Online Multi-Person Tracking-by-Detection from a Single, Uncalibrated Camera

Michael D. Breitenstein, *Student Member, IEEE,* Fabian Reichlin, Bastian Leibe, *Member, IEEE,* Esther Koller-Meier, *Member, IEEE,* and Luc Van Gool, *Member, IEEE*

Abstract—In this paper, we address the problem of automatically detecting and tracking a variable number of persons in complex scenes using a monocular, potentially moving, uncalibrated camera. We propose a novel approach for multi-person tracking-by-detection in a particle filtering framework. In addition to final high-confidence detections, our algorithm uses the continuous confidence of pedestrian detectors and online trained, instance-specific classifiers as a graded observation model. Thus, generic object category knowledge is complemented by instance-specific information. The main contribution of this paper is to explore how these unreliable information sources can be used for robust multi-person tracking. The algorithm detects and tracks a large number of dynamically moving persons in complex scenes with occlusions, does not rely on background modeling, requires no camera or ground plane calibration, and only makes use of information from the past. Hence, it imposes very few restrictions and is suitable for online applications. Our experiments show that the method yields good tracking performance in a large variety of highly dynamic scenarios, such as typical surveillance videos, webcam footage, or sports sequences. We demonstrate that our algorithm outperforms other methods that rely on additional information. Furthermore, we analyze the influence of different algorithm components on the robustness.

Index Terms—Multi-object tracking, tracking-by-detection, detector confidence particle filter, pedestrian detection, particle filtering, sequential monte carlo estimation, online learning, detector confidence, surveillance, sports analysis, traffic safety

1 Introduction

New video cameras are installed daily all around the world, as webcams, for surveillance, or for a multitude of other purposes. As this happens, it becomes increasingly important to develop methods that process such data streams automatically and in real-time, reducing the manual effort that is still required for video analysis. Of particular interest for many applications is the behavior of persons, *e.g.*, for traffic safety, surveillance, or sports analysis. As most tasks at semantically higher levels are based on trajectory information, it is crucial to robustly detect and track people in dynamic and complex real-world scenes. However, most existing multi-person tracking methods are still limited to special application scenarios. They require either multi-camera input, scene-specific knowledge, a static background, or depth information, or are not suitable for online processing.

In this paper, we address the problem of automatically detecting and tracking a variable number of targets in complex scenes from a single, potentially moving, uncalibrated camera, using a causal (or online) approach. This problem is very challenging, because there are many sources of uncertainty for the object locations such as measurement noise, clutter, changing background, and significant occlusions.

- *M. D. Breitenstein is with the Computer Vision Laboratory, ETH Zurich. E-mail: breitenstein@vision.ee.ethz.ch*
- *F. Reichlin is with LiberoVision AG, Zurich.*
- *B. Leibe is with the Mobile Multimedia Processing group, UMIC Research Centre, RWTH Aachen University.*
- *E. Koller-Meier is with the Computer Vision Laboratory, ETH Zurich.*
- *L. Van Gool is with the Computer Vision Laboratory, ETH Zurich, and with ESAT-PSI/IBBT, KU Leuven.*

In order to cope with those difficulties, *tracking-by-detection* approaches have become increasingly popular, driven by the recent progress in object detection. Such methods involve the continuous application of a detection algorithm in individual frames and the association of detections across frames. In contrast to *background modeling*-based trackers, they are generally robust to changing background and moving cameras.

The main challenge when using an object detector for tracking is that the detector output is unreliable and sparse, *i.e.*, detectors only deliver a discrete set of responses and usually yield false positives and missing detections. Thus, the resulting association problem between detections and targets is difficult. Several recent algorithms address this problem by optimizing detection assignments over a large temporal window in an offline step [1], [3], [26], [30]. They use information from future frames and locate the targets in the current frame with a temporal delay or after the entire sequence has been observed. In contrast, Sequential Monte Carlo methods offer a framework for representing the tracking uncertainty in a *causal* manner. By only considering information from past frames, such approaches are more suitable for time-critical, online applications.

Although a few methods exist for online *multi-target* tracking-by-detection, they rely only on the final, sparse output from the object detector [7], [33], [45]. In contrast, our approach is based on monitoring its *continuous detection confidence* and using this as a graded observation model. The intuition is that by forgoing the hard detection decision, we can render tracking more robust. Although such a combination appears desirable, available object detectors have only been optimized for accurate results at those locations passing the

final non-maximum suppression stage. This said, it is not guaranteed that the shape of the confidence volume in-between those locations will support tracking. In particular, a majority of the densities' local maxima correspond to false positives that may deteriorate the tracking results, especially during occlusions and when several interacting targets are present.

The main contribution of our work is the exploration how this unreliable information source can be used for robust *multi-person* tracking. Our algorithm achieves this robustness through a careful interplay between object detection, classification, and target tracking components. Typically, a *bottom-up* process deals with target representation and localization, trying to cope with changes in the appearance of the tracked targets, and a *top-down* process performs data association and filtering to deal with object dynamics. Correspondingly, our approach is based on a combination of a general, class-specific *pedestrian detector* to localize people and a *particle filter* to predict the target locations, incorporating a motion model. To complement the generic object category knowledge from the detector, our algorithm trains *person-specific classifiers* during run-time to distinguish between the tracking targets.

This paper makes the following contributions:

1) We combine a generic class-specific object detector and particle filtering for robust multi-person tracking suitable for online applications. The algorithm addresses the specific problems caused by the unreliable output from object detectors and the presence of multiple, possibly interacting targets.
2) To handle false positive detections, we learn target-specific classifiers at run-time, which are used to select high-confidence detections and associate them to targets.
3) To handle missing detections, we exploit the continuous confidence density output of detectors and classifiers.
4) We analyze and discuss the robustness of the method, in particular the influence of each part of the algorithm.
5) We experimentally validate our method on a large variety of highly dynamic scenarios. We quantitatively compare our method to other algorithms and demonstrate that ours outperforms several state-of-the-art algorithms that require multi-camera setups, scene knowledge, non-causal processing, or that rely on object detectors that are specifically trained for a specific application.

In contrast to our previous work [5], [6], we increase the robustness of the tracker by detecting re-appearing persons that temporally left the scene. Second, we discuss how the different observation model terms assist in handling difficult situations, and we quantitatively evaluate the influence of these terms. Third, we show additional results and experiments. Additionally, we provide a more comprehensive description of the algorithm as well as implementation details.

The paper is structured as follows. After discussing related work in the following section, Section 3 describes the algorithm and several important design choices. Section 4 presents a quantitative evaluation on a large variety of datasets and a comparison to other algorithms. In Section 5, the robustness of the observation model is discussed in detail. Section 6 concludes the paper with a summary and outlook.

2 Related Work

Particle Filtering. Particle filters were introduced to the vision community to estimate the multi-modal distribution of a target's state space [19]. Other researchers extended the framework for multiple targets by either representing all targets jointly in a particle filter [43] or by extending the state space of each target to include components of other targets [41]. In the first approach, a fixed number of particles represent a varying number of targets. Hence, new targets have to "steal" particles from existing trackers, reducing the accuracy of the approximation. In the second approach, the state space becomes increasingly large, which may require a very large number of particles for a good representation. Thus, the computational complexity increases exponentially with the number of targets. To overcome these problems, most methods employ one particle filter per target using a small state space and deal with interacting targets separately [21], [24], [38].

Tracking-by-Detection. While many tracking methods rely on background subtraction from one or several static cameras [3], [20], [24], [42], [49], recent progress in object detection has stimulated the interest in combining tracking and detection. In contrast to data association based tracking approaches, which link detection responses to trajectories by global optimization based on position, size and appearance similarity [1], [3], [18], [26], [30], [36], [48], the combination of object detectors and particle filtering results in algorithms that are more suitable for time-critical, online applications.

To this end, Okuma *et al.* [33] combine the algorithm of Vermaak *et al.* [43] with a boosted object detector. Cai *et al.* [7] extend this boosted particle filter using independent particle sets for each target to increase the robustness for multiple targets. Additionally, to handle occlusions more robustly, other researchers use 3D information [11], [15], train detectors for individual body parts [45], or apply application-specific motion models [35]. However, all of those approaches have in common that they rely only on the final, sparse output from the object detector. On the other hand, state-of-the-art object detectors all build up some form of confidence density as one stage of their pipeline, which could be used instead as a graded observation model to handle difficult situations more robustly.

Previous algorithms that exploit this intermediate output have been developed primarily for single-target tracking (mostly of faces) and have not been evaluated thoroughly for multiple, interacting targets [27]. For example, to apply their method to several targets, Li *et al.* [27] need to employ offline post-processing [29]. Similarly, tracking can be performed by exploiting a classifier trained to distinguish between object and background [2], [17]. Similar approaches exist that apply classifiers with different confidence thresholds [28], [46] or accumulate detection probabilities temporally [8], [40]. However, the extension of these methods to robust multi-target tracking is not trivial. Relying on the detector confidence in every situation can cause tracking errors, particularly during occlusions between interacting targets and in complex, cluttered scenes. This work presents a method to use this unreliable information source for robust multi-person tracking.

