# Efficient multi-target visual tracking using Random Finite Sets

Emilio Maggio, Murtaza Taj, Andrea Cavallaro

***Abstract*—We propose a filtering framework for multi-target tracking that is based on the Probability Hypothesis Density (PHD) filter and data association using graph matching. This framework can be combined with any object detectors that generate positional and dimensional information of objects of interest. The PHD filter compensates for missing detections and removes noise and clutter. Moreover, this filter reduces the growth in complexity with the number of targets from exponential to linear by propagating the first-order moment of the multi-target posterior, instead of the full posterior. In order to account for the nature of the PHD propagation, we propose a novel particle resampling strategy and we adapt the dynamic and observation models to cope with varying object scales. The proposed resampling strategy allows us to use the PHD filter when a priori knowledge of the scene is not available. Moreover, the dynamic and observation models are not limited to the PHD filter and can be applied to any Bayesian tracker that can handle State Dependent Variances (SDV). Extensive experimental results on a large standard video surveillance dataset using a standard evaluation protocol show that the proposed filtering framework improves the accuracy of the tracker, especially in cluttered scenes.**

***Index Terms*—Video surveillance, clutter, tracking, multi-target, PHD filter, Monte Carlo methods.**

## I. Introduction

The growth of adoption of video surveillance systems has been recently driven by hardware advances, such as camera miniaturization, digitization and increased availability of low-cost data storage. However, the opportunities offered by automated video surveillance are not yet exploited due to the lack of accurate and efficient algorithms for data-mining, content retrieval, event detection and behavior analysis. The extraction of high-level information from surveillance video mainly relies on the analysis of lower level video data like objects and their trajectories, which are generated by multi-target trackers. While reliable tracking is possible under constrained conditions, the problem of tracking in a generic unconstrained scenario (for example in a dense scene with uncontrolled illumination) is still unsolved.

The multi-target visual tracking problem can be decomposed into two main tasks, namely the detection of the objects of interests in each frame and the association of unique identities to the detections over time. The major challenge in the estimation of the number of targets and their position is that the estimate is based on a set of uncertain observations (i.e., the detections). A target may fail to generate an observation when *occluded*, an additional observation may be generated by *clutter*, and observations from actual targets may be corrupted by *noise*, thus affecting the state estimator. A multiple object tracker must also account for target *interactions* and for the time-varying number of targets in the scene by modeling their *birth* (when a new target appears in the scene or is a spawn from another target, such as a person stepping out of a car) and their *death*. Although the complete modeling of the multi-target problem is possible, its computational cost inevitably grows exponentially with the number of targets.

### A. Prior work

Bayesian recursion is a popular approach to filter noisy observations in single-target tracking [1], [2], [3]. The Bayes filter first predicts the target state based on a dynamical model and then updates the resulting density using the newly available observation. Two algorithms implementing this recursion are the Kalman Filter [4] and the Particle Filter (PF) [5]. Multi-target tracking requires the extension of these algorithms to cope with target birth and target death, clutter and missing observations (Tab. I). Although the multi-target state can be seen as a concatenation of single-target states, each modeled as a random variable [6], Bayes multi-target filtering is computationally intensive due to the increase of the state dimensionality with the number of targets. To alleviate this problem several approaches have been proposed, as described below.

One solution is to model the multi-target problem in the single-target state by propagating a mixture of single-target *pdf*s approximated by particles [7]. When a target appears in the scene, a new component of the mixture is initialized and then propagated independently. The birth event is governed by heuristics and it is not included in the filtering framework. The volume of the multi-target state sampled by PF can be reduced by assuming that the targets do not appear simultaneously and by modeling the birth as a Poisson process [8]. To reduce the computational cost, Markov Chain Monte Carlo methods can be used to better sample the multi-target density [9].

Although the above-mentioned approaches make the multi-target problem tractable, they do not account for clutter and missing observations. An attempt to alleviate these limitations is presented in [10], but in this case the number of visible targets is assumed to be known and fixed. Jump Markov Systems (JMS) approximated by PF have also been used to model the varying number of targets in the scene, clutter and missing detections [11], [12]. A JMS models the dependencies

E. Maggio, M. Taj and A. Cavallaro are with the Multimedia and Vision Group - Queen Mary, University of London, United Kingdom, E1 4NS, UK e-mail: {emilio.maggio, murtaza.taj, andrea.cavallaro}@elec.qmul.ac.uk. The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1.

in the multi-target state evolution thus allowing the design of an efficient importance sampling function for PF. A similar path is followed in [13] where the marginal association *pdf*s of the Joint Probability Data Association Filter (JPDAF) are sampled using PF. The approach is less complex than sampling the full multi-target state, as filtering is applied to independent association hypotheses pruned by a gating procedure. Recently, Rao-Blackwellization (RB) has been used to reduce the computational cost [14]. The RB multi-target filter integrates the state propagation in closed form, while Monte Carlo integration is used for data association. The data association problem can also be modeled using graph theory [16]. The graph structure accounts for target birth, death and missing detections, but a pre-filtering step is necessary to remove spatial noise and clutter.

A general Bayesian framework for multi-target tracking makes use of Finite Set Statistics (FISS) [17]. This framework considers the multi-target state as a single meta-target and the observations as a single set of measurements of the meta-sensor [15]. In this case, the multi-target state can be represented by a Random Finite Set (RFS), whose Bayesian propagation is similar to that of the single-target case. However, the dimensionality of the target state still grows with the number of targets. This means that the approximation of the RFS with Monte Carlo sampling requires a number of samples that grows exponentially, thus making the propagation of the full posterior impractical. A less computationally intensive alternative is to propagate the Probability Hypothesis Density (PHD) (i.e., the first-order moment of the multi-target posterior) [17]. The integrals of the PHD recursion can either have an exact solution by assuming the PHD to be a mixture of Gaussians (GM-PHD) [18] or can be approximated with the samples generated by a Sequential Monte Carlo (SMC) method (Particle-PHD) [15]. As the dimensionality of the PHD is that of the single-target state, efficient sampling requires a number of particles that is proportional to the expected number of targets, thus leading to linear complexity.

The cost for the lower complexity is the lack of information on the identity of the targets. For Particle-PHD a clustering step is necessary to associate the peaks of the PHD with target identities [19], [20]. Data association for the GM-PHD is easier as the identity can be associated directly with each Gaussian [21], [18]. However, these methods are limited by the linearity and Gaussianity assumptions on the transition and measurement models. Recently, Jump Markov Models have been used to extend GM-PHD to maneuvering targets [22], [23]. Filtering techniques based on the Particle PHD have been tested on synthetic data [15], [24], 3D sonar data [25], feature point filtering [26], and groups-of-humans detection [27]. However, as no data association is performed [15], [24], [26], [27] nor the target size is estimated [25], none of the above approaches can be applied to multi-target visual tracking.

