Online multi-target tracking with strong and weak detections

Ricardo Sanchez-Matilla, Fabio Poiesi and Andrea Cavallaro

Centre for Intelligent Sensing
Queen Mary University of London
{ricardo.sanchezmatilla,fabio.poiesi,a.cavallaro}@qmul.ac.uk

Abstract. We propose an online multi-target tracker that exploits both high- and low-confidence target detections in a Probability Hypothesis Density Particle Filter framework. High-confidence (strong) detections are used for label propagation and target initialization. Low-confidence (weak) detections only support the propagation of labels, i.e. tracking existing targets. Moreover, we perform data association just after the prediction stage thus avoiding the need for computationally expensive labeling procedures such as clustering. Finally, we perform sampling by considering the perspective distortion in the target observations. The tracker runs on average at 12 frames per second. Results show that our method outperforms alternative online trackers on the Multiple Object Tracking 2016 and 2015 benchmark datasets in terms tracking accuracy, false negatives and speed.

Keywords: Multi-Target Tracking, Probability Hypothesis Density, Particle Filter.

1 Introduction

Multi-target tracking-by-detection performs temporal association of target detections to estimate trajectories, while compensating for miss-detections and rejecting false-positive detections. Trajectories can be generated online [1], offline [2] or with a short latency [3]. *Online trackers* estimate the target state at each time instant as detections are produced. In case of miss-detections, online trackers may rely on predictive models to continue tracking until a matching detection is found [4]. *Offline trackers* use both past and future detections and can therefore better cope with miss-detections using re-identification [5].

An effective filter for online state estimation is the Probability Hypothesis Density (PHD) filter, which can cope with clutter, spatial noise and miss-detections [6,7]. The PHD filter estimates the state of multiple targets by building a positive and integrable function over a multi-dimensional state whose integral approximates the expected number of targets [6,8]. The posterior function can be computed based on a Bayesian recursion that leverages the set of (noisy) detections and it is approximated using Sequential Monte Carlo for computational efficiency via a set of weighted random samples (particles) [8]. This approximation is known as the PHD Particle Filter (PHD-PF) and involves four

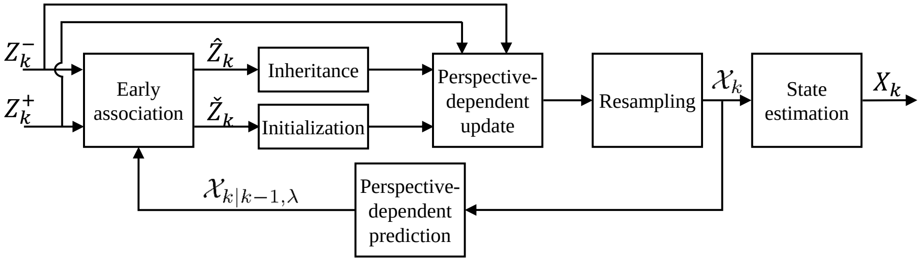


Fig. 1: Block diagram of the proposed multi-target tracking pipeline. At time k, the predicted particles $\mathcal{X}_{k|k-1,\lambda}$ are calculated with a perspective-dependent prediction. Strong Z_k^+ and weak Z_k^- detections are associated to the predicted states calculated from $\mathcal{X}_{k|k-1,\lambda}$. After the early association, two subsets of detections are used for tracking. Detections $\hat{Z}_k$ inherit the identity of the corresponding trajectories and are used for tracking existing states; $\check{Z}_k$ are un-associated strong detections and are used for initializing new states. After the perspective-dependent update, and resampling the particles $\mathcal{X}_k$ are used to estimate the states X_k.

main steps [6,8,7]: the *prediction* of particles over time; the *update* of the weights of the particles based on new detections; the *resampling* step to avoid that only few particles monopolize the whole mass; and *state estimation.* A PHD filter needs an additional mechanism to provide target identity information. For example, the particles can be clustered and labeled after resampling to enable the temporal association of the clusters with previous states [7,9,10]. This additional mechanism is computationally expensive and error prone.

In this paper, we formulate an early association strategy between trajectories and detections after the prediction stage, which allows us to perform target estimation and state labeling without any additional mechanisms. Our online multi-target tracker exploits both strong (certain) and weak (uncertain) detections. Strong detections have a higher confidence score and are used for initialization and tracking. Weak detections have a lower confidence score and are used only to support the continuation of an existing track when strong detections are missing. We also introduce a perspective-dependent sampling mechanism to create newborn particles depending on their distance from the camera. Fig. 1 shows the block diagram of the proposed online multi-target tracker.

In summary, our contributions include (i) a strategy for the effective exploitation of low-confidence (weak) detections; (ii) a procedure to label states via an early association strategy; (iii) the exploitation of perspective in prediction, update and newborn particle generation. The tracker works on average at 12 frames per second (fps) on an i7 3.40GHz, 16GB RAM computer, without any parallelization. We validate our tracking pipeline without using any appearance features and compare our method against state-of-the-art alternatives on the MOT15 and MOT16 benchmark datasets.

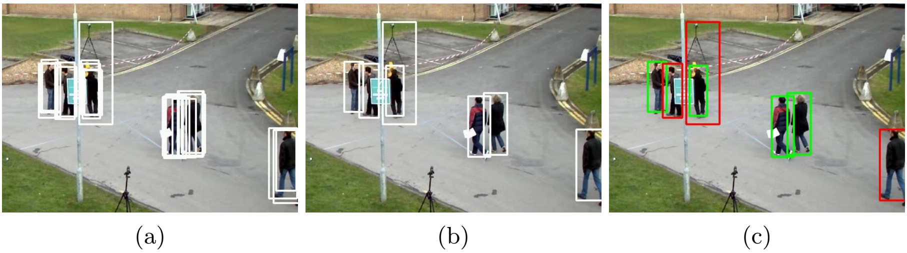


Fig. 2: Example of strong and weak detections at frame 43 (crop) in PETS09-S2L1. (a) Initial target detections Z_k^*; (b) combined detections Z_k; (c) strong detections (green) Z_k^+ and weak detections (red) Z_k^- after classification.

