Robust Tracking-by-Detection using a Detector Confidence Particle Filter

Michael D. Breitenstein[1] Fabian Reichlin[1] Bastian Leibe[1,2] Esther Koller-Meier[1] Luc Van Gool[1,3]
[1]Computer Vision Laboratory ETH Zurich [2]UMIC Research Centre RWTH Aachen [3]ESAT-PSI / IBBT KU Leuven

Abstract

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. A main contribution of this paper is the exploration of how these unreliable information sources can be used for multi-person tracking. The resulting algorithm robustly tracks a large number of dynamically moving persons in complex scenes with occlusions, does not rely on background modeling, and operates entirely in 2D (requiring no camera or ground plane calibration). Our Markovian approach relies only on information from the past and is suitable for online applications. We evaluate the performance on a variety of datasets and show that it improves upon state-of-the-art methods.

1. Introduction

The goal of the work presented in this paper is to automatically detect and track a variable number of targets in complex scenes using a monocular, potentially moving, uncalibrated camera. This is a very challenging problem, since there are many sources of uncertainty for the object locations, *e.g.*, measurement noise, clutter, changing background, and significant occlusions. In order to cope with those difficulties, *tracking-by-detection* approaches have become increasingly popular [2, 11, 16, 20, 26]. Such approaches involve the continuous application of a detection algorithm in individual frames and the association of detections across frames. The main challenges when using an object detector for tracking are that the resulting output is unreliable and sparse, *i.e.*, detectors only deliver a discrete set of responses and usually yield false positives and missing detections (see Fig. 1, left). Several recent multi-object tracking-by-detection algorithms address the resulting data association problem by optimizing detection assignments over a larger temporal window [1, 3, 12, 16, 21]. They use information from future frames and locate the targets in the current frame with a temporal delay.

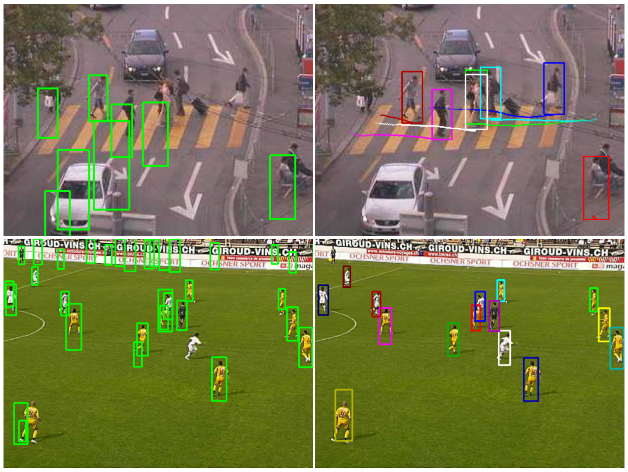


Figure 1: Using the output of a person detector (left), which typically contains many false positives and missing detections, our algorithm robustly tracks multiple targets in complex scenes (right).

Sequential Monte Carlo methods (or Particle Filters) [8] offer a framework for representing the tracking uncertainty in a *Markovian* manner by only considering information from past frames. Therefore, such an approach is more suitable for time-critical, online applications. Okuma *et al*. [20] and Cai *et al*. [5] combine tracking-by-detection with particle filtering by using final detections to initialize color based tracker samples. We also adopt a particle filtering framework, but we extend previous methods by several new ideas.

Most importantly, the above approaches only rely on the final, sparse output from the object detector. In contrast, our approach integrates the object detector itself into the tracking process by monitoring its *continuous detection confidence* and using it as a graded observation model. This idea follows the intuition that by forgoing the hard detection decision, we can impart tracking approaches with more flexibility to handle difficult situations. Although such a combination appears desirable, it raises a number of questions. As available object detectors have only been optimized for accurate results at those locations passing the final non-maximum suppression stage, it is not guaranteed that the shape of the underlying confidence volume in-between those locations will always support tracking. In addition, a

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

majority of the densities' local maxima correspond to false positives that may deteriorate the tracking results.

A main contribution of this paper is the exploration of 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: In addition to a general, class-specific *pedestrian detector* to localize people, we train *person-specific classifiers*. As our experiments show, the resulting approach yields a good tracking performance in a large variety of highly dynamic scenarios.

