Training Data**	**AUC**$^{\uparrow}$	**GPU**	**FPS**$^{\uparrow}$
$DPM_{v4}^{\oplus}$ [135], provided by [192, 193]	VOC_{09} [122]	$0.71_{(+0.03)}$		—
$DPM_{v5}^{\oplus}$ [135]	VOC_{10} [122]	$0.61_{(+0.02)}$		0.11 ± 0.02
DPM_{v5} [135]	VOC_{10} [122]	$0.42_{(+0.01)}$		0.33 ± 0.10
DPM_{v5} Person Grammar$^{\oplus}$ [135, 153]	VOC_{10} [122]	$0.59_{(+0.00)}$		0.06 ± 0.01
DPM_{v5} Person Grammar [135, 153]	VOC_{10} [122]	$0.50_{(+0.01)}$		0.21 ± 0.04
F-RCNN Inception-$ResNet_{v2}$ [362, 407]	COCO [276]	$0.79_{(+0.03)}$	✓	2.57 ± 0.15
F-RCNN $Inception_{v2}$ [362, 406]	COCO [276]	$0.75_{(+0.07)}$	✓	10.86 ± 0.55
F-RCNN NAS [362, 500]	COCO [276]	$0.75_{(+0.01)}$	✓	2.60 ± 0.15
F-RCNN $ResNet_{101}$ [181, 362]	COCO [276]	$0.79_{(+0.03)}$	✓	7.30 ± 0.44
F-RCNN $ResNet_{101}$ [181, 362]	KITTI [150]	$0.48_{(+0.09)}$	✓	12.43 ± 0.66
F-RCNN $ResNet_{50}$ [181, 362]	COCO [276]	$0.78_{(+0.04)}$	✓	7.90 ± 0.44
IKSVM$^{\oplus}$ [295]	INRIA [91]	$0.66_{(+0.61)}$		0.02 ± 0.01
IKSVM [295]	INRIA [91]	$0.33_{(+0.33)}$		0.12 ± 0.02
LDCF$^{\oplus}$ [322]	Caltech [108]	$0.42_{(+0.02)}$		1.04 ± 0.04
LDCF [322]	Caltech [108]	$0.44_{(+0.04)}$		3.36 ± 0.20
Poselets [56]	H3D [56]	$0.65_{(+0.05)}$		0.03 ± 0.01
R-FCN $ResNet_{101}$ [90, 181]	COCO [276]	$0.75_{(+0.03)}$	✓	9.20 ± 0.55
SSD $Inception_{v2}$ [280, 406]	COCO [276]	$0.50_{(+0.08)}$	✓	16.64 ± 1.22
SSD MobileNet [197, 280]	COCO [276]	$0.44_{(+0.10)}$	✓	17.79 ± 1.19
$YOLO_{v2}$ [358]	COCO [276]	$0.51_{(+0.07)}$	✓	63.50 ± 1.88

Table C.9: Evaluation of state-of-the-art pedestrian detectors on the PETS'09 S2L3 [136] dataset. The superscript $^{\oplus}$ indicates that the input images have been upsampled (to twice the size) in order to better match the object sizes used during training the detector model. Best, second best and third best results have been highlighted for each measure.

Detector	**Training Data**	**AUC**$^{\uparrow}$	**GPU**	**FPS**$^{\uparrow}$
ACF$^{\oplus}$ [109]	Caltech [108]	$0.34_{(+0.11)}$		3.16 ± 0.24
ACF [109]	Caltech [108]	$0.36_{(+0.12)}$		12.58 ± 2.20
ACF$^{\oplus}$ [109]	INRIA [91]	$0.63_{(+0.07)}$		9.14 ± 1.06
ACF [109]	INRIA [91]	$0.30_{(+0.17)}$		31.69 ± 4.10
$DPM_{v5}^{\oplus}$ [135]	INRIA [91]	$0.60_{(+0.09)}$		0.08 ± 0.00

Table continued on next page.

Table C.9: Pedestrian detection on PETS'09 S2L3 – *Continued from previous page.*

Detector	**Training Data**	**AUC**$^{\uparrow}$	**GPU**	**FPS**$^{\uparrow}$
DPM$_{v5}$ [135]	INRIA [91]	$0.46_{(+0.06)}$		0.28 ± 0.05
DPM$_{v5}{}^{\oplus}$ [135]	VOC$_{07}$ [122]	$0.59_{(+0.08)}$		0.10 ± 0.03
DPM$_{v5}$ [135]	VOC$_{07}$ [122]	$0.43_{(+0.08)}$		0.29 ± 0.14
DPM$_{v5}$ Person Grammar$^{\oplus}$ [135, 153]	VOC$_{07}$ [122]	$0.58_{(+0.09)}$		0.06 ± 0.01
DPM$_{v5}$ Person Grammar [135, 153]	VOC$_{07}$ [122]	$0.48_{(+0.07)}$		0.19 ± 0.05
DPM$_{v4}{}^{\oplus}$ [135], provided by [192, 193]	VOC$_{09}$ [122]	$0.60_{(+0.02)}$		—
DPM$_{v5}{}^{\oplus}$ [135]	VOC$_{10}$ [122]	$0.58_{(+0.09)}$		0.10 ± 0.03
DPM$_{v5}$ [135]	VOC$_{10}$ [122]	$0.43_{(+0.08)}$		0.28 ± 0.14
DPM$_{v5}$ Person Grammar$^{\oplus}$ [135, 153]	VOC$_{10}$ [122]	$0.56_{(+0.07)}$		0.06 ± 0.01
DPM$_{v5}$ Person Grammar [135, 153]	VOC$_{10}$ [122]	$0.49_{(+0.07)}$		0.19 ± 0.05
F-RCNN Inception-ResNet$_{v2}$ [362, 407]	COCO [276]	$0.68_{(+0.04)}$	✓	2.58 ± 0.18
F-RCNN Inception$_{v2}$ [362, 406]	COCO [276]	$0.56_{(+0.04)}$	✓	10.85 ± 0.71
F-RCNN NAS [362, 500]	COCO [276]	$0.64_{(+0.04)}$	✓	2.59 ± 0.18
F-RCNN ResNet$_{101}$ [181, 362]	COCO [276]	$0.68_{(+0.05)}$	✓	7.30 ± 0.55
F-RCNN ResNet$_{101}$ [181, 362]	KITTI [150]	$0.50_{(+0.14)}$	✓	12.41 ± 0.82
F-RCNN ResNet$_{50}$ [181, 362]	COCO [276]	$0.65_{(+0.04)}$	✓	7.87 ± 0.55
IKSVM$^{\oplus}$ [295]	INRIA [91]	$0.45_{(+0.45)}$		0.03 ± 0.01
IKSVM [295]	INRIA [91]	$0.22_{(+0.22)}$		0.13 ± 0.03
LDCF$^{\oplus}$ [322]	Caltech [108]	$0.41_{(+0.08)}$		1.06 ± 0.04
LDCF [322]	Caltech [108]	$0.42_{(+0.08)}$		3.59 ± 0.22
Poselets [56]	H3D [56]	$0.60_{(+0.09)}$		0.06 ± 0.03
R-FCN ResNet$_{101}$ [90, 181]	COCO [276]	$0.66_{(+0.03)}$	✓	9.22 ± 0.69
SSD Inception$_{v2}$ [280, 406]	COCO [276]	$0.35_{(+0.01)}$	✓	16.47 ± 1.38
SSD MobileNet [197, 280]	COCO [276]	$0.30_{(+0.04)}$	✓	17.75 ± 1.36
YOLO$_{v2}$ [358]	COCO [276]	$0.31_{(+0.03)}$	✓	62.56 ± 2.96

Table C.10: Evaluation of state-of-the-art pedestrian detectors on the TownCentre [36] dataset. Due to the significant object size variations (caused by the viewpoint) neither down- nor upscaling the image lead to notable improvements. Thus, only the results on the original input images are reported. Best, second best and third best results have been highlighted for each measure.

Detector	Training Data	AUC$^{\uparrow}$	GPU	FPS$^{\uparrow}$
ACF [109]	Caltech [108]	$0.36_{(+0.01)}$		2.50 ± 0.18
ACF [109]	INRIA [91]	$0.66_{(+0.03)}$		7.40 ± 0.35
ACF$^{\oplus}$ [109], provided by [256]	INRIA [91]	$0.48_{(+0.03)}$		—
DPM_{v5} [135]	INRIA [91]	$0.79_{(+0.04)}$		0.06 ± 0.00
DPM_{v5} [135]	VOC_{07} [122]	$0.78_{(+0.07)}$		0.11 ± 0.00
DPM_{v5} Person Grammar [135, 153]	VOC_{07} [122]	$0.77_{(+0.05)}$		0.06 ± 0.00
DPM_{v5} [135]	VOC_{10} [122]	$0.77_{(+0.07)}$		0.10 ± 0.00
DPM_{v5} Person Grammar [135, 153]	VOC_{10} [122]	$0.80_{(+0.06)}$		0.04 ± 0.00
F-RCNN Inception-$\text{ResNet}_{\text{v2}}$ [362, 407]	COCO [276]	$0.78_{(+0.02)}$	✓	2.43 ± 0.13
F-RCNN $\text{Inception}_{\text{v2}}$ [362, 406]	COCO [276]	$0.71_{(+0.06)}$	✓	10.29 ± 0.57
F-RCNN NAS [362, 500]	COCO [276]	$0.73_{(+0.00)}$	✓	2.62 ± 0.15
F-RCNN ResNet_{101} [181, 362]	COCO [276]	$0.73_{(+0.00)}$	✓	6.84 ± 0.52
F-RCNN ResNet_{101} [181, 362]	KITTI [150]	$0.61_{(+0.00)}$	✓	11.07 ± 0.75
F-RCNN ResNet_{50} [181, 362]	COCO [276]	$0.75_{(+0.00)}$	✓	7.64 ± 0.48
HOG [91], provided by [36]	INRIA [91]	$0.62_{(+0.02)}$		—
IKSVM [295]	INRIA [91]	$0.74_{(+0.66)}$		0.02 ± 0.00
LDCF [322]	Caltech [108]	$0.33_{(+0.02)}$		0.88 ± 0.05
Poselets [56]	H3D [56]	$0.82_{(+0.06)}$		0.01 ± 0.00
R-FCN ResNet_{101} [90, 181]	COCO [276]	$0.78_{(+0.04)}$	✓	8.52 ± 0.54
SSD $\text{Inception}_{\text{v2}}$ [280, 406]	COCO [276]	$0.45_{(+0.07)}$	✓	15.63 ± 1.47
SSD MobileNet [197, 280]	COCO [276]	$0.39_{(+0.10)}$	✓	17.13 ± 1.28
YOLO_{v2} [358]	COCO [276]	$0.49_{(+0.05)}$	✓	65.39 ± 0.99

C.3 Multiple Object Tracking Results

In the following, we report the detailed tracking results for our multiple object tracking approach. The results on the PETS'09 S2L1, S2L2 and S2L3 [136] sequences are shown in Table C.11, C.12 and C.13, respectively. Table C.14 lists the tracking results on the TownCentre [36] dataset. For each sequence, we report the results for our occlusion geodesics-based tracker (denoted OccGeo) using different off-the-shelf pedestrian detectors and compare these to state-of-the-art approaches published at major computer vision conferences and journals. Since raw tracking results are mostly not available, we show the results of state-of-the-art approaches reported within the corresponding publications or provided by the authors via personal correspondence – thus, these results should only be considered for reference but not for direct comparison as we cannot ensure the same evaluation protocol. In particular, despite using the standard CLEAR metrics, there are subtle differences which slightly effect the overall results, *e.g.* the way of counting identity switches [253, 273] or whether bounding box overlap (following the PASCAL criterion)

Table C.11: Tracking results on PETS'09 S2L1 [136]. We compare our tracker using different off-the-shelf detectors to various state-of-the-art approaches. The second and third column indicate if the corresponding tracker uses an instance-specific appearance model (A) and is causal (C), respectively. Best, second best and third best results have been highlighted for each measure.