2 Strong and weak detections

Let a (large) set of target detections $\mathbf{z}_k^* \in Z_k^*$, ideally without false negatives but potentially with multiple false positives, be generated at each time step k (Fig. 2 (a)). These detections can be produced for example by running in parallel multiple detectors, by changing the operational parameters of a detector or with a combination of these two approaches. During this 'over-detection' process a target is likely to generate multiple overlapping detections. Overlapping detections produced by the same target may be combined into a single detection $\mathbf{z}_k \in Z_k$ using, for example, non-maxima suppression [11,12] (Fig. 2 (b)) forming the set of combined detections Z_k. Let each combined target detection be defined as

$$\mathbf{z}_k = (x_k, y_k, w_k, h_k)^T , \tag{1}$$

where (x_k, y_k) is the center and (w_k, h_k) are the width and height of the bounding box of the detection on the image plane. Let each $\mathbf{z}_k$ be associated to a detection confidence-score $s_k \in [0, 1]$.

We categorize the set Z_k based on s_k into two subsets: strong (certain) and weak (uncertain) detections (Fig. 2 (c)). This categorization can be obtained using the score confidences or via learning certain metrics on a training dataset [13]. *Strong* detections $Z_k^+ = \{\mathbf{z}_k^+ : s_k \geq \tau_s\}$, where τ_s is a confidence threshold, are more likely to be true positives. We will use strong detections for trajectory initialization and for tracking existing targets. *Weak* detections $Z_k^- = \{\mathbf{z}_k^- : s_k < \tau_s\}$ are potential false positives. We will use weak detections for tracking existing targets to shorten the prediction time and to maintain the tracking uncertainty low. The value of τ_s influences the ratio between the false positives and the false negatives, as we discuss in Section 6.1.

3 Perspective-dependent prediction

Let Λ_k be the set of existing identities at time k whose elements are $\lambda \in \Lambda_k$. Let the state be defined as

$$\mathbf{x}_{k,\lambda} = \left(x_{k,\lambda}, \dot{x}_{k,\lambda}, y_{k,\lambda}, \dot{y}_{k,\lambda}, w_{k,\lambda}, h_{k,\lambda}\right)^T, \tag{2}$$

where $(x_{k,\lambda}, y_{k,\lambda})$ is the center, $(\dot{x}_{k,\lambda}, \dot{y}_{k,\lambda})$ are the horizontal and vertical components of the velocity, $(w_{k,\lambda}, h_{k,\lambda})$ are the width and height and λ is the identity of the estimated state. Let the set of all estimated states at k be X_k whose elements are $\mathbf{x}_{k,\lambda} \in X_k$. The elements of this set are obtained at each time step from the set of all existing particles $\mathcal{X}_k$ whose elements are $\mathbf{x}^i_{k,\lambda} \in \mathcal{X}_k$, where $\mathbf{x}^i_{k,\lambda}$ is the i^{th} particle.

The prediction step assumes the motion of a target to be independent from the others and propagates each particle $\mathbf{x}^i_{k-1,\lambda}$ as

$$\mathbf{x}^i_{k,\lambda} = G_k \mathbf{x}^i_{k-1,\lambda} + N_k, \tag{3}$$

where G_k is an affine transformation defined as

$$G_k = \begin{pmatrix} A_k & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & B_k & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \end{pmatrix}, \tag{4}$$

where $\mathbf{0}$ and $\mathbf{I}$ are the zero and identity matrices, respectively. N_k is an additive Gaussian noise defined as $N_k = \left(n^x_k, n^{\dot{x}}_k, n^y_k, n^{\dot{y}}_k, n^w_k, n^h_k\right)^T$, where each component of N_k is an independent Gaussian variable with zero mean and standard deviation proportional to the bounding box size in the previous frame.

As a target moving at constant velocity produces a smaller apparent displacement on the image plane when it is farther from the camera, we improve the model in Eq. 4 by considering the effect of foreshortening. To this end, we model N_k as a function of the distance from the camera. Specifically, we set the standard deviation of the noise for the horizontal and vertical components to be proportional to the width $w_{k-1,\lambda}$ and height $h_{k-1,\lambda}$ of the state, respectively.

In addition to the above, target acceleration variations, noisy detections and camera motion may generate erroneous predictions. To address these problems, instead of relying only on the previous time step [7], we average the past M states over a longer time interval $[t-M, t-1]$. Therefore, A_k and B_k dynamically update the position and velocity via the average velocity in the previous M frames:

$$A_k = \begin{pmatrix} 1 & \frac{u_{k,\lambda}}{\dot{x}_{k,\lambda}} \\ 0 & \frac{u_{k,\lambda}}{\dot{x}_{k,\lambda}} \end{pmatrix}, \; B_k = \begin{pmatrix} 1 & \frac{v_{k,\lambda}}{\dot{y}_{k,\lambda}} \\ 0 & \frac{v_{k,\lambda}}{\dot{y}_{k,\lambda}} \end{pmatrix}, \tag{5}$$

where $u_{k,\lambda}, v_{k,\lambda}$ are the average horizontal and vertical velocities of the estimated state $\mathbf{x}_{k,\lambda}$, respectively, whose values are computed as

$$\left(u_{k,\lambda}, v_{k,\lambda}\right) = \frac{1}{M} \sum_{j=1}^{M} \left(x_{k-j,\lambda}, y_{k-j,\lambda}\right), \tag{6}$$

where $M = \min\left(M_{k,\lambda}, M_{max}\right)$, $M_{k,\lambda}$ is the number of time steps since the target $\mathbf{x}_{k,\lambda}$ was initialized and M_{max} the maximum number of time steps.

