Multi-target tracking on confidence maps: an application to people tracking

Fabio Poiesi*, Riccardo Mazzon, Andrea Cavallaro

Centre for Intelligent Sensing,
Queen Mary University of London, London, UK.

Abstract

We propose a generic online multi-target *track-before-detect* (MT-TBD) that is applicable on confidence maps used as observations. The proposed tracker is based on particle filtering and automatically initializes tracks. The main novelty is the inclusion of the target ID into the particle state, enabling the algorithm to deal with unknown and large number of targets. To overcome the problem of mixing IDs of targets close to each other, we propose a probabilistic model of target birth and death based on a Markov Random Field (MRF) applied to the particle IDs. Each particle ID is managed using the information carried by neighboring particles. The assignment of the IDs to the targets is performed using Mean-Shift clustering and supported by a Gaussian Mixture Model. We also show that the computational complexity of MT-TBD is proportional only to the number of particles. To compare our method with recent state-of-the-art works, we include a postprocessing stage suited for multi-person tracking. We validate the method on real-world and crowded scenarios, and demonstrate its robustness in scenes presenting different perspective views and targets very close to each other.

Keywords: Track-before-detect, crowd, multi-target tracking, Markov Random Field, Gaussian Mixture Model, Likelihood modeling.

* *Corresponding author*
Email address: fabio.poiesi@eecs.qmul.ac.uk (Fabio Poiesi)
URL: http://www.eecs.qmul.ac.uk/∼andrea/ (Andrea Cavallaro)

1. Introduction

Multi-target tracking is a challenging task in real-world scenarios due to the variability of target movements, shapes, clutter and occlusions. Moreover, the computational cost may exponentially increase with the number of co-occurring targets and the maximum number of targets may have to be fixed *a priori*. Single-target tracking generally represents the state of each target with a single state vector [1]. In multi-target tracking the size of state vector increases with the number of targets [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] unless a single-target tracker is initialized for each target [13, 14, 15, 16, 17, 18, 19, 20]. We refer to the former approach as *one-state-per-target* (OSPT) and to the latter *one-filter-per-target* (OFPT). OSPT methods perform tracking optimization at each time step on the overall state space. Only a predefined number of targets can be tracked [14] or ad-hoc stages can be used to estimate the number of targets in the scene [2, 5]. OFPT methods perform tracking by a local optimization for each target, thus limiting their application to situations with a small number of targets that are easily distinguishable.

Target locations may be gathered from sensors (e.g. laser, sonar, camera) via confidence maps that provide multiple measurements per target and carry information in the form of intensity levels over space (Fig. 1). These intensity levels are affected by different types of noise on background areas and/or on the targets themselves, thus resulting in inaccurate position estimations. Tracking algorithms employ target locations as measurements, either directly as confidence maps (unthresholded data) [13, 21, 20, 22] or as binary maps (target/non-target information) obtained by thresholding the confidence values [3, 4, 6, 10]. Although the latter strategy is the most commonly used, relevant data may be lost with this process. Tracking-by-detection methods [20] perform target-tracker association, and initialization and termination of tracks with greedy algorithms. Track-before-detect (TBD) methods perform tracking of targets using unthresholded data [23] and target-tracker association is implicitly computed by the tracker. TBD is a Bayesian filter, generally built on the concept of particle filter, and commonly used for radar tracking [23, 24]. Multi-target tracking is performed on noisy intensity levels and the targets are assumed to be point targets. Initialization and termination of tracks are performed by the tracker using target *birth and death* models.

In this paper we propose a novel multi-target tracker based on TBD algorithm [23] and applied to confidence maps. To enable multi-target tracking, we develop a method where target IDs are assigned based on Mean-Shift clustering and Gaussian Mixture Model (GMM). The birth and death of targets are mod-

Figure 1: Sample confidence map that we use as input (observation) to simultaneously track multiple objects. In this example, the confidence map is obtained with a head localization method based on [25].

eled with a Markov Random Field (MRF). Unlike [24], we do not need to define the maximum number of targets *a priori* and, unlike [20], the initialization of a track may occur in any location of the image, thus making the multi-target *track-before-detect* (MT-TBD) automatic and flexible to different scenarios. MRF enables multi-target tracking without augmenting the state (OSPT methods, e.g. [2]) or the number of filters (OFPT methods, e.g. [13]), caused by an increase in the number of targets. Moreover, the use of MRF overcomes the limitations of [24] by allowing a reliable tracking of close targets without loss of performance and leads to a computational complexity depending only on the number of particles. Compared to the recent work by Benfold and Reid [10], the tracking accuracy of the proposed MT-TBD improves by 11% with 2 seconds of latency and by 10% with 4 seconds of latency on a publicly dataset from Oxford town center.

The paper is organized as follows. Section 2 discusses the related work for multi-person tracking. Section 3 gives an overview of the proposed approach and introduces MT-TBD. The ID management via MRF is explained in Sec. 4. Section 5 illustrates the application of MT-TBD to multi-person tracking. Section 6 discusses the experimental results, the comparisons with existing methods and the analysis of the computational complexity. Finally, in Sec. 7 we draw the conclusions and present possible research directions.

2. Related work

In this section we discuss recent works on multi-person tracking, we analyze their main contributions and classify each method in its corresponding category. Multi-target video trackers can be classified into causal and non-causal methods. *Causal* methods use information from past and present observations to estimate trajectories at the current time step. *Non-causal* methods use also information from future observations, thus resulting in a delayed decision. Although non-causal approaches are not suitable for time-critical applications, they can achieve a global optimum leading to more robust results during occlusions.

Examples of causal trackers are Bayesian filters [17, 10, 16, 15, 20]. Yang *et al.* [17] use a Bayesian-based detection association obtained by Convolutional Neural Network (CNN) trained on color histograms, elliptical head model, and bags of SIFTs. Benfold and Reid [10] find the optimum trajectories within a four-second window by a Minimum Description Length (MDL) method applied on trajectories from a forward and backward Kanade-Lucas-Tomasi (KLT) tracking and from a Markov Chain Monte Carlo Data Association (MCMCDA). Alternatively, the particle filter is used in [16, 15, 20]. Ali and Dailey [16] track heads obtained by Haar-like features and AdaBoost; whereas Xing *et al.* [15] employ the Hungarian algorithm for the optimization of short but reliable trajectories obtained by tracking the upper human body. Depending on the scenario, Breitenstein *et al.* [20] track people detected by Histogram of Oriented Gradients (HOG) or Implicit Shape Model (ISM). Here the association between detections and tracks is performed by a greedy algorithm and boosting. A different approach is presented in Rodriguez *et al.* [7] where tracking is obtained on four points per head by KLT and head detection is optimized by crowd density estimation and camera-scene geometry. Tag-and-track methods for high-density crowd are proposed in [26, 27], where targets are assumed to follow a learned crowd behavior. Ali and Shah [26] deal with crowds with coherent motion by modeling their global behavior, the environment structure and the local behavior of people. Rodriguez *et al.* [27] focus on crowds with non-coherent motion where the modeling is performed by Correlated Topic Model (CTM) that predicts the next position of a person by exploiting the optical flow. Note that among causal methods, only Benfold and Reid [10] and Rodriguez *et al.* [7] use an OSPT framework. This is because the OSPT is generally more complex than OFTP, but the modeling for multi-person tracking is more flexible and computationally cheaper [10].

As for non-causal trackers, short-term tracks (tracklets) [3, 4, 8, 6, 9, 11, 12] can be associated over time by using a modification of the Multi-Hypothesis

Tracking (MHT) algorithm [28], where the detections are obtained with a person detector [29]. Huang *et al.* [3] associate tracklets by Hungarian algorithm using position, time and appearance features, and then refine them using entry and exit points in the scenes, which are in turn learned from tracklets. Li *et al.* [4] show how the association can be improved by using a combination of RankBoost and AdaBoost in a hierarchical approach where longer trajectories are generated using a set of 14 features per tracklet by starting from the lower levels. In Yang *et al.* [8], the association is performed using RankBoost applied to an optimization of affinities and dependencies between tracklets by a Conditional Random Field (CRF). Kuo *et al.* [6] associate tracklets using an AdaBoost classifier that learns online the discriminative appearance of targets based on their color histogram, covariance matrix features and HOG. Kuo *et al.* [9] extract motion, time and appearance from different body parts of each target in order to perform a re-identification step to resolve long-term occlusions. Yang and Nevatia [11] learn online the non-linear motion of people and a Multiple Instance Learning (MIL) framework for the appearance modeling using the estimation of entry and exit regions. Furthermore, Yang and Nevatia [12] use CRF to model affinity relationships between tracklet pairs, where the association of tracklets is based on Hungarian algorithm and a heuristic search. Table 1 summarizes the methods covered in this section and the dataset on which these methods have been tested.

Similarly to Stalder *et al.* [21] and Breitenstein *et al.* [20], the proposed MT-TBD is a causal method that makes use of confidence maps as measurement for tracking. However, compared to [21], we use the confidence maps online without the need of any temporal processing and, compared to [20], an automatic assignment between confidence map and targets is performed. Moreover, unlike [20], which uses manually selected areas at the borders of the image to initialize tracks, we do not use any prior information about the scene. This becomes extremely advantageous when targets temporarily undergo a total occlusion in any position of the image. In addition to this, we overcome the limitations of OFTP approaches [20, 22] with a global and instantaneous optimization of target tracking in MT-TBD by employing a general likelihood function obtained from a controlled sequence (Sec. 5.1). Finally, unlike De Leat *et al.* [22], the use of multiple measurements per target is tested in various crowded scenes with different camera perspectives.

Table 1: Summary of state-of-the-art methods for multi-person tracking and datasets used (see text for details). Key: CM = Confidence Map; OSPT = one-state-per-target; CRF = Conditional Random Field; OLDAMs = Online Learning of Discriminative Appearance Models; PIRMPT = Person Identity Recognition based Multi-Person Tracking; MIL = Multiple Instance Learning; KLT = Kanade-Lucas-Tomasi feature tracker; MCMCDA = Markov-Chain Monte-Carlo Data Association; JPDA = Joint Probabilistic Data Association.

Ref.	Method	CM	OSPT	Causality	Dataset
[3]	Three-stage algorithm, Hungarian algorithm		✓		CAVIAR, iLids
[4]	HybridBoost		✓		CAVIAR, TRECVID
[8]	CRF, RankBoost		✓		TRECVID
[6]	AdaBoost on OLDAMs		✓		CAVIAR, TRECVID
[9]	PIRMPT		✓		CAVIAR, ETH, TRECVID
[11]	Learning of motion map, MIL for appearance		✓		CAVIAR, PETS2009, TRECVID
[12]	CRF, Hungarian algorithm/heuristic search		✓		ETH, TRECVID, TUD
[17]	Bayesian filter, Hungarian algorithm			✓	CAVIAR, TRECVID
[10]	KLT, MCMCDA	✓	✓	✓	iLids, PETS2007, TownCentre
[16]	Particle filter			✓	Bangkok station
[15]	Particle Filter, Hungarian algorithm			✓	CAVIAR, ETH
[20]	Particle filter, Greedy algorithm, Boosting	✓		✓	iLids, PETS2009, soccer, TDU campus, UBC Hockey
[7]	KLT points, Crowd density estimation	✓	✓	✓	Political rally
[26]	Floor fields			✓	Marathon, train station
[27]	Correlated Topic Model			✓	Mall, sport crowd
[22]	Automatic relevance detection, JPDA	✓	✓	✓	Ants, laser output
Our approach	Multi-target track-before-detect	✓		✓	Apidis, ETH, iLids, TownCentre, TRECVID

3. Sequential Monte Carlo estimation for multi-target track-before-detect

3.1. Confidence maps and track-before-detect

Let a confidence map $\mathfrak{M}$ provide the information on the estimated position of targets through spatially-localized intensity levels (Fig. 1). The ideal representation of the target position on a confidence map is a Dirac delta (a point target), with maximum confidence. In practice, such Dirac delta is a spread function centered in the target position and affecting neighboring pixels.