TABLE I

MODELING CAPABILITIES OF MULTI-TARGET TRACKING ALGORITHMS. MD: MISSING DETECTIONS; PF: PARTICLE FILTER; MCMC: MARKOV CHAIN MONTE CARLO; JMS: JUMP MARKOV SYSTEMS; JPDAF: JOINT PROBABILISTIC DATA ASSOCIATION FILTER

| **Ref.** | **Algorithm** | **Modeling capabilities** | | |
|---|---|---|---|---|
| | | **Birth** | **Clutter** | **MD** |
| [7] | PF mixture | Heuristic | No | No |
| [8] | Multi-target Condensation | One at a time | No | No |
| [9] | Multi-target MCMC-PF | No | No | No |
| [10] | Multi-target Condensation | No | Yes | Yes |
| [11] | JMS and PF | One at a time | Yes | Yes |
| [12] | JMS and PF | One spawn | No | No |
| [13] | JPDAF and PF | Yes | Yes | Yes |
| [14] | Rao-Blackwellized-PF | Yes | Yes | Yes |
| [15] | Particle PHD filter | Yes | Yes | Yes |
| [16] | Graph matching | Yes | No | Yes |

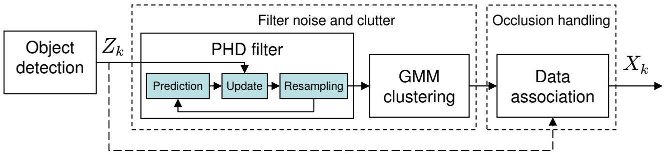


Fig. 1. Multiple target tracking scheme based on object detection and on Particle PHD filtering. The PHD filter removes spatio-temporal noise from the observations before the tracker performs data association.

## B. Contribution

In this paper we propose a complete multi-target visual tracking framework based on the PHD filter that addresses the problems of clutter, spatial noise and missing detections. Our main contribution is the adaptation of a filter based on Random Finite Sets to real–world visual tracking scenarios. These adaptations are not straightforward as, unlike conventional applications of the PHD filter, we have to account for non–punctual observations like those produced by video object detectors. Compared to our preliminary work in [28], we present here a novel resampling strategy, enhanced dynamic and observation models, and an evaluation on a larger dataset. Unlike the single-target particle filter, the multi-target PHD filter generates particles with two different purposes: (i) to propagate the state of existing targets and (ii) to model the birth of new ones. The proposed multi-stage resampling strategy accounts for the different nature of the particles and, compared to the multinomial strategy used in [28], improves the quality of the Monte Carlo estimation from a tracking perspective. As for the dynamic and observation models, we use State Dependent Variances (SDV) to account for the size of the targets. These models are not limited to the PHD recursion and can be implemented in any Bayesian recursive algorithms that can handle SDV.

We incorporate the PHD filter in an end-to-end flexible tracking framework that can deal with any detectors that generate a set of observations representing the position and the size of the targets. First, clutter and spatial noise are filtered by the particle PHD filter. Next, clustering is used on the samples of the PHD to detect filtered target positions. Finally, the cluster centers are processed by a data association algorithm based on the maximum path cover of a bi-partitioned graph. Figure 1 shows the block diagram of the proposed tracking framework. We demonstrate the multi-target framework using two different detectors, one based on background subtraction [29] and one based on Adaboost classifiers [30], and we objectively evaluate the results on a large outdoor surveillance dataset containing more than 1 hour and 20 minutes of annotated surveillance videos (the CLEAR-2007 dataset).

The paper is organized as follows. Section II describes the Particle PHD filter with the dynamic model, the resampling strategy and the particle clustering. Section III describes the data association method. In Sec. IV we show the results on surveillance and face tracking scenarios. In Sec. V we draw conclusions.

## II. Filtering detections with the Particle PHD

Let us approximate the target area in the image plane with a $w \times h$ rectangle centered at $\left(y^{(1)}, y^{(2)}\right)$. Let the single target state at time $k$ be $x_k = \left(y^{(1)}_{x_k}, \dot{y}^{(1)}_{x_k}, y^{(2)}_{x_k}, \dot{y}^{(2)}_{x_k}, w_{x_k}, h_{x_k}\right) \in E_s$, where $\dot{y}^{(1)}_{x_k}$ and $\dot{y}^{(2)}_{x_k}$ are the speed components of the target and $E_s$ is the state space. Finally, let the single-target observation $z_k = \left(y^{(1)}_{z_k}, y^{(2)}_{z_k}, w_{z_k}, h_{z_k}\right) \in E_o$ in the observation space $E_o$ be a rectangle generated by an object detector (e.g., a change detector or a face detector).

### A. Single-target recursive Bayes filtering

The single-target tracking problem can be modeled using the state and the measurement equations [5]

$$x_k = \mathbf{f}_k(x_{k-1}, n_k), \tag{1}$$

and

$$z_k = \mathbf{g}_k(x_k, v_k), \tag{2}$$

where $\mathbf{f}_k$ and $\mathbf{g}_k$ are non-linear, time-varying functions; and $\{n_k\}_{k=1,...}$ and $\{v_k\}_{k=1,...}$ are assumed to be independent and identically distributed stochastic processes. The goal of tracking is to estimate $p_{k|k}(x_k|z_{1:k})$, the *pdf* of the object being in state $x_k$, given all the observations $z_k$ up to time $k$, based on (1) and on (2). The estimation is performed recursively in two steps, namely prediction and update. The *prediction step* uses the dynamic model defined in (1) to obtain the prior *pdf* as

$$\begin{aligned} &p_{k|k-1}(x_k|z_{1:k-1}) \\ &= \int f_{k|k-1}(x_k|x_{k-1}) p_{k-1|k-1}(x_{k-1}|z_{1:k-1}) dx_{k-1}, \end{aligned} \tag{3}$$

with $p_{k-1|k-1}(x_{k-1}|z_{1:k-1})$ known from the previous iteration and the transition density $f_{k|k-1}(x_k|x_{k-1})$ determined by (1). The *update step* uses the Bayes' rule once the observation $z_k$ is available, so that

$$p_{k|k}(x_k|z_{1:k}) = \frac{g_k(z_k|x_k) p_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|x_k) p_{k|k-1}(x_k|z_{1:k-1}) dx_k}, \tag{4}$$