	Tracker	A	C	MOTA↑	MOTP↑	MT↑	ML↓	IDs↓	FM↓	FPS↑
Ours	OccGeo using DPM		✓	0.96	0.81	1.00	0.00	12	20	28.2
	OccGeo using R-FCN		✓	0.88	0.74	0.89	0.00	12	32	27.2
	OccGeo using ACF		✓	0.86	0.77	0.89	0.00	13	22	19.1
	OccGeo using Poselets		✓	0.84	0.76	0.79	0.00	19	28	24.9
	OccGeo using LDCF		✓	0.78	0.72	0.89	0.00	22	21	11.8
	OccGeo using IKSVM		✓	0.78	0.70	0.79	0.00	15	34	21.6
	OccGeo using F-RCNN		✓	0.68	0.65	0.79	0.00	21	55	27.2
	OccGeo using SSD		✓	0.67	0.67	0.68	0.00	44	49	15.6
	OccGeo using YOLO		✓	0.64	0.64	0.53	0.05	37	60	12.7
Major Literature	Hofmann *et al.* [193]			0.98	0.83	1.00	0.00	10	11	–
	Hofmann *et al.* [192]	✓		0.98	0.75	1.00	0.00	8	8	–
	Jiang *et al.* [213]	✓	✓	0.96	0.88	0.95	0.00	6	5	66.7
	Andriyenko *et al.* [13]			0.96	0.79	1.00	0.00	10	8	2.0
	Wu *et al.* [453]	✓	✓	0.93	0.74	1.00	0.00	8	11	1.7
	Izadinia *et al.* [210]	✓		0.91	0.76	–	–	–	–	–
	Milan *et al.* [308]	✓		0.91	0.80	0.91	0.04	11	6	–
	Milan *et al.* [307]			0.90	0.74	0.78	0.00	22	15	–
	Zamir *et al.* [476]	✓		0.90	0.69	0.90	0.00	10	54	–
	Henriques *et al.* [187]	✓		0.83	0.71	0.90	0.00	19	45	–
	Andriyenko and Schindler [12]			0.81	0.76	0.83	0.00	15	21	–
	Berclaz *et al.* [39]			0.80	0.72	0.74	0.09	13	22	–
	Breitenstein *et al.* [60]	✓	✓	0.80	0.56	–	–	–	–	1.2
	Yang *et al.* [465]	✓	✓	0.76	0.54	–	–	–	–	–

or ground plane distances (with a cut-off threshold of typically 1 [m]) are used to assign tracking results to ground truth annotations. For the sequences contained in the 3D MOT'15 [256] benchmark – *i.e.* PETS'09 S2L2 and TownCentre – we also compare to officially benchmarked trackers using the publicly available tracking results. For these trackers, we use the same evaluation protocol as for our OccGeo approach to ensure a fair comparison – please refer to Section 5.2.2 for details.

Table C.12: Tracking results on PETS'09 S2L2 [136]. We compare our tracker using different off-the-shelf detectors to various state-of-the-art approaches, including trackers with participated in the 3D MOT'15 benchmark [256]. The second and third column indicate if the corresponding tracker uses an instance-specific appearance model (A) and is causal (C), respectively. Best, second best and third best results have been highlighted for each measure.

	Tracker	A	C	MOTA↑	MOTP↑	MT↑	ML↓	IDs↓	FM↓	FPS↑
Ours	OccGeo using F-RCNN		✓	0.60	0.62	0.44	0.09	118	146	3.5
	OccGeo using DPM		✓	0.57	0.65	0.28	0.14	125	136	7.8
	OccGeo using ACF		✓	0.43	0.62	0.47	0.07	216	182	2.1
	OccGeo using R-FCN		✓	0.41	0.61	0.26	0.07	206	173	2.3
	OccGeo using IKSVM		✓	0.40	0.60	0.12	0.26	90	116	4.9
	OccGeo using Poselets		✓	0.37	0.61	0.19	0.12	195	196	2.5
	OccGeo using YOLO		✓	0.31	0.57	0.09	0.16	140	186	3.0
	OccGeo using LDCF		✓	0.30	0.62	0.19	0.12	165	134	2.7
	OccGeo using SSD		✓	0.26	0.62	0.05	0.37	101	127	4.5
3D MOT'15	GPR-DBN [232]	✓	✓	0.54	0.65	0.23	0.14	122	163	–
	STV [440]	✓		0.46	0.55	0.14	0.12	186	215	–
	LP-3D [255]			0.42	0.50	0.19	0.09	220	249	–
	LP-SFM [253]			0.39	0.52	0.05	0.19	173	208	–
	S-RNN [368]	✓	✓	0.31	0.53	0.00	0.14	515	677	–
	K-SFM [341]		✓	0.30	0.52	0.02	0.07	698	683	–
Major Literature	Hofmann *et al.* [193]			0.76	0.72	0.65	0.00	234	252	–
	Wu *et al.* [453]	✓	✓	0.73	0.73	0.69	0.04	122	113	1.3
	Hofmann *et al.* [192]	✓		0.57	0.56	0.40	0.18	67	59	–
	Milan *et al.* [308]	✓		0.57	0.59	0.38	0.16	99	73	–
	Jiang *et al.* [213]	✓	✓	0.51	0.67	0.60	0.18	119	146	23.0
	Milan *et al.* [307]			0.46	0.60	0.34	0.11	126	105	–
	Berclaz *et al.* [39]			0.24	0.61	0.10	0.54	22	38	–

Table C.13: Tracking results on PETS'09 S2L3 [136]. We compare our tracker using different off-the-shelf detectors to various state-of-the-art approaches. The second and third column indicate if the corresponding tracker uses an instance-specific appearance model (A) and is causal (C), respectively. Best, second best and third best results have been highlighted for each measure.

	Tracker	A	C	MOTA$^\uparrow$	MOTP$^\uparrow$	MT$^\uparrow$	ML$^\downarrow$	IDs$^\downarrow$	FM$^\downarrow$	FPS$^\uparrow$
Ours	OccGeo using F-RCNN		✓	0.51	0.70	0.35	0.40	23	22	25.7
	OccGeo using R-FCN		✓	0.47	0.56	0.21	0.16	84	89	26.8
	OccGeo using DPM		✓	0.46	0.48	0.14	0.28	67	99	17.8
	OccGeo using ACF		✓	0.45	0.63	0.30	0.23	61	80	24.6
	OccGeo using Poselets		✓	0.44	0.62	0.19	0.35	53	63	30.9
	OccGeo using IKSVM		✓	0.41	0.47	0.21	0.35	31	42	40.5
	OccGeo using LDCF		✓	0.41	0.63	0.23	0.44	52	47	24.3
	OccGeo using YOLO		✓	0.21	0.59	0.07	0.63	36	49	13.5
	OccGeo using SSD		✓	0.18	0.56	0.05	0.65	18	29	33.6
Major Lit.	Hofmann *et al.* [193]			0.63	0.71	0.55	0.11	225	217	–
	Wu *et al.* [453]	✓	✓	0.58	0.70	0.48	0.18	41	39	1.2
	Milan *et al.* [308]	✓		0.46	0.65	0.21	0.41	38	27	–
	Hofmann *et al.* [192]	✓		0.42	0.65	0.34	0.32	49	67	–
	Milan *et al.* [307]			0.40	0.65	0.18	0.43	27	22	–
	Berclaz *et al.* [39]			0.29	0.62	0.11	0.71	7	12	–

Table C.14: Tracking results on TownCentre [36]. We compare our tracker using different off-the-shelf detectors to various state-of-the-art approaches, including trackers with participated in the 3D MOT'15 benchmark [256]. The second and third column indicate if the corresponding tracker uses an instance-specific appearance model (A) and is causal (C), respectively. Best, second best and third best results have been highlighted for each measure.

	Tracker	A	C	MOTA$^\uparrow$	MOTP$^\uparrow$	MT/GT$^\uparrow$	ML/GT$^\downarrow$	IDS$^\downarrow$	FM$^\downarrow$	FPS$^\uparrow$
Ours	OccGeo using DPM		✓	0.43	0.57	0.25	0.26	225	234	7.2
	OccGeo using IKSVM		✓	0.38	0.59	0.13	0.33	185	218	10.2
	OccGeo using Poselets		✓	0.36	0.57	0.21	0.20	218	262	5.1
	OccGeo using ACF		✓	0.35	0.59	0.33	0.17	286	277	5.3
	OccGeo using R-FCN		✓	0.32	0.54	0.20	0.30	248	262	6.3
	OccGeo using F-RCNN		✓	0.28	0.53	0.13	0.41	164	236	6.7
	OccGeo using SSD		✓	0.17	0.49	0.04	0.41	302	307	5.9
	OccGeo using LDCF		✓	0.14	0.57	0.07	0.40	243	277	6.6
	OccGeo using YOLO		✓	0.12	0.52	0.04	0.42	229	323	6.2
3D MOT'15	GPR-DBN [232]	✓	✓	0.42	0.59	0.35	0.22	59	107	–
	LP-SFM [253]			0.22	0.53	0.18	0.23	223	259	–
	LP-3D [255]			0.15	0.53	0.26	0.15	267	293	–
	S-RNN [368]	✓	✓	0.11	0.55	0.03	0.40	270	376	–
	STV [440]	✓		0.11	0.55	0.14	0.28	197	224	–
	K-SFM [341]		✓	0.09	0.52	0.08	0.15	765	639	–

Table continued on next page.

Table C.14: MOT on TownCentre – *Continued from previous page.*

	Tracker	A	C	MOTA↑	MOTP↑	MT/GT↑	ML/GT↓	IDS↓	FM↓	FPS↑
Major Literature	Izadinia *et al.* [210]	✓		0.76	0.72	–	–	–	–	–
	Zamir *et al.* [476]	✓		0.76	0.72	–	–	–	–	–
	Leal-Taixé *et al.* [253]			0.71	0.72	0.59	0.07	165	363	–
	Wu *et al.* [453]	✓	✓	0.70	0.69	0.65	0.08	209	453	1.3
	Zhang *et al.* [483]			0.69	0.72	0.53	0.09	243	440	–
	Yamaguchi *et al.* [457]		✓	0.67	0.72	0.58	0.07	302	492	–
	Pellegrini *et al.* [341]		✓	0.66	0.72	0.59	0.07	288	499	–
	Benfold and Reid [36]		✓	0.64	0.80	0.67	0.07	222	343	–
	Jiang *et al.* [213]	✓	✓	0.63	0.72	0.51	0.16	154	356	16.7

Bibliography

Those who cannot remember the past are condemned to repeat it.

— Jorge Agustín Nicolás Ruiz de Santayana y Borrás (The Life of Reason)

[1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on pages 16, 64, 66 and 69.

[2] Jagdishkumar Keshoram Aggarwal and Michael S. Ryoo. Human Activity Analysis: A Review. *ACM Computing Surveys*, 43(3):1–43, 2011. Cited on page 19.

[3] Alexandre Alahi, Laurent Jacques, Yannick Boursier, and Pierre Vandergheynst. Sparsity Driven People Localization with a Heterogeneous Network of Cameras. *Journal of Mathematical Imaging and Vision (JMIV)*, 41(1-2):39–58, 2011. Cited on pages 20 and 21.

[4] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 20, 21, 24, 51 and 117.

[5] Saad Ali and Mubarak Shah. Floor Fields for Tracking in High Density Crowd Scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. Cited on pages 20 and 23.

[6] Roy Allen, Peter McGeorge, David Pearson, and Alan B. Milne. Attention and Expertise in Multiple Target Tracking. *Applied Cognitive Psychology*, 18(3):337–347, 2004. Cited on page 2.

[7] Nicolas Alt, Stefan Hinterstoisser, and Nassir Navab. Rapid Selection of Reliable Templates for Visual Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on page 16.

[8] Padmanabhan Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision (IJCV)*, 2(3):283–310, 1989. Cited on pages 2 and 13.

[9] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-Tracking-by-Detection and People-Detection-by-Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 19, 20, 21 and 23.

[10] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3D Pose Estimation and Tracking-by-Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 20, 23 and 97.

[11] Anton Andriyenko and Konrad Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. Cited on pages 20 and 21.

[12] Anton Andriyenko and Konrad Schindler. Multi-target Tracking by Continuous Energy Minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 50, 54 and 149.

[13] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-Continuous Optimization for Multi-Target Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 20, 50, 100 and 149.

[14] Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Abhijit Ogale, and Dave Ferguson. Real-Time Pedestrian Detection With Deep Network Cascades. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015. Cited on page 51.

[15] Shai Avidan. Support Vector Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. Cited on page 21.

[16] Shai Avidan. Ensemble Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(2):261–271, 2007. Cited on page 21.

[17] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual Tracking with Online Multiple Instance Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on pages 15 and 64.

[18] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(7):1324–1338, 2011. Cited on pages 15, 66, 69, 71, 83, 84 and 95.

[19] R. Venkatesh Babu, Patrick Pérez, and Patrick Bouthemy. Kernel-Based Robust Tracking for Objects Undergoing Occlusion. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2006. Cited on page 28.

[20] R. Venkatesh Babu, Patrick Pérez, and Patrick Bouthemy. Robust tracking with motion estimation and local Kernel-based color modeling. *Image and Vision Computing (IVC)*, 25 (8):1205–1216, 2007.