Let the state vector $\mathbf{x}_k \in \mathfrak{X}$, where $\mathfrak{X}$ is the state space, be defined as

$$\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ I_k]^T, \tag{1}$$

where (x_k, y_k) is the position, $(\dot{x}_k, \dot{y}_k)$ the velocity, I_k the intensity and T is the symbol for the transposed matrix. TBD is a time-discrete system that observes multiple moving targets on a 2D image. The evolution of the targets at each time step k is described by a discrete and linear Gaussian model [23]:

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \mathbf{v}_{k-1}. \tag{2}$$

The transition matrix F describes the evolution of the target at a constant velocity:

$$F = \begin{bmatrix} 1 & K & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & K & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where K denotes the sampling period. The noise of this evolution is normally distributed and defined as $\mathbf{v}_{k-1} \sim \mathcal{N}(0, Q)$, with

$$Q = \begin{bmatrix} \frac{q_1}{3}K^3 & \frac{q_1}{2}K^2 & 0 & 0 & 0 \\ \frac{q_1}{2}K^2 & q_1 K & 0 & 0 & 0 \\ 0 & 0 & \frac{q_1}{3}K^3 & \frac{q_1}{2}K^2 & 0 \\ 0 & 0 & \frac{q_1}{2}K^2 & q_1 K & 0 \\ 0 & 0 & 0 & 0 & q_2 K \end{bmatrix}, \tag{4}$$

where q_1 and q_2 are noise levels in target motion and intensity, respectively.

Let the spread function of the estimated positions of targets (over the 2D image) be modeled as

$$h_k^{(i,j)}(\mathbf{x}_k) = I_k \exp\left\{ -\frac{(i - x_k)^2 + (j - y_k)^2}{2\Sigma^2} \right\}, \tag{5}$$

where Σ is a known parameter that represents the amount of blurring (i.e. the spread of the confidence) and (i, j) is the pixel position.

The recursive Bayesian filtering involves the calculation of the *posterior* probability density function (pdf) $p(\mathbf{x}_k|\mathbf{Z}_k)$ of $\mathbf{x}_k$ given the observations up to time k, $\mathbf{Z}_k = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$. The posterior is calculated in two steps: *prediction* and *update*. In the prediction step, the probability density function is calculated through a prior distribution, which determines the state evolution through the motion model. In the update step, when the observation $\mathbf{z}_k$ is available, the prediction is updated using the likelihood function. The posterior pdf is thus obtained with the Bayesian recursion as

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}, \tag{6}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the *likelihood* function, $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ is the prediction density and $p(\mathbf{z}_k|\mathbf{Z}_{k-1})$ is a normalizing constant calculated as

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int_{\mathfrak{X}} p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k. \tag{7}$$

3.2. Multi-target identity

The framework for single-target tracking described in [23] (Ch. 11) includes in the state vector $\mathbf{x}_k$ an existence variable $E_k \in \{0, 1\}$, where 0 (1) indicates the absence (presence). The global existence over time (i.e. target birth and target death) of the target is modeled with a two-state Markov chain. The further extension to multi-target [14] leads to the expansion of the state vector $\mathbf{x}_k$ and of the Markov chain proportionally to the number of the targets. Since the number of states of a Markov chain is fixed, the maximum number of targets must be known *a priori*. In addition to this, the Markov chain may not allow transitions from zero to two targets, and vice versa [14]. Alternatively, birth and death of multiple targets can be modeled with greedy algorithms, where a target is declared born if the tracker receives its measurements within a certain period of time [30], or by a multi-Bernoulli distribution defining birth and death probabilities, and used to declare a target birth when the existence probability of a candidate target is larger than a certain threshold [31].

In order to be independent of the number of targets, we include in $\mathbf{x}_k$ the state variable ξ for representing the target identity (ID). IDs are represented by the set of random variables $\mathcal{L}_k = \{L_\xi\}_{\xi \in \Xi_k}$, where Ξ_k is the set of IDs at time k and $p(L_\xi = \xi) = p(L_\xi)$. The IDs within Ξ_k at time k depend on two factors: the IDs at $k-1$ and $\mathbf{x}_k$. Hence, we define $\Xi_k = g(\Xi_{k-1}, \mathbf{x}_k)$, where $g(\cdot)$ represents the function that (i) maintains target IDs; (ii) assigns new IDs to appearing targets (target births); and (iii) removes the IDs of disappeared targets (target deaths). Targets can move in any locations of the observed area and they might cross or move close to each other. By considering the IDs as random variables, we can assign to each target the probability of having the corresponding ID, such that

$$p(\mathbf{x}_k, L_\xi) = p(\mathbf{x}_k | L_\xi) p(L_\xi). \tag{8}$$

A target may spatially interact with other targets in its vicinity (neighborhood). When targets are close to each other, there is uncertainty in assigning IDs. The main goal is to keep their identities separate and associated to the correct targets by maximizing their probability of having the assigned ID. To this end, we take into account the selected targets with respect to the neighboring ones in the calculation of the probability $p(L_\xi)\ \forall\ \xi$. The probability of a target having an ID depends only on the spatially close targets and, hence, the dependencies for the calculation of the probability follow the Markovian property. For this reason, to consider the state and its neighborhood, we model the set $\mathcal{L}_k$ as a Markov Random Field (MRF). With such definition of $g(\cdot)$ and $p(L_\xi)$, the proposed method of target birth and death lies between greedy and probabilistic methods.

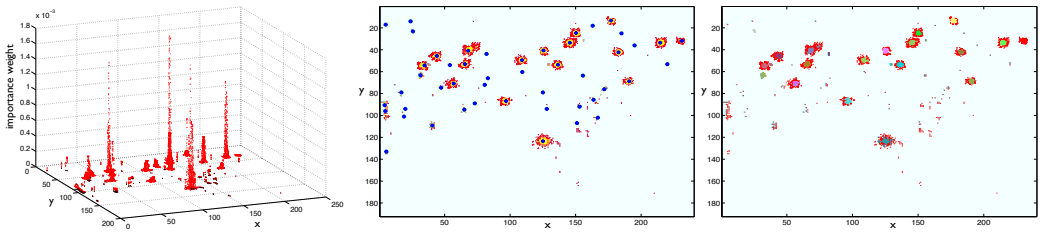


Figure 2: ID assignment, from prediction to state estimation. (a) Monte Carlo representation at the prediction step (red particles: existing particles propagated with the motion model from the previous time step; black particles: new-born particles). (b) Mean-Shift clustering result on the particles approximating the posterior distribution (blue markers: centroids of the clusters; yellow particles: particles kept in the resampling process). (c) Distribution of the particles with different IDs (color-coded) superimposed on the actual observation (the confidence map).

Let us denote the neighborhood of L_ξ as $\mathfrak{N}(\xi)$, hence the Markovian property of L_ξ is defined via local conditions

$$p(L_\xi |\, \mathcal{L}_k \setminus \xi) = p(L_\xi |\, \mathfrak{N}(\xi)). \tag{9}$$

The information on the target identity within the state leads to the calculation of the likelihood and the prediction depending on the set $\mathcal{L}_k$, such that

$$p(\mathbf{x}_k, \mathcal{L}_k | \mathbf{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathcal{L}_k) p(\mathbf{x}_k, \mathcal{L}_k | \mathbf{Z}_{k-1})}{p(\mathbf{z}_k | \mathbf{Z}_{k-1})}. \tag{10}$$

By construction $\mathcal{L}_k$ is conditionally independent of the time and the observations $\mathbf{Z}_k$, and hence Eq. 10 can be rewritten as

$$p(\mathbf{x}_k, \mathcal{L}_k | \mathbf{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}_{k-1}) p(\mathcal{L}_k)}{p(\mathbf{z}_k | \mathbf{Z}_{k-1})}, \tag{11}$$

where the prediction term $p(\mathbf{x}_k, \mathcal{L}_k | \mathbf{Z}_{k-1}) = p(\mathbf{x}_k | \mathbf{Z}_{k-1}) p(\mathcal{L}_k | \mathbf{Z}_{k-1}) = p(\mathbf{x}_k | \mathbf{Z}_{k-1}) p(\mathcal{L}_k)$ and the update term $p(\mathbf{z}_k | \mathbf{x}_k, \mathcal{L}_k) = p(\mathbf{z}_k | \mathbf{x}_k)$.

3.3. Sequential Monte Carlo estimation

In order to make the Bayesian recursion of Eq. 11 computationally tractable, we use the Sequential Monte Carlo estimation to approximate the probability densities with a set of particles [23] (Fig. 3). The N particles used to describe the

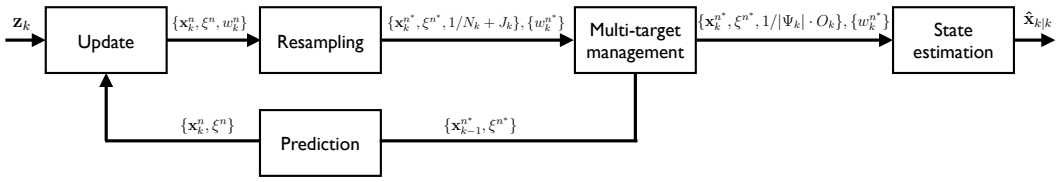


Figure 3: Block diagram of the proposed multi-target track-before-detect (MT-TBD). The filter receives as input the confidence map ($\mathbf{z}_k$) and draws the distribution of the target states using the Bayesian estimation with Monte Carlo approximation (particles). The weights of the particles are carried throughout the framework and used in the state estimation stage to find the target locations ($\hat{\mathbf{x}}_{k|k}$). The states marked with the superscript $*$ are generated after resampling. After the multi-target management stage, the weight distribution is uniform with respect to the number of the targets, O_k, at time k.

posterior $p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_k)$ at time k are denoted as $\{\mathbf{x}_k^n, \xi^n, w_k^n\}_{n=1}^N$, where w_k^n is the importance weight of the n-th particle.

In the prediction step there are two sets of particles: *existing* and *new-born* (Fig. 2(a)). The set of Q_k existing particles are drawn from the motion model of Eq. 2 and the set of J_k new-born particles are drawn from the proposal density $q_k(\mathbf{x}_k|\mathbf{z}_k)$; both are chosen *a priori*. Hence, $N = Q_k + J_k$. q_k distributes particles in $\mathbf{z}_k$ proportionally to the intensity values of the input confidence map, thus resulting in a high concentration of particles in high-intensity regions. The proposal density for the velocity is uniformly distributed for both x and y components, e.g. for x, $q_k(\dot{x}_k) = \mathcal{U}[-v_{max}, v_{max}]$, where v_{max} is the maximum target velocity. The proposal density for the intensity component is $q_k(I_k) = \mathcal{U}[I_{min}, I_{max}]$, where I_{min} and I_{max} are the minimum and maximum intensity values, respectively. ξ is initialized with null value.