Data Association. Using independent trackers requires solving a data association problem to assign detections to targets. Classical approaches include the Joint Probabilistic Data Association Filter (JPDAF) [13] and Multi Hypotheses Tracking (MHT) [39]. MHT considers multiple possible associations over several time steps, but its complexity usually limits the analysis to only few such steps. JPDAFs instead try to make the best possible assignment in each time step by jointly considering all possible associations between targets and detections, to the cost of an exponentially increasing complexity. Alternatively, the Hungarian algorithm [22] can be used to find the best assignment of possible detection-tracker pairs in a runtime that is cubic in the number of targets. In practice, a greedy approach is however often sufficient, as pointed out by [45].

We stick to a greedy scheme and focus on obtaining a good scoring function. Such an approach is also used by Cai *et al.* [7], but their assignments are made only based on the spatial distance, without considering target appearance. This can be problematic for complex scenes with many targets and difficult background, where many false positive detections occur. Additionally, color histograms can be learned (*e.g.*, separately for different body parts [45]), which however do not always distinguish very well between the targets. Instead, we employ target-specific classifiers that are trained at run-time. Song *et al.* [42] presented a tracking algorithm that also learns target-specific classifiers. However, their method relies on background modeling and employs classifiers only when targets merge and split (*i.e.*, during occlusions). In contrast, our method exploits the classifiers in each time-step similarly to the very recent work of Kuo *et al.* [23], using it both for data association and for the observation model.

3 Detector Confidence Particle Filter

For many tracking applications, only past observations can be used at a certain time step to estimate the location of objects. Within this context, Bayesian Sequential Estimation is a popular approach, which recursively estimates the time-evolving posterior distribution of the target locations conditioned on all observations seen so far. This filtering distribution can be approximated by Sequential Monte Carlo Estimation (or Particle Filtering), which represents the distribution with a set of weighted particles and consists of a dynamic model for prediction and an observation model to evaluate the likelihood of a predicted state [10].

As object detection has made impressive improvements over the last years, a promising strategy is to employ an object detector for the observation model. However, the resulting detections are often not reliable (Fig. 1), *i.e.*, not all persons are detected in each frame (*missing detections*) and some detections are not caused by a person (*false positive detections*). Furthermore, in cases where no depth or scene information (*e.g.*, ground plane) is available, the detector does not know where to expect objects of which size in the image. To address these problems, many recent methods rely on global optimization techniques instead of making successive, irreversible decisions at each time step, which is a major limitation for time-critical applications.

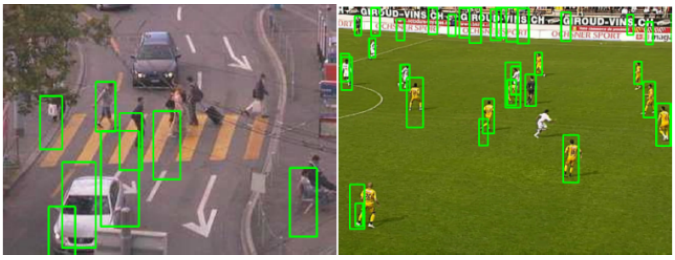


Fig. 1: The output of a person detector (right: ISM [25], left: HOG [9]) with false positives and missing detections.

3.1 Algorithm Overview

Our algorithm implements a first-order Markov model, considering only information from the current and the last time step, and integrates both class-specific and target-specific information in the observation model. A separate particle filter (*tracker*) is automatically initialized for each person detected with high confidence. To achieve the necessary robustness, the information from an *object detector* is integrated in two ways. First, the algorithm carefully assesses the high-confidence *detections* in each frame and selects maximally one to track one particular target. In order to resolve this *data association* problem, it evaluates a scoring function integrating *classifiers* that are trained during run-time for each target, the distance to the tracking target, and a probabilistic gating function accounting for the target size, motion direction, and velocity. If a detection is classified as reliable based on this function, it is mainly used to guide the associated tracker. Otherwise, the continuous *detector confidence* and output of the target-specific classifiers are mainly used. To evaluate the reliability of the detector confidence, we perform explicit inter-object occlusion reasoning.

Detector Confidence. At the core of our approach lies the *confidence density* built up by person detectors in some form. This is the case for both sliding-window based detectors such as HOG [9] and for feature-based detectors such as ISM [25]. In the sliding-window case, this density is implicitly sampled in a discrete 3D grid (location and scale) by evaluating the different detection windows with a classifier. In the ISM case, it is explicitly created in a bottom-up fashion through probabilistic votes cast by matching, local features.

In order to arrive at individual detections, both types of approaches search for local maxima in the density volume and then apply some form of non-maximum suppression. This reduces the result set to a manageable number of high-confidence hypotheses, but it also throws away potentially useful information. Figure 2 illustrates both types of output. As can be seen, there are situations where a detector did not yield a final detection, but a tracking algorithm could still be guided using the confidence density. On the other hand, both detectors also show a high detector confidence on certain background structures. Thus, relying on this intermediate output leads to tracking errors (*c.f.*, [27], [28], [46]).

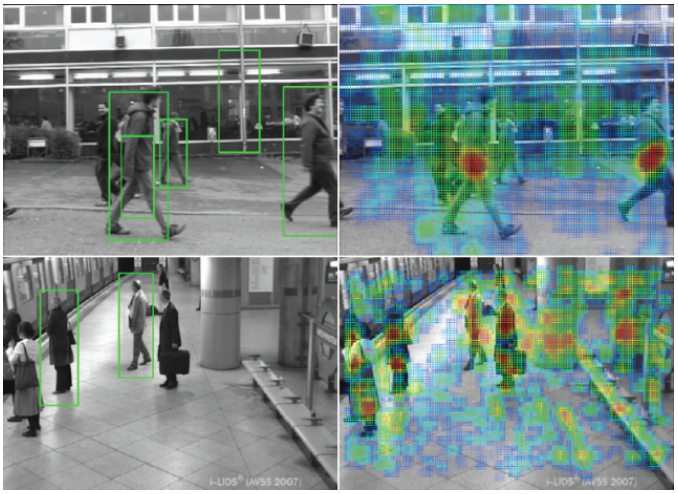


Fig. 2: Detector output (top: ISM [25], bottom: HOG [9]), showing high-confidence detections (left, green rectangles) and the detector confidence (right, shaded overlay). The confidence density often contains useful information at the location of missing detections, which we exploit for tracking.

3.2 Particle Filtering

Our tracking algorithm is based on estimating the distribution of each target state by a particle filter. The state $\boldsymbol{x} = \{x, y, u, v\}$ consists of the 2D image position (x, y) and the velocity components (u, v). We employ the *bootstrap filter*, where the state transition density (or *prior kernel*) is used as importance distribution to approximate the probability density function [16]. The importance weight w_t^i for each particle i at time step t is described by:

$$w_t^i \quad \propto \quad w_{t-1}^i \cdot p(o_t|\boldsymbol{x}_t^i). \tag{1}$$

Since re-sampling is carried out in each time step using a fixed number of $N = 100$ particles, $w_{t-1}^i = \frac{1}{N}$ is a constant and can be ignored. Thus, Eq. (1) reduces to the likelihood of a new observation o_t given the propagated particles $\boldsymbol{x}_t^i$, which we estimate as described in Sec. 3.4 (Eq. (6)).

Size and Position. Instead of including the size of the target in the state space of the particles, the target size is set to the average of the last four associated detections. In our experiments, this yielded better results, possibly because the number of particles necessary to estimate a larger state space is growing exponentially. Although represented by a (possibly multi-modal) distribution, a single position of the tracking target at the current time step is sometimes required (*e.g.*, for visualization or evaluation).

Motion Model. To propagate the particles, we use a constant velocity motion model:

$$(x, y)_t \quad = \quad (x, y)_{t-1} + (u, v)_{t-1} \cdot \Delta t + \varepsilon_{(x,y)} \tag{2}$$

$$(u, v)_t \quad = \quad (u, v)_{t-1} + \varepsilon_{(u,v)}. \tag{3}$$

The process noise $\varepsilon_{(x,y)}, \varepsilon_{(u,v)}$ for each state variable is independently drawn from zero-mean normal distributions. The initial variances $\sigma^2_{(x,y)}$ and $\sigma^2_{(u,v)}$ for position and velocity noise are set proportionally to the size of the tracking target.

Fig. 3: The initialization and termination region for a typical surveillance scenario (left). The initial particles are drawn from a normal distribution centered at the detection (middle). The weight of each particle is determined by evaluating the respective image patch (right).

During tracking, they decrease inversely proportional to the number of successfully tracked frames (down to a lower limit). Hence, the longer a target is tracked successfully, the less the particles are spread. Δt is dependent on the framerate of the sequence.

For sequences with abrupt, fast camera motion (which could be detected automatically), we apply the same motion model but additionally employ the *Iterative Likelihood Weighting* procedure [32]. To this end, the particles are divided into two sets, from which the first set is propagated normally. The particles from the second set are iteratively propagated and weighted several times (in our case, three times), to allow for more extreme particle movements within one time step.

Initialization and Termination. Object detection yields fully automatic initialization. The algorithm initializes a new tracker for an object that has subsequent detections with overlapping bounding boxes, which are neither occluded nor associated to an already existing tracker. In order to avoid persistent false positives from similar looking background structures (such as windows, doors, or trees), we only initialize trackers from detections that appear in a zone along the image borders for sequences where this is reasonable, such as for typical surveillance settings. This was the case for most experiments in Sec. 4, where the initialization region was comparable to Fig. 3 (left). For sequences where targets appear in the middle of the image, *e.g.*, for shorter sequences (TUD Crossing) or for sequences from moving cameras (UBC Hockey, Soccer), we initialized on the entire image.