The weights of the particles, $p^i_{k|k-1}$, are not modified during the prediction state, therefore

$$\pi^i_{k|k-1} = \pi^i_{k-1}, \qquad i = 1, ..., L_{k-1}, \tag{7}$$

where L_{k-1} is the number of existing particles at $k-1$.

4 Labeling

4.1 Early association (EA)

The PHD-PF estimates the state of each target without labels (i.e. without identity) [9]. Let π^i_{k-1} be the weight associated to particle $\mathbf{x}^i_{k-1,\lambda}$. The PHD-PF posterior $D_{k-1|k-1}(\cdot)$ is approximated as

$$D_{k-1|k-1}(\mathbf{x}_{k-1,\lambda}) \approx \sum_{i=1}^{L_{k-1}} \pi^i_{k-1} \delta\left(\mathbf{x}_{k-1,\lambda} - \mathbf{x}^i_{k-1,\lambda}\right), \tag{8}$$

where $\delta(\cdot)$ is the Kronecker's delta function. Various works have been published aiming to address the lack of identities in the PHD-PF: (i) clustering after resampling the particles on the right-hand side of Eq. 8 [7], (ii) keeping a separate tracker for each target and then perform 'peak-to-track' association [14], (iii) combining clustering techniques with the introduction of hidden identifiers to the samples of the PHD [15,10,8]. These solutions are computationally expensive and may introduce estimation errors. To avoid these problems, we move the association stage earlier in the pipeline.

We associate the elements of Z^+_k and Z^-_k to the predicted states using the Hungarian algorithm [16]. We refer to this association as *early association* because, unlike [8,15,7], it is performed before the update and resampling stages. The association cost, ω_k, between a detection $\mathbf{z}_k$ and the predicted state $\mathbf{x}_{k|k-1,\lambda}$ is

$$\omega_k = \frac{d_l(\mathbf{z}_k, \mathbf{x}_{k|k-1,\lambda})}{Q_l} \cdot \frac{d_s(\mathbf{z}_k, \mathbf{x}_{k|k-1,\lambda})}{Q_s}, \tag{9}$$

where $d_l(\cdot)$ and $d_s(\cdot)$ are the Euclidean distances between the position and bounding box size elements, respectively. Q_l is the diagonal of the image (i.e. the maximum position variation) and Q_s is the area of the image (i.e. the maximum size variation). Note that we multiply the normalized distances instead of averaging them to penalize when they are dissimilar (e.g. when two targets are far from each other in the scene but appear close to each other on the image plane).

When a trajectory is not associated to any (strong or weak) detections, the state is estimated using existing particles only. When the trajectory is not associated to any detections for a certain temporal interval (V frames, see Section 6.1), the state will be discarded before the EA and therefore the weight of its particles will gradually decrease toward zero.

EA enables the tracker to generate newborn particles that inherit the properties of its associated state (*inheritance*) or that produce a new identity (*initialization*).

4.2 Inheritance

Strong detections Z_k^+ generate J_k newborn particles to repopulate the area around existing states. The newborn particles are added to the L_{k-1} existing particles. In [8,7], the newborn particles are created from a newborn importance function $p_k(\cdot)$ [7], which can be independently modeled from the estimated states, as a Gaussian process:

$$\mathbf{x}_{k,\lambda}^i \sim p_k(\mathbf{x}_{k,\lambda}^i|\mathbf{z}_k^+) = \frac{1}{|Z_k^+|} \sum_{\forall \mathbf{z}_k^+ \in Z_k^+} \mathcal{N}(\mathbf{x}_{k,\lambda}^i; \mathbf{z}_k^+, \Sigma), \tag{10}$$

where $|\cdot|$ is the cardinality of a set, $\mathcal{N}(\cdot)$ is a Gaussian distribution and Σ is the covariance matrix. The covariance matrix can be dynamically updated based on parameters as detection size or video frame rate (see Section 6.1). Each newborn particle has an associated weight, π_k^i, defined as

$$\pi_k^i = \frac{1}{J_k} \frac{\gamma_k(\mathbf{x}_{k,\lambda}^i)}{p_k(\mathbf{x}_{k,\lambda}^i|\mathbf{z}_k^+)}, \quad i = L_{k-1}+1, ..., L_{k-1}+J_k, \tag{11}$$

where $\gamma_k(\cdot)$ is the birth intensity, which is assumed to be constant when no prior knowledge about the scene is available [7]. Typically, J_k is chosen to have, on average, ρ particles per newborn target [8]. The process described in Eq. 10 could create newborn particles that are dissimilar from the corresponding state as they are independently created.

Unlike [17,15] that included identities in the state, we consider the identity $\lambda \in \Lambda_k$ as attribute of the state and propagate it over time. Therefore, $|\Lambda_k|$ is the estimated number of targets at time k.

Let $\hat{Z}_k^+$ and $\hat{Z}_k^-$ be the sets that contain, respectively, strong and weak detections that are *associated* to one of the predicted states, i.e. to an existing trajectory. Let $\hat{Z}_k = \hat{Z}_k^+ \cup \hat{Z}_k^-$ be the set of detections that inherit the identity of the corresponding trajectories.

We create newborn particles from $\hat{Z}_k$ and inherit properties from their associated predicted states: the position and bounding box size are created from detections, whereas velocity and identity are inherited from the associated states.