where $g_k(z_k|x_k)$ is determined by (2). When (1) and (2) are linear and the stochastic processes are Gaussian, the recursion has a closed form solution known as Kalman filter [4]. A more generic approximation can be obtained using Monte Carlo estimation [5]. In this case the densities $p_{k|k}(x_k|z_{1:k})$ are approximated with a sum of $L$ Dirac $\delta$ functions (the particles) centered in $\left\{x^{(i)}_k\right\}^L_{i=1}$ as

$$p_{k|k}(x_k|z_{1:k}) \approx \sum_{i=1}^{L} \omega^{(i)}_k \delta\left(x_k - x^{(i)}_k\right), \tag{5}$$

where $\left\{\omega^{(i)}_k\right\}^L_{i=1}$ are the weights associated with the particles and are defined as

$$\omega^{(i)}_k \propto \frac{p_{k|k}(x^{(i)}_k|z_{1:k})}{q_k(x^{(i)}_k|z_{1:k})} \quad i = 1, \ldots, L. \tag{6}$$

$q_k(.)$ is the importance density function defined as the density that generated the current set of particles.

Let us assume that $p_{k-1|k-1}(x_{k-1}|z_{1:k-1})$ is approximated by the set of particles and associated weights $\left\{\omega^{(i)}_{k-1}, x^{(i)}_{k-1}\right\}^L_{i=1}$, as in (5). By substituting this approximation in (3) and by applying importance sampling and (4), we obtain a recursion to propagate the particles and their weights [5]. The filters, based on Monte Carlo sampling and recursive Bayes equations, are known as Particle Filters.

### B. Multi-target recursive Bayes filtering with RFS

In order to extend the single-target Bayes framework to multiple targets, let us define the multi-target state, $X_k$, and the multi-target state measurement, $Z_k$, as the finite collection of the states and observations of each target. If $M(k)$ is the number of targets in the scene at time $k$, then the multi-target state, $X_k$, is the set

$$X_k = \left\{x_{k,1}, ...x_{k,M(k)}\right\} \in \mathcal{F}(E_s). \tag{7}$$

The multi-target measurement, $Z_k$, is the set

$$Z_k = \left\{z_{k,1}, ...z_{k,N(k)}\right\} \in \mathcal{F}(E_o) \tag{8}$$

and is formed by the $N(k)$ observations. Note that some of these observations may be due to clutter. $\mathcal{F}(E)$ is the collection of all the finite subsets of $E$.

The uncertainty in the state and measurement is introduced by modeling the multi-target state and the multi-target measurement using two Random Finite Sets (RFS). Let $\Xi_k$ be the RFS associated with the multi-target state:

$$\Xi_k = S_k\left(X_{k-1}\right) \cup B_k\left(X_{k-1}\right) \cup \Gamma_k, \tag{9}$$

where $S_k\left(X_{k-1}\right)$ denotes the RFS of survived targets, while $B_k\left(X_{k-1}\right)$ is the RFS of targets spawned from the previous set of targets $X_{k-1}$, and $\Gamma_k$ is the RFS of the new-born targets [15]. The RFS $\Omega_k$ associated with the measurement is defined as

$$\Omega_k = \Theta_k\left(X_k\right) \cup K_k, \tag{10}$$

where $\Theta_k\left(X_k\right)$ is the RFS modeling the measurements generated by the targets $X_k$, and $K_k$ models clutter and false alarms.

Similarly to the single-target case, the dynamics of $\Xi_k$ are described by the multi-target transition density $f_{k|k-1}(X_k|X_{k-1})$, while $\Omega_k$ is described by the multi-target likelihood $g_k(Z_k|X_k)$. The recursive equations equivalent to (3) and (4) are

$$\begin{aligned} &p_{k|k-1}(X_k|Z_{1:k-1}) = \\ &\int f_{k|k-1}(X_k|X_{k-1}) p_{k-1|k-1}(X_{k-1}|Z_{1:k-1}) \mu(dX_{k-1}) \end{aligned} \tag{11}$$

and

$$p_{k|k}(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X_k)p_{k|k-1}(X_k|Z_{1:k-1})\mu(dX_k)}, \tag{12}$$

where $\mu$ is an appropriate dominating measure on $\mathcal{F}(E_s)$ (for a detailed description of RFSs, set integral and formulations of $\mu$, please refer to [17] and [15]). Although a Monte Carlo approximation of this recursion is possible [15], the number of particles required is exponentially related to the number of targets in the scene. For this reason, an approximation is necessary to make the problem computationally tractable. To this extent Mahler proposes to propagate the first-order moment of the multi-target posterior instead of the posterior itself [17]. The resulting filter is known as the Probability Hypothesis Density (PHD) filter.

### C. The PHD filter

The PHD is a function in the single-target state space whose peaks identify the likely position of the targets. The PHD, $\mathcal{D}_\Xi(x)$, is the first-order moment of a RFS, $\Xi$, and it is a function on $E_s$. The property of the PHD is that for any region $R \subseteq E_s$

$$E[|\Xi \cap R|] = \int_R \mathcal{D}_\Xi(x)dx, \tag{13}$$

where $|.|$ is used to denote the cardinality of a set. In practice, (13) means that by integrating the PHD on any region $R$ of the state space we obtain the expected number of targets in $R$.

If we denote $\mathcal{D}_{k|k}(x)$ as the PHD at time $k$ associated with the multi-target posterior density $p_{k|k}(X_k|Z_{1:k})$, then the Bayesian iterative prediction and update of $\mathcal{D}_{k|k}(x)$ is known as the PHD filter. The recursion of the PHD filter is based on three assumptions: *(i)* the targets evolve and generate measurements independently; *(ii)* the clutter RFS, $K_k$, is Poisson-distributed and *(iii)* the predicted multi-target RFS is Poisson-distributed. While the first two assumptions are common to most Bayesian multi-target trackers ([6], [10], [11], [13], [14]), the third is specific to the derivation of the PHD update operator.

The *PHD prediction* is defined as

$$\mathcal{D}_{k|k-1}(x) = \int \phi_{k|k-1}(x,\zeta)\mathcal{D}_{k-1|k-1}(\zeta)d\zeta + \gamma_k(x), \tag{14}$$

where $\gamma_k(.)$ is the intensity function of the new target birth RFS (i.e., the integral of $\gamma_k(.)$ over a region $R$ gives the expected number of new objects per frame appearing in $R$). $\phi_{k|k-1}(x,\xi)$ is the analogue of the state transition probability in the single-target case:

$$\phi_{k|k-1}(x,\xi) = e_{k|k-1}(\xi)f_{k|k-1}(x|\xi) + \beta_{k|k-1}(x|\xi), \tag{15}$$

where $e_{k|k-1}(\xi)$ is the probability that the target still exists at time $k$, and $\beta_{k|k-1}(.|\xi)$ is the intensity of the RFS that a target is spawned from the state $\xi$.