Our approach automatically initializes a separate particle filter for each person detected with high confidence. In order to resolve the data association of final high-confidence detections and trackers in each frame, our approach evaluates a scoring function integrating the online-trained classifier, the distance to the tracked target, and a probabilistic gating function accounting for the target size, motion direction, and velocity. If a final detection is classified as reliable based on this function, it is mainly used to guide the particles of the associated tracker. Otherwise, the continuous confidence of the class-specific detector and the instance-specific classifiers is mainly used. To evaluate the reliability of the detector confidence, we perform explicit inter-object occlusion reasoning. Finally, the algorithm computes the observation likelihood function of each particle filter using the associated detection, the intermediate output of the detector, and the classifier evaluated at each particle location. For computational efficiency, we limit ourselves to a first-order *Markov* model, *i.e.*, all data associations for time t are made at t and never reconsidered afterwards.

This paper makes the following contributions: (1) We present a novel approach for probabilistic tracking-by-detection in a Particle Filtering framework. (2) Our approach exploits the continuous detector confidence for robust multi-target tracking and integrates it into the observation model. (3) In order to deal with unreliable detections, we combine this input with online-trained classifiers to resolve the data association. The resulting combination effectively integrates generic category knowledge with person-specific information, thereby greatly improving tracker robustness and reducing classifier drift. (4) We present extensive experiments demonstrating that the proposed approach is applicable to a wide variety of tracking scenarios ranging from surveillance settings to highly dynamic sports scenes.

2. Related Work

A vast amount of work has been published on multi-target tracking, and a review is beyond the scope of this paper. While many approaches rely on background subtraction from one or several static cameras [3, 14, 23, 27], recent progress in object detection has increased interest in combining tracking and detection [2, 16, 20, 26]. However, many methods rely on global optimization to construct consistent trajectories from detections [1, 3, 12, 16, 21], which precludes their use in an online scenario.

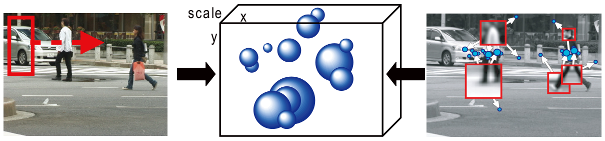
To better represent the state space uncertainty of a target, Particle Filters were introduced to the vision community by [13]. Later extensions include a representation of the joint state space for multiple targets [24] and the combination with an object detector for Markovian tracking-by-detection [10, 20]. As runtime directly scales with the number of particle evaluations, those approaches face a dilemma when additional targets appear. They can either spend an exponentially growing number of particles on representing the joint state space sufficiently well, or they can guarantee a constant runtime by keeping the number of particles fixed, at the price of lowering approximation accuracy. This can be solved by using independent particle sets for each target [5, 23], at the cost of potential problems with occlusions.

Using independent trackers requires solving a data association problem to assign measurements to targets. Classical approaches include JPDAF [9] and MHT [22]. However, the computational complexity grows exponentially with the number of targets and time steps, respectively. We stick to a greedy scheme for making the detection-tracker assignments and focus on obtaining a good scoring function instead. Such an approach is also used by [5, 26], but there the assignments are made only based on 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. [26] additionally learn color histograms for each part, which however do not always distinguish very well. In contrast, our method additionally evaluates the output of online-trained classifiers and a probabilistic gating function for reliable detection-tracker assignments. Recently, a background subtraction based tracker has been presented that also learns target-specific classifiers online [23], but employs them only when targets split and merge.

Previous approaches exist that exploit the detection confidence, but they are developed primarily for single-target tracking and do not show a thorough evaluation for multi-target tracking. *E.g.*, [2, 11] exploit the confidence map of a classifier, [17] apply classifiers with different confidence thresholds, and [6] accumulate detection probabilities temporally. In contrast, our algorithm is designed for robust multi-target tracking, addressing its specific problems.

3. Approach

For many tracking applications, approaches should rely only on past observations. Within this context, a Particle Filter (PF) estimates the time-evolving posterior distribution of the target locations with a weighted set of particles. PFs consist of a dynamic model for prediction and an observation model to evaluate the likelihood of a predicted state.



(a) Sliding window (b) Detector conf. density (c) Feature voting

Figure 2: The *detector confidence density*, which we use for the observation model of the particle filter, is common for sliding-window based and voting based detectors.