[21] Vijay Badrinarayanan, Patrick Pérez, Francois Le Clerc, and Lionel Oisel. Probabilistic Color and Adaptive Multi-Feature Tracking with Dynamically Switched Priority Between Cues. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. Cited on page 28.

[22] Seung-Hwan Bae and Kuk-Jin Yoon. Robust Online Multi-Object Tracking based on Tracklet Confidence and Online Discriminative Appearance Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on pages 20, 23 and 100.

[23] Seung-Hwan Bae and Kuk-Jin Yoon. Confidence-based Data Association and Discriminative Deep Appearance Learning for Robust Online Multi-object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(3):595–610, 2018. Cited on pages 20 and 23.

[24] Yancheng Bai and Ming Tang. Robust Tracking via Weakly Supervised Ranking SVM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 14 and 16.

[25] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004. Cited on page 13.

[26] Dana H. Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition (PR)*, 13(2):111–122, 1981. Cited on page 13.

[27] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 14, 15 and 69.

[28] Herbert Bay, Tinne Tuytelaars, and Luc van Gool. SURF: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006. Cited on page 11.

[29] Loris Bazzani, Nando de Freitas, Hugo Larochelle, Vittorio Murino, and Jo-Anne Ting. Learning Attentional Policies for Tracking and Recognition in Video with Deep Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011. Cited on pages 14 and 15.

[30] Vasileios Belagiannis, Falk Schubert, Nassir Navab, and Slobodan Ilic. Segmentation Based Particle Filtering for Real-Time 2D Object Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 28, 41 and 64.

[31] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Tracking Multiple People under Global Appearance Constraints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 19, 21 and 23.

[32] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(8):1614–1627, 2014. Cited on pages 19, 21, 22, 23 and 50.

[33] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc van Gool. Pedestrian Detection at 100 Frames per Second. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on page 52.

[34] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten Years of Pedestrian Detection, What Have We Learned? In *Proceedings of the Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving (CVRSUAD, in conjunction with ECCV)*, 2014. Cited on page 52.

[35] Ben Benfold and Ian Reid. Guiding Visual Surveillance by Tracking Human Attention. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. Cited on page 23.

[36] Ben Benfold and Ian Reid. Stable Multi-Target Tracking in Real-Time Surveillance Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 19, 20, 22, 23, 50, 97, 98, 110, 111, 113, 144, 148, 149, 151 and 152.

[37] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks (TNN)*, 5(2):157–166, 1994. Cited on page 24.

[38] Jérôme Berclaz, François Fleuret, and Pascal Fua. Robust People Tracking with Global Trajectory Optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on pages 20 and 23.

[39] Jérôme Berclaz, François Fleuret, Engin Türetken, and Pascal Fua. Multiple Object Tracking using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1806–1819, 2011. Cited on pages 19, 20, 21, 22, 23, 48, 50, 97, 149, 150 and 151.

[40] James R. Bergen, Padmanabhan Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical Model-Based Motion Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1992. Cited on page 2.

[41] Keni Bernardin and Rainer Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *European Association for Signal Processing (EURASIP) Journal on Image and Video Processing*, 2008(1):1–10, 2008. Cited on pages 97, 100 and 101.

[42] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. Staple: Complementary Learners for Real-Time Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 13, 14, 15, 16, 17, 26, 28, 29, 85, 87, 88 and 95.

[43] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-Convolutional Siamese Networks for Object Tracking. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ECCV)*, 2016. Cited on pages 13 and 14.

[44] Margrit Betke, Esin Haritaoglu, and Larry S. Davis. Real-time Multiple Vehicle Detection and Tracking from a Moving Vehicle. *Machine Vision and Applications (MVA)*, 12(2):69–83, 2000. Cited on page 19.

[45] Margrit Betke, Diane E. Hirsh, Angshuman Bagchi, Nickolay I. Hristov, Nicholas C. Makris, and Thomas H. Kunz. Tracking Large Variable Numbers of Objects in Clutter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. Cited on page 19.

[46] Anil Kumar Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943. Cited on page 28.

[47] Charles Bibby and Ian Reid. Robust Real-Time Visual Tracking using Pixel-Wise Posteriors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. Cited on pages 28 and 29.

[48] Adel Bibi, Matthias Mueller, and Bernard Ghanem. Target Response Adaptation for Correlation Filter Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. Cited on page 13.

[49] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, first edition, 2006. Cited on pages 10 and 12.

[50] Christopher M. Bishop and Julia Lasserre. *Bayesian Statistics*, volume 8, chapter Generative or Discriminative? Getting the Best of Both Worlds, pages 3–24. Oxford University Press, 2007. Cited on page 12.

[51] Michael J. Black and Alland D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision (IJCV)*, 26(1):63–84, 1998. Cited on pages 13 and 15.

[52] Andrew Blake and Michael Isard. The CONDENSATION Algorithm - Conditional Density Propagation and Applications to Visual Tracking. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 1996. Cited on page 22.

[53] Jules Bloomenthal and Jon Rokne. Homogeneous Coordinates. *The Visual Computer*, 11 (1):15–26, 1994. Cited on page 52.

[54] David S. Bolme, Bruce A. Draper, and J. Ross Beveridge. Average of Synthetic Exact Filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on page 13.

[55] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual Object Tracking using Adaptive Correlation Filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on pages 13, 14, 26, 29, 37, 64, 79 and 82.

[56] Lubomir Bourdev and Jitendra Malik. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009. Cited on pages 105, 107, 108, 109, 110, 111, 145, 146, 147 and 148.

[57] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting People Using Mutually Consistent Poselet Activations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. Cited on page 48.

[58] Kristian Bredies and Dirk Lorenz. *Mathematische Bildverarbeitung*. Springer (formerly Vieweg+Teubner), first edition, 2011. Cited on page 10.

[59] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc van Gool. Robust Tracking-by-Detection using a Detector Confidence Particle Filter. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009. Cited on pages 20, 21 and 22.

[60] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc van Gool. Online Multi-Person Tracking-by-Detection from a Single, Uncalibrated Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9): 1820–1833, 2011. Cited on pages 19, 20, 21, 22, 26, 48, 50, 62 and 149.

[61] William Brendel, Mohamed Amer, and Sinisa Todorovic. Multiobject Tracking as Maximum Weight Independent Set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on page 22.

[62] Duane C. Brown. Decentering Distortion of Lenses. *Photogrammetric Engineering*, 32(3): 444–462, 1966. Cited on page 53.

[63] François Burgeois and Jean-Claude Lassale. An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. *Communications of the ACM (CACM)*, 14 (12):802–804, 1971. Cited on page 57.

[64] Asad A. Butt and Robert T. Collins. Multi-target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on page 22.

[65] Q. Cai and Jagdishkumar Keshoram Aggarwal. Tracking Human Motion in Structured Environments Using a Distributed-Camera System. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 21(11):1241–1247, 1999. Cited on pages 20 and 23.

[66] Yizheng Cai, Nando de Freitas, and James J. Little. Robust Visual Tracking for Multiple Targets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006. Cited on pages 19, 20, 21 and 50.

[67] Zhaowei Cai, Longyin Wen, Jianwei Yang, Zhen Lei, and Stan Z. Li. Structured Visual Tracking with Dynamic Graph. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2012. Cited on pages 14 and 16.

[68] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos. Learning Complexity-Aware Cascades for Deep Pedestrian Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on page 51.

[69] Joshua Candamo, Matthew Shreve, Dmitry B. Goldgof, Deborah B. Sapper, and Rangachar Kasturi. Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algo-

rithms. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 11(1):206–224, 2010. Cited on page 19.

[70] Patrick Cavanagh and George A. Alvarez. Tracking Multiple Targets with Multifocal Attention. *Trends in Cognitive Sciences (TICS)*, 9(7):349–354, 2005. Cited on page 2.

[71] Luka Čehovin, Matej Kristan, and Aleš Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 16, 28 and 69.

[72] Luka Čehovin, Matej Kristan, and Aleš Leonardis. Robust Visual Tracking using an Adaptive Coupled-layer Visual Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(4):941–953, 2013. Cited on pages 14, 16, 28, 69, 83, 84 and 95.

[73] Luka Čehovin, Matej Kristan, and Aleš Leonardis. Is my new tracker really better than yours? In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, 2014. Cited on pages 69 and 71.

[74] Luka Čehovin, Aleš Leonardis, and Matej Kristan. Visual Object Tracking Performance Measures Revisited. *IEEE Transactions on Image Processing (TIP)*, 25(3):1261–1274, 2016. Cited on pages 69 and 71.

[75] Luka Čehovin, Aleš Leonardis, and Matej Kristan. Robust visual tracking using template anchors. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, 2016. Cited on pages 16 and 28.

[76] Dapeng Chen, Zeijan Yuan, Yang Wu, Geng Zhang, and Nanning Zheng. Constructing Adaptive Complex Cells for Robust Visual Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 14 and 16.

[77] Fan Chen and Christophe De Vleeschouwer. Personalized Production of Basketball Videos from Multi-sensored Data under Limited Display Resolution. *Computer Vision and Image Understanding (CVIU)*, 114(6):667–680, 2010. Cited on page 23.

[78] Hwann-Tzong Chen and Tyng-Luh Liu. Trust-Region Methods for Real-Time Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001. Cited on page 28.

[79] Nicolas Chenouard, Isabelle Bloch, and Jean-Christophe Olivo-Marin. Multiple Hypothesis Tracking for Cluttered Biological Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2736–2750, 2013. Cited on page 19.

[80] Wongun Choi. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 20 and 21.

[81] Wongun Choi and Silvio Savarese. Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[82] Wongun Choi and Silvio Savarese. A Unified Framework for Multi-Target Tracking and Collective Activity Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 20 and 22.

[83] Robert T. Collins. Mean-Shift Blob Tracking through Scale Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. Cited on page 28.

[84] Robert T. Collins, Yanxi Liu, and Marius Leordeanu. Online Selection of Discriminative Tracking Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(10):1631–1643, 2005. Cited on pages 16 and 28.

[85] Dorin Comaniciu and Peter Meer. Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24 (5):603–619, 2002. Cited on pages 2 and 28.

[86] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-Time Tracking of Non-Rigid Objects using Mean Shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000. Cited on page 28.

[87] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(5):564–577, 2003. Cited on pages 7, 26 and 28.

[88] Franklin C. Crow. Summed-Area Tables for Texture Mapping. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1984. Cited on page 37.

[89] Zhen Cui, Shengtao Xiao, Jiashi Feng, and Shuicheng Yan. Recurrently Target-Attending Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 13 and 14.

[90] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2016. Cited on pages 106, 107, 108, 109, 110, 111, 144, 145, 146, 147 and 148.

[91] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. Cited on pages 11, 15, 26, 48, 51, 52, 82, 98, 106, 107, 109, 110, 144, 145, 146, 147 and 148.

[92] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate Scale Estimation for Robust Visual Tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2014. Cited on pages 13, 14, 15, 16, 26, 27, 29, 37, 43, 68, 82, 84, 85, 86, 87, 88 and 95.

[93] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer. Adaptive Color Attributes for Real-Time Visual Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on pages 13, 14, 16, 26, 27, 29, 68, 84, 85, 86, 87, 88, 94, 95, 135, 136, 137, 138, 139, 140 and 141.

[94] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Learning Spatially Regularized Correlation Filters for Visual Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 13, 14, 16 and 17.

[95] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional Features for Correlation Filter Based Visual Tracking. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2015. Cited on pages 14 and 16.

[96] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive Decontamination of the Training Set: A Unified Formulation for Discriminative Visual Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on page 13.

[97] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In *Pro-*

ceedings of the European Conference on Computer Vision (ECCV), 2016. Cited on pages 14, 16, 26, 29, 85, 87, 88, 94 and 95.

[98] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient Convolution Operators for Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 13, 14, 26 and 29.

[99] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Discriminative Scale Space Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(8):1561–1575, 2017. Cited on pages 13, 14 and 26.

[100] Jesse Davis and Mark Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2006. Cited on page 106.

[101] Afshin Dehghan and Mubarak Shah. Binary Quadratic Programing for Online Tracking of Hundreds of People in Extremely Crowded Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(3):568–581, 2018. Cited on page 22.

[102] Afshin Dehghan, Yicong Tian, Philip H. S. Torr, and Mubarak Shah. Target Identity-aware Network Flow for Online Multiple Target Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 22.

[103] Caglayan Dicle, Mario Sznaier, and Octavia Camps. The Way They Move: Tracking Multiple Targets with Similar Appearance. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 20 and 23.

[104] Thang Ba Dinh, Nam Vo, and Gérard Medioni. Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 14, 29, 86, 89, 90, 91, 92, 95, 135, 142 and 143.

[105] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral Channel Features. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. Cited on page 52.