In the update step, the importance weights w_k^n are computed using the likelihood function. The likelihood modeling is performed in two steps: the extraction of intensity values of true and false target locations over time using ground-truth data from a training set (Sec. 5.1), and the fitting of a function on the collected data. The distribution of intensity values of true locations over time is referred to as *signal-plus-noise*, the distribution of intensity values of false locations as *noise*. The ideal case is with a Dirac delta in 0 for false locations (no noise) and a Dirac delta in 1 for true locations (clean signal). The likelihood is calculated as the ratio between the distribution of the target *signal-plus-noise* $p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^n)$ and the distribution of the *noise* $p_N(z_k^{(i,j)})$. In the former case, we use a Normal distribution and in the latter case a Pareto distribution [32].

Given the observation $\mathbf{z}_k$, the likelihood $\ell(z_k^{(i,j)}|\mathbf{x}_k^n)$ for the n-th particle at time k and position (i,j) is calculated as

$$\ell(z_k^{(i,j)}|\mathbf{x}_k^n) = \underset{i\in C_i(\mathbf{x}_k^n), j\in C_j(\mathbf{x}_k^n)}{\arg\max} \left\{ \frac{p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^n)}{p_N(z_k^{(i,j)})} \right\}, \tag{12}$$

where $C_i(\mathbf{x}_k^n)$ and $C_j(\mathbf{x}_k^n)$ are the set of pixels centered on pixel (i,j) and affected by the uncertainty mentioned in Sec. 3.1 during target localization. The importance weights are finally calculated as

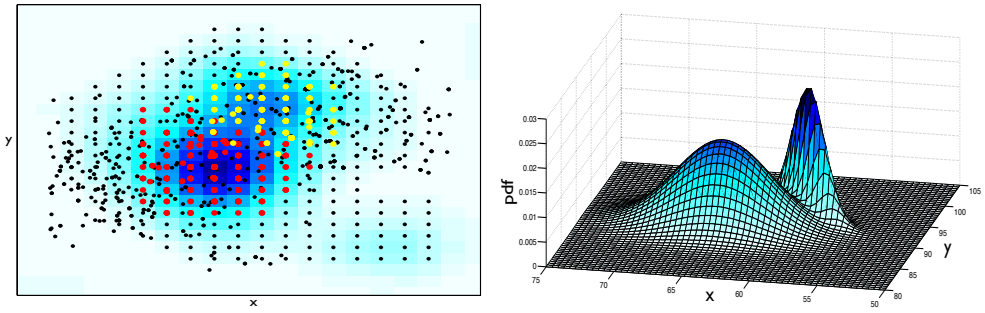
$$w_k^n = \frac{\ell(z_k^{(i,j)}|\mathbf{x}_k^n)}{\sum_{n=1}^N \ell(\mathbf{z}_k|\mathbf{x}_k^n)} \cdot \frac{p(L_{\xi^n})}{\sum_{n=1}^N p(L_{\xi^n})}, \tag{13}$$

where $p(L_{\xi^n})$ is the ID probability of the n-th particle (Sec. 4). The importance weights approximate the updated posterior $p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_k)$ whose modes represent the estimated state of the targets (Fig. 2(a)). To avoid the degeneracy problem [23], the particles are resampled using multinomial resampling. Resampling eliminates (duplicates) samples with low (high) importance weights. To retrieve the modes of the posterior distribution, we perform Mean-Shift (MS) clustering [33] using the position of the particles, i.e. $(x_k^n, y_k^n)\ \forall\ n$ (Fig. 2(b)), without any prior knowledge on the number of clusters or their shape, and with a fixed cluster size.

Let us define the size of the cluster as λ_Ψ and the set of clusters at time k as $\Psi_k = \{\psi_r\}_{r\in\mathcal{R}_k}$, with ψ_r the generic r-th cluster and $\mathcal{R}_k$ the set of cluster indexes. At this stage, the function $g(\cdot)$ introduced in Sec. 3.2 assigns a different ID to the particles belonging to different clusters at $k = 1$. At $k > 1$, if a cluster contains only new-born particles, they are all initialized with a new ID. Otherwise, the ID is assigned to the new-born particles with a method based on Gaussian Mixture Model (GMM), as explained in the next section.

4. ID management with Markov Random Field

We address now the issues of managing multiple target identities in the presence of interactions, target births and target deaths. We use the random variable L_ξ as a contribution to the posterior distribution of Eq. 11 for penalizing particles belonging to a target that either mix with particles of other targets or move far from the target they represent. Being the target location spatially spread in a kernel (i.e. $C_i(\mathbf{x}_k^n)$ and $C_j(\mathbf{x}_k^n)$), particles belonging to a target are in turn spread over the kernel (Fig. 2). Hence, when targets get close to each other, particles are



(a) (b)

Figure 4: Example of Gaussian fitting on the particle states of two close targets. The GMM is used to assign IDs to new-born particles within a cluster containing targets with different IDs. (a) Red and yellow represent existing particles belonging to different targets, whereas black represents new-born particles. (b) Corresponding Gaussian mixture fitting on the particles.

likely to mix (Fig. 4(a)), thus creating a challenging situation to manage in order to separately maintain the identity of multiple targets.

To address this problem, let us characterize the set $\mathcal{L}_k$ and the joint probability distribution $p(\mathcal{L}_k)$. Since $\mathcal{L}_k$ is a MRF, in order to construct the joint distribution of $\mathcal{L}_k$ considering the Markovian property of Eq. 9, we employ the Gibbs distribution [34],

$$p(\mathcal{L}_k) = \frac{1}{D} \exp\{U(\mathcal{L}_k)\}, \tag{14}$$

where D is a normalization factor and $U(\cdot)$ is the energy function

$$U(\mathcal{L}_k) = \sum_{\mathfrak{N}(\xi) \in \mathfrak{N}} V_{\mathfrak{N}(\xi)}(L_\xi), \tag{15}$$

where $\mathfrak{N}$ represents all the possible neighborhoods in the state space and $V_{\mathfrak{N}(\xi)}$ is the potential function defined for the neighborhood $\mathfrak{N}(\xi)$. Since a potential function is defined on a single neighborhood, it ensures that it is possible to factorize the joint probability such that the conditionally independent variables, for instance from non-connected neighborhoods, do not contribute to the same potential function.

Given a particle $\mathbf{x}_k^n$, the probability of ξ^n is $p(L_{\xi^n})$ and its neighborhood $\mathfrak{N}(\xi^n)$ corresponds to the domain defined by the pixels affected by the blurring introduced during target localization, i.e. $C_i(\mathbf{x}_k^n)$ and $C_j(\mathbf{x}_k^n)$ (Eq. 12).

The potential function of ξ^n at time k associated to particle $\mathbf{x}_k^n$ is calculated as

$$V_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) = V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) + V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n}), \tag{16}$$

where $V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ evaluates the agreement of the ID of particle n with respect to the IDs in $\mathfrak{N}(\xi^n)$ and $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ evaluates the distance between the ID of particle n and the center of mass of particles with the same ID of particle n. We define

$$V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) = \exp\left\{-\alpha_1(1-\delta_k^n)\frac{\iota_k^n}{\rho}\right\}, \tag{17}$$

where ι_k^n quantifies the agreement of the IDs and α_1 regulates the strength of the agreement. For instance, a high value of α_1 leads to a low probability of having an ID when a particle is surrounded by particles with different IDs, instead a low value of α_1 keeps the probability $p(L_{\xi^n})$ high when a particle is surrounded by particles with different IDs. ρ normalizes the agreement value over the number of particles in the neighborhood,

$$\iota_k^n = d_k^{\mathfrak{N}(\xi^n)} - a_k^{\mathfrak{N}(\xi^n)}, \tag{18a}$$

$$\rho = d_k^{\mathfrak{N}(\xi^n)} + a_k^{\mathfrak{N}(\xi^n)}, \tag{18b}$$

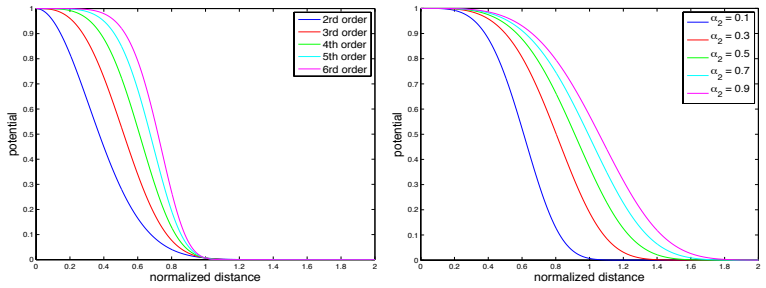
with $d_k^{\mathfrak{N}(\xi^n)}$ as the number of different IDs and $a_k^{\mathfrak{N}(\xi^n)}$ as the number of same IDs with respect to ξ^n within the neighborhood $\mathfrak{N}(\xi^n)$. δ_k^n is the Dirac function that indicates if n is a new-born particle or not,

$$\delta_k^n = \begin{cases} 1 & if \;\; \xi^n = 0 \\ 0 & if \;\; \xi^n \neq 0 \end{cases}. \tag{19}$$

In fact, if n is a new-born particle, then $p(L_{\xi^n}) = 1$ with null ID. The ID will be assigned to the new-born particles at the multi-target management stage (Fig. 3). The potential $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ is defined as

$$V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) = \exp\left\{\frac{-(1-\delta_k^n)(\gamma_k^n)^4}{2\alpha_2}\right\}, \tag{20}$$

where the rise $(\cdot)^4$ and α_2 are used to regulate the decreasing trend of the function. The higher α_2, the higher the probability of having an ID far from the group



(a) (b)

Figure 5: Potential $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ used for evaluating the distance between particle n and the center of mass of particles with the same ID. (a) Decreasing trend of the function when changing the order of γ_k^n. (b) Changes at different distances from the center of mass of the particles with the same ID.

of particles with the same ID. γ_k^n is the normalized Euclidean distance from the center of mass and δ_k^n is defined as in Eq. 19.

Figure 5 shows the trend of the function with different parameters: the horizontal axis represents the variation of γ_k^n and the vertical axis represents $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ as a function of γ_k^n. Figure 5(a) shows the decreasing trend of the potential function when changing the order of γ_k^n, whereas Fig. 5(b) shows how the potential $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ changes at different distances from the center of mass. The center of mass is calculated by utilizing the geometric mean of the position of the particles with the same ξ^n and the normalization is calculated by taking into account the area of the pixels affected by the blurring introduced during target localization,

$$\gamma_k^n = \frac{1}{4\Sigma}\sqrt{\left(x_k^n - \sqrt[M]{\prod_{m=1}^{M} x_k^m}\right)^2 + \left(y_k^n - \sqrt[M]{\prod_{m=1}^{M} y_k^m}\right)^2}, \tag{21}$$

where the normalizing factor 4Σ takes into account the 95% of the area affected by the blurring and $M = |\mathfrak{N}(\xi^n)|$ is the number of neighbors of the n-th particle. Finally, the value D in Eq. 14 used to normalize the energy function for each particle is defined as

$$D(L_{\xi^n}) = \exp\{\alpha_1(1 - \delta_k^n)\}. \tag{22}$$

The computation of the probability of ξ^n leads to the ID assignment to the new-born particles. The general concept is to assign to the new-born particles that belong to a cluster the same ID of the existing particles within the same cluster. This assignment is based on the probability of existing IDs. When existing particles are already initialized with an ID in a cluster, the ID assignment is performed