The *PHD update* is defined as

$$\mathcal{D}_{k|k}(x) = \left[p_M(x) + \sum_{z\in Z_k} \frac{\psi_{k,z}(x)}{\kappa_k(z) + \langle\psi_{k,z}, \mathcal{D}_{k|k-1}\rangle}\right]\mathcal{D}_{k|k-1}(x), \tag{16}$$

where $p_M(x)$ is the missing detection probability; $\psi_{k,z}(x) = (1-p_M(x))g_k(z|x)$, and $g_k(z|x)$ is the single-target likelihood defining the probability that $z$ is generated by a target with state $x$; $\langle f,g\rangle = \int f(x)g(x)dx$, and $\kappa_k(.)$ is the clutter intensity.

No generic closed form solution exists for the integral of (14) and (16). Under the assumptions of Gaussianity and linearity one can obtain a filter that in principle is similar to the Kalman filter. This filter is known as the Gaussian Mixture PHD filter (GM-PHD) [21]. However, given the limitations on the dynamic and observation models (Sec. II-E), we prefer the Monte Carlo implementation of the PHD recursion, known as the Particle PHD filter.

### D. The Particle PHD filter

A numerical solution for the integrals in (14) and (16) is obtained using a Sequential Monte Carlo method that approximates the PHD with a (large) set of weighted random samples (see (5)). A more detailed explanation of the procedure is available in [15].

Given the set $\left\{\omega_{k-1}^{(i)}, x_{k-1}^{(i)}\right\}_{i=1}^{L_{k-1}}$ of $L_{k-1}$ particles and associated weights approximating the PHD at time $k-1$ as

$$\mathcal{D}_{k-1|k-1}(x) \approx \sum_{i=1}^{L_{k-1}} \omega_{k-1}^{(i)}\delta\left(x - x_{k-1}^{(i)}\right), \tag{17}$$

an approximation of the predicted PHD, $\mathcal{D}_{k|k-1}(x)$, with weighted particles $\left\{\tilde{\omega}_k^{(i)}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_{k-1}+J_k}$ is obtained by substituting (17) into (14) and then applying separately importance sampling to both terms on the r.h.s.. In practice, first we draw $L_{k-1}$ samples from the importance function $q_k(.|x_{k-1}^{(i)}, Z_k)$ to propagate the tracking hypotheses from the samples at time $k-1$; we then draw $J_k$ samples from the new-born importance function $p_k(.|Z_k)$ to model the state hypotheses of new targets appearing in the scene. This last set also defines the configuration of the particles at initialization. We will discuss the choice of $q_k(.|x_{k-1}^{(i)}, Z_k)$ and $p_k(.|Z_k)$ in Section II-E. The values of the weights $\tilde{\omega}_{k|k-1}^{(i)}$ are computed as

$$\tilde{\omega}_{k|k-1}^{(i)} = \begin{cases} \frac{\phi_k\left(\tilde{x}_k^{(i)}, x_{k-1}^{(i)}\right)\omega_{k-1}^{(i)}}{q_k\left(\tilde{x}_k^{(i)}|x_{k-1}^{(i)}, Z_k\right)} & i = 1, ..., L_{k-1} \\ \frac{\gamma_k(\tilde{x}_k^{(i)})}{J_k p_k\left(\tilde{x}_k^{(i)}|Z_k\right)} & i = L_{k-1}+1, \dots, L_{k-1}+J_k \end{cases}. \tag{18}$$

Once the new set of observations is available, by substituting the approximation of $\mathcal{D}_{k|k-1}(x)$ into (16), the weights $\left\{\tilde{\omega}_{k|k-1}^{(i)}\right\}_{i=1}^{L_{k-1}+J_k}$ are updated according to

$$\tilde{\omega}_k^{(i)} = \left[p_M(\tilde{x}_k^{(i)}) + \sum_{z\in Z_k} \frac{\psi_{k,z}(\tilde{x}_k^{(i)})}{\kappa_k(z) + C_k(z)}\right]\tilde{\omega}_{k|k-1}^{(i)}, \tag{19}$$

where $C_k(z) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{k,z}(\tilde{x}_k^{(i)})\tilde{\omega}_{k|k-1}^{(j)}$.

The Particle PHD filter was originally designed to track targets generating punctual observations (radar tracking [17]). To deal with targets from videos, we have to adapt the dynamic and the observation models to account for the size of the target on the image plane. In the following we describe how to account for the two additional dimensions, the width and the height of a target, in the observation.

### *E. Dynamic and observation models*

In order to compute the PHD filter recursion, the probabilistic model needs information regarding object dynamics and sensor noise. The information contained in the dynamic and observation models is used by the PHD filter to classify as clutter, detections not fitting these priors.

The magnitude of the motion of an object in the image plane depends on the distance of the object from the camera. Since acceleration and scale variations in the camera far-field are usually smaller than those in the near-field, we model the state transition $f_{k|k-1}(x_k|x_{k-1})$ as a first-order Gaussian dynamic with State Dependent Variances (SDV). This model assumes that each target has constant velocity between consecutive time steps and acceleration and scale changes approximated by random processes with standard deviations proportional to the object size at time $k-1$, i.e.

$$x_k = \overbrace{\begin{bmatrix} A & 0_2 & 0_2 \\ 0_2 & A & 0_2 \\ 0_2 & 0_2 & I_2 \end{bmatrix}}^{G} x_{k-1} + \begin{bmatrix} B_1 & 0_2 \\ B_2 & 0_2 \\ 0_2 & B_3 \end{bmatrix} \begin{bmatrix} n_k^{(1)} \\ n_k^{(2)} \\ n_k^{(w)} \\ n_k^{(h)} \end{bmatrix}, \tag{20}$$

with

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \; B_1 = w_{x_{k-1}} \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \end{bmatrix},$$

$$B_2 = h_{x_{k-1}} \begin{bmatrix} 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix}, \; \textit{and} \;\; B_3 = \begin{bmatrix} Tw_{x_{k-1}} & 0 \\ 0 & Th_{x_{k-1}} \end{bmatrix},$$

where $0_n$ and $I_n$ are the $n \times n$ zero and identity matrices, and $\{n_k^{(1)}\}$,$\{n_k^{(2)}\}$, $\{n_k^{(w)}\}$ and $\{n_k^{(h)}\}$ are independent white Gaussian noises with standard deviations $\sigma_{n^{(1)}}$,$\sigma_{n^{(2)}}$,$\sigma_{n^{(w)}}$ and $\sigma_{n^{(h)}}$, respectively. $\{n_k^{(1)}\}$ and $\{n_k^{(2)}\}$ model the acceleration of the target, while $\{n_k^{(w)}\}$ and $\{n_k^{(h)}\}$ model the variation in size. $T = 1$ is the interval between two consecutive steps ($k-1$ and $k$), which we take to be constant when the frame rate is constant. For simplicity, no spawning of targets is considered in the dynamic model.