[106] Piotr Dollár, Serge Belongie, and Pietro Perona. The Fastest Pedestrian Detector in the West. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[107] Piotr Dollár, Ron Appel, and Wolf Kienzle. Crosstalk Cascades for Frame-Rate Pedestrian Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on page 52.

[108] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(4):743–761, 2012. Cited on pages 52, 106, 109, 110, 144, 145, 146, 147 and 148.

[109] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perone. Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(8):1532–1545, 2014. Cited on pages 48, 52, 58, 103, 105, 107, 108, 109, 110, 111, 144, 145, 146 and 148.

[110] Dawei Du, Honggang Qi, Wenbo Li, Longyin Wen, Qingming Huang, and Siwei Lyu. Online Deformable Object Tracking Based on Structure-Aware Hyper-Graph. *IEEE Transactions on Image Processing (TIP)*, 25(8):3572–3584, 2016. Cited on pages 14 and 16.

[111] Wei Du and Justus Piater. A Probabilistic Approach to Integrating Multiple Cues in Visual Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. Cited on pages 16 and 28.

[112] Genquan Duan, Haizhou Ai, Song Cao, and Shihong Lao. Group Tracking: Exploring Mutual Relations for Multiple Object Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 19, 20, 21 and 51.

[113] Stefan Duffner and Christophe Garcia. PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 14, 16, 28, 84, 94 and 95.

[114] Stefan Duffner and Christophe Garcia. Fast Pixelwise Adaptive Visual Tracking of Non-Rigid Objects. *IEEE Transactions on Image Processing (TIP)*, 26(5):2368–2380, 2017. Cited on pages 16 and 28.

[115] Ahmed Elgammal, David Harwood, and Larry S. Davis. Non-parametric Model for Background Subtraction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2000. Cited on page 15.

[116] Ahmed Elgammal, Ramani Duraiswami, and Larry S. Davis. Probabilistic Tracking in Joint Feature-Spatial Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. Cited on page 28.

[117] Christian Ertler, Horst Possegger, Michael Opitz, and Horst Bischof. Pedestrian Detection in RGB-D Images from an Elevated Viewpoint. In *Proceedings of the Computer Vision Winter Workshop (CVWW)*, 2017. Cited on page 51.

[118] Ran Eshel and Yeal Moses. Tracking in a Dense Crowd Using Multiple Cameras. *International Journal of Computer Vision (IJCV)*, 88(1):129–143, 2010. Cited on pages 19 and 20.

[119] Andreas Ess, Bastian Leibe, and Luc van Gool. Depth and Appearance for Mobile Scene Analysis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. Cited on page 23.

[120] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. A Mobile Vision System for Robust Multi-Person Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 20 and 23.

[121] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. Robust Multi-Person Tracking from a Mobile Platform. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(10):1831–1846, 2009. Cited on pages 20, 23 and 97.

[122] Mark Everingham, Luc van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. Cited on pages 39, 51, 69, 109, 110, 144, 145, 146, 147 and 148.

[123] Mark D. Fairchild. *Color Appearance Models*. Wiley, third edition, 2013. Cited on page 72.

[124] Heng Fan and Haibin Ling. SANet: Structure-Aware Network for Visual Tracking. In *Proceedings of the Workshop on Deep Vision: Deep Learning in Computer Vision (DVW, in conjunction with CVPR)*, 2017. Cited on pages 13, 14, 39 and 45.

[125] Heng Fan and Haibin Ling. Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 13 and 14.

[126] Jialue Fan, Xiaohui Shen, and Ying Wu. Scribble Tracker: A Matting-Based Approach for Robust Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(8):1633–1644, 2012. Cited on pages 12 and 28.

[127] Gunnar Farnebäck. Very High Accuracy Velocity Estimation using Orientation Tensors, Parametric Motion, and Simultaneous Segmentation of the Motion Field. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001. Cited on page 2.

[128] Gunnar Farnebäck. Two-Frame Motion Estimation Based on Polynomial Expansion. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, 2003. Cited on page 2.

[129] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, second edition, 2004. Cited on page 2.

[130] Tom Fawcett. Introduction to ROC analysis. *Pattern Recognition Letters (PRL)*, 27(8): 861–874, 2006. Cited on page 69.

[131] Federal Highway Administration (FHWA). Pedestrian Characteristics. Technical Report FHWA-HRT-05-099, U.S. Department of Transportation, 2006. Cited on page 98.

[132] Michael Felsberg. Enhanced Distribution Field Tracking using Channel Representations. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2013. Cited on pages 14, 15, 16, 82, 83 and 95.

[133] Michael Felsberg, Amanda Berg, Jörgen Ahlberg, Gustav Häger, Matej Kristan, Jiří Matas, Aleš Leonardis, Luka Čehovin, Gustavo Fernández, Tomáš Vojíř, Georg Nebehay, Roman Pflugfelder, *et al.* The Visual Object Tracking VOT-TIR2015 challenge results. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2015. Cited on pages 14, 16 and 17.

[134] Michael Felsberg, Matej Kristan, Jiří Matas, Aleš Leonardis, Roman Pflugfelder, Gustav Häger, Amanda Berg, Abdelrahman Eldesokey, Jörgen Ahlberg, Luka Čehovin, Tomáš Vojíř, Alan Lukežič, Gustavo Fernández, *et al.* The Visual Object Tracking VOT-TIR2016 challenge results. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ECCV)*, 2016. Cited on pages 14, 16 and 17.

[135] Pedro Felipe Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, 2010. Cited on pages 45, 48, 51, 58, 105, 106, 107, 108, 109, 110, 111, 112, 113, 144, 145, 146, 147 and 148.

[136] James M. Ferryman and Ali Shahrokni. PETS 2009: Dataset and Challenge. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS)*, 2009. Cited on pages 23, 61, 97, 98, 109, 113, 144, 145, 146, 149, 150 and 151.

[137] François Fleuret, Jérôme Berclaz, Richard Lengagne, and Pascal Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):267–282, 2008. Cited on pages 20, 21, 22, 23, 48, 50, 62 and 97.

[138] Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffé. Multi-Target Tracking using Joint Probabilistic Data Association. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 1980. Cited on page 22.

[139] Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffé. Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983. Cited on pages 22 and 50.

[140] Daniel Freedman and Tao Zhang. Active Contours for Tracking Distributions. *IEEE Transactions on Image Processing (TIP)*, 13(4):518–526, 2004. Cited on page 28.

[141] Pierre F. Gabriel, Jacques G. Verly, Justus H. Piater, and André Genon. The State of the Art in Multiple Object Tracking Under Occlusion in Video Sequences. In *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2003. Cited on page 9.

[142] Jürgen Gall and Victor Lempitsky. Class-Specific Hough Forests for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on page 52.

[143] Jürgen Gall, Nima Razavi, and Luc van Gool. On-line Adaption of Class-specific Codebooks for Instance Tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010. Cited on pages 20 and 23.

[144] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Multi-Channel Correlation Filters. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 13 and 14.

[145] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Correlation Filters with Limited Boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on pages 13 and 14.

[146] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for Speed: A Benchmark for Higher Frame Rate Object Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 14, 17 and 64.

[147] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning Background-Aware Correlation Filters for Visual Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on page 14.

[148] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer Learning Based Visual Tracking with Gaussian Process Regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on pages 14 and 16.

[149] Dariu M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding (CVIU)*, 73(1):82–98, 1999. Cited on page 9.

[150] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 23, 145, 146, 147 and 148.

[151] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on page 51.

[152] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(1):142–158, 2016. Cited on page 51.

[153] Ross B. Girshick, Pedro F. Felzenszwalb, and David McAllester. Object Detection with Grammar Models. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2011. Cited on pages 108, 110, 144, 145, 146, 147 and 148.

[154] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on pages 13, 23, 45, 46, 106 and 107.

[155] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv CoRR*, abs/1311.2524, 2014. URL http://arxiv.org/abs/1311.2524. Cited on pages 45 and 46.

[156] Martin Godec, Peter M. Roth, and Horst Bischof. Hough-based Tracking of Non-rigid Objects. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 13, 14, 16, 28, 41, 64, 69 and 94.

[157] Martin Godec, Peter M. Roth, and Horst Bischof. Hough-based Tracking of Non-rigid Objects. *Computer Vision and Image Understanding (CVIU)*, 117(10):1245–1256, 2013. Cited on pages 13, 14, 16, 28, 41, 83, 94 and 95.

[158] William J. Godinez and Karl Rohr. Tracking Multiple Particles in Fluorescence Time-Lapse Microscopy Images via Probabilistic Data Association. *IEEE Transactions on Medical Imaging (TMI)*, 34(2):415–432, 2015. Cited on page 19.

[159] Daniel Gordon, Ali Farhadi, and Dieter Fox. Re^3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects. *IEEE Robotics and Automation Letters*, 3(2):788–795, 2018. Cited on pages 13 and 14.

[160] Neil Gordon and David Salmond. Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter. *Journal of Guidance, Control, and Dynamics (JGCD)*, 18(6): 1434–1443, 1995. Cited on page 22.

[161] Helmut Grabner and Horst Bischof. On-line Boosting and Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on page 21.

[162] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time Tracking via On-line Boosting. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2006. Cited on pages 15 and 21.

[163] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised On-line Boosting for Robust Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. Cited on page 15.

[164] Helmut Grabner, Jiří Matas, Luc van Gool, and Philippe Cattin. Tracking the Invisible: Learning Where the Object Might be. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on page 29.

[165] Michael Grabner, Helmut Grabner, and Horst Bischof. Learning Features for Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. Cited on pages 15 and 16.

[166] Gösta H. Granlund. An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In *Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle (AFPAC)*, 2000. Cited on page 82.

[167] Shawn Green and Daphne Bavelier. Action Video Game modifies Visual Selective Attention. *Nature*, 423(6939):534–537, 2003. Cited on page 2.

[168] Stephen Grossberg. Competitive Learning: From Interactive Activation to Adaptive Resonance. *Cognitive Science*, 11(1):23–63, 1987. Cited on page 78.

[169] Li Guan, Jean-Sébastien Franco, and Marc Pollefeys. Multi-Object Shape Estimation and Tracking from Silhouette Cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 20, 21 and 23.

[170] Erhan Gündoğdu and Aydın Alatan. Good Features to Correlate for Visual Tracking. *IEEE Transactions on Image Processing (TIP)*, 27(5):2526–2540, 2018. Cited on page 16.

[171] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning Dynamic Siamese Network for Visual Object Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on page 26.

[172] Gregory D. Hager and Peter N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(10):1025–1039, 1998. Cited on pages 13 and 15.

[173] Gregory D. Hager, Maneesh Dewan, and Charles V. Stewart. Multiple Kernel Tracking with SSD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. Cited on page 28.

[174] Mei Han, Wei Xu, Hai Tao, and Yihong Gong. An Algorithm for Multiple Object Trajectory Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. Cited on pages 20 and 21.

[175] Dan Witzner Hansen and Qiang Ji. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32 (3):478–500, 2010. Cited on page 11.

[176] Sam Hare, Amir Saffari, and Philip H. S. Torr. Struck: Structured Output Tracking with Kernels. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 14, 16, 64, 82, 83, 84, 86, 89, 90, 91, 92, 94 and 95.

[177] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L. Hicks, and Philip H. S. Torr. Struck: Structured Output Tracking with Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(10):2096–2109, 2016. Cited on pages 14, 16, 85, 87, 88, 94 and 95.

[178] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W^4: Who? When? Where? What? A Real Time System for Detecting and Tracking People. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 1998. Cited on page 21.

[179] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. Cited on pages 2, 10, 52 and 56.

[180] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009. Cited on pages 10 and 12.

[181] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 108, 109, 110, 145, 146, 147 and 148.

[182] Shengfeng He, Qingxiong Yang, Rynson W. H. Lau, Jiang Wang, and Ming-Hsuan Yang. Visual Tracking via Locality Sensitive Histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 14 and 16.

[183] Janne Heikkilä and Olli Silvén. A Four-step Camera Calibration Procedure with Implicit Image Correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997. Cited on page 53.

[184] David Held, Sebastian Thrun, and Silvio Savarese. Learning to Track at 100 FPS with Deep Regression Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. Cited on pages 13, 14 and 26.

[185] Hermann Ludwig Ferdinand von Helmholtz. *Handbuch der Physiologischen Optik*. Verlag Leopold Voß, second edition, 1896. Cited on page 2.

[186] Cher Keng Heng, Samantha Yue Ying Lim, Zhi Heng Niu, and Bo Li. Single Scale Pixel based LUT Tracker (PLT). Published as appendix and presented at the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV), 2013. Cited on pages 14, 16, 82, 83 and 95.

[187] João F. Henriques, Rui Caseiro, and Jorge Batista. Globally Optimal Solution to Multi-Object Tracking with Merged Measurements. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 19, 20, 21, 48, 50, 62 and 149.