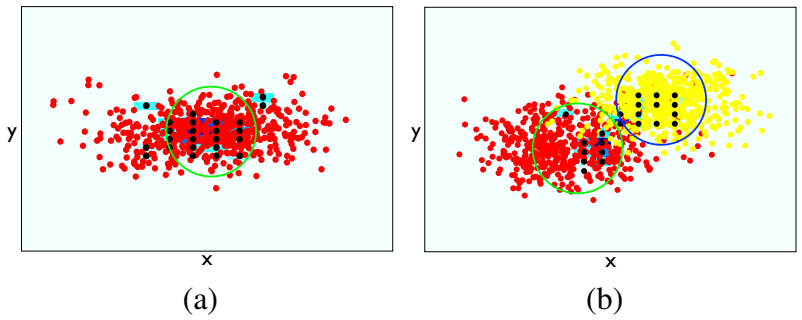


Figure 6: Two sample cases of ID assignment to the new-born particles (black) using Mean-Shift clustering. (a) Cluster (green) containing existing particles with the same ID (red) that is assigned to all the new-born particles within the cluster. (b) Cluster (green) containing existing particles with different IDs (red and yellow) that are assigned to the new-born particles within this cluster using a GMM approach (see text for details).

by considering two cases: (i) clusters with existing particles and the same ID (Fig. 6(a)), and (ii) clusters with existing particles and different IDs (Fig. 6(b)). In the former case, when a cluster contains new-born particles and existing particles sharing the same ID, the ID assigned to the new-born particles is the same as that of the existing particles. In the latter case, when in a cluster there are new-born particles and existing particles with different IDs, we use a method of ID assignment based on GMM[1]. By fitting a GMM with mean components placed on the center of mass of each group of particles with same ID and variance proportional to the probability of the respective ID, we ensure a *fair* assignment of IDs to the new-born particles located in the cluster. As shown in Fig. 4, the widest GMM component belongs to the target with widest spread function. Vice versa the narrowest GMM component belongs to the target with narrowest spread function. Each fitted Gaussian approximates the spatial distribution of particles sharing the same IDs, and the assignment of the ID to each new-born particle within the cluster is performed according to the Maximum A Posteriori (MAP).

Let us define the set $\mathcal{X}_r = \{(x_k^n, y_k^n, \xi^n) : x_k^n, y_k^n \in \psi_r\}$ of particle locations and IDs belonging to the r-th cluster at time k. Using $\mathcal{X}_r$, we calculate the mean position of the respective IDs, $\theta_\xi \, \forall \, \xi^n \in \mathcal{X}_r$. Let us denote the set of mean positions as $\Theta_r = \{\theta_\xi : \xi^n \in \mathcal{X}_r\}$. We then define the covariance matrices using the

[1]We choose a probabilistic model, rather than an ad-hoc assignment, since it can be easily extended or replaced with other probabilistic models in case of different applications of the tracker.

total probability of each ID $p(L_{\xi^n})$, such that

$$\phi_\xi = p(L_{\xi^n}) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{23}$$

and, as for the mean positions, we define the set of covariance matrices $\Phi_r = \{\phi_\xi : \xi^n \in \mathcal{X}_r\}$. In this way, the fitting is performed using Gaussians with covariances proportional to the probability of the IDs within $\mathcal{X}_r$. Note that $|\Theta_r| = |\Phi_r|$, where $|\cdot|$ is the cardinality of a set. The GMM is defined as a weighted sum of Gaussian densities given by

$$f(\mathcal{X}_r) = \sum_{m=1}^{|\Theta_r|} \pi_m \mathcal{N}(\mathcal{X}_r | \Theta_{r,m}, \Phi_{r,m}), \tag{24}$$

where $\Theta_{r,m}$ and $\Phi_{r,m}$ denote the m-th mean and covariance component of the corresponding sets, respectively, and each ID $\xi \in \mathcal{X}_r$ is associated to each component m, i.e. $\xi \rightarrow m$. The best fitting of the mixture is performed through the Expectation-Maximization algorithm [35]. Figure 4(b) shows an example of GMM fitting when two nearby targets are present. Once the GMM is fitted to the particle locations, the affiliation of the new-born particles to the targets is derived through the calculation of the MAP and the ID assignment is performed with respect to such information. Hence, $\forall\ (x_k^n, y_k^n, \xi^n) \in \mathcal{X}_r$ with $\xi^n = 0$, the ID is assigned using the MAP

$$\xi^n = \bar{\xi} \quad : \quad \bar{\xi} \rightarrow m', \ \ m' = \underset{m=1,\ldots,|\Theta_r|}{\arg\max} \left\{ p(m|(x_k^n, y_k^n)) \right\}, \tag{25}$$

where $\bar{\xi}$ is the ID associated to the component with the highest probability and

$$p(m|(x_k^n, y_k^n)) = \frac{p(m)p((x_k^n, y_k^n)|m)}{p((x_k^n, y_k^n))} = \frac{\pi_m \mathcal{N}((x_k^n, y_k^n)|\Theta_{r,m}, \Phi_{r,m})}{\sum_{m=1}^{|\Theta_r|} \pi_m \mathcal{N}((x_k^n, y_k^n)|\Theta_{r,m}, \Phi_{r,m})}. \tag{26}$$

The state estimate $\hat{\mathbf{x}}_{k|k} = (\hat{x}_{k|k}, \hat{y}_{k|k})$ is finally calculated using the weighted sum of the particle positions on their relative weights,

$$\hat{\mathbf{x}}_{k|k,\xi} = \frac{\sum_n w_{k,\xi}^n \cdot [x_{k,\xi}^n \ y_{k,\xi}^n]^T}{\sum_n w_{k,\xi}^n}, \tag{27}$$

where the subscript ξ is used to indicate that the state estimate is calculated among particles sharing the same ID.

Once the IDs are assigned, the resampling of the particle weights is performed for each cluster independently by assigning the same number of particles to each cluster. Ideally, each cluster contains a single target, hence by resampling each cluster independently we ensure that all clusters/targets evolve over time with the same number of particles.

5. Application to people tracking

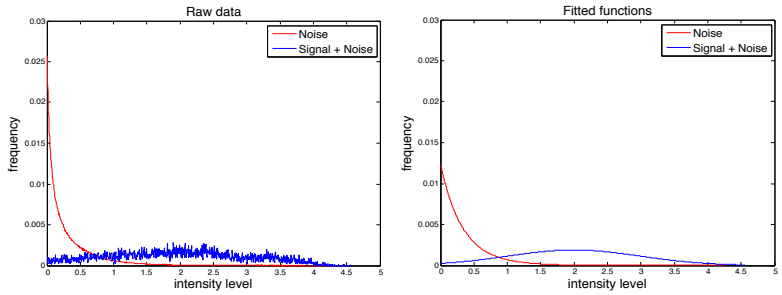
In this section, we model the likelihood function and develop a postprocessing stage for people tracking applications. The postprocessing makes use of track duration in case of moving cameras, and of background information and people appearance in case of static cameras.

5.1. Likelihood modeling

The likelihood function (Eq. 12) for MT-TBD is modeled using automatically generated confidence maps filtered by ground-truth information (Sec. 3.3). The intensity distribution of true locations is referred to as *signal-plus-noise*, since manifold factors may affect the response of the target localization method, such as objects with similar shape or color. The intensity distribution of false locations is referred to as *noise*. Ideally, a specific likelihood function should be modeled for each scenario. However, in order to demonstrate the flexibility of the proposed MT-TBD in different scenarios and for different targets, a single likelihood function is defined and used throughout our experiments. In particular, we model the likelihood function using highly noisy data, such as head locations obtained by Support Vector Machine (SVM) [36] and by using HOG features [25] in the TRECVID dataset. The distribution of head/non-head confidences is shown in Fig. 7(a). Figure 7(b) shows the fitted curves on the data for modeling the likelihood function. The *signal-plus-noise* distribution is fitted by a Normal distribution and the *noise* distribution by a Pareto distribution [32]. Since the exponential function goes quicker to zero than the Pareto function, the Pareto distribution is more suited for modeling the *noise* in Eq. 12 (at the denominator). In fact, very high values of likelihood for high values of observed intensities would lead to the divergence in the estimation of the posterior distribution (Eq. 11).

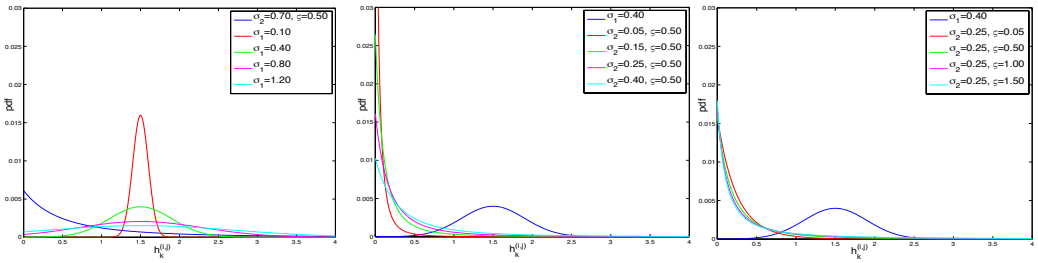
The final likelihood ratio (Eq. 12) is calculated as

$$\begin{aligned} &\frac{p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^n)}{p_N(z_k^{(i,j)})} = \\ &= \frac{\sigma_2}{\sqrt{2\pi}\sigma_1}\left(1+\varsigma\frac{z_k^{(i,j)}}{\sigma_2}\right)^{(1+\frac{1}{\varsigma})}\exp-\frac{(z_k^{(i,j)}-h_k^{(i,j)}(\mathbf{x}_k))}{2\sigma_1^2}, \end{aligned} \tag{28}$$



(a) (b)

Figure 7: (a) Distribution of the *signal-plus-noise* (blue) and *noise* (red) extracted from real data represented by the head locations [25] on the TRECVID dataset. (b) Normal distribution fitted on *signal-plus-noise* (blue) and Pareto distribution on *noise* (red).



(a) (b) (c)

Figure 8: Variation of the parameters of the fitted distributions for the likelihood function (Eq. 28): (a) σ_1, (b) σ_2 and (c) ς.

where σ_1 is the standard deviation of the Normal distribution, and σ_2 and ς are the scale and tail parameters of the Pareto distribution, respectively.

Figure 8 shows the effect of the parameter variations on the numerator and denominator of Eq. 28. When $p_N(z_k^{(i,j)})$ quickly decreases to zero, i.e. small σ_2 and small ς, the likelihood ratio gives high values. Vice versa, when $p_N(z_k^{(i,j)})$ slowly decreases to zero, i.e. if σ_2 and ς are large, the likelihood gets more biased on the value of the numerator.

5.2. Data-driven postprocessing

We use a shifting temporal window of τ frames overlapping of one frame over time. The tracks within this temporal window are collected into the set

$$\mathfrak{T}_k^\tau = \{\mathfrak{T}_{\mathfrak{K},\xi}^\tau : \xi \in \Xi_k, \mathfrak{K} = [k_s, k_e], \mathfrak{K} \subseteq [k-\tau+1, k]\}, \tag{29}$$

where $\mathfrak{T}_{\mathfrak{K},\xi}^\tau$ is the generic track with ID ξ within the interval $\mathfrak{K} = \mathfrak{K}_\xi^\tau = [k_s, k_e]$ and k_s, k_e are the starting and ending instants of the track within the temporal window, respectively.

Each track is defined as $\mathfrak{T}_{\mathfrak{K},\xi}^\tau = \{(\hat{\mathbf{x}}_{k,\xi}, b_{k,\xi}) : \xi \in \Xi_k, k \in \mathfrak{K}_\xi^\tau\}$[2], where the state estimate $\hat{\mathbf{x}}_{k,\xi}$ corresponds to the top-left corner of the bounding box and $b_{k,\xi}$ is the bounding box associated to the state $\hat{\mathbf{x}}_{k,\xi}$ retrieved using the scene calibration information. Note that the postprocessing introduces a delay in the tracking output that is analyzed in details in Sec. 6.5.