We use a mixture of Gaussians as importance function to sample position and bounding box size elements from the detections as

$$\mathbf{x}_{k,\lambda}^i \sim p_k(\mathbf{x}_{k,\lambda}^i|\hat{\mathbf{z}}_k) = \frac{1}{|\hat{Z}_k|} \sum_{\forall \hat{\mathbf{z}}_k \in \hat{Z}_k} \mathcal{N}(C\mathbf{x}_{k,\lambda}^i; C\hat{\mathbf{z}}_k, C\Sigma_k), \tag{12}$$

where

$$C = \begin{pmatrix} D & 0_{4\times2} \\ 0_{2\times4} & I_{2\times2} \end{pmatrix}, \ D = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,0 \end{pmatrix}, \tag{13}$$

and

$$\Sigma_k = diag(\sigma_k^x, \sigma_k^{\dot{x}}, \sigma_k^y, \sigma_k^{\dot{y}}, \sigma_k^w, \sigma_k^h)^T, \tag{14}$$

where Σ_k is a time-variant standard deviations matrix that is defined based on the size of the detection bounding box and the weight of the newborn particles is calculated as in Eq. 11. This solution allows us to address the perspective distortion during the generation of newborn particles. The values of these standard deviations are learned from a training dataset (see Section 6.1).

The velocities and the identity are inherited from the trajectory as

$$\begin{aligned} \dot{x}_{k,\lambda}^i &= \dot{x}_{k-1,\lambda} + n_k^{\dot{x}}, \\ \dot{y}_{k,\lambda}^i &= \dot{y}_{k-1,\lambda} + n_k^{\dot{y}}, \\ \lambda_k &= \lambda_{k-1}, \end{aligned} \tag{15}$$

where $(\dot{x}_{k-1,\lambda}, \dot{y}_{k-1,\lambda})$ are the velocity components of a trajectory X_λ (i.e. each state with identity λ for all k) and $n_k^{\dot{x}}$ and $n_k^{\dot{y}}$ are Gaussian noises that model the velocity variations of a target.

Fig. 3 shows the benefit of weak detections. Without weak detections, miss-detections produce false negative trajectories and identity switches (Fig. 3, second row). When weak detections are used the targets are correctly tracked (Fig. 3, third row).

4.3 Initialization

While un-associated *weak* detections are discarded after EA, un-associated *strong* detections form the set $\check{Z}_k = Z_k^+ \backslash \hat{Z}_k^+$ and initialize new target identities. Newborn particles associated to a *new target* are generated in a limited volume of the state space around the un-associated strong detections. The same new identity is assigned to each newborn particle.

We treat spawning targets as new targets. The newborn importance function $p_k(\cdot)$ in Eq. 10 can regulate where targets are likely to spawn or enter in a scene [18]. Each detection in $\check{Z}_k$ initializes a *new trajectory* and generates newborn particles using a mixture of Gaussians as

$$\mathbf{x}_{k,\lambda}^i \sim p_k(\mathbf{x}_{k,\lambda}^i | \check{\mathbf{z}}_k) = \frac{1}{|\check{Z}_k|} \sum_{\forall \check{\mathbf{z}}_k \in \check{Z}_k} \mathcal{N}(\mathbf{x}_{k,\lambda}^i; \check{\mathbf{z}}_k, \Sigma_k), \tag{16}$$

where Σ_k is defined in Eq. 14 and the weights of the particles are calculated as in Eq. 11.

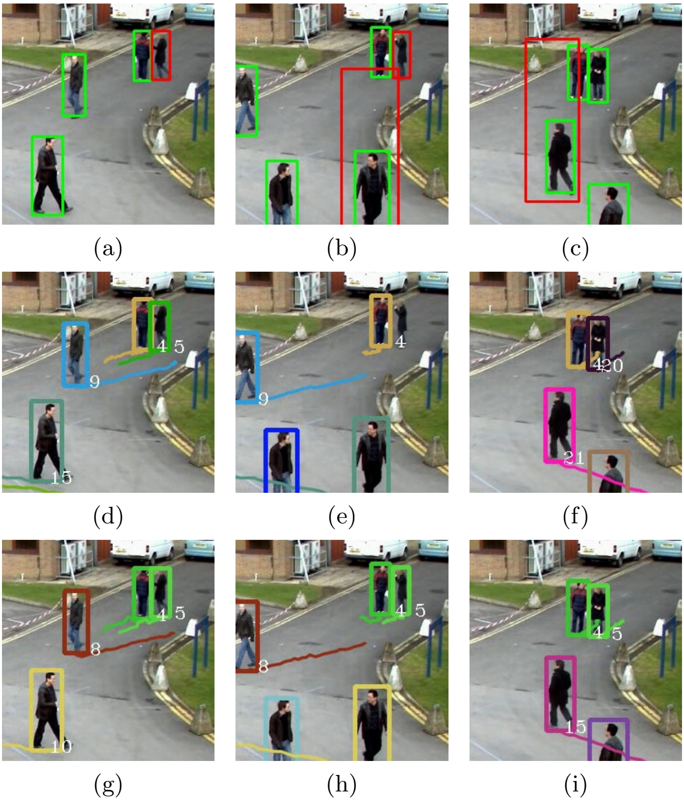
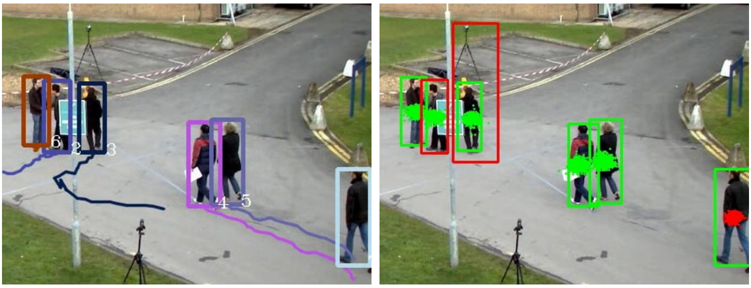


Fig. 3: Examples of tracking at frames 178, 193 and 240 (crops) in PETS09-S2L1 (not) using weak detections. (a), (b), (c) Strong (green) and weak (red) detections. (d), (e), (f) Without using weak detections target 5 is lost and a new trajectory is later initialized with identity 20. (g), (h), (i) Using weak detections target 5 is correctly tracked.