In this paper, we explore the idea of using the output of an object detector for the observation model. A general problem with this is the reliability of the resulting detections; *i.e.*, not all persons are detected in each frame (*missing detections*) and some detections are not caused by a person (*false positive detections*). These problems are demonstrated in Fig. 1. To address them, many recent methods rely on expensive global optimization techniques instead of making successive, irreversible decisions at each time step, which is a major limitation for time-critical applications.

In contrast, our algorithm for tracking-by-detection implements a first-order Markov model (*i.e.*, it only relies on information from the current and the last time step). It achieves the necessary robustness by integrating the information from object detection in two ways: First, our algorithm carefully assesses the detections in each frame and maximally selects one to guide the tracker for a particular target. For this purpose, we train a classifier for each target during runtime (see Sec. 3.2). Second, our algorithm additionally exploits the intermediate, continuous confidence density of an object detector (see next paragraph). To assess its reliability, our algorithm performs inter-object occlusion reasoning (see Sec. 3.3). Finally, the tracker integrates both types of detector output into the observation likelihood.

Detection Confidence Density. At the core of our approach lies the realization that current state-of-the-art person detectors all build up some form of *confidence density* as one stage of their pipeline. This is true for both sliding-window based detectors such as HOG [7] and for feature-based detectors such as ISM [15]. 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 (see Fig. 2). In order to arrive at individual detections, both types of approaches try to find 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-quality hypotheses, but it also throws away potentially useful information.

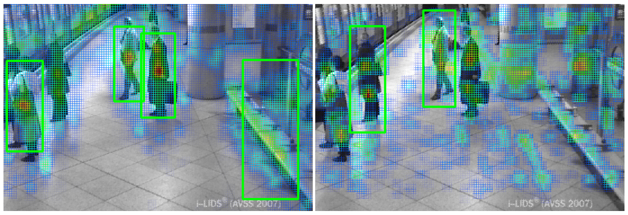


Figure 3: Detector output of ISM (left) and HOG (right). We show final detections as green bounding boxes and the detector confidence density in blue (low) to red (high). The density often contains useful information at the location of missing detections.

Fig. 3 illustrates both types of output for ISM (left) and HOG (right). 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 intermediate output. On the other hand, both detectors also show a high confidence density on certain background structures.

3.1. Particle Filtering

Bootstrap Filter. 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). Therefore, we employ the *bootstrap filter*, where the state transition density (or "prior kernel") is used as importance distribution to approximate the probability density function. For sequences with abrupt, fast camera motion, we apply iterative likelihood weighting [19]. For details about particle filtering, we refer to [8]. The importance weight $w_t^{(i)}$ for each particle i at time step t is then described by:

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

Since re-sampling is carried out in each time step using a fixed number of N particles, $w_{t-1}^{(i)} = \frac{1}{N}$ is a constant and can be ignored. Thus, (1) reduces to the conditional likelihood of a new observation y_t given the propagated particles $\boldsymbol{x}_t^{(i)}$, which we estimate as described in Sec. 3.3.

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

$$\begin{aligned}(x, y)_t &= (x, y)_{t-1} + (u, v)_{t-1} \cdot \Delta t + \varepsilon_{(x,y)} &(2)\\ (u, v)_t &= (u, v)_{t-1} + \varepsilon_{(u,v)}. &(3)\end{aligned}$$

The process noise $\varepsilon_{(x,y)}, \varepsilon_{(u,v)}$ for each state variable is independently drawn from zero-mean Normal distributions. The variance $\sigma^2_{(x,y)}$ for the position noise changes with the size of the tracking target, whereas the variance $\sigma^2_{(u,v)}$ for the velocity noise is inversely proportional to the number of successfully tracked frames. Hence, the longer a target is tracked successfully, the less the particles are spread. Δt is dependent on the frame-rate of the sequence. The size of the target is estimated as described below.

Tracker Initialization and Termination. Object detection yields fully automatic initialization. The algorithm initializes a new tracker for an object that has subsequent

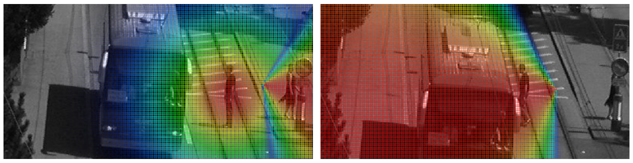


Figure 4: The result of the probabilistic gating function (Eq. 5) depends on the velocity of the target, resulting in radial isolines (left) or different 2D cone angles (left, right).

detections with overlapping bounding boxes, which are neither occluded nor associated to an already existing tracker (see Sec. 3.2). 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 (see Sec. 4). The initial sample positions are drawn from a Normal distribution centered at the detection center. 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.