The postprocessing stage for multi-person tracking is divided into (i) track pruning to remove tracks with a score s less than 3 within a temporal window $\tau_1 = 25$ frames, (ii) track fusion within a temporal window $\tau_2 = \tau$ and directly proportional to τ_1, and (iii) track pruning to remove fused tracks with score less than $\tau_2/10$ for a temporal window of τ_2.

For track pruning, let us consider a generic track $\mathfrak{T}_{\mathfrak{K},\xi}^{\tau_1}$ with generic ID ξ that exists within the temporal interval τ_1. A score $s_\xi^{\tau_1}$ is assigned to each track, such that

$$s_{\mathfrak{K},\xi}^{\tau_1} = \sum_{k \in \mathfrak{K}_\xi^{\tau_1}} r(\mathfrak{T}_{k,\xi}^{\tau_1}), \tag{30}$$

where $r : \mathbb{R}^m \to \{0, 1\}$ and m is a set of rules used to define the score. This leads to $s_{k,\xi}^{\tau_1}$ being equal to the duration of a track (in frames) if $r(\mathfrak{T}_{k,\xi}^{\tau_1}) = 1 \; \forall \; k \in \mathfrak{K}_\xi^{\tau_1}$, otherwise, if $r(\mathfrak{T}_{k,\xi}^{\tau_1}) = 0$ for some $k \in \mathfrak{K}_\xi^{\tau_1}$, the score $s_{k,\xi}^{\tau_1}$ is smaller than the duration of the track. The same process is performed into the temporal window τ_2.

In case of moving cameras, the function $r(\cdot)$ only evaluates the duration of the track in frames. In case of static cameras, the function $r(\cdot)$ is modeled as a *logic AND* of two rules, $r_1(\cdot)$ and $r_2(\cdot)$, obtained from a background subtraction step. Given $\mathcal{B}(\mathfrak{T}_{k,\xi}^{\tau_1})$, a patch within each bounding box from the difference image

[2]The conditional dependency on k of $\hat{\mathbf{x}}_{k|k,\xi}$ is omitted for simplicity in the notation.

between the current frame $f_k(\mathfrak{T}^{\tau_1}_{k,\xi})$ and the background, we define

$$r_1(\mathfrak{T}^{\tau_1}_{k,p}) = \begin{cases} 0 & \text{if } \mu(\mathcal{B}(\mathfrak{T}^{\tau_1}_{k,\xi})) < T_1 \\ 1 & otherwise \end{cases}, \tag{31}$$

where $\mu(\cdot)$ calculates the mean pixel intensity and $T_1 = 20$-25 depending on the contrast between targets and background, and

$$r_2(\mathfrak{T}^{\tau_1}_{k,p}) = \begin{cases} 0 & \text{if } \sigma(f_k(\mathfrak{T}^{\tau_1}_{k,\xi})) > T_2 \\ 1 & otherwise \end{cases}, \tag{32}$$

where $\sigma(\cdot)$ calculates the standard deviation of the pixel intensities in gray level and $T_2 = 5$ to remove false positive tracks on flat surfaces such as walls. For the specific case of head tracking, we define an additional rule, $r_3(\cdot)$, to calculate the relative distance and size between bounding boxes in order to remove false tracks originated due to shoulders, when they are erroneously detected as heads.

We formulate the track fusion process as a re-identification problem. The last available position of a track, the velocity and the color extracted from the upper-body patch [37] are used to find the best match between the final position of a track and the initial position of another track ahead in time.

Let us define the function $\kappa(\cdot)$ that calculates the cost between each track pair within the temporal window τ_2: $\kappa(\mathfrak{T}^{\tau_2}_{\hat{\kappa},\xi}, \mathfrak{T}^{\tau_2}_{\hat{\kappa},\xi'})$ is the affinity between track ξ and track ξ', $\forall\ \xi' \in \Xi_k \setminus \xi$. Using the temporal gap between two tracks and the last available position of $\mathfrak{T}^{\tau_2}_{\hat{\kappa},\xi}$, we predict the target position with a linear motion model. The affinity is thus calculated from the end point of a track ($\mathfrak{T}^{\tau_2}_{k_e,\xi}$) to the start point of another track ($\mathfrak{T}^{\tau_2}_{k_s,\xi'}$), with $k_s > k_e$. The calculation of the affinities is based on the Euclidean distance between predicted and current starting point, and the Bhattacharyya distance of the image patch at k_e and that at k_s. The Munkres algorithm[3] is then iteratively computed to associate all the possible track pairs.

6. Experiments and results

6.1. Datasets and algorithms

In this section, the proposed MT-TBD is tested as multi-person tracking on confidence maps generated by six person localization algorithms (see Dallar *et*

[3]http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html. Last accessed: March 2012.

al. [38] for a complete survey on person localization). In particular, we retrieve person locations using information of: full-body [25, 29], head [25, 10], full-body based on parts [39] and full-body from multiple views [40]. We firstly use reliable confidence maps obtained (i) from head locations guided by the ground truth and (ii) from multiple views of the same scene. Then, we compare the proposed method with the state of the art by employing automatically generated confidence maps on single-view.

The experiments are performed on one sport video, four surveillance videos and two videos obtained from a moving camera. The first set of reliable confidence maps are extracted on 2400 frames of size 720×576 pixels from Camera 1 of the London Gatwick airport dataset[4] that is recorded at 25Hz. The confidence maps are generated as the output of a SVM trained with HOG features [25], where false positive confidences are removed using ground-truth information. Let us call this dataset TRECVID-HOG+GT. In addition to this, we perform tracking on two different cameras of a basketball scenario (APIDIS dataset[5]) composed of 800 frames of size 800×600 pixels and recorded at 25Hz. Let us call them APIDISC1 and APIDISC2. Here, the reliable sets of confidence maps are obtained by a multi-layered homography method [40] that exploits the seven cameras available in the dataset. Results on TRECVID-HOG+GT, APIDISC1 and APIDISC2 are reported in Sec. 6.4.

In Sec. 6.5, MT-TBD is then tested on automatically generated confidence maps on single views. Firstly, we use the TownCentre dataset[6] composed of 4500 frames of size 1980×1080 pixels, recorded in Oxford (UK) town center at 25Hz. For a fair comparison with Benfold and Reid [10], we use the head locations provided by the authors, which are generated using HOG features and SVM. As the provided person locations have already been thresholded, they are not in the form of intensity levels. For this reason, the input to MT-TBD is a confidence map with 2D Dirac delta in correspondence to each localized head. Moreover, we use the iLids Easy[7] dataset composed of 5220 frames of size 720×576 pixels recorded at the London Westminster subway station at 25Hz, where we obtain person locations using an approach based on body-parts proposed by Felzenszwalb *et al.* [39]. Another localization method based on HOG features and SVM [25] is trained on

[4]iLIDS, Home Office multiple camera tracking scenario definition (UK), 2008.

[5]http://www.apidis.org/Dataset/. Last accessed: March 2012.

[6]http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html. Last accessed: March 2012.

[7]http://www.eecs.qmul.ac.uk/ andrea/avss2007_d.html. Last accessed: March 2012.

Table 2: Summary of the datasets and person localization methods used for validation. Key: H: Head; B: Body; P-B: part-based.

Dataset	**Image size**	**Localization method**	**Body part**
TRECVID-HOG+GT	720×576	HOG + SVM [25] + GT	H
APIDIS	800×600	Multi-layer homography [40]	B
TownCentre	1920×1080	Binary (HOG + SVM) [25]	H
iLids Easy	720×576	HOG + SVM [39]	B, P-B
TRECVID	720×576	HOG + SVM [25]	H
ETH	640×480	Binary (Edges + Weak Classfier) [29]	B

head patches of 24×24 pixels, and applied to the London Gatwick airport dataset that has the same specifications as above. Let us call this dataset TRECVID to distinguish it from TRECVID-HOG+GT. Finally, we test MT-TBD on two videos from the ETH dataset[8] recorded from a moving camera at 13-14Hz on outdoor scenarios, and composed of 353 and 999 frames of size 640×480 pixels. For a fair comparison with Kuo *et al.* [9] and Yang and Nevatia [12] in this dataset, we employ their full-body locations[9], and, as for the TownCentre dataset, the input of MT-TBD for ETH is a confidence map with 2D Dirac delta since the provided locations have already been thresholded.

Table 2 summarizes the datasets and the localization methods used for testing.

6.2. Parameters

This section describes the parameters used for MT-TBD. Similarly to Breitenstein *et al.* [20], some parameters are set experimentally.

The choice of the maximum values of velocity, v_{max}, used to propagate the particles by the proposal density $q_k(\cdot)$ (Sec. 3.3) depends on the frame resolution. Higher resolutions lead to higher values of the maximum velocity. TRECVID and iLids Easy datasets have the same frame resolution and, because they contain walking people, the variance of motion is low. For this reason, we set $q_1 \approx 0.3$ and $v_{max} \approx 3$. Similarly, the TownCentre dataset contains walking people, but the frame resolution is much higher (Tab. 2), thus leading to larger displacements on the image plane. Hence, we set $q_1 = 4$ and $v_{max} = 12$. Since the ETH dataset is recorded with a moving camera and at low frame rate, the displacement for walking people is larger than TRECVID and iLids Easy, and we set $q_1 = 2$ and

[8] http://www.vision.ee.ethz.ch/∼aess/dataset/. Last accessed: March 2012.

[9] http://iris.usc.edu/people/yangbo/downloads.html. Last accessed: March 2012.

$v_{max} = 14$. Finally, in the APIDIS dataset, we set $q_1 = 3$ and $v_{max} = 12$ because people movements can be subject to sudden variations.

The noise q_2 associated to the intensity of the confidence map is then chosen according to the specific confidence map given in input to MT-TBD. The confidence maps of TRECVID, iLids Easy and APIDIS datasets are not thresholded, and we set $I_{min} = 1$, $I_{max} = 3$ and $q_2 \approx 0.3$ for all of them. In case of ETH and TownCentre datasets, the confidence maps are thresholded (there is no variation of intensity) and we set $I_{min} = I_{max} = 2$ with noise $q_2 = 10^{-5}$.

The amount of blurring introduced in the target localization process is modeled by Σ in Eq. 5: its value is dependent on the precision of the person localization method and on the resolution of the confidence map where higher resolution leads to a higher spread in intensity values. For example, $\Sigma = 1.3$ for both TRECVID and iLids Easy datasets that have the same person localization method and the same frame resolution. On the contrary, in case of the 2D Dirac delta confidence maps where blurring is absent, $\Sigma = 4$ in order to have a similar spread of the particles over space.

The values of α_1 and α_2 for the MRF modeling (Sec. 4) depend on the desired strength level for maintaining the particles alive in case of mixing with different IDs. We use $\alpha_1 = 0.2$ and $\alpha_2 = 0.02$ for all the datasets.