The initial sample positions are drawn from a normal distribution around the detection center (Fig. 3, middle). The initial size corresponds to the detection size, and the motion direction is set to be orthogonal to the closest image border.

A tracker only survives a limited number of frames without associated detection and is then automatically terminated. However, to re-detect a target that temporally leaves and later re-enters the field of view, the trackers are only deactivated (*c.f.*, [6]). Thus, instead of immediately initializing a new tracker, the algorithm checks first if the same target has already been observed before. For this purpose, the classifier of each deactivated tracker is evaluated.

Algorithm 1 Greedy data association.

T : set of all trackers
D : set of all detections
$S(tr, d)$: scores for each tracker-detection pair, Eq. (4)
$A(tr, d) = 0$: final associations of detection d to tracker tr
Require: $\forall tr \in T : \sum_i A(tr, i) \leq 1$
Require: $\forall d \in D : \sum_j A(j, d) \leq 1$
while $T \neq \varnothing \wedge D \neq \varnothing$ **do**
  $(tr^*, d^*) = \arg\max_{tr \in T, d \in D} S(tr, d)$
  if $S(tr^*, d^*) \geq \tau$ **then**
    $A(tr^*, d^*) = 1$
  $T = \{T \setminus tr^*\}$
  $D = \{D \setminus d^*\}$

3.3 Data Association

In order to decide which detection should guide which tracker, we solve a data association problem, assigning at most one detection to at most one target. The optimal single-frame assignment can be obtained by the Hungarian algorithm [22]. In our experiments, we however found that a greedy algorithm achieves similar results at lower computational cost.

Greedy Data Association. The matching algorithm works as follows (see Algorithm 1): First, a matching score matrix S for each pair (tr, d) of tracker tr and detection d is computed as described below. Then, the pair (tr^*, d^*) with maximum score is iteratively selected, and the rows and columns belonging to tracker tr and detection d in S are deleted. This is repeated until no further valid pair is available. Finally, only the associated detections with a matching score above a threshold are used, ensuring that a selected detection actually is a good match to a target. Consequently, the chances are high that often no detection will be associated with a target, but if one is, it can be used to strongly influence the tracker.

Matching Score. Our data association method evaluates a matching function $S(tr, d)$ for each tracker-detection pair (tr, d). The higher the score, the better the match between detection and tracking target. It employs a classifier $c_{tr}(d)$ trained for tr, which is evaluated for d:

$$S(tr, d) = g(tr, d) \cdot (c_{tr}(d) + \alpha \cdot \sum_{p \in tr}^{N} p_{\mathcal{N}}(d - p)), \qquad (4)$$

where $p_{\mathcal{N}}(d-p) \sim \mathcal{N}(pos_d - pos_p; 0, \sigma^2)$ denotes the normal distribution evaluated for the distance between the position of detection d and a particle p, and $g(tr, d)$ is a gating function described next. The last term of (Eq. (4)) measures the density of the particle distribution, rewarding associations where the particles are densely distributed around the detection.

Gating Function. Not only the distance of a detection to the tracker is important, but also its location with respect to the motion direction. Therefore, a *gating function* $g(tr, d)$ additionally assesses each detection. It consists of the product of two factors:

$$g(tr, d) = p(size_d|tr)p(pos_d|tr) \qquad (5)$$

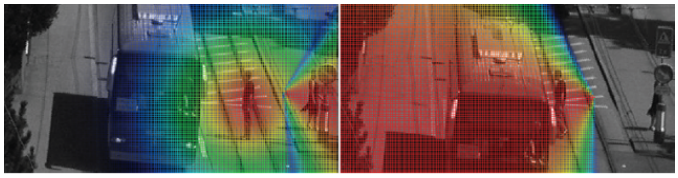


Fig. 4: The gating function depends on the velocity of the target, resulting in different 2D cone angles or a radial decay.

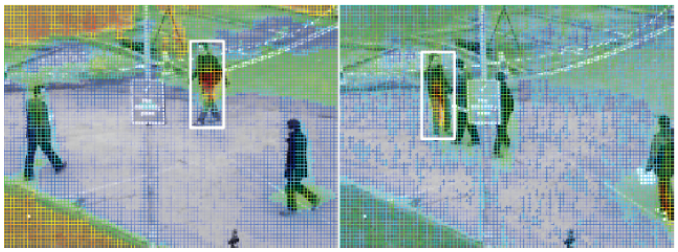


Fig. 5: The classifier response (heat map) visualized for one tracking target (white). As the classifier is adapted continuously, it becomes more discriminative (right: 20 frames later).

$$= \begin{cases} p_{\mathcal{N}}(\frac{size_{tr} - size_d}{size_{tr}}) \cdot p_{\mathcal{N}}(|d - tr|) & \text{if } |\boldsymbol{v}_{tr}| < \tau_v \\ p_{\mathcal{N}}(\frac{size_{tr} - size_d}{size_{tr}}) \cdot p_{\mathcal{N}}(dist(d, \boldsymbol{v}_{tr})) & \text{otherwise.} \end{cases}$$

The first normal distribution measures the agreement between the bounding box height of target and detection. The second normal distribution follows the intuition that fast-moving objects cannot change their course abruptly because of inertia. Therefore, the term depends on the velocity of the target. If the velocity $|\boldsymbol{v}_{tr}|$ is below a threshold τ_v, it is ignored and the term is proportional to the distance from the center of the main mode of tracker tr to detection d. In this case of a (almost) motionless target, the function decays radially (Fig. 4).

Otherwise, the second term depends on the distance between the detection d and the line $\boldsymbol{v}_{tr} = (u, v)$ given by the position of the tracker and the direction component of the velocity. The variance for this term is chosen such that it is proportional to the distance from the tracker to the detection projected to $\boldsymbol{v}_{tr}$. Thus, a detection d_1 with the same distance to the line $\boldsymbol{v}_{tr}$ than another detection d_2, but which is closer to the tracker tr, gets a lower score. Hence, the isolines of Eq. (5) then form a 2D cone (Fig. 4). Furthermore, the angle of the 2D cone is made smaller the higher the speed of the target.[1]

Boosted Classifiers. To assess the similarity of a tracker-detection pair, we use the algorithm from Grabner *et al.* [17]. We train a boosted classifier c_{tr} of weak learners for each tracking target against all others during run-time. Each weak learner represents a feature computed for both a positive and a negative training image (see Sec. 4.2 for a description of the features). For each classifier, weak learners are selected using AdaBoost. During evaluation, a classifier computes the similarity between the input and all its weak learners using a k-Nearest Neighbor classification approach.

1. The second term of Eq. (5) is equivalent to an angular error that is correctly measured by the von Mises distribution, but can be closely approximated by a Gaussian distribution in the 1D case [31].

Positive training examples are patches sampled from the bounding box of the associated detection (the sampling probability is higher the closer a patch is to the vertical center line). The negative training set is sampled from nearby targets, augmented by background patches. The classifier is only updated if a detection does not overlap with another detection. After each update step, we keep a constant number of the most discriminative weak learners. Thus, the classifier is continuously adapted, becoming more and more discriminative (Fig. 5). This framework has several advantages: it allows us to include different features, it automatically selects the most discriminative feature set, and it provides a natural way to adapt to appearance changes of the targets.

3.4 Observation Model

To compute the weight $w_{tr,p}$ for a particle p of the tracker tr, our algorithm estimates the likelihood of a particle. For this purpose, we combine different sources of information, namely the associated detection d^*, the intermediate output of the detection algorithm, and the output of the classifier c_{tr}:

$$w_{tr,p} = \underbrace{\beta \cdot \mathcal{I}(tr) \cdot p_{\mathcal{N}}(p - d^*)}_{\text{detection}} + \underbrace{\gamma \cdot d_c(p) \cdot p_o(tr)}_{\text{det. confidence}} + \underbrace{\eta \cdot c_{tr}(p)}_{\text{classifier}} \tag{6}$$

where the parameters β, γ, η are set experimentally and remain fixed during tracking (see Sec. 3.5). Each term is described below in detail.

Detection Term. The first term computes the distance between the particle p and the associated detection d^*, evaluated under a normal distribution $p_{\mathcal{N}}$. $\mathcal{I}(tr)$ is an indicator function that returns 1 if a detection was associated to the tracker and 0 otherwise by the data association procedure described in Section 3.3. When a matching detection is found, this term robustly guides the particles.

Detector Confidence Term. The second term evaluates the intermediate output of the object detector by computing the detector confidence density $d_c(p)$ at the particle position. To estimate $d_c(p)$ for the ISM detector, we compute the local density ρ in the Hough voting space using a cubic kernel adapted to the target size and scaled with $f = 1 - exp(-\rho)$ to $[0, 1]$. For the HOG detector, $d_c(p)$ corresponds to the raw SVM output before applying non-maximum suppression, which is also scaled to $[0, 1]$.