[188] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the Circulant Structure of Tracking-by-detection with Kernels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 13, 14, 16, 26, 64, 82, 86, 89, 90, 91, 92 and 95.

[189] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(3):583–596, 2015. Cited on pages 13, 14, 15, 16, 17, 26, 27, 29, 37, 43, 68, 79, 82, 84, 85, 86, 87, 88 and 95.

[190] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. A Novel Multi-Detector Fusion Framework for Multi-Object Tracking. *arXiv CoRR*, abs/1705.08314, 2017. URL https://arxiv.org/abs/1705.08314. Cited on pages 20, 21 and 24.

[191] Charles F. Hester and David Casasent. Multivariant Technique for Multi-Class Pattern Recognition. *Applied Optics*, 19(11):1758–1761, 1980. Cited on page 13.

[192] Martin Hofmann, Michael Haag, and Gerhard Rigoll. Unified Hierarchical Multi-Object Tracking using Global Data Association. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2013. Cited on pages 20, 21, 50, 109, 144, 146, 147, 149, 150 and 151.

[193] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. Hypergraphs for Joint Multi-View Reconstruction and Multi-Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 20, 23, 48, 50, 54, 62, 100, 109, 144, 146, 147, 149, 150 and 151.

[194] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015. Cited on pages 13, 14 and 26.

[195] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. MUlti-Store Tracker (MUSTer): a Cognitive Psychology Inspired Approach to Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 14.

[196] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1–3):185–203, 1981. Cited on page 2.

[197] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Adam Hartwig. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv CoRR*, abs/1704.04861, 2017. URL https://arxiv.org/abs/1704.04861. Cited on pages 145, 146, 147 and 148.

[198] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (TSMCC)*, 34(3):334–352, 2004. Cited on page 9.

[199] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (TSMCC)*, 34(3):334–352, 2004. Cited on page 19.

[200] Weiming Hu, Xi Li, Wenhan Luo, Xiaoqin Zhang, Stephen Maybank, and Zhongfei Zhang. Single and Multiple Object Tracking using Log-Euclidean Riemannian Subspace and Block-Division Appearance Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(12):2420–2440, 2012. Cited on pages 20 and 21.

[201] Yang Hua, Karteek Alahari, and Cordelia Schmid. Online Object Tracking with Proposal Selection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 14, 15 and 17.

[202] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust Object Tracking by Hierarchical Association of Detection Responses. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. Cited on pages 20, 21, 22, 48 and 50.

[203] Chen Huang, Simon Lucey, and Deva Ramanan. Learning Policies for Adaptive Tracking with Deep Feature Cascades. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 12, 14, 15, 26, 29 and 46.

[204] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on page 106.

[205] Stephen S. Intille and Aaron F. Bobick. Visual Tracking Using Closed-Worlds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1995. Cited on pages 3, 19, 20 and 21.

[206] Stephen S. Intille, James W. Davis, and Aaron F. Bobick. Real-Time Closed-World Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997. Cited on pages 3, 20 and 21.

[207] Michael Isard and Andrew Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1996. Cited on page 22.

[208] Michael Isard and Andrew Blake. CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998. Cited on pages 2 and 22.

[209] Michael Isard and John Philip MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001. Cited on page 28.

[210] Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. $(\mathrm{MP})^2\mathrm{T}$: Multiple People Multiple Parts Tracker. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 20, 21, 149 and 152.

[211] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust Online Appearance Models for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(10):1296–1311, 2003. Cited on page 16.

[212] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 14, 15, 86, 89, 90, 91, 92 and 95.

[213] Huaizu Jiang, Jinjun Wang, Yihong Gong, Na Rong, Zhenhua Chai, and Nanning Zheng. Online Multi-Target Tracking with Unified Handling of Complex Scenarios. *IEEE Transactions on Image Processing (TIP)*, 24(11):3464–3477, 2015. Cited on pages 100, 149, 150 and 152.

[214] Frédéric Jurie and Michel Dhome. Hyperplane Approximation for Template Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):996–1000, 2002. Cited on pages 2 and 13.

[215] Samira Ebrahimi Kahou, Vincent Michalski, Roland Memisevic, Christopher Pal, and Pascal Vincent. RATM: Recurrent Attentive Tracking Model. In *Proceedings of the Workshop on Brave New Motion Representations (BNMW, in conjunction with CVPR)*, 2017. Cited on pages 14 and 15.

[216] Zdenek Kalal, Krystian Mikolajczyk, and Jiří Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(7):1409–1422, 2012. Cited on pages 14, 64 and 83.

[217] Rudolf Emil Kálmán. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. Cited on pages 22 and 37.

[218] Ahmed T. Kamal, Jay A. Farrell, and Amit K. Roy-Chowdhury. Information Consensus for Distributed Multi-Target Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 19, 20 and 23.

[219] Rangachar Kasturi, Dmitry Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John Garofolo, Rachel Bowers, Matthew Boonstra, Valentina Korzhova, and Jing Zhang. Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(2):319–336, 2009. Cited on page 100.

[220] Vera Kettnaker and Ramin Zabih. Bayesian Multi-camera Surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999. Cited on pages 20 and 23.

[221] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A Multi-cut Formulation for Joint Segmentation and Tracking of Multiple Objects. *arXiv CoRR*, abs/1607.06317, 2016. URL `https://arxiv.org/abs/1607.06317`. Cited on page 24.

[222] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Andrew D. Bagdanov, Maria Vanrell, and Antonio M. Lopez. Color Attributes for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on page 29.

[223] Fahad Shahbaz Khan, Joost van de Weijer, and Maria Vanrell. Modulating Shape Features by Color Attention for Object Recognition. *International Journal of Computer Vision (IJCV)*, 98(1):49–64, 2012. Cited on page 29.

[224] Rahat Khan, Joost van de Weijer, Fahad Shahbaz Khan, Damien Muselet, Christophe Ducottet, and Cecile Barat. Discriminative Color Descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on page 29.

[225] Saad M. Khan and Mubarak Shah. A Multiview Approach to Tracking People in Crowded Scenes using a Planar Homography Constraint. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006. Cited on pages 20, 21 and 23.

[226] Saad M. Khan and Mubarak Shah. Tracking Multiple Occluding People by Localizing on Multiple Scene Planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(3):505–519, 2009. Cited on pages 19, 20, 21 and 23.

[227] Sohaib Khan, Omar Javed, Zeeshan Rasheed, and Mubarak Shah. Human Tracking in Multiple Cameras. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001. Cited on pages 20 and 23.

[228] Zia Khan, Tucker Balch, and Frank Dellaert. MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(11):1805–1819, 2005. Cited on pages 19 and 69.

[229] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple Hypothesis Tracking Revisited. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on page 24.

[230] Genshiro Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics (JCGS)*, 5(1):1–25, 1996. Cited on page 22.

[231] Tobias Klinger, Franz Rottensteiner, and Christian Heipke. Probabilistic Multi-Person Tracking using Dynamic Bayes Networks. In *Proceedings of the ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. Cited on page 111.

[232] Tobias Klinger, Franz Rottensteiner, and Christian Heipke. Probabilistic Multi-Person Localisation and Tracking in Image Sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127(Geospatial Week 2015):73–88, 2017. Cited on pages 20, 24, 111, 112, 114, 150 and 151.

[233] Stefan Kluckner, Thomas Mauthner, and Horst Bischof. A Covariance Approximation on Euclidean Space for Visual Tracking. In *Proceedings of the Workshop of the Austrian Association for Pattern Recognition (AAPR)*, 2009. Cited on page 28.

[234] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust Multiple Car Tracking with Occlusion Reasoning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1994. Cited on page 19.

[235] Adam R. Kosiorek, Alex Bewley, and Ingmar Posner. Hierarchical Attentive Recurrent Tracking. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2017. Cited on pages 14 and 15.

[236] Matej Kristan, Janez Perš, Matej Perše, and Stanislav Kovačič. Closed-World Tracking of Multiple Interacting Targets for Indoor-Sports Applications. *Computer Vision and Image Understanding (CVIU)*, 113(5):598–611, 2009. Cited on pages 3, 19, 20, 23 and 69.

[237] Matej Kristan, Stanislav Kovačič, Aleš Leonardis, and Janez Perš. A Two-Stage Dynamic Model for Visual Tracking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (TSMCB)*, 40(6):1505–1520, 2010.

[238] Matej Kristan, Roman Pflugfelder, Aleš Leonardis, Jiří Matas, Fatih Porikli, Luka Čehovin, Georg Nebehay, Gustavo Fernandez, Tomáš Vojíř, *et al.* The Visual Object Tracking VOT2013 challenge results. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2013. Cited on pages 14, 16, 26, 64, 65, 68, 72, 82, 95, 135 and 136.

[239] Matej Kristan, Roman Pflugfelder, Aleš Leonardis, Jiří Matas, Luka Čehovin, Georg Nebehay, Tomáš Vojíř, Gustavo Fernández, *et al.* The Visual Object Tracking VOT2014 challenge results. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ECCV)*, 2014. Cited on pages 17, 18, 26, 27, 32, 36, 65, 68, 82, 95, 135 and 137.

[240] Matej Kristan, Jiří Matas, Aleš Leonardis, Michael Felsberg, Luka Čehovin, Gustavo Fernández, Tomáš Vojíř, Gustav Häger, Georg Nebehay, Roman Pflugfelder, *et al.* The Visual Object Tracking VOT2015 challenge results. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2015. Cited on pages 65 and 85.

[241] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomáš Vojiř, Gustav Häger, Alan Lukežič, Gustavo Fernández, *et al.* The Visual Object Tracking VOT2016 challenge results. In *Proceedings of the IEEE Workshop*

on Visual Object Tracking Challenge (VOT, in conjunction with ECCV), 2016. Cited on pages 14, 65, 68, 69, 72, 82, 85, 95, 135, 138 and 140.

[242] Matej Kristan, Jiří Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernández, Georg Nebehay, Fatih Proikli, and Luka Čehovin. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(11):2137–2155, 2016. Cited on pages 9, 65 and 71.

[243] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomáš Vojíř, Gustav Häger, Alan Lukežič, Abdelrahman Eldesokey, Gustavo Fernández, *et al.* The Visual Object Tracking VOT2017 challenge results. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2017. Cited on pages 14, 16, 17, 18, 64 and 69.

[244] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Everest Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2012. Cited on pages 13 and 51.

[245] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-Camera Multi-Person Tracking for EasyLiving. In *Proceedings of the IEEE International Workshop on Visual Surveillance (VS)*, 2000. Cited on pages 20 and 21.

[246] Bhagavatula Vijaya Kumar, Abhijit Mahalanobis, and Richard D. Juday. *Correlation Pattern Recognition*. Cambridge University Press, first edition, 2005. Cited on page 13.

[247] Cheng-Hao Kuo and Ramakant Nevatia. How does Person Identity Recognition help Multi-Person Tracking? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 20, 21, 48 and 50.

[248] Junseok Kwon and Kyoung Mu Lee. Tracking of a Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on pages 14, 16, 28, 64 and 69.

[249] Junseok Kwon and Kyoung Mu Lee. Visual Tracking Decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on pages 14, 15, 28, 64, 66, 86, 89, 90, 91, 92 and 95.

[250] Junseok Kwon and Kyoung Mu Lee. Tracking by Sampling Trackers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 14, 16, 86, 89, 90, 91, 92 and 95.

[251] Junseok Kwon and Kyoung Mu Lee. Highly Non-Rigid Object Tracking via Patch-based Dynamic Appearance Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(10):2427–2441, 2013. Cited on pages 14, 16, 28, 64 and 69.

[252] Julia A. Lasserre, Christopher M. Bishop, and Thomas P. Minka. Principled Hybrids of Generative and Discriminative Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on page 12.

[253] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *Proceedings of the Workshop on Modeling, Simulation and Visual Analysis of Large Crowds (in conjunction with ICCV)*, 2011. Cited on pages 98, 111, 112, 149, 150, 151 and 152.

[254] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 20 and 21.

[255] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an Image-based Motion Context for Multiple People Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on pages 111, 112, 150 and 151.

[256] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv CoRR*, abs/1504.01942, 2015. URL `http://arxiv.org/abs/1504.01942`. Cited on pages 19, 23, 97, 98, 100, 102, 111, 112, 144, 145, 148, 150 and 151.

[257] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by Tracking: Siamese CNN for Robust Target Association. In *Proceedings of the Workshop on Deep Vision: Deep Learning in Computer Vision (DVW, in conjunction with CVPR)*, 2016. Cited on pages 20 and 21.