The value of σ_1, σ_2 and ς of Eq. 28 are provided in Tab. 3. The values of σ_1 used in TRECVID-HOG+GT and TRECVID datasets are similar because the same person localization method is used in both datasets, while the variation of σ_2 and ς is due to the employment of the ground-truth information in TRECVID-HOG+GT. Since in TRECVID-HOG+GT, the *noise* due to false localizations is absent, we set σ_2 and ς such that the numerator (*signal-plus-noise*) of the likelihood function is predominant on the denominator (*noise*). Vice versa, in case of TRECVID, the confidence maps are more noisy, and σ_2 and ς are set in order to take into account also the contribution of the denominator. The person localization method used in iLids Easy [39] provides a more stable *signal-plus-noise* compared to the method used in TRECVID, thus leading to a smaller variance of the confidence values and hence to a smaller σ_1. However, the *noise* is still high and σ_2 is set as for TRECVID. The value of ς is large, in order to avoid the divergence of the likelihood function in case of large confidence values. For APIDIS, TownCentre and ETH datasets the confidence maps are provided as 2D Dirac delta functions and this justifies the similarity of σ_1 and σ_2 values. The parameters are chosen such that the likelihood function does not diverge. Unlike TownCentre and ETH datasets where the 2D Dirac deltas are binary, in APIDIS the 2D Dirac deltas represent confidence values and, similarly to the iLids Easy,

Table 3: Parameters of the likelihood function (Eq. 28) used in the experiments.

Dataset	$\boldsymbol{\sigma_1}$	$\boldsymbol{\sigma_2}$	ς
TRECVID-HOG+GT	0.70	0.10	0.60
TRECVID	0.60	0.30	0.15
iLids Easy	0.15	0.40	1.70
APIDIS	0.70	0.16	0.25
TownCentre	0.80	0.20	0.04
ETH	0.80	0.22	0.05

we keep the value of ς large in order to avoid the divergence of the likelihood function for large confidence values.

6.3. Evaluation procedure

Given a bounding box for each target along with the confidence map at each time step, a true positive track is defined as the one having a bounding box overlapping at least 25% the ground-truth box in case of heads as targets, and at least 50% in case of full bodies as targets [10]. Let tp be the number of all the true positive tracks in a video sequence, fp all the false positive tracks, fn all the false negative tracks, IDS the number of all ID switches, and N_G the number of ground-truth targets. Performance scores are obtained by calculating the Multiple Object Tracking Accuracy (MOTA), the Multiple Object Tracking Precision (MOTP), Precision and Recall [41]. MOTA is calculated as

$$MOTA = 1 - \frac{(N_G - tp) + fp + IDS}{N_G} \tag{33}$$

and MOTP as

$$MOTP = \frac{O_t}{N_m}, \tag{34}$$

where O_t quantifies the overlap between the tracked bounding boxes and the ground-truth bounding boxes, and N_m is the number of ground-truth targets mapped with the tracking output for the whole video sequence. Precision, P, is calculated as

$$P = \frac{tp}{tp + fp} \tag{35}$$

and Recall, R, as

$$R = \frac{tp}{tp + fn}. \tag{36}$$

The results on the ETH dataset are evaluated using the toolbox provided by Li *et al.* [4].

6.4. Validation

The validation for the robustness of the proposed method is performed using the datasets TRECVID-HOG+GT, APIDISC1 and APIDISC2, and in particular, we analyze the tracking results generated by (i) MT-TBD without any postprocessing, (ii) track pruning on the tracks from MT-TBD, (iii) track fusion on the tracks from the previous track pruning, and (iv) track pruning on the tracks from the previous track fusion.

The analysis of the tracking results generated by MT-TBD without any postprocessing demonstrates the proposed filter, especially in situations with close targets where the MRF modeling helps avoiding particles of different targets to be mixed together. The first dataset we employ is the TRECVID-HOG+GT. In Fig. 9, a situation of a significant overlap ($> 50\%$) between two targets is shown. In Fig. 9(a), all targets in the scene are correctly tracked. Subsequently, when two targets get closer (Fig. 9(b)), the target far apart from the camera gets almost completely occluded, however, since the confidence map still localizes the target, MT-TBD correctly tracks it. In Fig. 9(c), when the targets are completely overlapped, the intensity levels on the confidence map appear as a single target with a large spread. Even if the tendency of mixing of particles with different IDs is visible, the MRF modeling consistently assigns the correct ID to each particle. Figures 9(d-f) finally show how the particles remain associated to the correct target over time.

Figure 10 shows an example of incorrect ID assignment leading to an ID switch generated by MT-TBD without any postprocessing. In this case, the confidence intensities are completely overlapped with a mixing of IDs. Initially, two close targets move in the same direction (Fig. 10(a)) and suddenly one target changes direction and becomes completely occluded (Fig. 10(b)). Although both IDs remain alive for a few time steps, the particles with magenta ID die (Fig. 10(d)) and the green particles move on the visible target. When the occluded target becomes visible again on the confidence map (Fig. 10(e-f)), MT-TBD immediately initializes a new track and correctly tracks the target in the following frames. Note that MT-TBD is not designed to reinitialize a target track with a previously existing ID, hence a different ID is assigned to a target that disappears and reappears in a scene, thus leading to an ID switch (Sec. 6.3).

Quantitative results for MT-TBD and postprocessing are reported in Fig. 11(a). After the first track pruning, Recall and MOTA are slightly decreased because the short tracks are removed due to their low score. However, after track fusion has been applied, Recall reaches a higher value because short but reliable tracks are

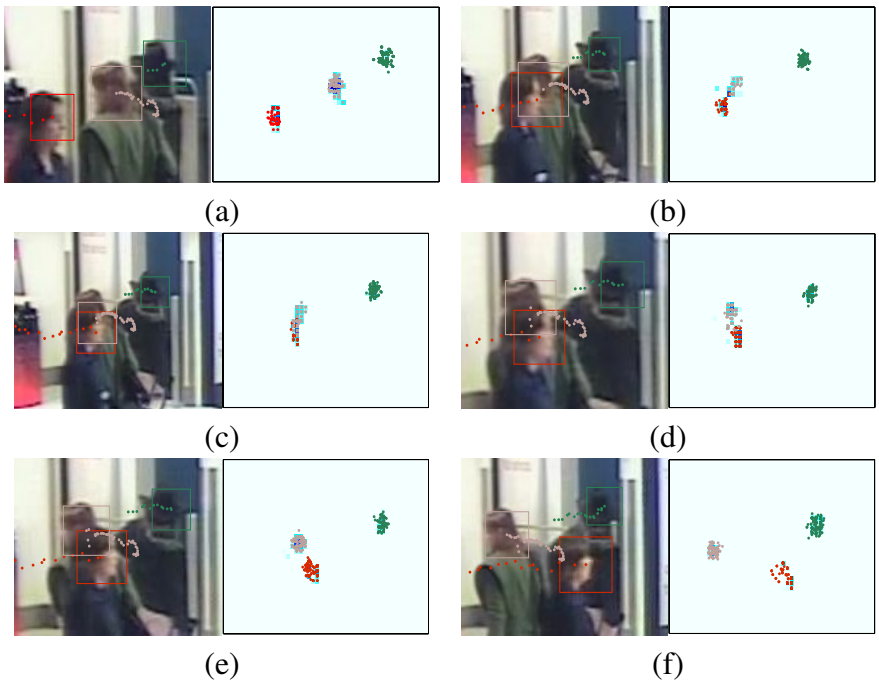


Figure 9: Example from TRECVID-HOG+GT dataset which represents a situation of a significant overlap ($> 50\%$) between two targets (red and gray color-codes). Before the occlusion occurs (a) the targets are correctly tracked with unique IDs. When the occlusion starts (b) particles start mixing but the IDs are still well-separated. During the occlusion (c), particles and IDs are mixed, but it is possible to notice that the mixing remains limited. When the targets start splitting (d), there is a tendency of the particles to mix (the red particles mix to the grey particles). When the split of targets occurs (e-f), the particles are again well-separated with their own IDs.

correctly fused. Lastly, by pruning the unreliable tracks generated by the fusion stage, it is possible to keep the same value of Recall while increasing Precision.

The second validation of MT-TBD and post-processing is presented using APIDISC1 and APIDISC2 (Fig. 11). By analyzing the results shown in Fig. 14(a-d), we see the tracking succeeding in most cases even while players are very close to each other. The main challenges here are the sudden movements of players. Recall is larger than 90% in both datasets even if some of the tracks are lost (Fig. 14(d)). A possible solution for this problem is the use of multi-dynamic model particle filters [23], which are able to perform nonlinear filtering with switching of dynamic models.

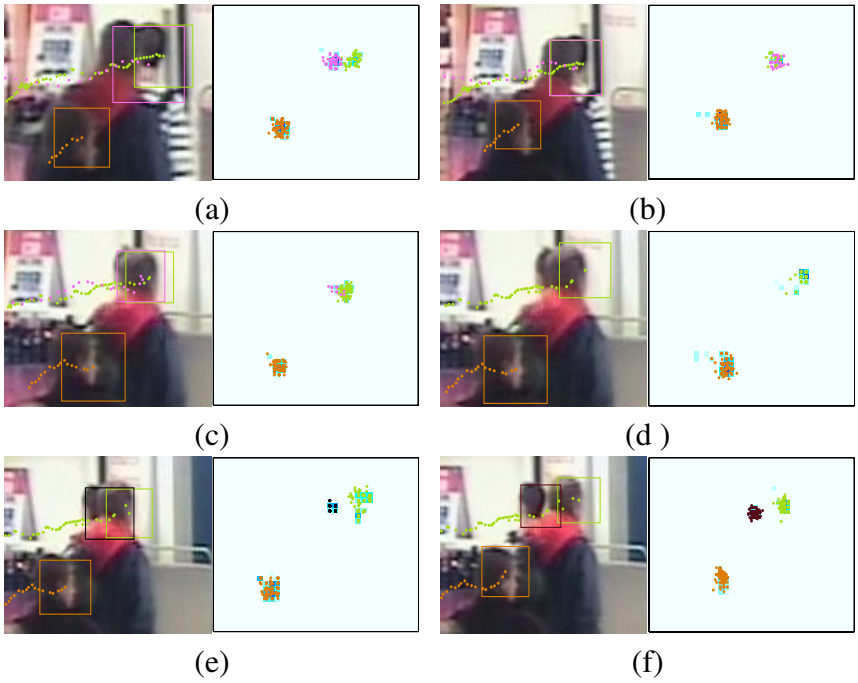
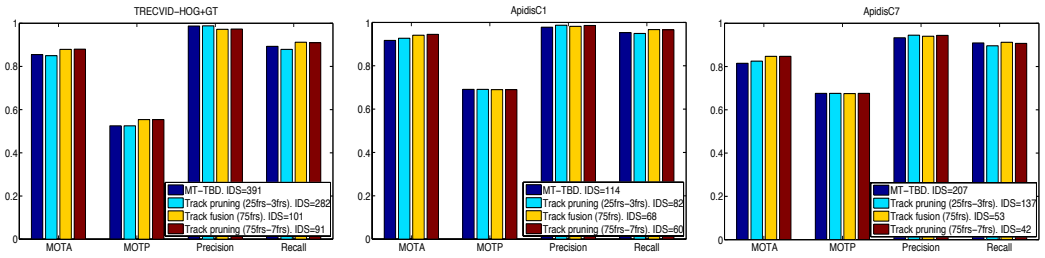


Figure 10: Example from TRECVID-HOG+GT dataset which represents a situation of significant overlap ($\approx 100\%$) between two targets where an ID switch occurs. Before the occlusion (a), the targets are correctly tracked with unique IDs. During the occlusion (b), the particles are mixed and the algorithm cannot maintain the correct IDs. When the targets start splitting (c), the number of magenta particles start getting smaller. Then particles belonging to the magenta target die (d) and the green particles swap target (they get attached to the target in front). When the target that is behind becomes visible, MT-TBD immediately starts tracking it again but with a new ID (e-f).