Fig. 4 shows an example of how newborn particles are created. The target on the right is initialized because of the presence of an un-associated strong detection. The target with identity number 2 is localized with a weak detection. The weak detection in this case is a false positive that is discarded because it is not associated with any predicted states.

5 Perspective-dependent update, resampling and state estimation

Let the set of particles that share the same identity be $\mathcal{X}_{k,\lambda}$ whose elements are $\mathbf{x}^i_{k,\lambda} \in \mathcal{X}_{k,\lambda}$. After new detections are generated, the weights of the particles, π^i_k, are recalculated for allowing the particles to update the estimation [9,19,7].



(a) (b)

Fig. 4: Example of newborn particles generated at frame 43 (crop) in PETS09-S2L1. (a) Color-coded target identities; (b) existing particles (green dots) and newborn particles (red dots). The newborn particles initialize a new trajectory from an un-associated strong detection.

The weights at k are *updated* as

$$\pi_k^i = \left[p_M + \sum_{\forall \mathbf{z}_k \in Z_k} \frac{(1-p_M) g_k(\mathbf{z}_k|\mathbf{x}_{k,\lambda}^i)}{\kappa_k(\mathbf{z}_k) + C_k(\mathbf{z}_k)} \right] \pi_{k|k-1}^i, \tag{17}$$

where p_M is the probability of miss-detection, $\kappa_k(\cdot)$ is the clutter intensity associated to a detection $\mathbf{z}_k$ [19,7], $C_k(\mathbf{z}_k)$ is defined as

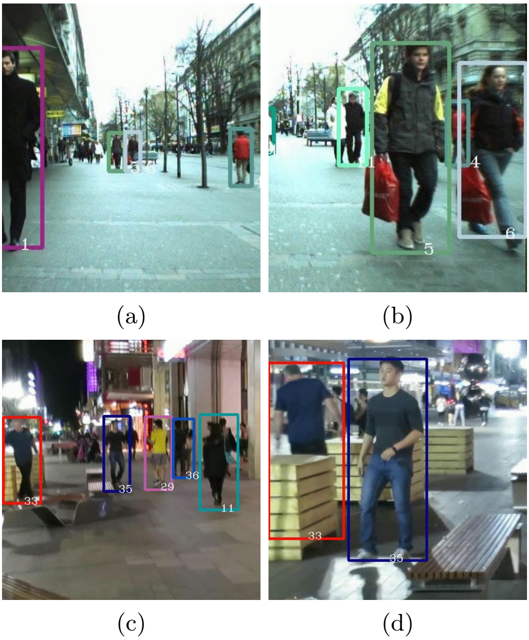
$$C_k(\mathbf{z}_k) = \sum_{i=1}^{L_{k-1}+J_k} (1-p_M) g_k(\mathbf{z}_k|\mathbf{x}_{k,\lambda}^i)\, \pi_{k|k-1}^i, \tag{18}$$

and $g_k(\mathbf{z}_k|\mathbf{x}_{k,\lambda}^i)$ is the likelihood function defined as

$$g_k(\mathbf{z}_k|\mathbf{x}_{k,\lambda}^i) = \mathcal{N}(C\mathbf{z}_k; C\mathbf{x}_{k,\lambda}^i, C\Sigma_k), \tag{19}$$

where C is defined in Eq. 13. The likelihood function, $g_k(\cdot)$ computes the location and bounding box similarities. Unlike [7,10] where $\Sigma = \Sigma_k$ is fixed, we define Σ_k in Eq. 14 as a time-variant matrix that regulates the location and bounding box similarity between particles and detections (i.e. the particles of an object far from the camera will be less spread than those of a closer object due to the perspective). Fig. 5 shows examples of the use of the proposed perspective-dependent approach.

After the update step, *resampling* helps avoiding the degeneracy problem [20]. The standard multinomial resampling [20,8] splits particles proportionally to their weights, frame-by-frame independently. Because newborn particles have in general a lower weight than existing particles, new targets may not be initialized due to repetitive deletion of their particles during resampling. To allow newborn particles to grow over time and reach a comparable weight to that of



(a) (b)

(c) (d)

Fig. 5: Examples of tracking under perspective changes at frames 32 and 102 (crops) in ETH-Bahnhof, and at frames 178 and 375 (crops) in ADL-Rundle-8. Targets 5 and 6 (see (a), (b)) and targets 33 and 35 (see (c), (d)) are correctly tracked despite considerable perspective changes.

existing particles, newborn particles are resampled independently from existing particles using a Multi-stage Multinomial Resampling step [7]. Finally, each state $\mathbf{x}_{k,\lambda} \in X_k$ is estimated as the average of all resampled particles sharing the same identity:

$$\mathbf{x}_{k,\lambda} = \frac{1}{|\mathcal{X}_{k,\lambda}|} \sum_{\forall \mathbf{x}^i_{k,\lambda} \in \mathcal{X}_{k,\lambda}} \mathbf{x}^i_{k,\lambda}. \tag{20}$$

6 Results

6.1 Experimental setup

We validate the proposed tracker[1], the Early Association Probability Hypothesis Density Particle Filter (EA-PHD-PF), and compare it against state-of-the-art online tracking methods on the MOT15 and MOT16 benchmark datasets (`motchallenge.net`) [21,22]. We use the *public detections* provided by the MOT benchmark and our *private detections* produced by combining detections from

[1] Results are available at: `http://www.eecs.qmul.ac.uk/~andrea/eamtt.html`.

state-of-the-art person detectors. We refer to the tracker using the public detections from MOT benchmark as EA-PHD-PF(Pub) and to the tracker using the private detections as EA-PHD-PF(Priv).