The observation model is derived from the following considerations: when an object is partially detected (e.g., the body of a person is detected while her/his head is not detected), the magnitude of the error is dependent on the object size. Moreover, the error on the estimation of the target size is twice the error on the estimation of the centroid. This is equivalent to assuming that the amount of noise on the observations is proportional to the size of the targets, and that the standard deviation of the noise on the centroid is half that on the size. To this extent we define the single-target likelihood as a Gaussian SDV model, such that

$$g_k(z|x) = \mathcal{N}(z; Cx, \Sigma(x)), \tag{21}$$

where $\mathcal{N}(z; Cx, \Sigma(x))$ is a Gaussian function evaluated in $z$, centered in $Cx$ and with covariance matrix $\Sigma(x)$. $C$ is defined as

$$C = \begin{bmatrix} D & 0_{2\times 3} \\ 0_{2\times 4} & I_2 \end{bmatrix}, \quad \text{with } D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and $0_{n\times m}$ is the $n \times m$ zero matrix. $\Sigma(x)$ is a diagonal covariance matrix defined as

$$diag(\Sigma(x)) = \left[\frac{\sigma_{v^{(w)}}}{2} w_x, \frac{\sigma_{v^{(h)}}}{2} h_x, \sigma_{v^{(w)}} w_x, \sigma_{v^{(h)}} h_x\right].$$

Note that the SDV models described in (20) and (21) do not allow closed-form solution of the PHD filter recursive equations (see (14) and (16)). They require an algorithm such as the Particle PHD that can handle generalized state space models. In order to use GM-PHD [18] with SVD, an approximation based on the Extended Kalman filter or on the Unscented transformation is necessary [21]. The other functions that define the PHD recursion are defined below.

In the absence of any prior knowledge about the scene, we assume that the missing detection probability, $p_M(x)$, the probability of survival, $e_{k|k-1}(x)$, and the birth intensity $\gamma_k(x)$ are constant over $x$. To this extent, we decompose $\gamma_k(x)$ as $\bar{s}b(x)$, where $\bar{s}$ is the average birth events per frame and $b(x)$ is the probability density of a birth that we take to be uniform on the state space. Similarly, we define the clutter intensity $\kappa_k(z)$ as $\bar{r}c(z)$, and we assume the clutter density $c(z)$ to be uniform over the observation space.

In order to complete the definition of the Particle PHD filter recursion we need to design the importance sampling functions for the Monte Carlo approximation. On the one hand, $L_{k-1}$ old particles are propagated, as in CONDENSATION [1], according to the dynamics (i.e., $q_k(.|.) \propto f_{k|k-1}(.|.)$). On the other hand, drawing the $J_k$ new-born particles is not straightforward as the tracker should be able to reinitialize after an unexpected lost track or target occlusion. When prior knowledge on the scene is available, the samples could be drawn from a localized $\gamma_k(.)$. However, no target birth would be possible in state regions with low $\gamma_k(.)$, as no particles would be sampled in these areas. When no prior knowledge is available, drawing from a uniform non-informative $\gamma_k(.)$ (as in the one we use) would require too many particles to obtain a dense sampling on a 6D state space. To avoid this problem, we assume that the birth of a target happens in a limited volume around the measurements; thus we draw the $J_k$ new-born particles from a mixture of Gaussians centered on the components of the set $Z_k$. Hence, we define the importance sampling function for new-born targets $p_k(.|Z_k)$ as

$$p_k(x|Z_k) = \frac{1}{N(k)} \sum_{z \in Z_k} \mathcal{N}(x; [z, 0, 0], \Sigma_b(z)), \tag{22}$$

where the elements of the $6 \times 6$ diagonal covariance matrix $\Sigma_b$ are proportional to $w_z$ and $h_z$, and are defined as

$$\begin{aligned} diag(\Sigma_b(z)) \quad &= \left[\sigma_{b,y^{(1)}} w_z, \sigma_{b,\dot{y}^{(1)}} w_z, \sigma_{b,y^{(2)}} h_z, \ldots \right. \\ &\quad \left. \ldots, \sigma_{b,\dot{y}^{(2)}} h_z, \sigma_{b,w} w_z, \sigma_{b,h} h_z\right]. \end{aligned}$$

Although drawing new-born particles from (22) allows dense sampling around regions where a birth is possible, the Particle PHD recursion is also influenced by the resampling strategy used to select the most promising hypotheses. In the next section we discuss the resampling issues for the Particle PHD filter that accounts for the different nature of the particles.

## F. Resampling

At each iteration, $J_k$ new particles are added to the old $L_{k-1}$ particles. To limit the growth of the number of particles, a resampling step is performed after the update step. If classical multinomial resampling is applied ([15], [5]), then $L_k$ particles are resampled with probabilities proportional to their weights from $\left\{\tilde{\omega}_k^{(i)}/\hat{M}_{k|k}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_k+J_k}$, where $\hat{M}_{k|k}$ is the total mass. This resampling procedure gives greater chance to tracking hypotheses with higher likelihood to propagate by pruning from the set unlikely hypotheses.

$L_k$ is usually chosen to keep the number of particles per target, $\rho$, constant. At each time step, a new $L_k$ is computed so that $L_k = \rho\hat{M}_{k|k}$. Hence the computational cost of the algorithm grows *linearly* with the number of targets in the scene. After resampling, the weights of $\left\{\omega_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$ are normalized to preserve the total mass.

Although multinomial resampling is appropriate for a single-target Particle Filter, this strategy poses a series of problems when applied to the PHD filter. The prediction stage of the PHD (see (14)) generates two different sets of particles: (i) the $L_{k-1}$ particles propagated from the previous steps to model the state evolution of existing targets, with weights proportional to $\omega_{k-1}^{(i)}$, and (ii) the remaining $J_k$ particles modeling the birth of new targets, with weights proportional to the birth intensity $\gamma_k(.)$.