Tracker Position and Size. Although represented by a (possibly multi-modal) distribution, a single position of the tracking target is sometimes required (*e.g.*, for visualization). We estimate the position by the strongest mode of the distribution, found using mean-shift. 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 yields better results, possibly because of the exponentially growing number of particles necessary for estimating a larger state space.

3.2. 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. In our experiments, we however found that the following greedy algorithm achieves equivalent results at lower computational cost (also reported by [26]).

The matching algorithm works as follows: 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 is associated with a target, but if one is, it can be used to strongly influence the tracker.

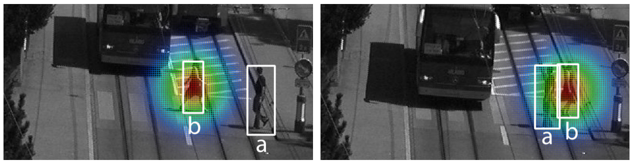


Figure 5: The detector confidence reliability function (Eq. 7) evaluated for tracker a returns a high value if another tracker b with associated detection is nearby (right).

Matching Score. Our data association method evaluates a matching function for each tracker-detection pair (tr, d) (the higher the score, the better the match between detection and tracking target). The matching function $s(tr, d)$ evaluates the distance between the detection d and each particle p of tracker tr. It employs a classifier $c_{tr}(d)$ trained for tr and evaluated for d (see the following paragraphs):

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

where $p_{\mathcal{N}}(d - p) \sim \mathcal{N}(d - p; 0, \sigma^2_{det})$ denotes the Normal distribution evaluated for the distance between d and p, and $g(tr, d)$ is a gating function described next.

Gating Function. Not only the distance of a detection to the tracker is important, but also its location with respect to velocity and motion direction of the target. Therefore, a *probabilistic gating function* $g(tr, d)$ additionally assesses each detection. It consists of the product of two probabilities, both drawn from Normal distributions.

$$\begin{aligned} g(tr, d) &= p(size_d | tr) p(pos_d | tr) \qquad (5) \\ &= \begin{cases} p_{\mathcal{N}}(\frac{size_{tr} - size_d}{size_{tr}}) \cdot p_{\mathcal{N}}(|d - tr|) & \text{if } |\boldsymbol{v}_{tr}| < \tau_v \\ p_{\mathcal{N}}(\frac{size_{tr} - size_d}{size_{tr}}) \cdot p_{\mathcal{N}}(dist(d, \boldsymbol{v}_{tr})) & \text{otherwise.} \end{cases} \end{aligned}$$

The first factor measures the agreement between the sizes of target and detection. The second term implements the intuition that fast moving objects cannot change their course so abruptly because of their inertia. Therefore, it depends on the velocity of the target; if it is below a threshold τ_v, the velocity is ignored and the term is proportional to the distance from the tracker position tr to the detection d. In this case of an (almost) motionless target, the isolines of the function are radial (see Fig. 4, left). Otherwise, the second term depends on $dist(d, \boldsymbol{v}_{tr})$, which is the shortest distance between the detection d and the line given by the motion direction $\boldsymbol{v}_{tr} = (u, v)$ of the tracker. The variance for the second term is chosen such that it is proportional to the distance from the tracker to the detection projected to $\boldsymbol{v}_{tr}$. In this case, the isolines of Eq. 5 form a 2D cone (see Fig. 4). The variance is made inversely proportional to the velocity, such that the angle of the 2D cone is smaller the higher the speed of a target is (see Fig. 4).[1]

[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 [18].

Boosted Classifiers. To assess the similarity of a tracker-detection pair, we train a boosted classifier of weak learners for each tracking target. The classifier is similar to [11] and is trained online on one target against all others. Patches used as positive training examples are sampled from the bounding box of the associated detection. The negative training set is sampled from nearby targets, augmented by background patches. The classifier is only updated on non-overlapping detections. After each update step, we keep a constant number of the most discriminative weak learners. The output of the classifier is linearly scaled to the range $[-1, 1]$. The weak learners (feature types) are selected by evaluating the classifier for different combinations of color and appearance features (see Sec. 4).