In the specific implementation presented here, the combined detection has position and bounding box size equal to the weighted average of the position and bounding box size of the detections that contributed to the combination. We use detections generated by Discriminatively Trained Deformable Part Models (DT-DPM) [12], Scale Dependent Pooling (SDP) [23], Aggregate Channel Features (ACF) [24] trained on INRIA (ACF-I) and Caltech (ACF-C) datasets. We reward detections generated by the combination of a larger number of detectors (possible true positives) and penalize isolated detections (possible false positives). We normalize the confidence score of each detector using the 99^{th} percentile of the detection scores generated by each detector over the training set (and truncating to 1). Then we combine all detections via voting when their overlap area divided by their union area exceeds $\tau_f = 1/3$. Given the normalized detection confidence of each detection, s_j, the confidence score of the combined detection, $s_k \in [0,1]$, is $s_k = \frac{U}{D^2}\sum_{j=1}^{U} s_j$, where U is the number of *contributing* detectors and D is the total number of detectors.

We allow to perform association between detections and predicted states only if their overlap area divided by their union area exceeds $\tau_a = 1/3$. The parameter that controls when a trajectory will not seek for more detections is $V = \lceil f \rceil / 1s$, where f is the frame-rate of the video sequence. The parameter that controls the maximum possible number of frames to consider in the prediction model is $M_{max} = \lceil f/2 \rceil / 1s$.

We train the parameters of our method on the MOT15 and MOT16 training datasets and then use these parameters in MOT15 and MOT16 testing sequences, respectively. For the set of public detections $\tau_s = 0.39$ in MOT15 and $\tau_s = 0.20$ in MOT16. For the set of private detections $\tau_s = 0.35$ in both datasets[2]. The number of particles per target, ρ, is 500. The standard deviation values used for the prediction, update and newborn particle generation are modelled as a function of the bounding box size as

$$\sigma^x = w_{x_k} std\left(\left\{\frac{1}{w_k^g}\frac{d^2 x_k^g}{dk^2}\right\}_{\forall g}\right), \quad \sigma^y = h_{x_k} std\left(\left\{\frac{1}{h_k^g}\frac{d^2 y_k^g}{dk^2}\right\}_{\forall g}\right),$$

$$\sigma^{\dot{x}} = w_{x_k} std\left(\left\{\frac{1}{w_k^g}\frac{d^3 x_k^g}{dk^3}\right\}_{\forall g}\right), \quad \sigma^{\dot{y}} = h_{y_k} std\left(\left\{\frac{1}{h_k^g}\frac{d^3 y_k^g}{dk^3}\right\}_{\forall g}\right),$$

$$\sigma^w = w_{x_k} std\left(\left\{\frac{1}{w_k^g}\frac{d^2 w_k^g}{dk^2}\right\}_{\forall g}\right), \quad \sigma^h = h_{x_k} std\left(\left\{\frac{1}{h_k^g}\frac{d^2 h_k^g}{dk^2}\right\}_{\forall g}\right),$$

where $g \in [1, G]$ indicates a state element of a ground-truth trajectory, $std(\cdot)$ is the standard deviation operation, $\frac{d^2(\cdot)}{dk^2}$ is the second derivative that quantifies

[2] Larger values of τ_s reduce the number of false positives and lead to a more conservative initialization of the trajectories.

Table 1: Online tracking results on the MOT15 (TOP-7 trackers) and on the MOT16 (all available trackers) test datasets. Dark gray indicates the best and light gray indicates the second best scores.

Dataset	Tracker	Det	MOTA	MOTP	FAF	MT (%)	ML (%)	FP	FN	IDS	Frag	Hz
MOT15	AMPL [21]	Priv	51.9	75.0	1.2	26.4	24.8	6,963	22,225	372	1,130	2.8
	LKDAT_CNN [21]	Priv	49.3	74.5	1.0	20.8	28.4	6,009	24,550	563	1,155	1.2
	MDP_SubCNN [27]	Priv	47.5	74.2	1.5	30.0	18.6	8,631	22,969	628	1,370	2.1
	justry [21]	Priv	45.2	74.7	2.4	40.6	16.0	14,117	18,769	764	1,413	2.6
	kalman_mdp [21]	Pub	37.2	74.5	2.8	38.7	13.3	16,196	20,328	2,065	1,856	21.8
	mLK [21]	Pub	35.1	71.5	1.0	12.3	38.3	5,678	33,815	383	1,175	1.0
	HybridDAT [21]	Pub	35.0	72.6	1.5	11.4	42.2	8,455	31,140	358	1,267	4.6
	EA-PHD-PF	Pub	22.3	70.8	1.4	5.4	52.7	7,924	38,982	833	1,485	12.2
	EA-PHD-PF	Priv	53.0	75.3	1.3	35.9	19.6	7,538	20,590	776	1,269	11.5
MOT16	AMPL [22]	Priv	50.9	77.0	0.5	16.7	40.8	3,229	86,123	196	639	1.5
	olCF [22]	Pub	43.2	74.3	1.1	11.3	48.5	6,651	96,515	381	1,404	0.4
	OVBT [22]	Pub	38.4	75.4	1.9	7.5	47.3	11,517	99,463	1,321	2,140	0.3
	GMPHD_HDA [22]	Pub	30.5	75.4	0.9	4.6	59.7	5,169	120,970	539	731	13.6
	EA-PHD-PF	Pub	38.8	75.1	1.4	7.9	49.1	8,114	102,452	965	1,657	11.8
	EA-PHD-PF	Priv	52.5	78.8	0.7	19.0	34.9	4,407	81,223	910	1,321	12.2

the noise in the variation of x, y, w and h over time, and $\frac{d^3(\cdot)}{dk^3}$ is the third derivative that quantifies the noise in the variation of $\dot{x}$ and $\dot{y}$ over time. We use the bounding box size at time k, w_{x_k} and h_{x_k}, in order to adapt the noise to the scale of the bounding box. Note that estimated states are used in the prediction step (Section 3), whereas detections are used in the generation of newborn particles (Section 4) and update step (Section 5).