For multi-dimensional state spaces where the birth event is sparse (i.e., low $\gamma_k(.)$), the predicted weights $\tilde{\omega}_{k|k-1}^{(i)}$ of the new-born particles may be several orders of magnitude smaller than the weights of the propagated particles. In this case, as the probability of resampling is proportional to $\tilde{\omega}_k^{(i)}$ and thereby to $\tilde{\omega}_{k|k-1}^{(i)}$, it is possible that none of the new-born particles is resampled and propagated to the next step. Although the approximation of the PHD is still asymptotically correct, the birth of a new target also depends on combinatorial factors. Furthermore, when one or a few new-born particles are finally propagated, the PHD is not densely sampled around the new-born target, thus reducing the quality of the spatial filtering effect. Increasing the number of particles per target, $\rho$, is not effective as the value should be very large and comparable with $1/\gamma_k(.)$.

To overcome this problem, we construct a multi-stage pipeline that resamples the new-born particles independently from the others. The idea is to separately apply multinomial resampling to the new-born particles by segregating them for a fixed number $N_s$ of time steps. In this way we allow the weights to grow till they reach the same magnitude as those associated with particles modeling older targets. The proposed multi-stage multinomial resampling strategy for the particle PHD filter is summarized in Algorithm 1. Figure 2 shows an example of the multi-stage resampling pipeline when $N_s = 3$.

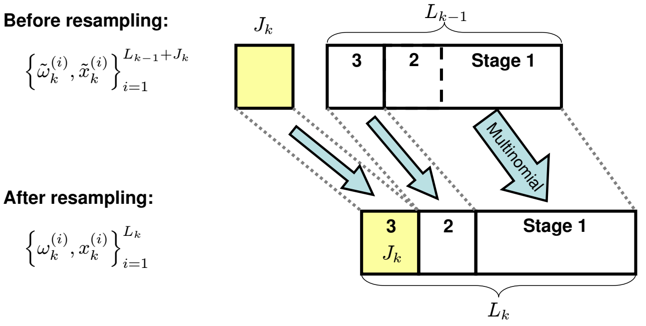

Before resampling:
$\left\{\tilde{\omega}_k^{(i)}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_{k-1}+J_k}$
$J_k$
$L_{k-1}$
3
2
Stage 1
Multinomial
After resampling:
$\left\{\omega_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$
3
$J_k$
2
Stage 1
$L_k$

Fig. 2. Schema of the multi-stage resampling strategy for the three-stages case. The $J_k$ particles modeling the birth of new targets are resampled separately from the older ones for a fixed number of time steps.

**Algorithm 1** Multi-stage multinomial resampling

```
    $\left\{\tilde{\omega}_k^{(i)}, \tilde{x}_k^{(i)}\right\}_{i=1}^{L_{k-1}+J_k} \rightarrow \left\{\omega_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$
1: if $k = 0$ then
2:     $S_i = 0 \quad \forall i = 1, \dots N_s$
3: else if $k \geq 1$ then
4:     $S_{N_s} = S_{N_s-1} + J_k$
5:     Compute the stage mass $\hat{M}_{S_1} = \sum_{i=1}^{S_1} \tilde{\omega}_k^i$
6:     Compute the number of particles $\tilde{S}_1 = \hat{M}_{S_1}\rho$
7:     Multinomially resample $\left\{\tilde{\omega}_k^{(i)}/\hat{M}_{S_1}, \tilde{x}_k^{(i)}\right\}_{i=1}^{S_1}$ to get $\left\{\omega_k^{(i)} = 1/\hat{M}_{S_1}, x_k^{(i)}\right\}_{i=1}^{S_1}$
8:     for $j = 2 : N_s$ do
9:         Compute the stage mass $\hat{M}_{S_j} = \sum_{i=S_{j-1}+1}^{S_j} \tilde{\omega}_k^i$
10:        Compute the number of particles $\tilde{S}_j = \tilde{S}_{j-1} + \max\{\hat{M}_{S_j}\rho,\ S_j - S_{j-1}\}$
11:        Multinomially resample $\left\{\tilde{\omega}_k^{(i)}/\hat{M}_{S_j}, \tilde{x}_k^{(i)}\right\}_{i=S_{j-1}+1}^{S_j}$ to get $\left\{\omega_k^{(i)} = 1/\hat{M}_{S_j}, x_k^{(i)}\right\}_{i=\tilde{S}_{j-1}+1}^{\tilde{S}_j}$
12:    end for
13:    $L_k = \tilde{S}_{N_s}$
14:    $S_1 = \tilde{S}_1 + \tilde{S}_2$
15:    $S_i = \tilde{S}_{i+1} \quad \forall i = 2, \dots N_s - 1$
16: end if
```

The multi-stage multinomial resampling preserves the total mass of whole set of particles $\hat{M}_{k|k}$ (this is a requirement of the PHD filter), as it preserves the total mass of the particles in each stage (see Step 7 and Step 11 of Algorithm 1). As we model proposal density $p_k\left(\tilde{x}_k^{(i)}|Z_k\right)$ of the new-born particles with a mixture of Gaussians centered on the observations (see (22)), we can take $J_k = N(k)\cdot\tau$, where $\tau$ is the number of new-born particles per observation. The overall computational cost of the algorithm grows linearly with the number of targets $X_k$, and linearly with the number of observations $Z_k$.

In order to compare the proposed resampling strategy with the standard multinomial resampling, we analyze the statistics of the delay in the response of the filter produced by the resulting Monte Carlo approximations. To ensure that the difference is generated only by the resampling, we produce a synthetic scenario where the targets move according to the model described in Sec. II-E. We fix one target in the center of the scene and then we generate new targets uniformly distributed over the state space and according to a Poisson process. The two components of the speed of the new targets are uniformly drawn over the ranges $[-4\sigma_{b,\dot{y}^{(1)}}w_z, 4\sigma_{b,\dot{y}^{(1)}}w_z]$ and $[-4\sigma_{b,\dot{y}^{(2)}}h_z, 4\sigma_{b,\dot{y}^{(2)}}h_z]$ respectively. This also produces targets in regions of the state space with low density of new-born particles (see (22)). We collect the measurements $Z_k$