3.3. Observation Model

To compute the weight $w_{tr,p}$ for a particle p of the tracker tr, our algorithm estimates the conditional likelihood of the new observation given the propagated 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}:

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

where the parameters β, γ, η are set experimentally. 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. When a matching detection is found, this term robustly guides the particles.

Detector Confidence Density 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 [15], we compute the local density ρ in the Hough voting space using a cubical kernel adapted to the target object size and scale it with $f = 1 - exp(-\rho)$ to $[0, 1]$. For the HOG detector [7], $d_c(p)$ corresponds to the raw SVM confidence output before applying non-maximum suppression.

Unfortunately, the detector confidence density is not always reliable; often, an erroneously high value is caused by background structures (see Fig. 3). 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 density around this image location is most probably caused by the foreground and not by background structure. Consequently, the detector probably did not find both targets because of the occlusion. In this case, we assume that the detection confidence density is meaningful in this image area and can be used to guide the tracker.

To assess the *reliability of the detector confidence density*, our tracking algorithm evaluates a function,

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

which is used in (6) to weight the influence of the detector confidence density. The closer tr' is, the more reliable is the detector output at the position of tracker tr. In Fig. 5, the function values are illustrated for the tracker of the person entering the scene from the right.

Classifier Term. For the third term of Eq. (6), the classifier trained for the target tr is evaluated for the image patch at the particle location with the corresponding size. This term uses color and texture information (see Sec. 3.2 and 4) 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 (*c.f.* [11]), this adds additional robustness to our particle filter approach. In addition, the combination of generic category knowledge with person-specific information makes our approach more robust to classifier drift than *e.g.* [11].

Influence of Observation Likelihood Terms. The influence of the different observation likelihood terms in (6) is demonstrated on a simple sequence in Fig. 6. As can be seen, certain background structures (tram tracks and road markings) cause false positive detections. In Fig. 6(a), both targets are correctly associated with a detection (dashed bounding boxes), based on which the trackers are mainly guided. In contrast, Fig. 6(b) shows a state where no detection is assigned to them; as a result, the particles are weighted (almost) uniformly. In Fig. 6(c), the particles of the blue tracker are weighted mainly based on the detector confidence density term, because no detection is available. As the red tracker is nearby and is associated with a detection, the detector confidence for the blue tracker is assumed to be high. Between the images in Figs. 6(d) and 6(e), a bus causes a long occlusion (for 50 frames) during which false positive detections are rejected by the classifier. This is essential for the success of the tracker.

Fig. 7 shows erroneous tracking results if the density or classifier term in Eq. (6) have too much influence (controlled by the parameters β, γ, η). In Fig. 7 (left), the tracker is misguided by the detection confidence density term. In Fig. 7 (right), the tracker is misguided because a part of the roof of the bus was visible in detections used for updating the classifier. In our experiments, we kept β and η fixed and only adapted γ for some sequences (as described below).

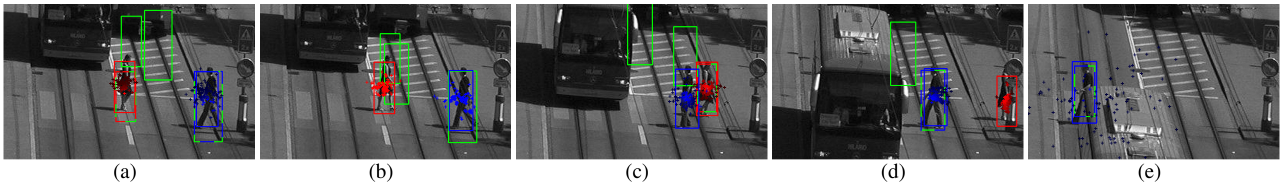


Figure 6: Example tracking results demonstrating the influence of the different observation likelihood terms in Eq. (6) (see text; *green*: ISM detection bounding boxes; *red* and *blue*: tracker particle sets, weights are proportional to color intensity).

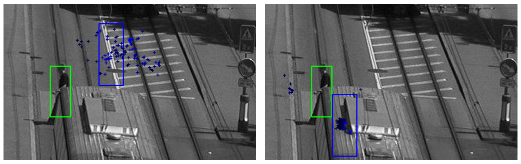


Figure 7: Erroneous tracking results if (a) the detector confidence density or (b) the classifier have too much influence (*c.f.* Fig. 6(e)).