The evaluation measures are Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP) [25], False Alarm per Frame (FAF), Mostly Tracked targets (MT), Mostly Lost targets (ML) [26], Fragmented trajectories (Frag), False Positives (FP), False Negatives (FN), Identity Switches (IDS) and tracker speed in Hz. For a detailed description of each metric, please refer to the MOT website and [21].

6.2 Discussion

Table 1 compares the tracking results of our proposed method using both public and private detections with other online trackers submitted to the MOT15 and MOT16 benchmark[3]. The upper part of the table (MOT15) shows that EA-PHD-PF(Priv) outperforms AMPL, LKDAT_CNN, MDP_SubCNN and *justry*, in terms of MOTA. The number of FN and the ML percentage are overall lower than the other trackers. This is due to the ability of EA-PHD-PF(Priv) to robustly perform state estimation exploiting weak detections without relying on the prediction only when (strong) detections are missing. The higher number of IDS compared to the other methods is due to the fact that we rely only on the position and size of the bounding box inferred from the detections and *we are not using any appearance models* to discriminate nearby targets. Moreover, we do not model spawning targets. Therefore, identity switches are more likely

[3] Last accessed on 10th August 2016.

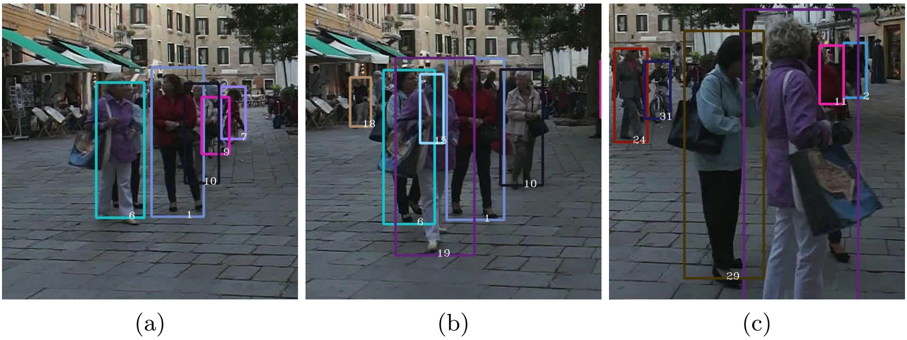


Fig. 6: Examples of tracking under multiple occlusions at frames 22, 97 and 261 (crops) in Venice-1/MOT16-01 using EA-PHD-PF(Priv). (a) Target 6 is correctly tracked while it occludes another target. (b) The occluded target becomes visible and trajectory 6 drifts towards it. Target 6 is reinitialized as target 19 at frame 22. (c) Intermittent detections cause target 6 to be reinitialized as target 29 at frame 97.

in crowded scenes, as shown in Fig. 6. The bottom part of Table 1 (MOT16) shows that EA-PHD-PF(Priv) outperforms AMPL, olCF and OVBT in terms of MOTA, FN and FP. However, the number of IDS is higher than AMPL and olCF, because the features they use are better able to discriminate targets. The results using public detections rank our tracker EA-PHD-PF(Pub) at half-rank overall as it generates a high amount of FN.

Fig. 7 compares sample tracking results using public (first row) and private (second row) detections. We can observe along the first row how the target firstly initialized as 115 is then reinitialized, lost and reinitialized again due to the high number of FN in the public dataset. However, the (same) target firstly initialized as 41 in the second row is correctly tracked along the whole sequence. We can observe the presence of false-positive trajectories (i.e. green, purple and red targets in Fig. 7b). These false-positive trajectories are difficult to remove because they are caused by persistent false-positive detections appearing for a few consecutive frames and the confidence scores of those detections are as high as those of true positive detections. With EA-PHD-PF(Priv) these detections are filtered out without adding any false-negative trajectories.

7 Conclusion

We presented an online multi-target tracker that exploits strong and weak detections in a Probability Hypothesis Density Particle Filter framework. Strong detections are used for trajectory initialization and tracking. Weak detections are used for tracking existing targets only to reduce the number of false negatives without increasing the false positives. Moreover, we presented a method to

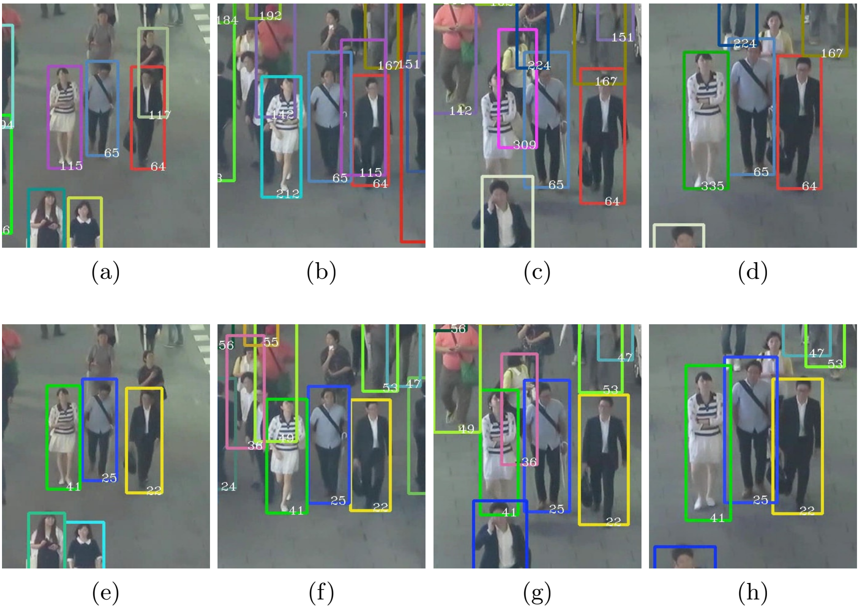


Fig. 7: Examples of tracking at frames 240, 461, 645 and 717 (crops) in MOT16-03 using public (first row) and private (second row) detections. (a)-(d) The target identified as 115 is reinitialized multiple times due to occlusions and lack of detections. (e)-(h) The (same) target, identified as 41, is correctly tracked.