TABLE II
COMPARISON OF FILTERING RESPONSE STATISTICS BETWEEN THE STANDARD MULTINOMIAL (MUL) RESAMPLING AND THE PROPOSED (PROP) MULTI-STAGE MULTINOMIAL RESAMPLING

| | **Delay** | | **Never detected %** | | | | |
|---|---|---|---|---|---|---|---|
| | Avg | Std dev | **0-.25** | 0-.05 | .05-.1 | .1-.15 | .15-.2 |
| Mul | 11.2 | 10.5 | 37.2 | 23.8 | 29.4 | 45.6 | 50.0 |
| Prop | 5.1 | 3.8 | 14.8 | 9.9 | 8.7 | 16.2 | 27.5 |

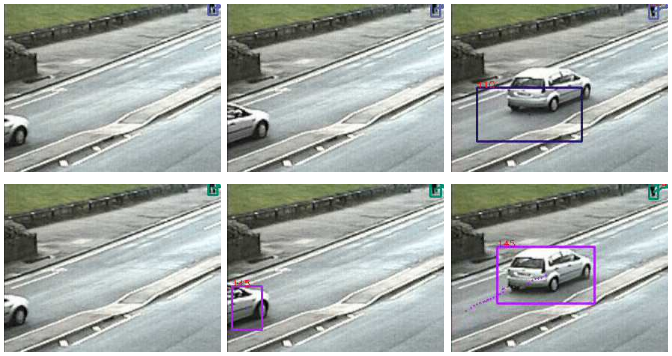


Fig. 3. Sample tracking results using multinomial and multi-stage multinomial resampling (CLEAR-2007 dataset, sequence 102a03, frames 1354, 1359 and 1385). The multinomial resampling (top row) delays the initialization of the track and introduces an error in the state estimation due to the low number of available samples. These behaviors are corrected by the proposed multi-stage resampling strategy (bottom row).

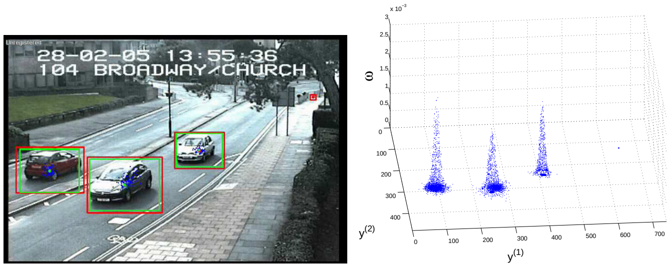


Fig. 4. Visualization of the particles approximating the PHD (before resampling step) on the frame at the left when the vehicles are the targets (red boxes: input detections; green boxes: cluster centers).

for 1000 synthetic targets. We then give the measurements as input to the approximated PHD recursions using the two resampling strategies. Table II shows the statistics related to the time delay in validating the new-born targets (expressed in frames), and the percentage of never–detected targets with respect to the speed ranges expressed as ratios between speed and object size. Higher ratios are associated with regions of the state space where filtering is more difficult as the density of sampled particles (see (22)) is lower. Also, faster targets are more likely to leave the scene before the PHD filter manages to produce a target birth. The standard deviation of the filtering delay (Tab. II) shows that the multi-stage resampling strategy has a beneficial effect in stabilizing the behavior of the filter (lower standard deviation). The higher average delay produced by multinomial resampling is due to those situations where none of the new-born particles is propagated to the next time-step. This is also confirmed by the higher percentage of never–detected targets produced by multinomial resampling.

A comparison between the proposed resampling strategy and the standard multinomial resampling is shown in Fig. 3. The top row shows a delayed target birth (box) caused by the standard multinomial resampling. In this situation, dense sampling is made more difficult by the fast motion of the vehicle. Note that 30 frames of consecutive coherent detections are not enough to validate the target. Furthermore, when the first particles are resampled and propagated, the filtering result is poor due to the low number of samples available. Figure 3, bottom row, shows how the proposed resampling strategy improves the quality of the PHD approximation when new targets appear in the scene. The proposed multi-stage multinomial resampling that uses the same birth intensity validates the track in 4 frames only, despite the motion of the target.

### G. Particle clustering

After the resampling step, the PHD is represented by a set of particles, $\left\{\omega_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$, defined in the single-target state space. An example of PHD approximated by particles is shown in Fig. 4. The peaks of the PHD are on the detected vehicles and the mass $\hat{M}_{k|k} \approx 3$ estimates the number of targets. The local mass of the particles is larger where the tracking hypotheses are validated by consecutive detections. Note that although the set of particles carries information about the expected number of targets and their location in the scene, the PHD does not hold information about the identity of the targets. A clustering algorithm is required to detect the peaks of the PHD. These peaks define the set of candidate states $\bar{X}_k = \left\{\bar{x}_{k,1}, ...\bar{x}_{k,\bar{M}(k)}\right\}$ of the targets in the scene and are the input of the data association algorithm. The information carried on by $\bar{X}_k$ is richer than the one carried on by the original set of detections $Z_k$, as the elements of $\bar{X}_k$ are filtered in space and time by the PHD and include an estimate of the target velocity. Further information is also carried on by the total mass $\hat{M}_{k|k}$ estimating the expected number of targets in the scene. However, $\hat{M}_{k|k}$ may be composed of several clusters of particles with mass smaller than 1 and therefore the real number of clusters may be larger than $\hat{M}_{k|k}$.

To avoid underestimating the number of clusters, we propose a top-down procedure based on Gaussian Mixture Models (GMM) that accounts for the new set of particles associated with target births, and updates the cluster parameters by means of Expectation Maximization (EM). The intuitive reason for using GMM is that both state dynamics and observation models are Gaussian, and therefore also the clusters of particles tend to be Gaussian-distributed. The procedure works as follows: first, the set of clusters obtained at the previous step, $k-1$, is augmented with new clusters initialized on the observations to model candidate new-born targets. Next, a hypothesis test is conducted to discard the new clusters that are similar to old ones. The parameters of the remaining clusters are optimized running EM on a set of $\check{L}_k = \rho_{GM}\hat{M}_{k|k}$ particles multinomially resampled from $\left\{\omega_k^{(i)}/\hat{M}_{k|k}, x_k^{(i)}\right\}_{i=1}^{L_k}$ [5]. Resampling is performed to obtain particles with uniform weights and also to reduce the computational cost of the EM recursion. After convergence we discard small clusters (with