4. Experimental Results

Experimental Setup. We evaluate the tracking algorithm on five challenging sequences: ETHZ Central [16], TUD Crossing [1], i-Lids AB [12], UBC Hockey [20], and our own Soccer dataset[2]. They are taken from both static and moving cameras and vary with respect to viewpoint, type of movement, and amount of occlusion, which demonstrates the robustness of our approach (see Fig. 9 and the videos). Unfortunately, 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.

In all sequences, we only use 2D information and do not assume any scene knowledge (*e.g.*, ground plane calibration or entry/exit zones, *c.f.*, [12, 16]). Sticking to the detectors originally used with these sequences, we employ the ISM detector [15] for ETH Central, TUD crossing, and UBC Hockey, and the HOG detector [7] for i-Lids and Soccer. We use the publicly available, pre-trained versions (*i.e.*, not specifically trained for any test sequence, *c.f.*, [20]). Given the detector output, the runtime of our unoptimized code is 2–0.4 fps (Intel Core2Duo 2.13GHz), depending on the number of detections and targets in a sequence.[3]

All parameters have been set experimentally, but most remained identical for all sequences. This was the case for the variances σ^2 in Eqs. (4)-(7), as well as for β, η in Eq. (6). γ was increased for TUD Crossing to overcome long-lasting overlaps between detections. Overall, β, γ, η were chosen such that the ratio between the respective terms are about 20:2:1 for a tracker with associated detection. On average, a detection is selected and associated to a tracker every 2–10 frames, depending on the sequence. To handle abrupt motion changes in the sports sequences, we increased σ^2 in Eqs. (2), (3) to make the motion model more flexible.

Dataset	Prec.	Accur.	F. Neg.	F. Pos.	ID Sw.
ETH Centr.	70.0%	**72.9%**	26.8%	0.3%	0
ETH Centr. [16]	66.0%	33.8%	51.3%	14.7%	5
UBC Hockey	57.0%	**76.5%**	22.3%	1.2%	0
UBC Hockey [20]	51.0%	67.8%	31.3%	0.0%	11
i-LIDS easy	67.0%	**78.1%**	16.4%	5.3%	18
i-LIDS med*	66.0%	**76.0%**	22.0%	2.0%	2
i-LIDS [12]	-	68.4%	29.0%	13.7%	-
i-LIDS [26]	-	55.3%	37.0%	22.8%	-
TUD Cross.	71.0%	**84.3%**	14.1%	1.4%	2
Soccer	67.0%	**85.7%**	7.9%	6.2%	4

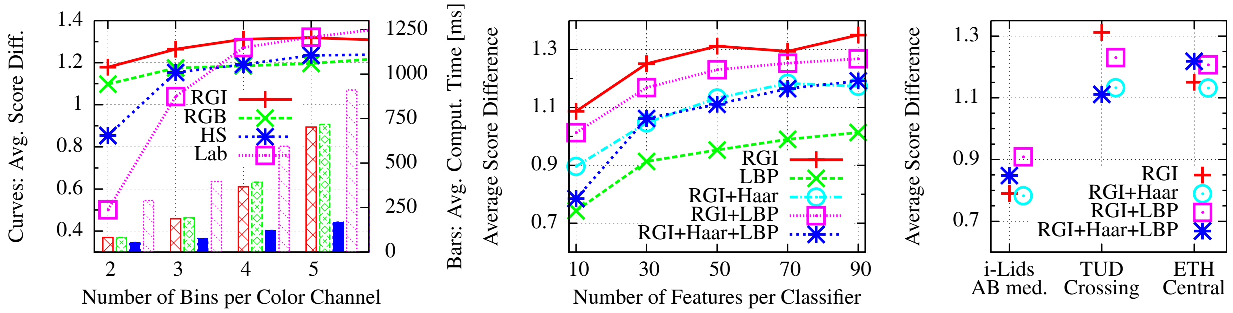
Table 1: CLEAR MOT results on 5 datasets demonstrate the performance of our algorithm compared to state-of-the-art methods.