Unfortunately, the detector confidence is not always reliable; often, an erroneously high value is caused by background structures (Fig. 2). To assess its reliability, our algorithm therefore performs *inter-object occlusion reasoning* using the following rationale: if another tracker tr' is nearby that is associated with a detection, the detector confidence at this image location and in its proximity is most probably caused by the foreground and not by background structure. Consequently, it is likely that the detector did not find both targets because of the occlusion. In this case, we assume that the detection confidence is meaningful in this image area and can be used to guide the tracker. Hence, the function $p_o(tr)$ increases the influence of the detector confidence for tracker tr in Eq. (6) the closer another tracker tr' is:

$$p_o(tr) = \begin{cases} 1 & \text{if } \mathcal{I}(tr) = 1 \\ \max\limits_{tr':\mathcal{I}(tr')=1} p_{\mathcal{N}}(tr - tr') & \text{else if } \exists \mathcal{I}(tr') = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Fig. 6: Visualization of the detector confidence reliability function, which returns for tracker a a higher value (right) if another tracker b with associated detection is close.

Note that the region defined by $p_{\mathcal{N}}$ is rather large, as shown in Fig. 6, where the function is evaluated for person a entering the scene from the right. Thus, the confidence map is only completely ignored when no track passes by even close to the corresponding image region, which is only rarely the case in practice.

Classifier Term. For the third term of Eq. (6), the classifier c_{tr} trained for target tr (Sec. 3.3) is evaluated for the image patch at the particle location with the corresponding size (Fig. 3, right). This term uses color and texture information to assess the new particle position and complements the terms from the detector output. While other tracking methods are purely based on such classifier output (*e.g.*, [2], [17]), this adds additional robustness to our particle filter approach, especially during partial occlusions. In addition, the combination of generic category knowledge and person-specific information makes our approach more robust to classifier drift.

3.5 Implementation

Detectors. For all experiments, we employ either the HOG detector [9] or the ISM detector [25], which are publicly available and not trained specifically for our tracking scenarios (*c.f.*, [7], [33]). We apply the provided ISM model trained on side-views of persons with size 80×200 pixels, operating on Hessian-Laplace interest points. The HOG detector is trained on the INRIA Person Dataset, resized to 48×96 pixels for a better correspondence of the person size in the test data.

Algorithm Parameters. All parameters have been set experimentally and most remained identical for all experiments with different sequences. This was the case for the variances σ^2 in Eqs. (4)–(7), for α in Eq. (4), for β and η in Eq. (6), and for τ in Algorithm 1. Only γ was increased for one sequence (TUD Crossing, see Section 5) to overcome very long-lasting overlaps between detections by the detector confidence. β, γ, η were chosen experimentally and set such that the ratio between the respective terms in Eq. (6) are approximately 20:2:1 for a tracker *with* associated detection. Hence, if a reliable detection is found, the first term of the observation model mainly guides the particles, which is the case every 2–10 frames on

average, depending on the sequence. During a typical tracking cycle, the contribution of each of the individual observation model terms to the total particle weight can however differ significantly. We analyze the influence of each term to the overall robustness in Sec. 5.

The initial target size corresponds to the size of the detection ($scale_{det}$ is the size compared to the detector training size). The initial sample positions are drawn from a normal distribution with standard deviation $\sigma = 6 \cdot scale_{det}$ pixels, centered at the detection bounding box center. The standard deviations for the position and velocity noise are set to $\sigma = 4 \cdot scale_{det}$ and $\sigma = 12 \cdot scale_{det}$ pixels (*i.e.*, about 10 and 30 pixels for a target with a height of 180 pixels ($scale_{det} = 2.5$)). The initial motion direction is set to be orthogonal to the closest image border with magnitude $v = 24 \cdot scale_{det}$ pixels. To handle abrupt motion changes in sports sequences, we increased σ^2 in Eqs. (2)–(3) to make the motion model more flexible.

4 Experiments

4.1 Datasets

There is no generally accepted benchmark available for multi-person tracking. Therefore, most related publications have carried out experiments on their own sequences, which we have tried to combine. Thus, we evaluate on a large variety of challenging sequences: ETHZ Central [26], TUD Campus and TUD Crossing [1], i-Lids AB [18], [45], UBC Hockey [7], [33], PETS'09 S2.L1–S2.L3 [12], ETHZ Standing [14], and our own Soccer dataset.[2]

These sequences are taken from both static and moving cameras, and they vary with respect to viewpoint, type of movement, and amount of occlusion. While some datasets show rather classical surveillance and security scenarios from an elevated viewpoint, others are captured at eye level and are typical for robot / car navigation and traffic safety applications, while some are sports sequences with abrupt motion changes of the players and moving cameras. For all sequences, we use only a single camera (*c.f.*, [3]), we do not assume any scene knowledge such as ground plane calibration (*c.f.*, [14], [26]) or scene-specific entry/exit zones (*c.f.*, [18]), do not employ an object detector specifically trained for a certain application scenario (*c.f.*, [7], [33]), and process the sequences in a causal way (*i.e.*, without using information from future frames, *c.f.*, [3], [14], [18]).

We use the detectors originally used with these sequences: we employ the ISM detector for ETHZ Central, TUD Crossing, TUD Campus, ETHZ Standing and UBC Hockey. For i-Lids, PETS'09 and Soccer, we use the HOG detector, since it is not only trained on side-views of persons in contrast to the ISM detector. For the PETS'09 dataset, the input images are resized from originally 768×576 pixels to 1280×960 pixels, such that the size of the persons better corresponds to the detector training size (analogously for the Soccer dataset).

2. The references indicate publications with state-of-the-art results. Please watch our result videos: http://www.vision.ee.ethz.ch/~bremicha/tracking

4.2 Classifier Features

To select features for the boosted classifier (*i.e.*, number, type, combination of features), we evaluate the ability of the classifiers to distinguish between the correct target and all others. To this end, we compare the classifiers on different sequences using annotated ground truth. Ideally, the classifier returns a score of $+1$ for the target it is trained for, and -1 for all other targets. Hence, the larger the difference between the classifier score for a correct and the other targets, the better the classifier can distinguish between them. In the Figures 7(a)–7(d), we plot the difference of the classifier score on the annotated targets and the highest score on all other targets.

We performed experiments with color histogram features in RGB (red-green-blue), HS (hue-saturation), RGI (red-green-intensity) and Lab space, and with texture features LBP (local binary patterns) and Haar wavelets. Each feature is computed on a patch with random size and position, sampled from within the bounding box of a detection or tracker main mode.

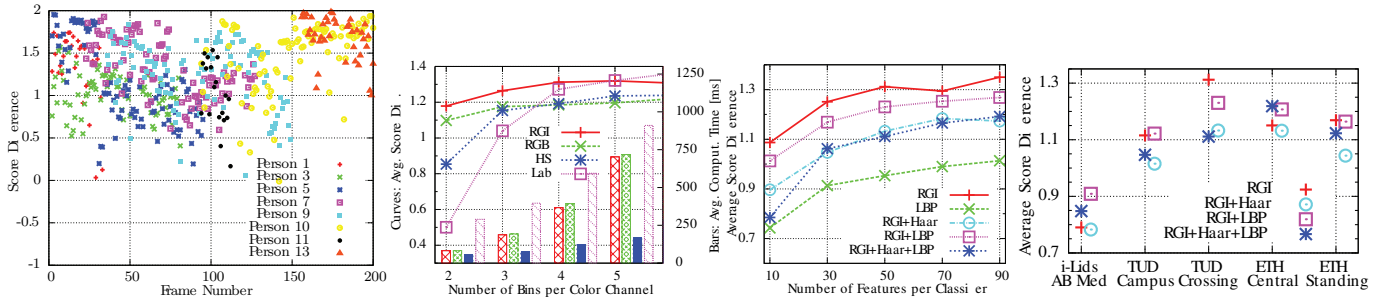
In Fig. 7(a), we plot the score difference for 200 frames of the TUD Crossing sequence [1], using 50 RGI color features with 3 bins per color channel. The score difference is large for most targets and throughout most frames. Importantly, it is never negative, *i.e.*, two targets are never mixed up. Some targets are more difficult to distinguish than others because of their (similar) clothing. The score difference declines sometimes when a new target enters the scene, against which the other classifiers are not trained yet. Also, when the appearance of a target changes (*e.g.*, during an occlusion), the classifier needs some time to adapt, causing the performance to drop.

In Fig. 7(b), we investigated the impact of different color features on classification accuracy and speed, averaged over all targets and frames. The accuracy increases if more bins are used for the color histogram. However, also the computation time (incl. training and testing) increases. As a compromise, we chose the RGI feature with 3 bins per color channel. Fig. 7(c) shows the evaluation for feature combinations and numbers of features (*i.e.*, weak learners). We use 50 features per classifier. Fig. 7(d) shows the effect of combining different features for different sequences. The combination of RGI and LBP features often outperforms color or textural features alone and other combinations. We use RGI and LBP features for all sequences.

4.3 Qualitative Analysis

ETHZ Central. The output of the ISM detector is very noisy for the ETHZ Central dataset (Fig. 8, top). The cars and road markings produce many false positives, and pedestrians are often not detected. Only a few detections consistently match the targets throughout the sequence (*e.g.*, the blue tracker in Fig. 8, bottom, gets assigned a detection only every 30 frames). Thus, the trackers often rely on the detector and classifier confidence. Furthermore, there are many occlusions, *e.g.*, when people walk in parallel. Hence, the correct association of detections to trackers is a key factor of our algorithm.