perform early association between trajectories and detections, which eliminates the need for a clustering step for labeling. Finally, we exploited perspective information in prediction, update and newborn particle generation. Results show that our method outperforms alternative online trackers on the Multiple Object Tracking 2016 and 2015 benchmark datasets in terms tracking accuracy, false negatives and speed. The tracker works at an average speed of 12 fps. Future work will involve using appearance features, such as color histograms, to reduce trajectory fragmentation.

Acknowledgments

This work was supported in part by the ARTEMIS JU and the UK Technology Strategy Board (Innovate UK) through the COPCAMS Project, under Grant 332913.

References

1. Solera, F., Calderara, S., Cucchiara, R.: Learning to divide and conquer for online multi-target tracking. In: Proc. of International Conference on Computer Vision, Santiago, CL (December 2015)
2. Wang, B., Wang, G., Chan, K., Wang, L.: Tracklet association with online target-specific metric learning. In: Proc. of Computer Vision and Pattern Recognition, Columbus, OH, USA (June 2014)
3. Poiesi, F., Cavallaro, A.: Tracking multiple high-density homogeneous targets. IEEE Trans. on Circuits and Systems for Video Technology **25**(4) (April 2015) 623–637
4. Possegger, H., Mauthner, T., Roth, P., Bischof, H.: Occlusion geodesics for online multi-object tracking. In: Proc. of Computer Vision and Pattern Recognition, Columbus, OH, USA (June 2014)
5. Shu, G., Dehghan, A., Oreifej, O., Hand, E., Shah, M.: Part-based multiple-person tracking with partial occlusion handling. In: Proc. of Computer Vision and Pattern Recognition, Rhode Island, USA (June 2012)
6. Mahler, R.: A theoretical foundation for the Stein-Winter Probability Hypothesis Density (PHD) multitarget tracking approach. In: Proc. of MSS National Symposium on Sensor and Data Fusion, San Diego, CA, USA (June 2002)
7. Maggio, E., Taj, M., Cavallaro, A.: Efficient multitarget visual tracking using random finite sets. IEEE Trans. on Circuits and Systems for Video Technology **18**(8) (August 2008) 1016–1027
8. Vo, B.N., Singh, S., Doucet, A.: Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. In: Proc. of Information Fusion. Volume 2., Queensland, AU (July 2003)
9. Mahler, R.: PHD filters of higher order in target number. IEEE Trans. on Aerospace and Electronic Systems **43**(4) (October 2007) 1523–1543
10. Panta, K., Vo, B., Singh, S.: Improved probability hypothesis density (PHD) filter for multitarget tracking. In: 2005 3rd International Conference on Intelligent Sensing and Information Processing. (December 2005)
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of Computer Vision and Pattern Recognition, San Diego, CA, USA (June 2005)
12. Felzenszwalb, P.F., Girshick, R., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. on Pattern Analysis and Machine Intelligence **32**(9) (September 2010) 1627–1645
13. Smedt, F.D., Goedeme, T.: Open framework for combinated pedestrian detection. In: Proc. of Computer Vision, Imaging and Computer Graphics Theory and Applications, Berlin, GE (March 2015)
14. Lin, L., Bar-Shalom, Y., Kirubarajan, T.: Data association combined with the Probability Hypothesis Density Filter for multitarget tracking. In: Proc. SPIE. (August 2004)
15. Panta, K., Vo, B.N., Singh, S., Doucet, A.: Probability Hypothesis Density filter versus multiple hypothesis tracking. In: Proc. of SPIE. (August 2004)
16. Kuhn, H., Yaw, B.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2** (1955) 83–97
17. Poiesi, F., Mazzon, R., Cavallaro, A.: Multi-target tracking on confidence maps: An application to people tracking. Computer Vision Image Understanding **117**(10) (October 2013) 1257–1272

18. Maggio, E., Cavallaro, A.: Learning scene context for multiple object tracking. IEEE Trans. on Image Processing **18**(8) (August 2009) 1873–1884
19. Vo, B.N., Singh, S., Doucet, A.: Sequential Monte Carlo methods for multitarget filtering with random finite sets. IEEE Trans. on Aerospace and Electronic Systems **41**(4) (October 2005) 1224–1245
20. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. on Signal Processing **50**(2) (August 2002) 174–188
21. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 [cs] (April 2015)
22. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv:1603.00831 [cs] (March 2016)
23. Yang, F., Choi, W., Lin, Y.: Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers. In: Proc. of Computer Vision and Pattern Recognition, Las Vegas, NV, USA (June 2016)
24. Dollar, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. on Pattern Analysis and Machine Intelligence **36**(8) (August 2014) 1532–1545
25. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. IEEE Trans. on Pattern Analysis and Machine Intelligence **31**(2) (February 2009) 319–336
26. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: Proc. of Computer Vision and Pattern Recognition, Miami, FL, USA (June 2009)
27. Xiang, Y., Alahi, A., Savarese, S.: Learning to Track: Online Multi-Object Tracking by Decision Making. In: Proc. of International Conference on Computer Vision, Santiago, CL (December 2015)