Classifier Comparison. 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 other targets. For this purpose, we compare the classifiers on different sequences using annotated ground truth. Ideally, a classifier returns a score of $+1$ for the bounding box of the target it is trained for and -1 for all other input. Fig. 8 shows the difference between the classifier score on the annotated target and the highest score on all other targets for different features and combinations (RGI/RGB/HS/Lab=red-green-intensity/RGB/hue-saturation/Lab histograms; LBP=local binary patterns; Haar=Haar wavelets). The higher the score difference, the better is the ability of the classifier to distinguish between targets. In Figs. 8(a) and 8(b), we show a detailed evaluation for TUD Crossing. As can be seen, the number of features is not critical. However, the choice of color feature type and number of histogram bins heavily affects the result and the average computation time (which includes training and testing). Fig. 8(c) compares different feature combinations on three test sequences. Based on these evaluations, we use 50 RGI and LPB features with 3 bins per color channel for all sequences.

Tracking Evaluation. We use the CLEAR MOT metrics for evaluation. It returns a precision score (intersection over union of bounding boxes) and an accuracy score (composed of false negative rate, false positive rate, and number of ID switches). As can be seen in table 1, the sequences are tracked with high precision and accuracy. The false nega-

[2] The references indicate publications with state-of-the-art results.

[3] Note that the HOG detector can be implemented in real-time [25].



(a) Color feature type for TUD Crossing.

(b) Number of features for TUD Crossing.

(c) Feature combinations.

Figure 8: Evaluation for classifier feature selection. The plots show the difference between the classifier score on the annotated target and the highest score on all other targets for different features, averaged over all frames and targets.

tives occur if persons are annotated but not detected. This happens if a person is very close to another one (ETH Central, TUD Crossing), is sitting (ETH Central), or is partially outside of the image (i-Lids). The ID switches in i-Lids happen mainly if a person is occluded (*e.g.*, by the pillar) and a new tracker is initialized for a reappearing target. For sports sequences, the ability of the classifier to differentiate between players is decreased because of their similar appearance.

We compare our method with the state-of-the-art results reported for these sequences[4] (see Tab. 1): On ETH Central with [16] (using provided trajectories), on UBC Hockey with [20] (obtained using their publicly available Matlab code on their data), and on i-Lids as reported by [12][5]. In all cases, our precision and accuracy results outperform the previously published results, even though our algorithm does not use global optimization [12, 16], a detector specifically trained for the appearance in the sequence [20], camera calibration [16], or a scene model [12].

5. Conclusion

We have presented a novel approach for tracking-by-detection in a particle filtering framework. As our experiments show, the proposed approach achieves robust tracking performance in a large variety of application scenarios and outperforms previous methods. The key factors for this performance are: (1) a careful selection of the detections that influence a tracker; (2) use of an online trained classifier for data association; and (3) a combination of final detections, continuous detector confidence and classifier output to guide particles. While our approach only uses 2D information, additional knowledge about the scene (*e.g.*, a ground plane to improve detections), about the appearance of persons, or about the camera motion would be beneficial.

Acknowledgments: We acknowledge support by the EU project HERMES (IST-027110). We thank H. Grabner, K. Schindler, A. Ess, M. Andriluka and K. Okuma for their code or data. 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 *CVPR*, 2008.

[2] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

[3] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, 2006.

[4] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Markovian tracking-by-detection from a single, uncalibrated camera. In *CVPR PETS*, 2009.

[5] Y. Cai, N. de Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *ECCV*, 2006.

[6] R. Choudhury, C. Schmid, and K. Mikolajczyk. Face detection and tracking in a video by propagating detection probabilities. *PAMI*, 25(10):1215–1228, 2003.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2001.

[9] 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.

[10] J. Giebel, D. Gavrila, and C. Schnörr. A bayesian framework for multi-cue 3d object tracking. In *ECCV*, 2004.

[11] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.

[12] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008.

[13] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[14] O. Lanz. Approximate Bayesian multibody tracking. *PAMI*, 28(9):1436–1449, 2006.

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

[16] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

[4] Videos: www.vision.ee.ethz.ch/~bremicha/tracking. See additionally [4] for an evaluation on the PETS'09 dataset.

[5] We tested on i-Lids AB easy and the first half of i-Lids medium, for which we added annotations for fully visible, sitting persons (i-Lids med*).