TUD Campus. The ISM detections are more accurate for the TUD Campus dataset. On average, a tracker is associated



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

Fig. 7: (a) Classifier evaluation on the TUD Crossing sequence with 50 RGI features and 3 bins per color channel. We plot the difference between the classifier score on the correct target and the highest score on all other targets. (b) Evaluation of performance (left scale) and computation time (bars, right scale) for different color features. (c) Evaluation of the number of features per classifier. (d) Evaluation of feature combinations for some datasets.

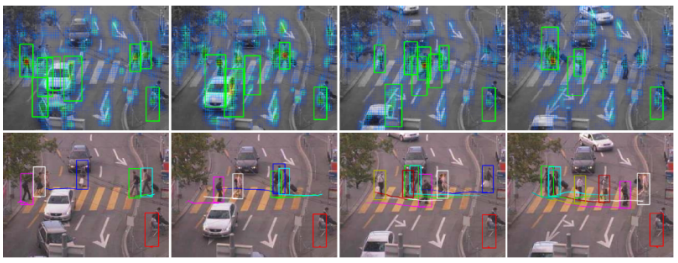


Fig. 8: Result for the ETHZ Central dataset (top: final ISM detections (green) and detector confidence (heat map)), tracking result (bottom).

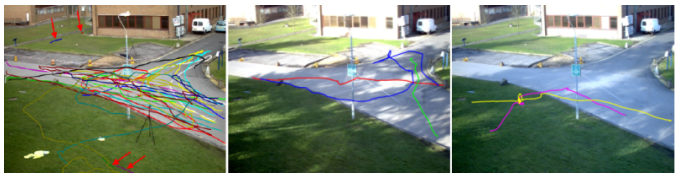


Fig. 10: The resulting trajectories for the PETS'09 tasks S2.L1, S2.L2 and S2.L3 (false positives denoted by red arrow).

with a detection in every second frame. Since the different persons have different sizes, it is easier to assign detections to trackers (Fig. 9(a)). However, many long inter-object occlusions occur, *e.g.*, the cyan tracker fully occludes four other persons temporarily. During these occlusions, the particles of the other trackers are attracted by the high detector confidence around the cyan target, until the other targets reappear.

TUD Crossing. In contrast, most persons in the TUD Crossing dataset have a similar size (Fig. 9(b)); thus, the detection sizes are not useful to simplify data association. Additionally, most persons walk at similar speeds, so this cue also cannot be used to resolve ambiguities. By increasing the influence of the detector confidence term (*i.e.*, γ in Eq. (6), as described in Sec. 3.5), all persons are however successfully tracked through the long inter-object occlusions.

AVSS i-Lids AB Medium. Due to the elevated camera viewpoint, the persons occlude each other frequently, and their visible sizes differ substantially (Fig. 9(c)). This makes the sequence challenging for both the detector and the tracker. The classifier and detector confidence terms are therefore particularly important. They keep the particles from drifting and locking onto other targets. Furthermore, a persistent foreground object (a pillar) occludes many targets immediately after entering the scene. This makes the initialization more difficult, as the classifiers are trained with only few samples before the target is occluded. However, even though no scene-specific information is used, the tracker manages to handle these problems in most cases. The HOG detector causes many false positives when many people enter the scene that are already partially occluded (*e.g.*, if a train arrives), making initialization difficult. Because of these frequent inter-object occlusions, a part-based detector trained on individual body parts would be advantageous for this sequence.

PETS'09. The PETS'09 dataset is recorded from several synchronized cameras, from which we only use one (view 1). In contrast to the first task S2.L1, only two (predetermined) targets need to be tracked for the tasks S2.L2 and S2.L3. Since our algorithm automatically initializes for all detected targets, we manually select the corresponding trajectories for the evaluation after running our algorithm completely. In Fig. 9(d)–(f) and Fig. 10, we show the results and all trajectories, respectively.

For S2.L1, all persons are tracked. The HOG detector finds about 80% of all persons throughout the sequence, while about 50% of all detections are false positives. Although the size of the targets changes significantly, no identity switches occur (Fig. 10). A second challenge are the significant complete and partial occlusions caused by the traffic sign and by other tracking targets, which are handled robustly (Fig. 9(d)). Third, the motion of some targets is highly dynamic, as they are suddenly stopping, moving backwards, or in circles. Over the whole sequence (of about 90 seconds), our method returns 4 short false positive trajectories (marked by the red arrows in Fig. 10), which are caused by erroneously initialized trackers due to persistent false positive detections at the image borders. For most targets that temporally leave the field of view and re-enter the scene, the respective tracker can be re-activated (*e.g.*, in Fig. 9(d), the purple target in the first and second

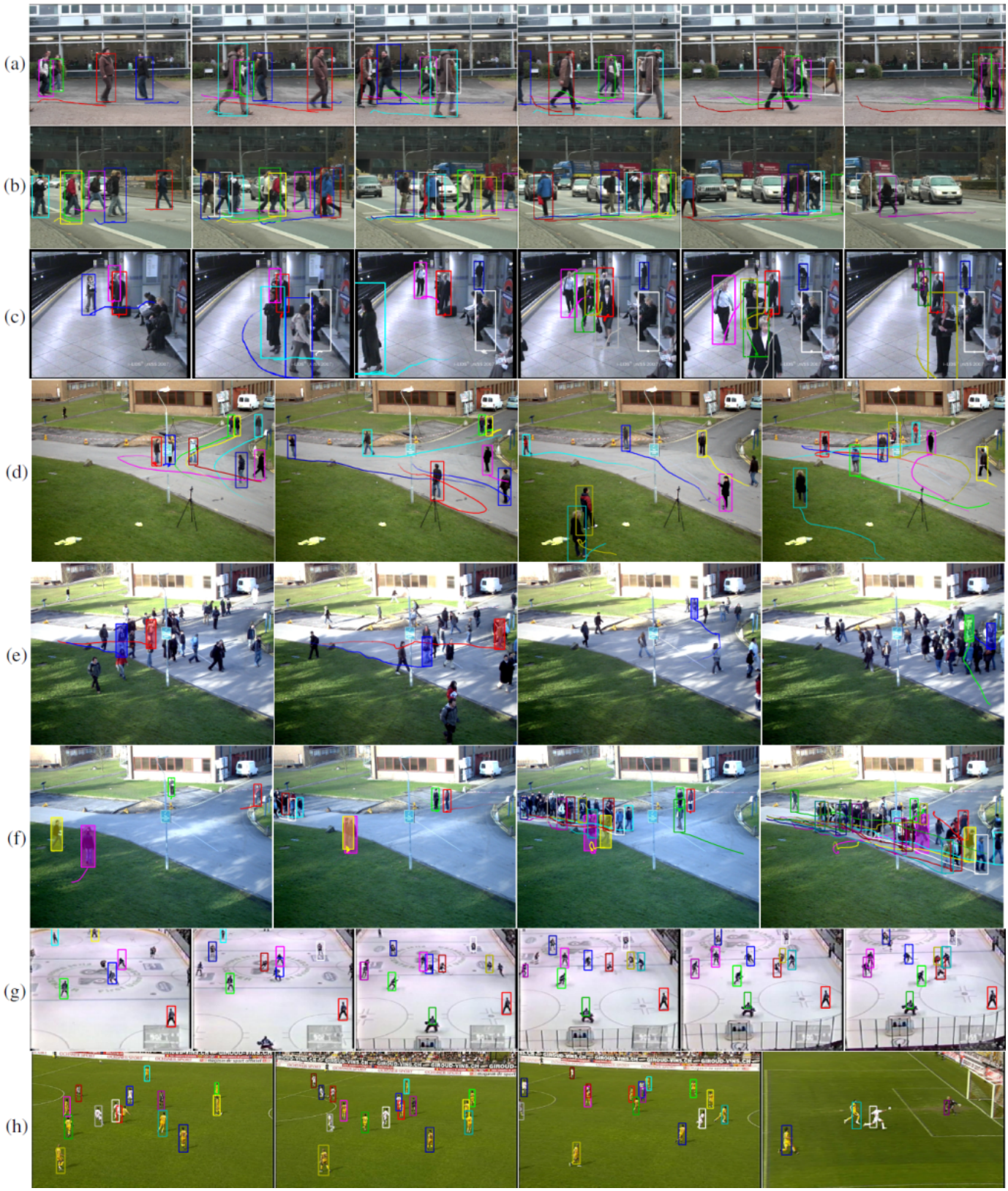


Fig. 9: Tracking output on the TUD Campus (a), TUD Crossing (b), AVSS i-Lids AB Medium (c), PETS'09 S2.L1 (d), S2.L2 (e), S2.L3 (f), UBC Hockey (g), and Soccer dataset (h).For visualization purposes, the shown trajectories are computed by averaging over the last three positions of the tracker bounding box. However, only the bounding boxes are used for evaluation.

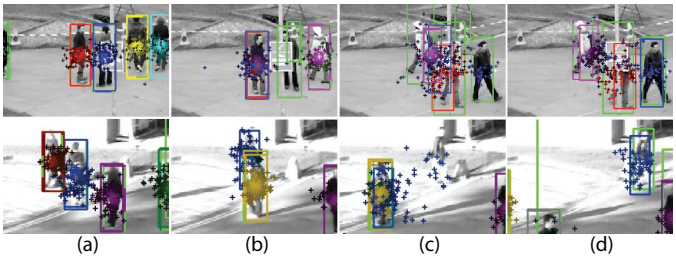


Fig. 11: Particle filter output (particles and main modes) and HOG detections (green) for PETS'09 S2.L1 (top) and S2.L2 (bottom). The tracking algorithm recovers after the occlusion (top) and the appearance change (bottom).