[258] Karel Lebeda, Simon Hadfield, Jiří Matas, and Richard Bowden. Tracking the Untrackable: How to Track When Your Object is Featureless. In *Proceedings of the Workshop on Detection and Tracking Challenging Environments (DTCE, in conjunction with ACCV)*, 2012. Cited on pages 14 and 15.

[259] Karel Lebeda, Simon Hadfield, Jiří Matas, and Richard Bowden. Long-Term Tracking Through Failure Cases. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ICCV)*, 2013. Cited on pages 14 and 15.

[260] Bastian Leibe, Konrad Schindler, and Luc van Gool. Coupled Detection and Trajectory Estimation for Multi-Object Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. Cited on pages 20, 21, 22 and 23.

[261] Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc van Gool. Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(10):1683–1698, 2008. Cited on pages 19, 20, 21 and 23.

[262] Christian Leistner, Amir Saffari, and Horst Bischof. MIForests: Multiple Instance Learning with Randomized Trees. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. Cited on page 15.

[263] Philip Lenz, Andreas Geiger, and Raquel Urtasun. FollowMe: Efficient Online Min-Cost Flow Tracking with Bounded Memory and Computation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 20, 21 and 22.

[264] Evgeny Levinkov, Jonas Uhrig, Siyu Tang, Mohamed Omran, Eldar Insafutdinov, Alexander Kirillov, Carsten Rother, Thomas Brox, Bernt Schiele, and Bjoern Andres. Joint Graph Decomposition and Node Labeling: Problem, Algorithms, Applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 20, 21 and 24.

[265] Annan Li, Min Lin, Yi Wu, Ming-Hsuan Yang, and Shuicheng Yan. NUS-PRO: A New Visual Tracking Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(2):335–349, 2016. Cited on pages 14 and 64.

[266] Hanxi Li, Chunhua Shen, and Qinfeng Shi. Real-time visual tracking using compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 14, 15 and 69.

[267] Longzhen Li, Tahir Nawaz, and James M. Ferryman. PETS 2015: Dataset and Challenge. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2015. Cited on pages 17 and 97.

[268] Shuxiao Li, Ou Wu, Chengfei Zhu, and Hongxing Chang. Visual Object Tracking using Spatial Context Information and Global Tracking Skills. *Computer Vision and Image Understanding (CVIU)*, 125:1–15, 2014. Cited on page 28.

[269] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton van den Hengel. A Survey of Appearance Models in Visual Object Tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4:58):1–48, 2013. Cited on page 9.

[270] Xiaokun Li and William G. Wee. An Efficient Method for Eye Tracking and Eye-Gazed FOV Estimation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2009. Cited on page 11.

[271] Yang Li and Jianke Zhu. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In *Proceedings of the IEEE Workshop on Visual Object Tracking Challenge (VOT, in conjunction with ECCV)*, 2014. Cited on pages 13, 14, 15, 16, 82, 84, 85, 86, 87, 88 and 95.

[272] Yang Li, Jianke Zhu, and Steven C. H. Hoi. Reliable Patch Trackers: Robust Visual Tracking by Exploiting Reliable Patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 16.

[273] Yuan Li, Chang Huang, and Ramakant Nevatia. Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on pages 20, 21, 50, 100 and 149.

[274] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding Color Information for Visual Tracking: Algorithms and Benchmark. *IEEE Transactions on Image Processing (TIP)*, 24 (12):5630–5644, 2015. Cited on pages 14 and 64.

[275] Martijn Liem and Dariu M. Gavrila. Multi-Person Tracking with Overlapping Cameras in Complex, Dynamic Environments. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. Cited on pages 20, 21 and 23.

[276] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Charles Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on pages 109, 110, 117, 144, 145, 146, 147 and 148.

[277] Baiyang Liu, Junzhou Huang, Lin Yang, and Casimir Kulikowsk. Robust Tracking Using Local Sparse Appearance Model and K-Selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 14 and 15.

[278] Jia Liu, Xiaofeng Tong, Wenlong Li, Tao Wang, Yimin Zhang, Hongqi Wang, Bo Yang, Lifeng Sun, and Shiqiang Yang. Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2007. Cited on pages 20 and 21.

[279] Ting Liu, Gang Wang, and Qingxiong Yang. Real-time part-based visual tracking via adaptive correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 13.

[280] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. Cited on pages 106, 107, 108, 109, 110, 111, 144, 145, 146, 147 and 148.

[281] Stephan Liwicki, Stefano Zafeiriou, Georgios Tzimiropoulos, and Maja Pantic. Efficient Online Subspace Learning With an Indefinite Kernel for Visual Tracking and Recognition. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 23(10):1624–1636, 2012. Cited on page 15.

[282] Chen Long, Ai Haizhou, Shang Chong, Zhuang Zijie, and Bai Bo. Online Multi-Object Tracking with Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2017. Cited on pages 23 and 24.

[283] Hugh Christopher Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293:133–135, 1981. Cited on page 2.

[284] David Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. Cited on page 11.

[285] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1981. Cited on pages 2 and 13.

[286] Alan Lukežič, Tomáš Vojíř, Luka Čehovin Zajc, Jiří Matas, and Matej Kristan. Discriminative Correlation Filter with Channel and Spatial Reliability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 13, 14, 15, 16, 28 and 29.

[287] Alan Lukežič, Luka Čehovin Zajc, and Matej Kristan. Deformable Parts Correlation Filters for Robust Visual Tracking. *IEEE Transactions on Cybernetics (TCYB, formerly TSMCB)*, 2017. In press. Cited on pages 12, 13, 14 and 28.

[288] Wenhan Luo, Tae-Kyun Kim, Björn Stenger, Xiaowei Zhao, and Roberto Cipolla. Bi-label Propagation for Generic Multiple Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on page 19.

[289] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Xiaowei Zhao, and Tae-Kyun Kim. Multiple Object Tracking: A Literature Review. *arXiv CoRR*, abs/1409.7618, 2017. URL https://arxiv.org/abs/1409.7618. Cited on page 19.

[290] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical Convolutional Features for Visual Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 14 and 26.

[291] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term Correlation Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on pages 13 and 14.

[292] Cong Ma, Zhenjiang Miao, and Xiao-Ping Zhang. Saliency Prior Context Model for Visual Tracking. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2016. Cited on page 29.

[293] Emilio Maggio, Fabrizio Smeraldi, and Andrea Cavallaro. Combining Colour and Orientation for Adaptive Particle Filter-based Tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2005. Cited on page 28.

[294] Abhijit Mahalanobis, Bhagavatula Vijaya Kumar, and David Casasent. Minimum average correlation energy filters. *Applied Optics*, 26(17):3633–3640, 1987. Cited on page 13.

[295] Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification Using Intersection Kernel Support Vector Machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 105, 107, 108, 109, 110, 111, 145, 146, 147 and 148.

[296] Rok Mandeljc, Stanislav Kovačič, Matej Kristan, and Janez Perš. Non-Sequential Multi-View Detection, Localization and Identification of People using Multi-Modal Feature Maps. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2012. Cited on page 20.

[297] Rok Mandeljc, Stanislav Kovačič, Matej Kristan, and Janez Perš. Tracking by Identification Using Computer Vision and Radio. *SENSORS*, 13(1):241–273, 2013. Cited on pages 20, 23 and 97.

[298] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, first edition, 2008. Cited on page 33.

[299] Mario Edoardo Maresca and Alfredo Petrosino. MATRIOSKA: A Multi-level Approach to Fast Tracking by Learning. In *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, 2013. Cited on page 16.

[300] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc van Gool. Traffic Sign Recognition – How far are we from the solution? In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 2013. Cited on page 51.

[301] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The Template Update Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(6):810–815, 2004. Cited on pages 15 and 78.

[302] Thomas Mauthner and Horst Bischof. A Robust Multiple Object Tracking for Sport Applications. In *Proceedings of the Workshop of the Austrian Association for Pattern Recognition (AAPR)*, 2007. Cited on page 19.

[303] Nigel J. B. McFarlane and Charles Patrick Schofield. Segmentation and Tracking of Piglets in Images. *Machine Vision and Applications (MVA)*, 8(3):187–193, 1995. Cited on page 19.

[304] Stephen J. McKenna, Yogesh Raja, and Shaogang Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing (IVC)*, 17(1):225–231, 1999. Cited on page 28.

[305] Xue Mei and Haibin Ling. Robust Visual Tracking using ℓ_1 Minimization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. Cited on pages 14 and 15.

[306] Xue Mei, Haibin Ling, Yi Wu, Erik Blasch, and Li Bai. Minimum Error Bounded Efficient ℓ_1 Tracker with Occlusion Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 14 and 15.

[307] Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and Trajectory-Level Exclusion in Multiple Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 20, 22, 50, 149, 150 and 151.

[308] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous Energy Minimization for Multi-Target Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(1):58–72, 2014. Cited on pages 19, 20, 21, 48, 50, 98, 100, 111, 149, 150 and 151.

[309] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A Benchmark for Multi-Object Tracking. *arXiv Corr*, abs/1603.00831, 2016. URL `http://arxiv.org/abs/1603.00831`. Cited on pages 19, 24, 97 and 102.

[310] Anurag Mittal and Larry S. Davis. $\mathrm{M_2}$Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002. Cited on pages 20 and 21.

[311] Anurag Mittal and Larry S. Davis. $\mathrm{M_2}$Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision (IJCV)*, 51 (3):189–203, 2003. Cited on pages 20 and 21.

[312] Dennis Mitzel and Bastian Leibe. Taking Mobile Multi-object Tracking to the Next Level: People, Unknown Objects, and Carried Items. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on page 23.

[313] Thomas B. Moeslund and Erik Granum. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding (CVIU)*, 81(3):231–268, 2001. Cited on page 9.

[314] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU)*, 104(2–3):90–126, 2006. Cited on page 9.

[315] Matthias Mueller, Neil Smith, and Bernard Ghanem. A Benchmark and Simulator for UAV Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. Cited on page 14.

[316] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-Aware Correlation Filter Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on page 13.

[317] James Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. Cited on pages 22, 56, 100 and 106.

[318] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 11(2):300–311, 2010. Cited on page 11.

[319] Hyeonseob Nam and Bohyung Han. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 13, 14, 16, 26, 39, 45, 81, 85, 86, 87, 88, 94 and 95.

[320] Hyeonseob Nam, Seunghoon Hong, and Bohyung Han. Online Graph-Based Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on pages 14, 16, 84 and 94.

[321] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and Propagating CNNs in a Tree Structure for Visual Tracking. *arXiv CoRR*, abs/1704.06326, 2017. URL `https://arxiv.org/abs/1704.06326`. Cited on pages 14, 16, 17, 26, 39, 45, 81, 85, 87, 88, 94 and 95.

[322] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local Decorrelation for Improved Detection. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2014. Cited on pages 52, 105, 108, 109, 110, 111, 145, 146, 147 and 148.

[323] Georg Nebehay and Roman Pflugfelder. Consensus-based Matching and Tracking of Keypoints for Object Tracking. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, 2014. Cited on pages 14, 16, 84, 85, 87, 88 and 94.

[324] Georg Nebehay and Roman Pflugfelder. Clustering of Static-Adaptive Correspondences for Deformable Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on pages 14, 16 and 94.

[325] S. M. Shahed Nejhum, Jeffrey Ho, and Ming-Hsuan Yang. Visual Tracking with Histograms and Articulating Blocks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 13, 14 and 16.

[326] S. M. Shahed Nejhum, Jeffrey Ho, and Ming-Hsuan Yang. Online Visual Tracking with Histograms and Articulating Blocks. *Computer Vision and Image Understanding (CVIU)*, 114(8):901–914, 2010. Cited on pages 13, 14 and 16.

[327] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on page 117.

[328] Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2001. Cited on page 12.

[329] Peter Nillius, Josephine Sullivan, and Stefan Carlsson. Multi-Target Tracking - Linking Identities using Bayesian Network Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on pages 19 and 20.

[330] Guanghan Ning, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017. Cited on pages 14 and 15.

[331] Katja Nummiaro, Esther Koller-Meier, and Luc van Gool. Object Tracking with an Adaptive Color-Based Particle Filter. In *Proceedings of the Annual Symposium of Pattern Recognition (DAGM)*, 2002. Cited on pages 26 and 28.

[332] Katja Nummiaro, Esther Koller-Meier, and Luc van Gool. An Adaptive Color-Based Particle Filter. *Image and Vision Computing (IVC)*, 21(1):99–110, 2003. Cited on pages 26 and 28.

[333] Katja Nummiaro, Esther Koller-Meier, and Luc van Gool. Color Features for Tracking Non-Rigid Objects. *Acta Automatica Sinica (Chinese Journal of Automation)*, 29(3):345–355, 2003. Cited on page 28.

[334] Songhwai Oh, Stuart Russel, and Shankar Sastry. Markov Chain Monte Carlo Data Association for Multiple-Target Tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, 2009. Cited on page 50.