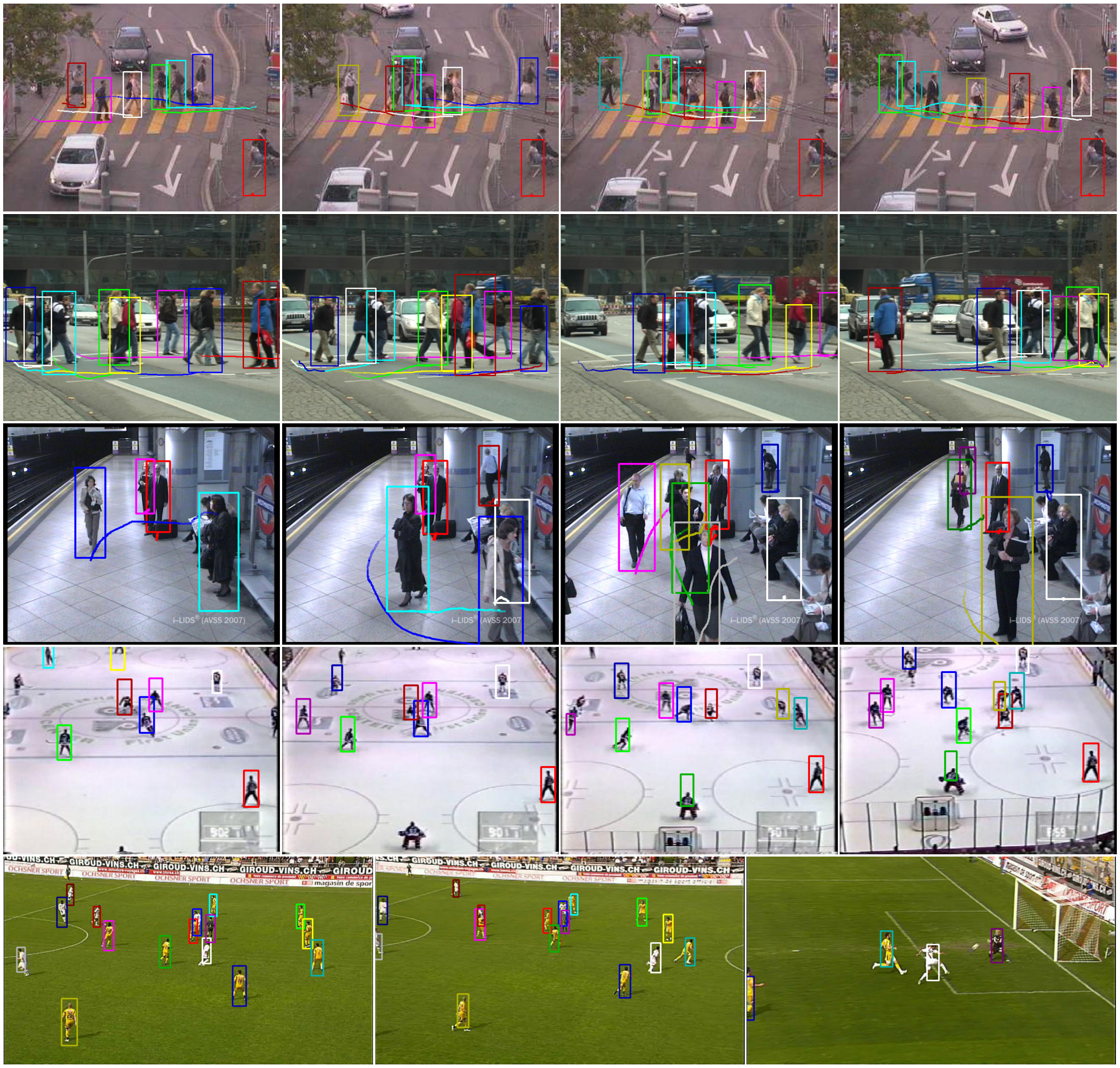


Figure 9: Tracking results on the ETHZ Central [16], TUD Crossing [1], i-Lids AB medium [12], UBC Hockey [20] and Soccer dataset.

[17] 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. *PAMI*, 30(10):1728–1740, 2008.

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

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

[20] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[21] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006.

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

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

[24] J. Vermaak, A. Doucet, and P. Perez. Maintaining multimodality through mixture tracking. In *ICCV*, 2003.

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

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

[27] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *PAMI*, 26(9):1208–1221, 2004.