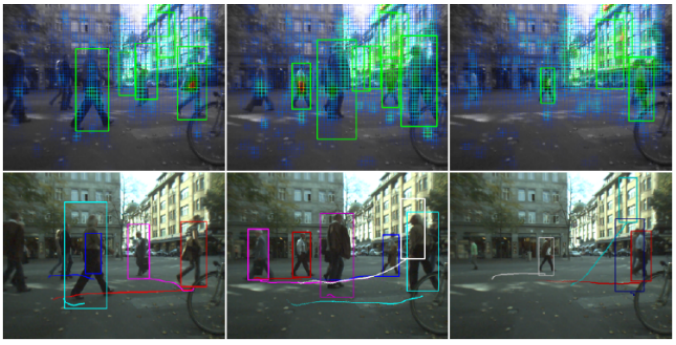


Fig. 12: For the ETHZ Standing dataset, the trackers often in the right part of the image (bottom), because the ISM detector confidence is very high on background structures (top).

image, and the cyan target in the third and fourth image).

The sequences S2.L2 and S2.L3 mainly pose two additional challenges. First, target appearance changes heavily, caused by different lighting conditions in different image areas, or when a target turns with respect to the camera position. Second, the persons in the crowd walk very closely together, regularly occluding each other. Our algorithm manages to robustly handle most of these problems. As can be seen from Fig. 10, one target person leaves and later re-enters the scene. Here, the respective tracker could not be re-activated because the classifier score is too low.

In Fig. 11, we show a sequence of frames to illustrate how the algorithm handles situations with severe occlusions (top: S2.L1) and appearance changes (below: S2.L2). First, all trackers are associated with a detection (Fig. 11, top, image a). The person represented by the blue tracker then moves towards the road sign and becomes occluded (b). Since no detection is available, the particles propagate towards nearby areas of high detector confidence (*i.e.*, to the target of the red tracker). After 50 frames, the person reappears from behind the road sign (c) and is represented by the respective tracker again (d), thanks to the classifier. In the second example (Fig. 11, bottom), the blue person is occluded (image b) while entering a brightly illuminated image area, thus changing its appearance. As a result, the classifier is not updated and does not adapt. However, because of the particle filter's multi-modality, some particles remain on the correct target (c), and the tracker recovers (d).

UBC Hockey. In contrast to the typical pedestrian sequences shown before, sports videos impose additional difficulties to a tracking algorithm. First, the camera is usually not static, *i.e.*, it is not clear from the 2D image information alone whether the motion is caused by camera movement or by a moving target. Second, player motion may change more abruptly, which makes data association more challenging. Furthermore, the hockey players' appearance differs substantially from the dataset used to train the detectors. The final detections are therefore very unreliable, and the detector and classifier confidence is primarily used for tracking. Although the players' appearance (*i.e.*, their jersey color) is very similar, mismatches are avoided thanks to the gating function used for data association (Sec. 3.3).

Soccer. In an even more challenging setting, the Soccer dataset was recorded with a strongly moving camera that additionally zooms in. The player sizes therefore change considerably (Fig. 9(h)). The players are interacting and look similar, making data association difficult. Fortunately, two nearby targets are often from rivaling teams, hence the colors of their jerseys are different. As can be seen from Fig. 9(h), the classifiers are not very robust in the beginning of the sequence, causing identity switches. However, after a while, the tracker finds and differentiates all targets, even during a fast pan of the camera.

ETHZ Standing. Fig. 12 (bottom) shows the result for the ETHZ Standing sequence, illustrating the limitations of the method. The ISM detector confidence is very high on background structures (Fig. 12, top), producing regular false positive detections. Hence, the trackers fail to robustly find the targets after the long occlusions, or they are not properly terminated and drift to these image regions. Here, scene knowledge or depth information probably is necessary for robust tracking (as used by others, *e.g.*, [14]).

Summary. We have demonstrated on a variety of sequences that the tracker robustly handles different challenges. The remaining failures occur mainly when the detector output is extremely noisy during initialization, termination, and long occlusions. Other cases are when partially occluded targets enter the scene, or when the appearance of a target changes while it is not detected (*e.g.*, during an occlusion).

4.4 Quantitative Analysis

We use the CLEAR MOT metrics [4] to evaluate the tracking performance. This returns a precision score MOTP (intersection over union of bounding boxes) and an accuracy score MOTA (composed of false negative rate, false positive rate, and number of identity switches). The results for the sequences discussed in Sec. 4.3 are shown in Tab. 1. Where available, the results of the state-of-the-art methods are also shown.

As for the precision (MOTP), we consider a score of above 50% as reasonable for tracking. The same threshold is used to accept detections for the prominent Pascal VOC challenge [34]. The accuracy (MOTA) consists of the false negative (FN) and false positive (FP) rate, and the number of identity switches (ID Sw.). The false negatives occur when

Dataset	MOTP	MOTA	FN	FP	ID Sw.
ETHZ Central	70.0%	**72.9%**	26.8%	0.3%	0
Leibe *et al.* [26]	66.0%	33.8%	51.3%	14.7%	5
UBC Hockey	57.0%	**76.5%**	22.3%	1.2%	0
Okuma *et al.* [33]	51.0%	67.8%	31.3%	0.0%	11
i-Lids Easy	67.0%	**78.1%**	16.4%	5.3%	18
i-Lids Medium*[3]	66.0%	**76.0%**	22.0%	2.0%	2
Huang *et al.* [18]	-	68.4%	29.0%	13.7%	-
Wu and Nevatia [45]	-	55.3%	37.0%	22.8%	-
TUD Campus	67.0%	**73.3%**	26.4%	0.1%	2
TUD Crossing	71.0%	**84.3%**	14.1%	1.4%	2
Soccer	67.0%	**85.7%**	7.9%	6.2%	4
PETS'09 S2.L1	56.3%	**79.7%**	-	-	-
PETS'09 S2.L1*[4]	56.7%	74.9%	-	-	-
Yang *et al.* [47]	53.8%	75.9%	-	-	-
Berclaz *et al.* [3][5]	ca. 60%	ca. 66%	-	-	-
PETS'09 S2.L2	51.3%	**50.0%**	-	-	-
PETS'09 S2.L3	52.1%	**67.5%**	-	-	-

TABLE 1: CLEAR MOT [4] evaluation results, showing precision (MOTP), accuracy (MOTA), false negative rate (FN), false positive rate (FP), and the number of ID switches (ID Sw.). Where available, state-of-the-art results are also shown.

persons are annotated but not detected. This happens for persons that are very close to another person (ETHZ Central, TUD Crossing), that are sitting (ETHZ Central), or that are partially outside of the image (i-Lids). False positives are caused by trackers that drift during an occlusion (*e.g.*, due to the pillar in i-Lids) or that lose their target (*e.g.*, due to strong camera motion in the Soccer dataset). If a target leaves the scene while a new target enters, the tracker may switch their identities, which happens only rarely thanks to the online trained classifiers. The remaining identity switches are due to cases where a person that was only shortly visible is occluded (*e.g.*, in i-Lids) or for newly appearing persons with similar appearances that are close together (*e.g.*, in Soccer). In these cases, the motion model and classifier for the targets are not sufficiently adapted yet. A low number of ID switches is one of the most important properties of a good tracking algorithm.

We compare our method with the state-of-the-art results reported for these sequences (Tab. 1): On ETHZ Central with Leibe *et al.* [26] (using provided trajectories), on UBC Hockey with Okuma *et al.* [33] (obtained using their publicly available Matlab code on their data), and on i-Lids as reported by Huang *et al.* [18][3]. In all those cases, our precision and accuracy results outperform the previously published results, even though our algorithm does not use global optimization (*c.f.*, [18], [26]), nor a detector specifically trained for the appearance in the sequence (*c.f.*, [33]), camera calibration

3. [18] did not report all CLEAR MOT evaluation numbers. We tested on i-Lids Easy and the first half of i-Lids Medium, for which we added annotations for fully visible, sitting persons, reported as i-Lids Medium*. For the second half, many persons are only partially visible, and the HOG detector therefore did not yield reasonable results. In contrast, [18] used the part-based detector of [45], which works better for such situations but is not publicly available. Thus, a direct comparison is not possible.

4. Without re-using previous trackers for re-entering targets, from [5].

5. The numbers are extracted from Fig. 3 of the PETS 2009 report [12].

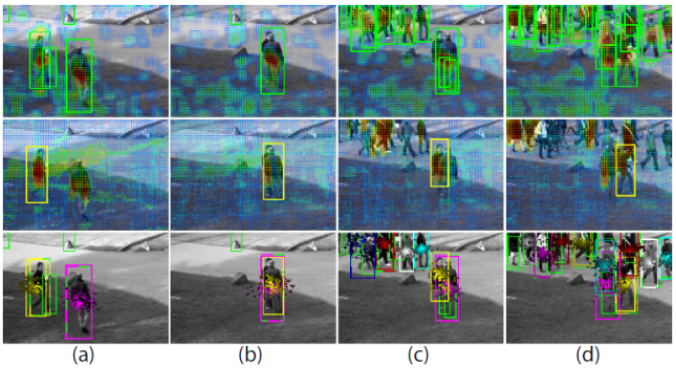


Fig. 13: Visualization of detector output (top), classifier output for the yellow target (middle), and particle filter output (bottom; dashed bounding boxes are detections associated to the tracker with the respective color).