6.5. Comparisons and discussion

As far as TownCentre dataset is concerned, we show how our method outperforms the recent work by Benfold and Reid [10] by using the same observations for tracking. This scenario is fairly challenging as it contains very close targets and the field-of-view of the camera is very large, hence ID switches are likely to be frequent. For comparison, we present the results with the same latency used in [10] for postprocessing and, in particular, of 1, 2, 3, and 4 seconds (1 second = 25 frames). In order to show the global improvement of our proposed method, we also include the performance of MT-TBD without any postprocessing. Note that, unlike our tracker, the work in [10] cannot work with no latency.

Figure 12 shows the quantitative results. The superior performance of the



(a) (b) (c)

Figure 11: Tracking results of the proposed method at different stages of computation: MT-TBD, Track pruning τ_1, Track fusion τ_2 and Track pruning τ_2. Dataset: (a) TRECVID-HOG+GT, (b) APIDISC1 and (c) APIDISC2. The numbers between round brackets in the legend of the bar plot refer to the length of the temporal window and to the threshold applied on the minimum track length in the track pruning stage. IDS: ID Switches.

Table 4: Summary of the comparison with the best results obtained using the proposed method. The number of frames between round brackets represents the temporal window duration τ. Key: IDS: ID Switches.

Sequence	**Method**	**MOTA**	**MOTP**	**Precision**	**Recall**	**IDS**
TownCentre	MT-TBD (100frs)	0.546	0.637	0.783	0.762	285
	Benfold2011 [10]	0.454	0.508	0.738	0.710	-
iLids Easy	MT-TBD (100frs)	0.622	0.695	0.914	0.690	35
	Breitenstein2011 [20]	0.781	0.670	0.947	0.836	18
	Stalder2010 [21]	-	-	0.894	0.533	-
	Benfold2011 [10]	0.599	0.736	0.803	0.820	-

proposed method is highlighted by the value of Recall that is consistently higher than [10] at various latencies. For MT-TBD without latency (and no postprocessing), the value of Recall is already comparable with that of 4-second latency. However, Precision in this case is lower due to the short and false tracks generated by the temporally-consistent false positives head locations. By applying the proposed postprocessing, Precision drastically increases. Table 4 summarizes the final results. Figure 14(i-l) shows sample tracking results: it is possible to notice that the method is robust under severe occlusions with few fragmented tracks.

The results of iLids Easy and TRECVID are quantitatively evaluated in Fig. 13. For these two cases, the input confidence maps to MT-TBD are given as intensity levels. For this reason, it is possible to analyze the results in detail by com-

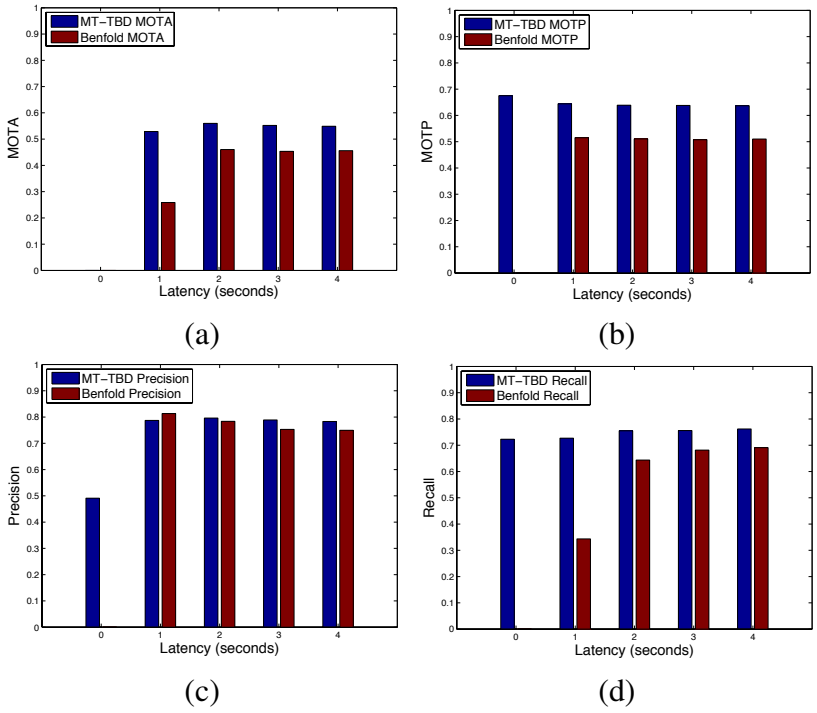


Figure 12: Comparison of our results on TownCentre dataset with the Benfold and Reid method [10]. The graphs show the variation of the scores as a function of the latency introduced by the postprocessing: (a) MOTA, (b) MOTP, (c) Precision and (d) Recall.

paring the accuracy of target localizations with the accuracy of MT-TBD. In the graphs of Fig. 13(c)-(d), the variation of Precision and Recall of the localization results with respect to the threshold variation on the confidence maps is shown, and the improvement that MT-TBD carries out can be appreciated. With the iLids Easy dataset, an indoor video surveillance scenario is analyzed where the main challenges are due to (i) the perspective of the scene (which leads to occlusions among targets), (ii) a column in the middle of the scene (which causes complete occlusions), and (iii) a dynamic background (which does not allow an effective background subtraction). Since a full-body person detector [39] is used, half-visible people in the scene cannot be localized, thus leading to the failure of our multi-person tracking in the lower part of the image (Fig. 14(h)). The graph in Fig. 13(c) shows that the maximum value of Precision is about 0.6 in person localization and the maximum value of Recall is about 0.8. The MT-TBD, in this case, can achieve Precision of 0.490 and Recall of 0.676, while the postprocessing

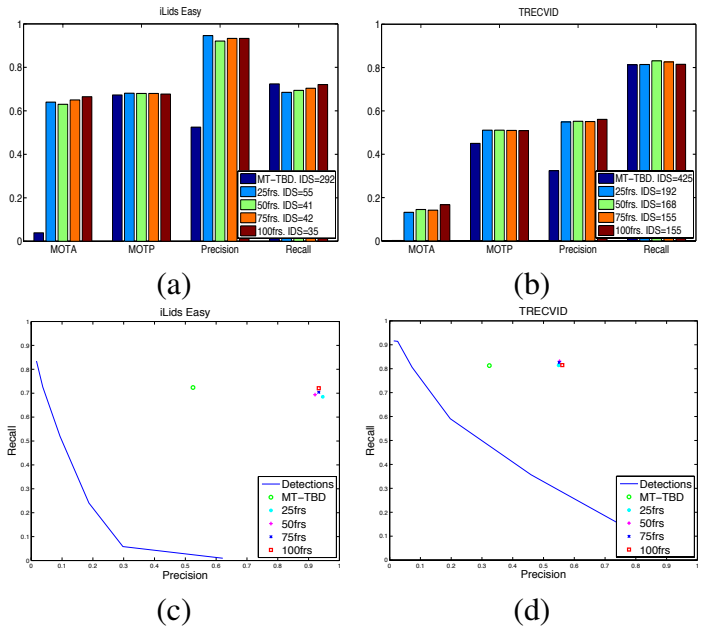


Figure 13: Results of the proposed tracker and person localization methods obtained on iLids Easy and TRECVID datasets. (a-b) Bar plots of MOTA, MOTP, Precision and Recall values by varying the temporal window τ used in the postprocessing. (c-d) Precision and Recall rates for the thresholded confidence map plotted along with the tracking scores that show tracking performance with respect to the input with varying threshold. The duration of the temporal window is indicated in frames (frs) within the legend. IDS: ID Switches.

considerably increases Precision while maintaining high values of Recall. These results are compared with the recent state-of-the-art methods and are summarized in Tab. 4. Considering the difficulty of localizing people in the lower part of the image, the difference between Recall of the proposed method and Breitenstein *et al.* [20] is 0.146 (such method uses the more effective Implicit Shape Model (ISM) for person localization) while a comparable Precision is obtained. In Fig. 14(e-f)) it is possible to see how track fusion allows tracking in case of complete occlusions.

With the TRECVID dataset, we validate the proposed method using a confidence map built on head localizations. The head localization reduces the effect of occlusions among targets in crowded scenarios, but since many objects in the scene have shape similar to heads (e.g. bags, shoulders and luggages), the localization contains a large number of false positives (Fig. 13(d)). Comparatively with

Table 5: Comparison of results on the ETH dataset using the evaluation toolbox provided by Li *et al.* [4]. The number of frames between round brackets represents the temporal window duration τ. Key: GT: Ground-Truth trajectories; MT: Mostly Tracked; PT: Partially Tracked; ML: Mostly Lost; Frag: Fragments; IDS: ID Switches.

Method	**Recall**	**Precision**	**GT**	**MT**	**PT**	**ML**	**Frag**	**IDS**
MT-TBD (75frs)	0.787	0.855	125	0.624	0.296	0.080	69	45
Kuo2011 [9]	0.768	0.866	125	0.584	0.336	0.080	23	11
Yang2012 [12]	0.790	0.904	125	0.680	0.248	0.072	19	11

iLids dataset (Fig. 13), Precision remains higher in TRECVID since the spread function of the localized heads is smaller than the person localizations in iLids. Hence, head localization turns out to have higher Precision than that for bodies at same Recall values. Qualitative tracking results are shown in Fig. 14(m-p): it is possible to notice the long tracks belonging to the heads and the false positive tracks. The quantitative evaluation is given in Fig. 13(b,d). The improvement of the tracker with respect to the confidence map is shown in Fig. 13(d), where Recall of 0.813 and Precision of 0.324 are achieved. Then, the postprocessing phase improves the Precision rate by around 20% at the cost of a slight decrease of Recall.

The last dataset we use for validation is ETH (Fig. 14(q-t)), where full-bodies [9] are represented as 2D Dirac deltas over the space. The experiments on this dataset are run with a latency of 3 seconds for the proposed approach and compared with the recent *offline* methods proposed by Kuo *et al.* [9] and Yang and Nevatia [12]. The performance comparison is shown in Tab. 5. The performance of the proposed method is comparable with the state-of-the-art methods, despite the diversity of the working modalities. Recall is at the same level as the one obtained in Yang and Nevatia [12] and Precision is slightly lower. Note that, since our method is not offline, some short tracks may not be fused together leading to a higher number of fragmented tracks (second-last column in Tab. 5).