[335] Kenji Okuma, Ali Taleghani, Nando de Freitas, James J. Little, and David G. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004. Cited on pages 20, 21 and 50.

[336] Peter Olofsson and Mikael Andersson. *Probability, Statistics, and Stochastic Processes*. Wiley, second edition, 2012. Cited on page 10.

[337] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. Locally Orderless Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on page 28.

[338] Kazuhiro Otsuka and Naoki Mukawa. Multiview Occlusion Analysis for Tracking Densely Populated Objects Based on 2-D Visual Angles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. Cited on page 20.

[339] Luis Patino, Tom Cane, Alain Vallee, and James M. Ferryman. PETS 2016: Dataset and Challenge. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2016. Cited on pages 17 and 97.

[340] Luis Patino, Tahir Nawaz, Tom Cane, and James M. Ferryman. PETS 2017: Dataset and Challenge. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2017. Cited on pages 17 and 97.

[341] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc van Gool. You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking. In *Proceedings of the IEEE*

International Conference on Computer Vision (ICCV), 2009. Cited on pages 20, 21, 51, 111, 112, 150, 151 and 152.

[342] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-Based Probabilistic Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002. Cited on pages 26 and 28.

[343] Federico Pernici and Alberto Del Bimbo. Object Tracking by Oversampling Local Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(12):2538–2551, 2014. Cited on pages 14 and 16.

[344] Roman Pflugfelder and Horst Bischof. Localization and trajectory reconstruction in surveillance cameras with non-overlapping views. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(4):709–721, 2010. Cited on page 23.

[345] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 20, 21 and 22.

[346] Georg Poier, Samuel Schulter, Sabine Sternig, Peter M. Roth, and Horst Bischof. Hough Forests Revisited: An Approach to Multiple Instance Tracking from Multiple Cameras. In *Proceedings of the German Conference on Pattern Recognition (GCPR, formerly DAGM)*, 2014. Cited on pages 20, 21 and 23.

[347] Fatih Porikli. Achieving Real-Time Object Detection and Tracking Under Extreme Conditions. *Journal of Real-Time Image Processing*, 1(1):33–40, 2006. Cited on page 9.

[348] Fatih Porikli, Oncel Tuzel, and Peter Meer. Covariance Tracking using Model Update Based on Lie Algebra. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on page 28.

[349] Horst Possegger, Sabine Sternig, Thomas Mauthner, Peter M. Roth, and Horst Bischof. Robust Real-Time Tracking of Multiple Objects by Volumetric Mass Densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 19, 20, 21, 23, 26, 48 and 97.

[350] Horst Possegger, Thomas Mauthner, Peter M. Roth, and Horst Bischof. Occlusion Geodesics for Online Multi-Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on pages 26 and 49.

[351] Horst Possegger, Thomas Mauthner, and Horst Bischof. In Defense of Color-based Model-free Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on pages 14, 28, 37, 38, 42, 83, 84, 85, 87 and 88.

[352] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged Deep Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on page 14.

[353] Qian Qu and Gérard Medioni. Multiple-Target Target Tracking by Spatiotemporal Monte Carlo Markov Chain Data Association. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2196–2210, 2009. Cited on page 22.

[354] Wei Qu, Dan Schonfeld, and Magdi Mohamed. Real-Time Interactively Distributed Multi-Object Tracking Using a Magnetic-Inertia Potential Model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005. Cited on page 19.

[355] Ali Rahimi, Brian Dunagan, and Trevor Darrell. Simultaneous Calibration and Tracking with a Network of Non-Overlapping Sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. Cited on pages 20 and 23.

[356] Yogesh Raja, Stephen J. McKenna, and Shaogang Gong. Tracking and Segmenting People in Varying Lighting Conditions using Colour. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 1998. Cited on page 28.

[357] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proceedings of the Workshop on Deep Vision: Deep Learning in Computer Vision (DVW, in conjunction with CVPR)*, 2014. Cited on page 13.

[358] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 108, 109, 110, 145, 146, 147 and 148.

[359] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 13, 106, 107, 110 and 111.

[360] Donald B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. Cited on pages 22 and 50.

[361] Vladimir Reilly, Haroon Idrees, and Mubarak Shah. Detection and Tracking of Large Number of Targets in Wide Area Surveillance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. Cited on page 19.

[362] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2015. Cited on pages 13, 48, 51, 58, 105, 107, 108, 109, 110, 111, 144, 145, 146, 147 and 148.

[363] Travis Rose, Jonathan Fiscus, Paul Over, John Garofolo, and Martial Michel. The TRECVid 2008 Event Detection Evaluation. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, 2009. Cited on page 97.

[364] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision (IJCV)*, 77(1-3): 125–141, 2008. Cited on pages 14, 15, 64, 66, 83 and 84.

[365] Peter M. Roth, Sabine Sternig, Helmut Grabner, and Horst Bischof. Classifier Grids for Robust Adaptive Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on page 52.

[366] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts. In *ACM Transactions on Graphics (TOG)*, 2004. Cited on pages 41 and 81.

[367] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. Cited on pages 39 and 51.

[368] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking The Untrackable: Learning To Track Multiple Cues with Long-Term Dependencies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 20, 23, 24, 111, 112, 150 and 151.

[369] Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. On-line Random Forests. In *Proceedings of the Workshop on On-line Computer Vision (OLCV, in conjunction with ICCV)*, 2009. Cited on page 24.

[370] Jakob Santner, Markus Unger, Thomas Pock, Christian Leistner, Amir Saffari, and Horst Bischof. Interactive Texture Segmentation using Random Forests and Total Variation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. Cited on page 81.

[371] Jakob Santner, Christian Leistner, Amir Saffari, Thomas Pock, and Horst Bischof. PROST: Parallel Robust Online Simple Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Cited on pages 16 and 66.

[372] Jakob Santner, Thomas Pock, and Horst Bischof. Interactive Multi-Label Segmentation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2010. Cited on page 41.

[373] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Transactions on Signal Processing (TSP)*, 56(8): 3447–3457, 2008. Cited on page 100.

[374] Samuel Schulter, Christian Leistner, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Alternating Regression Forests for Object Detection and Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 51 and 52.

[375] Samuel Schulter, Christian Leistner, Peter M. Roth, and Horst Bischof. Accurate Object Detection with Joint Classification-Regression Random Forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on pages 51 and 52.

[376] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep Network Flow for Multi-Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 19, 20 and 21.

[377] Paul Scovanner and Marshall F. Tappen. Learning Pedestrian Dynamics from the Real World. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009. Cited on page 51.

[378] Aleksandr V. Segal and Ian Reid. Latent Data Association: Bayesian Model Selection for Multi-target Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 19, 20 and 21.

[379] Taiki Sekii. Robust, Real-Time 3D Tracking of Multiple Objects with Similar Appearances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on page 23.

[380] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. Cited on page 51.

[381] Laura Sevilla-Lara and Erik Learned-Miller. Distribution Fields for Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 14, 15 and 82.

[382] Khurram Shafique, Mun Wai Lee, and Niels Haering. A Rank Constrained Continuous Formulation of Multi-frame Multi-target Tracking Problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on page 22.

[383] Jianbo Shi and Carlo Tomasi. Good Features to Track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994. Cited on page 13.

[384] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. Part-based Multiple-Person Tracking with Partial Occlusion Handling. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2012. Cited on pages 20, 21, 22, 26 and 97.

[385] Tomáš Sixta and Boris Flach. Multiple Object Segmentation and Tracking by Bayes Risk Minimization. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2016. Cited on page 19.

[386] Arnold W. M. Smeulders, Dung M. Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual Tracking: an Experimental Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1442–1468, 2014. Cited on pages 14, 26, 64, 69 and 71.

[387] Kevin Smith, Daniel Gatica-Perez, and Jean-Marc Odobez. Using Particles to Track Varying Numbers of Interacting People. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. Cited on pages 20 and 23.

[388] Kevin Smith, Daniel Gatica-Perez, Jean-Marc Odobez, and Sileye Ba. Evaluating Multi-Object Tracking. In *Proceedings of the Workshop on Empirical Evaluation Methods in Computer Vision (EEMCV, in conjunction with CVPR)*, 2005. Cited on page 100.

[389] Francesco Solera, Simone Calderara, and Rita Cucchiara. Learning to Divide and Conquer for Online Multi-Target Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 20 and 23.

[390] Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K. Roy-Chowdhury. A Stochastic Graph Evolution Framework for Robust Multi-target Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. Cited on pages 20 and 21.

[391] Hyun Oh Song, Ross Girshick, Stefan Zickler, Christopher Geyer, Pedro Felzenszwalb, and Trevor Darrell. Generalized Sparselet Models for Real-Time Multiclass Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(5):1001–1012, 2015. Cited on page 51.

[392] Shuran Song and Jianxiong Xiao. Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 14 and 64.

[393] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson W. H. Lau, and Ming-Hsuan Yang. CREST: Convolutional Residual Learning for Visual Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 12, 13, 14 and 26.

[394] Milan Šonka, Václav Hlaváč, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. CL Engineering, third edition, 2007. Cited on page 10.

[395] Christian Soto, Bi Song, and Amit K. Roy-Chowdhury. Distributed Multi-Target Tracking In A Self-Configuring Camera Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on pages 20 and 23.

[396] Severin Stalder, Helmut Grabner, and Luc van Gool. Cascaded Confidence Filtering for Improved Tracking-by-Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. Cited on pages 20 and 21.

[397] Chris Stauffer and W. Eric L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999. Cited on page 15.

[398] Chris Stauffer and W. Eric L. Grimson. Learning Patterns of Activity using Real-Time Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22 (8):747–757, 2000. Cited on page 15.

[399] Sabine Sternig, Peter M. Roth, and Horst Bischof. Inverse Multiple Instance Learning for Classifier Grids. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2010. Cited on page 52.

[400] Sabine Sternig, Thomas Mauthner, Arnold Irschara, Peter M. Roth, and Horst Bischof. Multi-camera Multi-object Tracking by Robust Hough-based Homography Projections. In *Proceedings of the IEEE International Workshop on Visual Surveillance (VS)*, 2011. (in conjunction with CVPR). Cited on pages 20 and 23.

[401] Sabine Sternig, Peter M. Roth, and Horst Bischof. On-line Inverse Multiple Instance Boosting for Classifier Grids. *Pattern Recognition Letters (PRL)*, 33(7):890–897, 2011. Cited on page 52.

[402] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, John Garofolo, Djamel Mostefa, and Padmanabhan Soundararajan. The CLEAR 2006 Evaluation. In *Proceedings of the International Evaluation Conference on Classification of Events, Activities and Relationships (CLEAR)*, 2006. Cited on pages 97, 100 and 101.

[403] Thomas M. Strat and Martin A. Fischler. Context-Based Vision: Recognizing Objects Using Information from Both 2-D and 3-D Imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 13(10):1050–1065, 1991. Cited on page 3.

[404] Josephine Sullivan, Andrew Blake, Michael Isard, and John Philip MacCormick. Object localization by Bayesian correlation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1999. Cited on page 28.

[405] Xin Sun, Ngai-Man Cheung, Hongxun Yao, and Yiluan Guo. Non-Rigid Object Tracking via Deformable Patches using Shape-Preserved KCF and Level Sets. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on pages 13 and 26.

[406] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jon Shlens. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 108, 109, 110, 145, 146, 147 and 148.

[407] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017. Cited on pages 108, 110, 144, 146, 147 and 148.

[408] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, first edition, 2010. Cited on page 10.

[409] Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele. Detection and Tracking of Occluded People. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012. Cited on page 110.

[410] Siyu Tang, Mykhaylo Andriluka, Anton Milan, Konrad Schindler, Stefan Roth, and Bernt Schiele. Learning People Detectors for Tracking in Crowded Scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on pages 20, 21 and 110.

[411] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Subgraph Decomposition for Multi-Target Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 22.

[412] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple People Tracking by Lifted Multicut and Person Re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 20, 22 and 24.

[413] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese Instance Search for Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 14, 15, 39 and 45.

[414] Zhu Teng, Junliang Xing, Qiang Wang, Congyan Lang, Songhe Feng, and Yi Jin. Robust Object Tracking based on Temporal and Spatial Deep Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on page 26.

[415] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991. Cited on page 13.

[416] Zhuowen Tu. Learning Generative Models via Discriminative Approaches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. Cited on page 12.

[417] Engin Türetken, Xinchao Wang, Carlos J. Becker, Carsten Haubold, and Pascal Fua. Network Flow Integer Programming to Track Elliptical Cells in Time-Lapse Sequences. *IEEE Transactions on Medical Imaging (TMI)*, 36(4):942–951, 2017. Cited on page 19.

[418] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013. Cited on page 51.