(*c.f.*, [26]), or a scene model (*c.f.*, [18]).

The evaluation of the PETS'09 tracking results was performed by the organizers of the workshop [12], who did not provide the scores for FN, FP and ID switches. Surprisingly, our algorithm outperforms the multi-camera system of Berclaz *et al.* [3] in terms of accuracy, even though the latter uses 5 camera views, scene-specific knowledge (a ground plane), and delivers the results with a temporal delay. As expected, Berclaz *et al.* achieve a slightly higher precision score. Our method also outperforms the well-engineered system of Yang *et al.* [47] that is based on background modeling and relies on a static background. As shown in Tab. 1, the accuracy score drops by about 5% when trackers are immediately terminated and not reused for re-entering targets (due to the higher number of identity switches).

Summary. Both accuracy and precision of our method are reasonably high. The algorithm outperforms other state-of-the-art methods, even though many of them rely on simplifying assumptions or additional information, limiting their applicability. Our method relies only on information from the past and is thus suitable for time-critical, online applications.

4.5 Runtime Performance

The entire system is implemented in C++, without taking advantage of GPU processing. On a workstation with an Intel Core2Duo 2.13GHz and 2GB of memory, we achieve processing times of 2–0.4 frames per second (given the detector output), depending on the number of detections and targets in a sequence. While the current bottleneck is the detection stage, we want to point out that for the HOG detector, real-time GPU implementations exist [37], [44]. As not all speedup possibilities are explored yet, the current run-time raises hope that real-time experiments will not be too far away.

5 Discussion

We first discuss how the different observation model terms assist in handling difficult situations. Then, the influence of each term is evaluated quantitatively. Third, we demonstrate

Observation Models	MOTP	MOTA	FN	FP	ID Sw
(a): Det+Conf+Class	70.0%	72.9%	26.8%	0.3%	0
(b): Det+Conf	64.0%	54.5%	28.2%	17.2%	5
(c): Det+Class	65.0%	55.3%	31.3%	13.4%	0
(d): Conf+Class	68.0%	49.0%	37.7%	13.1%	5
(e): Det	67.0%	40.9%	30.7%	28.0%	10
(f): Conf	64.0%	47.6%	33.0%	19.1%	8
(g): Class	48.0%	25.3%	46.2%	27.9%	17
(h): N=25	63.0%	45.0%	33.4%	21.4%	6
(i): N=15	53.0%	23.4%	41.4%	34.7%	12
(j): N=10	51.0%	-5.6%	50.8%	53.9%	23
(k): N=5	40.0%	-59.4%	53.6%	104.7%	31
(l): N=1	36.0%	-104.1%	57.4%	144.8%	52
(m): $\tau = 0.5$	69.0%	60.1%	31.4%	8.4%	3
(n): $\tau = 0.2$	65.0%	32.0%	34.5%	33.2%	5

TABLE 2: CLEAR MOT evaluation results on the ETHZ Central dataset, using (a-g) different observation models (see also Fig. 14), (h-l) different numbers of particles N for a tracker, or (m-n) different values for parameter τ. For the original result (a), the complete observation model and the parameters N=100 and $\tau = 1$ are used.

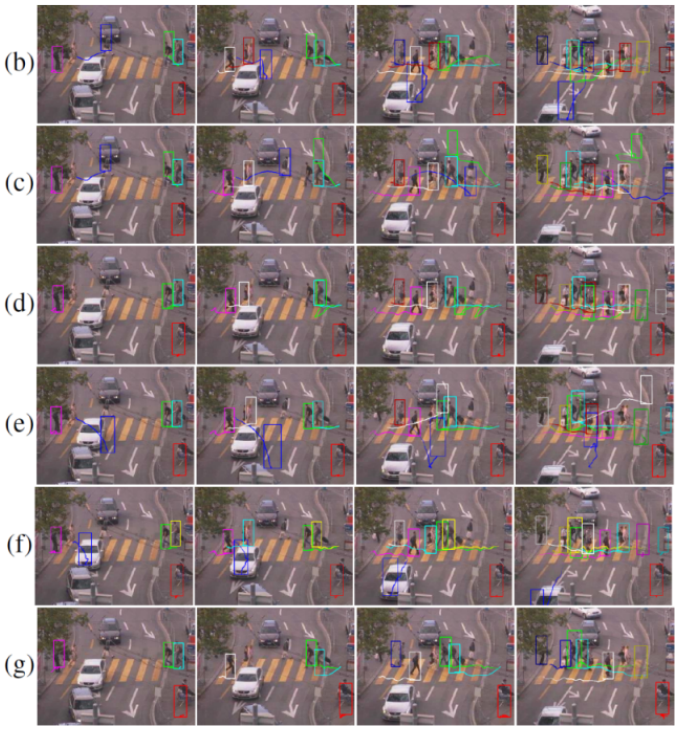


Fig. 14: Tracking output on ETHZ Central with different observation models, acc. to Tab. 2 (original result in Fig. 8).

the contribution of the particle filter. Last, we show how tracking performance varies when relying on discrete detections.

Handling Difficult Situations. In Fig. 13(a)–(d), we show the detector output (top), classifier output for one target (middle), and particle filter output (bottom) for frames 18, 105, 151 and 175 of the PETS'09 S2.L3 sequence. In Fig. 13(a), good detections are available as both tracking targets are fully visible (top). Two trackers (yellow, magenta) have been initialized, and detections are associated to them (bottom), hence primarily guiding the trackers.

In contrast, one target is fully occluded by the other in Fig. 13(b). The detection is associated to the correct tracker (magenta, bottom). The particle weights for the yellow tracker are primarily computed from the detector confidence, since another tracker (magenta) is nearby and associated with a detection, keeping the particles from drifting (Fig. 13, bottom).

Occasionally, as in Fig. 13(c), the target of the yellow tracker becomes partially visible. However, the detector cannot accumulate enough evidence to detect the target, and it issues many detections on the approaching crowd (top). The data association algorithm manages to distinguish between the targets using the classifier. Thus, the detections are not wrongly associated to the yellow tracker, preventing the algorithm from switching identities. Moreover, the yellow tracker accurately locates the partially visible target (bottom), thanks to the classifier confidence output (middle). The particles are guided temporally by the detector confidence (Fig. 13, top), until a high-confidence detection is associated again (Fig. 13(d)).

In such situations, both a pure classifier based tracking approach (*e.g.*, [2], [17]) and a pure detector confidence based approach (*e.g.*, [8], [27], [46]) would probably fail, resulting in lost targets and identity switches.

Robustness of Observation Model. We demonstrate the influence of each observation model term by evaluating all possible combinations of terms on the ETH Central dataset. To this end, the respective weights β, γ, ν in Eq. (6) were set to zero, while the others remained identical. Table 2 and the respective rows in Fig. 14 show the results (Tab. 2(a) repeats the original result from Fig. 8).

Overall, the performance is highest when using all terms, and it decreases the more terms are removed. When removing the classifier term, three effects can be observed (Tab. 2(b), Fig. 14(b)): target localization is not as precise as before; some targets are lost (*e.g.*, the blue tracker), which increases the false positive and false negative rates; and the number of identity switches increases, as new trackers are initialized for lost targets. If the detector confidence term is removed instead (c), the number of false negatives is higher. In contrast, the number of identity switches and false positives remain lower, as the classifier confidence helps distinguish the targets. Tab. 2(d) and Fig. 14(d) show that without the discrete detections, more targets are missed, while the precision of the targets that are tracked remains quite high.

When relying only on the discrete detections alone, the tracker fails regularly, especially in the case of frequent occlusions and ambiguities (e). The number of false positives is lower (f) when using only the detection confidence instead, as the tracker is not immediately misguided by wrong detections. However, the tracker hardly recovers after failure. Finally, the performance is lowest when relying on the classifier term alone (g), probably because the classifiers are primarily trained to distinguish between the tracking targets, not between background and targets.

Number of Particles. To demonstrate the contribution of the particle filter, we evaluate the algorithm on the ETH

Central dataset and decrease the number of particles N from originally 100 to 25, 15, 10, 5, and 1. As can be seen from the results in Tab. 2(h)-(l), especially the false positive rate and the number of identity switches drastically increase, as many targets are lost and thus new trackers are initialized. Therefore, the capability of a particle filter to estimate a multi-modal distribution seems to be important for correctly tracking targets in such challenging scenarios. Although other, probably more powerful statistical frameworks exist, we refer to specific papers on this topic, as a detailed evaluation of different frameworks is beyond the scope of this paper.

Trust in Detections. As described in Sec. 3.5, our algorithm uses only few discrete detections for tracking. Since the detector output is often very noisy and thus not reliable, our algorithm aims at selecting those detections that are a good match. In Tab. 2(m)-(n), we show how tracking performance varies as a function of the detection threshold τ from Algorithm 1 (all other experiments are carried out with $\tau = 1$). As more detections are associated to trackers by decreasing τ to 0.5 and 0.2, the number of misguided trackers increases, which can be seen from the false positive rate.