6.6. Computational cost

The overall complexity of MT-TBD with N particles has an upper bound of $O(N\log(N))$ operations. Specifically, for the motion model, the proposal distribution and the multinomial resampling (Sec. 3.1 and 3.3) the cost is $O(N)$, as these operations are sequential on the number of particles. For the neighborhood search of Eq. 15, we give as input a set of spatially ordered particles at the cost of $O(N\log(N))$ and we use a method based on binary search [42] whose cost

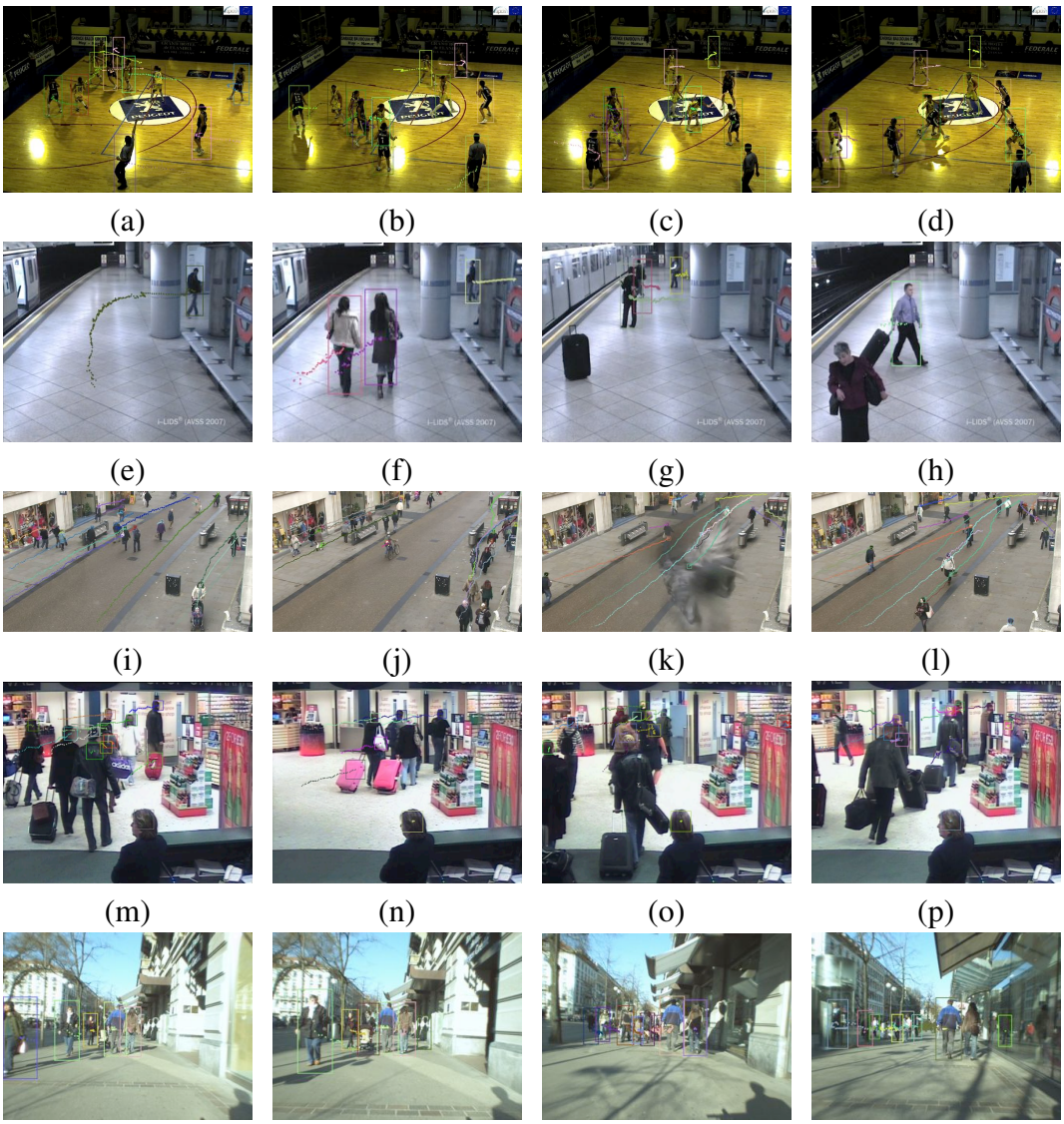


Figure 14: Sample tracking results of the proposed method on (a-d) APIDISC2 dataset, (e-h) iLids-Easy, (i-l) TownCentre, (m-p) TRECVID and (q-t) ETH datasets. The visualization of tracks for APIDISC2 and TRECVID are truncated to the last 50 frames to make the examples clearer. The tracks for TownCentre, iLids and ETH are shown from the initialization of the track.

is $O(\log(N))$. For the Mean-Shift clustering, the operation is performed on the complete set of N particles with complexity $O(N \log(N))$ [43].

7. Conclusions

We presented a Bayesian method for multi-object tracking based on *track-before-detect*, which utilizes a Markov Random Field applied on the particles to perform tracking of unknown and large number of targets, and by probabilistically managing the ID assignment to avoid ID switches of close targets. The state estimate of a target is performed via Mean-Shift clustering and supported by Mixture of Gaussians in order to enable an accurate assignment of IDs within each single cluster. The birth and death of the targets at each iteration of the filter is modeled with a Markov Random Field. The computational complexity is proportional to the number of particles only. The robustness of our algorithm was demonstrated by applying the method on sport and surveillance datasets with different perspective views, partial and full occlusions of targets, different backgrounds, variable number of people and moving cameras. We showed the flexibility of the proposed tracker by giving as input the results of different target localization methods, and by obtaining comparable or better results compared to recent methods from the state of the art.

As future work, the proposed method will be improved by including a multi-dynamic switching model [23] to deal with different motions of the observed targets and by developing an automatic method for estimating the filter parameters, such as the variance of the target spread function, the motion model and the likelihood function.

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (2006) 1–45.

[2] Y. Boers, J. Driessen, Multitarget particle filter track-before-detect applications, IEE Proc. Radar, Sonar and Navigation 151 (2004) 351–357.

[3] C. Huang, B. Wu, R. Nevatia, Robust object tracking by hierarchical association of detection responses, in: Proc. of European Conference on Computer Vision, Marseille, France, 2008, pp. 788–801.

[4] Y. Li, C. Huang, R. Nevatia, Learning to associate: Hybridboosted multi-target tracker for crowded scene, in: Proc. of IEEE Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 2953–2960.

[5] E. Maggio, A. Cavallaro, Learning scene context for multiple object tracking, IEEE Trans. on Image Processing 18 (2009) 1873–1884.

[6] C. Kuo, C. Huang, R. Nevatia, Multi-target tracking by on-line learned discriminative appearance models, in: Proc. of IEEE Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 685–692.

[7] M. Rodriguez, I. Laptev, J. Sivic, J. Audibert, Density-aware person detection and tracking in crowds, in: Proc. of IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2423–2430.

[8] B. Yang, C. Huang, R. Nevatia, Learning affinities and dependencies for multi-target tracking using a CRF models, in: Proc. of IEEE Computer Vision and Pattern Recognition, Colorado Springs, USA, 2011, pp. 1233–1240.

[9] C. Kuo, R. Nevatia, How does person identity recognition help multi-person tracking?, in: Proc. of IEEE Computer Vision and Pattern Recognition, Colorado Springs, USA, 2011, pp. 1217–1224.

[10] B. Benfold, I. Reid, Stable multi-target tracking in real-time surveillance video, in: Proc. of IEEE Computer Vision and Pattern Recognition, Colorado Springs, USA, 2011, pp. 3457–3464.

[11] B. Yang, R. Nevatia, Multi-target tracking by online learning of non-linear motion patterns and robust appearance model, in: Proc. of IEEE Computer Vision and Pattern Recognition, Providence, Rhode Island, USA, 2012, pp. 1918–1925.

[12] B. Yang, R. Nevatia, An online learned CRF model for multi-target tracking, in: Proc. of Computer Vision and Pattern Recognition, Providence, Rhode Island, USA, pp. 2034–2041.

[13] Z. Khan, T. Balch, F. Dellaert, MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements, IEEE Trans. on Pattern Analysis and Machine Intelligence 28 (2006) 1960–1972.

[14] J. Czyz, B. Ristic, B. Macq, A particle filter for joint detection and tracking of color objects, Image and Vision Computing 25 (2007) 1271–1281.

[15] J. Xing, H. Ai, S. Lao, Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses, in: Proc. of IEEE Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 1200–1207.

[16] I. Ali, M. N. Dailey, Multiple human tracking in high-density crowds, in: Proc of Conference on Advanced Concepts for Intelligent Vision Systems, Bordeaux, France, 2009, pp. 540–549.

[17] M. Yang, F. Lv, W. Xu, Y. Gong, Detection driven adaptive multi-cue integration for multiple human tracking, in: Proc. of IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 1554–1561.

[18] B.-N. Vo, B.-T. Vo, N.-T. Pham, D. Suter, Joint detection and estimation of multiple objects from image observations, IEEE Trans. on Signal Processing 58 (2010) 5129–5241.

[19] H. Tong, H. Zhang, H. Meng, X. Wang, Multitarget tracking before detection via probability hypothesis density filter, in: Int. Conference on Electrical and Control Engineering, Wuhan, China, 2010, pp. 1332–1335.

[20] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. V. Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, IEEE Trans. on Pattern Analysis and Machine Intelligence 33 (2011) 1820–1833.

[21] S. Stalder, H. Grabner, L. V. Gool, Cascaded confidence filtering for improved tracking-by-detection, in: Proc. of European Conference on Computer Vision, Crete, Greece, 2010, pp. 369–382.

[22] T. D. Laet, H. Bruyninckx, J. D. Schutter, Shape-based online multitarget tracking and detection for targets causing multiple measurements: Variational bayesian clustering and lossless data association, IEEE Trans. on Pattern Analysis and Machine Intelligence 33 (2011) 2477–2491.

[23] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter: particle filters for tracking applications, Artech House, Boston, 2004.

[24] S. Buzzi, M. Lops, L. Venturino, M. Ferri, Track-before-detect procedures in a multi-target environment, IEEE Trans. of Aerospace and Electronic Systems 44 (2008) 1135–1150.

[25] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. of IEEE Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005, pp. 886–893.

[26] S. Ali, M. Shah, Floor fields for tracking in high density crowd scenes, in: Proc. of European Conference on Computer Vision, Marseille, France, 2008, pp. 1–14.

[27] M. Rodriguez, S. Ali, T. Kanade, Tracking in unstructured crowded scenes, in: Proc. of IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 1389–1396.

[28] D. Reid, An algorithm for tracking multiple targets, IEEE Trans. on Automatic Control 24 (1979) 843–854.

[29] B. Wu, R. Nevatia, Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors, International Journal of Computer Vision 75 (2007) 247–266.

[30] W. Ng, J. Li, S. Godsill, J. Vermaak, A review of recent results in multiple target tracking, in: Proc. of Image and Signal Processing and Analysis, Trieste, Italy, 2005, pp. 40–45.

[31] R. Hoseinnezhad, B.-N. Vo, D. Suter, B.-T. Vo, Multi-object fitlering from image sequence without detection, in: Proc. of International Conference on Acoustic, Speech and Signal Processing, Dallas, Texas, USA, 2010, pp. 1154–1157.

[32] J. Hosking, J. Wallis, Parameter and quantile estimation for the generalized Pareto distribution, Technometrics 29 (1987) 339–349.

[33] D. Comaniciu, P. Meer, Distribution free decomposition of multivariate data, IEEE Trans. on Pattern Analysis and Machine Intelligence 2 (1999) 22–30.

[34] R. Kindermann, J. L. Snell, Markov Random Fields and their applications, American Mathematical Society, Providence, Rhode Island, 2000.

[35] R. Hogg, J. McKean, A. Craig, Introduction to Mathematical Statistics, Upper Saddle River, NJ, 2005.

[36] R. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley & Sons, Singapore, 2001.

[37] R. Mazzon, S. F. Tahir, A. Cavallaro, Person re-identification in crowd, Pattern Recognition Letters DOI: 10.1016/j.bbr.2011.03.031 (Available online 1 March 2012).

[38] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Trans. on Pattern Analysis and Machine Intelligence 34 (2012) 743–761.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. on Pattern Analysis and Machine Intelligence 32 (2010) 1627–1645.

[40] R. Eshel, Y. Moses, Homography based multiple camera detection and tracking of people in a dense crowd, in: Proc. of IEEE Computer Vision and Pattern Recognition, Anchorage, AK, US, 2008, pp. 1–8.

[41] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: The CLEAR MOT metrics, EURASIP Journal on Image and Video Processing 2008 (2008) 246–309.

[42] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to algorithms, MIT press, Cambridge, MA, 1990.

[43] B. Georgescu, I. Shimshoni, P. Meer, Mean-Shift based clustering in high dimensions: a texture classification example, in: Proc. of IEEE International Conference on Computer Vision, Beijing, China, 2003, pp. 456–463.