[419] Shimon Ullman. The Interpretation of Structure from Motion. *Proceedings of the Royal Society of London, B*, 203(1153):405–426, 1979. Cited on page 2.

[420] Markus Unger, Thomas Pock, Werner Trobin, Daniel Cremers, and Horst Bischof. TVSeg - Interactive Total Variation Based Image Segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2008. Cited on pages 41 and 81.

[421] Jack Valmadre, Luca Bertinetto, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. End-to-end representation learning for Correlation Filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 14 and 15.

[422] Joost van de Weijer and Cordelia Schmid. Coloring Local Feature Extraction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006. Cited on page 29.

[423] Joost van de Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning Color Names for Real-World Applications. *IEEE Transactions on Image Processing (TIP)*, 18(7): 1512–1524, 2009. Cited on pages 26, 29 and 82.

[424] Vladimir Naumovich Vapnik. *Statistical Learning Theory*. Wiley, first edition, 1998. Cited on pages 10 and 12.

[425] Jaco Vermaak, Arnaud Doucet, and Patrick Pérez. Maintaining Multi-Modality through Mixture Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003. Cited on pages 19 and 50.

[426] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. Cited on pages 37 and 52.

[427] Paul Viola, Michael Jones, and Daniel Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003. Cited on page 52.

[428] Tomáš Vojíř and Jiří Matas. Robustifying the Flock of Trackers. In *Proceedings of the Computer Vision Winter Workshop (CVWW)*, 2011. Cited on pages 14, 16, 82, 83, 84, 85, 87, 88 and 95.

[429] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Online Object Tracking with Sparse Prototypes. *IEEE Transactions on Image Processing (TIP)*, 22(1):314–325, 2013. Cited on page 66.

[430] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual Tracking with Fully Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on pages 13 and 14.

[431] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. STCT: Sequentially Training Convolutional Networks for Visual Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 13 and 14.

[432] Lituan Wang, Lei Zhang, and Zhang Yi. Trajectory Predictor by Using Recurrent Neural Networks in Visual Tracking. *IEEE Transactions on Cybernetics (TCYB, formerly TSMCB)*, 47(10):3172–3183, 2017. Cited on pages 14 and 15.

[433] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia. Understanding and Diagnosing Visual Tracking Systems. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. Cited on page 16.

[434] Shaofei Wang and Charless C. Fowlkes. Learning Optimal Parameters for Multi-target Tracking with Contextual Interactions. *International Journal of Computer Vision (IJCV)*, 122(3): 484–501, 2016. Cited on pages 20 and 21.

[435] Xiaogang Wang. Intelligent Multi-Camera Video Surveillance: A Review. *Pattern Recognition Letters (PRL)*, 34(1):3–19, 2013. Cited on page 19.

[436] Xinchao Wang, Engin Türetken, François Fleuret, and Pascal Fua. Tracking Interacting Objects Optimally Using Integer Programming. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on page 20.

[437] Xinchao Wang, Engin Türetken, François Fleuret, and Pascal Fua. Tracking Interacting Objects Using Intertwined Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(11):2312–2326, 2016. Cited on pages 20, 21 and 23.

[438] Longyin Wen, Zhaowei Cai, Zhen Lei, Dong Yi, and Stan Z. Li. Online Spatio-temporal Structural Context Learning for Visual Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on page 29.

[439] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z. Li. Multiple Target Tracking based on Undirected Hierarchical Relation Hypergraph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on page 100.

[440] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-Camera Multi-Target Tracking with Space-Time-View Hyper-graph. *International Journal of Computer Vision (IJCV)*, 122(2):313–333, 2017. Cited on pages 111, 112, 150 and 151.

[441] Juyang Weng, Paul Cohen, and Marc Herniou. Camera Calibration with Distortion Models and Accuracy Evalution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(10):965–980, 1992. Cited on page 53.

[442] Oliver Williams, Andrew Blake, and Roberto Cipolla. A Sparse Probabilistic Learning Algorithm for Real-Time Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003. Cited on pages 2 and 16.

[443] Paul Wohlhart, Samuel Schulter, Martin Koestinger, Peter M. Roth, and Horst Bischof. Discriminative Hough Forests for Object Detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012. Cited on page 51.

[444] Nicolai Wojke and Dietrich Paulus. Global Data Association for the Probability Hypothesis Density Filter using Network Flows. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016. Cited on pages 20, 21 and 24.

[445] Thomas Woodley, Bjorn Stenger, and Roberto Cipolla. Tracking Using Online Feature Selection and a Local Generative Model. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2007. Cited on page 12.

[446] Bo Wu and Ramakant Nevatia. Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. Cited on pages 100 and 101.

[447] Bo Wu and Ramakant Nevatia. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision (IJCV)*, 75(2):247–266, 2007. Cited on pages 19, 21, 22 and 50.

[448] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online Object Tracking: A Benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 14, 64, 65, 66 and 69.

[449] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1834–1848, 2015. Cited on pages 14, 64, 65, 68, 69, 72, 86, 89, 90, 91, 92, 95, 135 and 142.

[450] Zheng Wu, Nickolay I. Hristov, Tyson L. Hedrick, Thomas H. Kunz, and Margrit Betke. Tracking a Large Number of Objects from Multiple Views. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009. Cited on page 19.

[451] Zheng Wu, Thomas H. Kunz, and Margrit Betke. Efficient Track Linking Methods for Track Graphs using Network-flow and Set-cover Techniques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on page 22.

[452] Zheng Wu, Ashwin Thangali, Stan Sclaroff, and Margrit Betke. Coupling Detection and Data Association for Multiple Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 20 and 21.

[453] Zheng Wu, Jianming Zhang, and Margrit Betke. Online Motion Agreement Tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2013. Cited on pages 149, 150, 151 and 152.

[454] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to Track: Online Multi-Object Tracking by Decision Making. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on pages 20 and 21.

[455] Jingjing Xiao, Rustam Stolkin, and Aleš Leonardis. Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 15.

[456] Junliang Xing, Haizhou Ai, and Shihong Lao. Multi-Object Tracking through Occlusions by Local Tracklets Filtering and Global Tracklets Association with Detection Responses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Cited on page 22.

[457] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who are you with and Where are you going? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on pages 51 and 152.

[458] Bo Yang and Ramakant Nevatia. Multi-Target Tracking by Online Learning of Non-linear Motion Patterns and Robust Appearance Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on page 20.

[459] Bo Yang and Ramakant Nevatia. Online Learned Discriminative Part-Based Appearance Models for Multi-Human Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on page 20.

[460] Bo Yang and Ramakant Nevatia. An Online Learned CRF Model for Multi-Target Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on page 22.

[461] Bo Yang and Ramakant Nevatia. Multi-Target Tracking by Online Learning a CRF Model of Appearance and Motion Patterns. *International Journal of Computer Vision (IJCV)*, 107 (2):203–217, 2014. Cited on page 19.

[462] Bo Yang, Chang Huang, and Ramakant Nevatia. Learning Affinities and Dependencies for Multi-Target Tracking using a CRF Model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. Cited on page 22.

[463] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. Robust Superpixel Tracking. *IEEE Transactions on Image Processing (TIP)*, 23(4):1639–1651, 2014. Cited on page 66.

[464] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011. Cited on page 9.

[465] Jun Yang, Patricio A. Vela, Zhongke Shi, and Jochen Teizer. Probabilistic Multiple People Tracking through Complex Situations. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2009. Cited on page 149.

[466] Ming Yang, Ting Yu, and Ying Wu. Game-Theoretic Multiple Target Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. Cited on pages 20 and 21.

[467] Ming Yang, Ying Wu, and Gang Hua. Context-Aware Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(7):1195–1209, 2009. Cited on pages 14 and 29.

[468] Kwang Moo Yi, Hawook Jeong, Byeongho Heo, Hyung Jin Chang, and Jin Young Choi. Initialization-Insensitive Visual Tracking Through Voting with Salient Local Features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. Cited on page 14.

[469] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object Tracking: A Survey. *ACM Journal of Computing Surveys (CSUR)*, 38(4):Article No. 13, 2006. Cited on page 9.

[470] Dong Hyun Yoo and Myung Jin Chung. A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Computer Vision and Image Understanding (CVIU)*, 98(1):25–51, 2005. Cited on page 11.

[471] David P. Young and James M. Ferryman. PETS Metrics: On-Line Performance Evaluation Service. In *Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VSPETS)*, 2005. Cited on page 97.

[472] Qian Yu, Gérard Medioni, and Isaac Cohen. Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. Cited on pages 19, 20 and 22.

[473] Xianguo Yu and Qifeng Yu. Online Structural Learning with Dense Samples and a Weighting Kernel. *Pattern Recognition Letters (PRL)*, 2017. In press. Cited on page 17.

[474] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on pages 14, 15, 26 and 46.

[475] Luka Čehovin Zajc, Alan Lukežič, Aleš Leonardis, and Matej Kristan. Beyond standard benchmarks: Parameterizing performance evaluation in visual object tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. Cited on page 64.

[476] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 20, 21, 62, 149 and 152.

[477] Beibei Zan, Dorothy N. Monekosso, Paolo Remagnino, Sergio A. Velastin, and Li-Qun Xu. Crowd Analysis: A Survey. *Machine Vision and Applications (MVA)*, 19(5):345–357, 2008. Cited on page 19.

[478] Harry Zhang. The Optimality of Naive Bayes. In *Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2004. Cited on page 12.

[479] Jianming Zhang, Liliana Lo Presti, and Stan Sclaroff. Online Multi-Person Tracking by Tracker Hierarchy. In *Proceedings of the IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, 2012. Cited on pages 20 and 21.

[480] Jianming Zhang, Shugao Mao, and Stan Sclaroff. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on pages 14, 16 and 28.

[481] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time Compressive Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. Cited on pages 14, 15, 69, 83 and 95.

[482] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang. Fast Visual Tracking via Dense Spatio-Temporal Context Learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on page 29.

[483] Li Zhang, Yuan Li, and Ramakant Nevatia. Global Data Association for Multi-Object Tracking Using Network Flows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. Cited on pages 20, 21, 22, 50 and 152.

[484] Lu Zhang and Laurens van der Maaten. Structure Preserving Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. Cited on pages 16, 19, 20, 21 and 29.

[485] Lu Zhang and Laurens van der Maaten. Preserving Structure in Model-Free Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(4):756–769, 2014. Cited on pages 19, 20, 21, 27 and 29.

[486] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Filtered Channel Features for Pedestrian Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 51.

[487] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. How Far are We from Solving Pedestrian Detection? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on page 52.

[488] Shengping Zhang, Hongxun Yao, Xin Sun, and Xiusheng Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition (PR)*, 46(7):1772–1788, 2013. Cited on pages 9 and 15.

[489] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust Visual Tracking via Multi-Task Sparse Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Cited on pages 14 and 15.

[490] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust Visual Tracking via Structured Multi-Task Sparse Learning. *International Journal of Computer Vision (IJCV)*, 101(2):367–383, 2013. Cited on pages 14 and 15.

[491] Tianzhu Zhang, Bernard Ghanem, Si Liu, Changsheng Xu, and Narendra Ahuja. Robust Visual Tracking via Exclusive Context Modeling. *IEEE Transactions on Cybernetics (TCYB, formerly TSMCB)*, 46(1):51–63, 2016. Cited on page 29.

[492] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task Correlation Particle Filter for Robust Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Cited on page 14.

[493] Tao Zhao and Ramakant Nevatia. Tracking Multiple Humans in Crowded Environment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. Cited on pages 20 and 21.

[494] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust Object Tracking via Sparse Collaborative Appearance Model. *IEEE Transactions on Image Processing (TIP)*, 23(5):2356–2368, 2014. Cited on pages 86, 89, 90, 91, 92 and 95.

[495] Gao Zhu, Fatih Porikli, and Hongdong Li. Beyond Local Search: Tracking Objects Everywhere with Instance-Specific Proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Cited on pages 14, 15, 16, 17, 85, 87 and 88.

[496] Guibo Zhu, Jinqiao Wang, Yi Wu, and Hanqing Lu. Collaborative Correlation Tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015. Cited on page 14.

[497] Guibo Zhu, Jinqiao Wang, Chaoyang Zhao, and Hanqing Lu. Weighted Part Context Learning for Visual Tracking. *IEEE Transactions on Image Processing (TIP)*, 24(12):5140–5151, 2015. Cited on page 29.

[498] Charles Lawrence Zitnick and Piotr Dollár. Edge Boxes: Locating Object Proposals from Edges. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. Cited on page 51.

[499] Zoran Zivkovic and Ben Kröse. An EM-like algorithm for color-histogram-based object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. Cited on page 28.

[500] Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. Cited on pages 108, 109, 145, 146, 147 and 148.