6 Conclusion

We have presented a novel method for online multi-object tracking-by-detection, exploring the capabilities of an approach that relies only on 2D image information from one single, uncalibrated camera, without any additional scene knowledge. The main challenge for tracking algorithms are unreliable measurements, *i.e.*, in the case of tracking-by-detection, false positives and missing detections. The contribution of our work is thus to explore how this unreliable information source can be used for robust multi-person tracking. The key factors of our algorithm are: (1) careful selection and association of final detections using target-specific classifiers trained during run-time, (2) utilization of the continuous output of detector and classifier, and (3) robust combination of unreliable information for multi-person tracking using particle filtering.

While the data association algorithm handles false positive detections, different observation model terms help overcome problems with missing detections. They are complementary, as they are trained on different features and training data. While instance-specific information is beneficial to resolve ambiguous situations between different targets, class-specific knowledge helps differentiate between object and background.

For this purpose, the detector confidence term guides the particles of the filter primarily when no discrete high-confidence detection is issued by the detector. Although this is beneficial for situations with missing detections, it can also misguide trackers to image areas with high confidence on background structures. On the other hand, the classifier term helps localize particles more accurately, adapting online to the appearance of the targets. However, the classifier requires some amount of training data to work reliably and hence does neither help in situations shortly after initialization nor if the appearance of a target changes heavily during occlusions.

Our experiments have shown that the method achieves a good performance on a large variety of application scenarios, outperforming other state-of-the-art algorithms, some of which rely on scene-specific information, multiple calibrated cameras, or global optimization. To increase the robustness during partial occlusions, a part-based detector would be beneficial. Also, the detector could be trained for specific applications and the motion model could be specialized, *e.g.*, for applications in sports television broadcasting. Furthermore, if applied to a specific scenario, scene-specific information could be used to help resolve ambiguities, restricting motion to a ground plane or providing information about obstacles. Finally, the method could be enhanced by taking advantage of a more sophisticated estimation framework than particle filtering.

Acknowledgments

This research has been funded by the EU project HERMES (IST-027110). We thank H. Grabner, K. Schindler, A. Ess, M. Andriluka and K. Okuma for their code or dataset. The soccer dataset is courtesy of LiberoVision and Teleclub.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Comp. Vision and Pattern Rec.*, 2008.
[2] S. Avidan. Ensemble tracking. *IEEE T. Pattern Anal. and Machine Intell.*, 29(2):261–271, 2007.
[3] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *IEEE Comp. Vision and Pattern Rec.*, 2006.
[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *J. Image and Video Processing*, (3):1–10, 2008.
[5] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Markovian tracking-by-detection from a single, uncalibrated camera. In *IEEE Workshop Performance Evaluation of Tracking and Surveillance*, 2009.
[6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE Int. Conf. Comp. Vision*, 2009.
[7] Y. Cai, N. de Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *Eur. Conf. Comp. Vision*, 2006.
[8] R. Choudhury, C. Schmid, and K. Mikolajczyk. Face detection and tracking in a video by propagating detection probabilities. *IEEE T. Pattern Anal. and Machine Intell.*, 25(10):1215–1228, 2003.
[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Comp. Vision and Pattern Rec.*, 2005.
[10] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.
[11] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust multi-person tracking from a mobile platform. *IEEE T. Pattern Anal. and Machine Intell.*, 31(10):1831–1846, 2009.
[12] J. Ferryman. IEEE Workshop Performance Evaluation of Tracking and Surveillance, 2009.
[13] T. Fortmann, Y. Bar Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Oceanic Engineering*, 8(3):173–184, 1983.
[14] S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, and L. Van Gool. Articulated multibody tracking under egomotion. In *Eur. Conf. Comp. Vision*, 2008.
[15] J. Giebel, D. Gavrila, and C. Schnörr. A bayesian framework for multi-cue 3d object tracking. In *Eur. Conf. Comp. Vision*, 2004.
[16] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE P. Radar Signal Processing*, 140:107–113, 1993.
[17] H. Grabner and H. Bischof. On-line boosting and vision. In *IEEE Comp. Vision and Pattern Rec.*, 2006.
[18] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Eur. Conf. Comp. Vision*, 2008.
[19] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *Int. J. Comp. Vision*, 29(1):5–28, 1998.
[20] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Eur. Conf. Comp. Vision*, 2006.

[21] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE T. Pattern Anal. and Machine Intell.*, 27(11):1805–1918, 2005.

[22] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quaterly*, 2:83–87, 1955.

[23] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *IEEE Comp. Vision and Pattern Rec.*, 2010.

[24] O. Lanz. Approximate Bayesian multibody tracking. *IEEE T. Pattern Anal. and Machine Intell.*, 28(9):1436–1449, 2006.

[25] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int. J. Comp. Vision*, 77(1-3):259–289, 2008.

[26] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE T. Pattern Anal. and Machine Intell.*, 30(10):1683–1698, 2008.

[27] Y. Li, H. Ai, C. Huang, and S. Lao. Robust head tracking based on a multi-state particle filter. In *IEEE Conf. Face and Gesture Rec.*, 2006.

[28] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. *IEEE T. Pattern Anal. and Machine Intell.*, 30(10):1728–1740, 2008.

[29] Y. Li, C. Huang, and H. Ai. Tsinghua face detection and tracking for clear 2007 evaluation. In *Proceedings of International Workhop on Classification of Events, Activities and Relationships (CLEAR)*, 2007.

[30] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *IEEE Comp. Vision and Pattern Rec.*, 2009.

[31] K. Mardia. *Statistics of directional data*. Acad. Press, 1972.

[32] S. J. McKenna and H. Nait-Charif. Tracking human motion using auxiliary particle filters and iterated likelihood weighting. *Image and Vision Computing*, 25(6):852–862, 2007.

[33] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Eur. Conf. Comp. Vision*, 2004.

[34] PASCAL VOC, 2009. http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

[35] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE Int. Conf. Comp. Vision*, 2009.

[36] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *IEEE Comp. Vision and Pattern Rec.*, 2006.

[37] V. Prisacariu and I. Reid. Fasthog - a real-time GPU implementation of HOG. Technical Report 2310/09, Department of Engineering Science, Oxford University, 2009.

[38] C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE T. Pattern Anal. and Machine Intell.*, 23(6):560–576, 2001.

[39] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, 24(6):843–854, 1979.

[40] X. Ren. Finding people in archive films through tracking. In *IEEE Comp. Vision and Pattern Rec.*, 2008.

[41] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE Int. Conf. Robotics and Automation*, 2001.

[42] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *Eur. Conf. Comp. Vision*, 2008.

[43] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *IEEE Int. Conf. Comp. Vision*, 2003.

[44] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele. Sliding-windows for rapid object class localization: A parallel technique. In *DAGM Annual Pattern Rec. Symposium*, 2008.

[45] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comp. Vision*, 75(2):247–266, 2007.

[46] B. Wu, L. Zhang, V. K. Singh, and R. Nevatia. Robust object tracking based on detection with soft decision. In *IEEE Workshop Motion and Video Computing*, 2008.

[47] J. Yang, Z. Shi, P. Vela, and J. Teizer. Probabilistic multiple people tracking through complex situations. In *IEEE Workshop Performance Evaluation of Tracking and Surveillance*, 2009.

[48] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Comp. Vision and Pattern Rec.*, 2008.

[49] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE T. Pattern Anal. and Machine Intell.*, 26(9):1208–1221, 2004.

Michael D. Breitenstein received the MSc and PhD degrees from ETH Zurich in 2006 and 2009. He did part of his research at Mitsubishi Research and Oxford University. Michael has published over ten articles in peer-reviewed journals and major conferences, and is inventor of several patents. He received the Innovation Award 2009 from the Swiss IT Society for his research. Michael is currently working for a professional services firm.

Fabian Reichlin received the MSc degree in electrical engineering and information technology from ETH Zurich in 2009. During his studies, he attended courses at KTH Stockholm, Sweden, and worked as a software engineer for an IT services company in Switzerland. His research interests include object tracking, camera calibration and information retrieval. He is currently with LiberoVision AG, Zurich, Switzerland.

Bastian Leibe is an assistant professor at RWTH Aachen University. He received the diploma, MSc, and PhD degrees from the University of Stuttgart (2001), the Georgia Institute of Technology (1999), and ETH Zurich (2004). He received several awards for his research work, including the ETH medal and DAGM Main Prize in 2004, the CVPR Best Paper Award in 2007, the DAGM Olympus Prize in 2008, and the ICRA Best Vision Paper Award in 2009.

Esther Koller-Meier received her Master's degree in Computer Science in 1995 and the PhD in 2000 from ETH Zurich. Currently, she is working as a Postdoc within the Computer Vision Lab of ETH Zurich. Her research interests include object tracking, gesture analysis and multi-camera systems.

Luc Van Gool received the PhD degree in electrical engineering from the Katholieke Universiteit Leuven in 1991. He is currently a full professor of computer vision at the Katholieke Universiteit Leuven and ETH Zurich. He has coauthored more than 250 papers and received best paper awards at ICCV 98 and CVPR 07. He has been a program committee member and area chair of several major vision conferences and was program cochair of ICCV 05.