**Algorithm 2** Particle clustering

```
$\left\{\theta_{k-1}, Z_k, \left\{\omega_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}\right\} \rightarrow \left\{\theta_k, \bar{X}_k\right\}$
1: Multinomially resample $\left\{\omega_k^{(i)}/\hat{M}_{k|k}, x_k^{(i)}\right\}_{i=1}^{L_k}$ to get $\left\{\check{\omega}_k^{(i)} = 1/\hat{M}_{k|k}, \check{x}_k^{(i)} =\right\}_{i=1}^{\check{L}_k}$
2: $\forall z \in Z_k$ initialize a new cluster $\{1/N_k, [z, 0, 0], \Sigma_b(z)\}$ and add it to $\theta_N$
3: $\forall$ clusters $c_j = \{\bar{x}_j, \Sigma_j\} \in \theta_N$ compute hypothesis test that $\bar{x}_{k-1,i} \in \theta_{k-1}$ is in the 99 percentile of $c_j$, and remove $c_j$ from $\theta_N$ if $\exists i|$ the test is positive
4: Add the clusters to $\theta_{k-1}$ and, to obtain $\tilde{\theta}_k$, run EM till convergence on the particles obtained at step 1
5: Prune from $\tilde{\theta}_k$ the small clusters with $\tilde{\pi}_{k,j} < S$
6: Merge similar clusters (with the procedure defined in [21]) thus obtaining $\theta_k$
7: Create the set of cluster centers $\bar{X}_k$ according to $\bar{X}_k = \{\bar{x}_{k,i}, i = 1, \ldots N_{c,k}|\pi_{k,i}\hat{M}_{k|k} < T_M\}$
```

mass below a threshold $S$) as they are usually associated with disappeared targets. Finally, we merge similar clusters according to a criterion based on hypothesis testing [21].

Let us define the parameters of the GMM at time $k$ as

$$\theta_k = \{\pi_{k,1}, \bar{x}_{k,1}, \Sigma_{k,1}, \ldots, \pi_{k,N_{c,k}}, \bar{x}_{k,N_{c,k}}, \Sigma_{k,N_{c,k}}\},$$

where $\pi_{k,i}$ is a weight coefficient of the mixture, $\bar{x}_{k,i}$ is the cluster center, $\Sigma_{k,i}$ is the covariance matrix and $N_{c,k}$ is the number of clusters at time $k$. Given the cluster parameters $\theta_{k-1}$ at $k-1$, the observation $Z_k$ and the set of particles $\left\{\omega_k^{(i)}, x_k^{(i)}\right\}_{i=1}^{L_k}$, the clustering procedure that outputs the new set of clusters $\theta_k$ and the set of states $\bar{X}_k$ is detailed in Algorithm 2.

## III. Data association

To obtain a consistent identity of each target over time we use an optimized data association procedure based on graphs [16]. Although this algorithm does not account for clutter and spatial noise, it is less computationally intensive than other probabilistic techniques (e.g., the Multi Hypotheses Tracker [31]) and produces comparable or better results [16]. This choice is motivated by the fact that we do not need to handle clutter measurements and sensor noise at this stage as they have been already treated by the PHD filter.

Let a cluster center $\bar{x}_k \in \bar{X}_k$ be represented by a vertex $v(\bar{x}_k) \in V_k$ of the graph $G$, where $V_k$ is the set of vertices representing the targets at time $k$. The tentative associations between candidate targets at different instants of time are described by the gain associated with each edge in $G$. The graph is formed by iteratively creating new edges from the old set of vertices, $\{V_{k-j}\}_{j=1\ldots W}$, to the new set of vertices, $V_k$, associated with cluster centers of frame $k$. The possible combinations of set of edges represent multiple track hypotheses, which account also for possible missing detections and occlusions (i.e., edges between two vertices $v(\bar{x}_k)$ and $v(\bar{x}_{k-j})$, with $j > 1$). The final tracks are identified by the best set of edges generated by the path cover of $G$ with the maximum gain. We define the gain $g(\bar{x}_k, \bar{x}_{k-j})$ of the edge between $\bar{x}_k$ and $\bar{x}_{k-j}$ as

$$g(\bar{x}_k, \bar{x}_{k-j}) = \log(\mathcal{N}(\bar{x}_k; G\bar{x}_{k-j}, \Sigma_g(\bar{x}_{k-j}))P_m^{k-j-1}) - T_g, \tag{23}$$

where $P_m \leq 1$ penalizes shorter trajectories, $G$ is defined in (20), $\Sigma_g(x)$ is a diagonal matrix with

$$\begin{aligned} diag(\Sigma_g(x)) &= [\sigma_{g,y^{(1)}}w_x, \sigma_{g,\dot{y}^{(1)}}w_x, \sigma_{g,y^{(2)}}h_x, \ldots \\ &\quad \ldots, \sigma_{g,\dot{y}^{(2)}}h_x, \sigma_{g,w}w_x, \sigma_{g,h}h_x] \end{aligned}$$

and $T_g$ is a gating threshold defined by the 99 percentile of the Gaussian. An edge is added to $V_k$ if $g(\bar{x}_k, \bar{x}_{k-j}) > 0$. Estimating the maximum path cover (i.e., the maximum sum of edges given the tracking constraints) of the graph corresponds to maximizing the likelihood over the set of edges (i.e., correspondences) represented in the graph. To this end, we enforce a bi-partitioning of the graph and solve the maximization problem by means of the algorithm from Hopcroft and Karp [32]. The complexity of this algorithm is $O(n^{2.5})$, where $n$ is the number of vertices in $V_k$.

## IV. Experimental results

In this section we report on tests of the proposed multi-target tracking framework on real-world scenarios. In particular, we assess the contribution of the Particle PHD filter and of the dynamic and observation models with state-dependent variances on the tracking result. To test the flexibility of the proposed framework we use two different detectors, namely a change detector and a face detector.

*The parameters used in the simulations are the same for all test sequences* and, unless otherwise stated, they are the same for the two detectors. The values of the parameters are empirically chosen and a sensitivity analysis for these choices is given later in this section. The particle PHD filter uses $\rho = 2000$ particles per target and $\tau = 500$ particles per detection. The standard deviations of the dynamic model defining target acceleration and scale changes are: $\sigma_{n^{(1)}} = \sigma_{n^{(2)}} = \sigma_{n^{(w)}} = \sigma_{n^{(h)}} = 0.04$. The standard deviations of the Gaussian observation noise are: $\sigma_{v^{(w)}} = \sigma_{v^{(h)}} = 0.15$ for the change detector and $0.1$ for the face detector. Larger spatial noise is used in the change detector case as we have to cope with the errors related to merging and splitting of the blobs. The birth intensity parameter defining the number of new targets per frame is $\bar{s} = 0.005$. The number of observations due to clutter is set to $\bar{r} = 2.0$ clutter points per frame. The missing detection probability $P_M = 0.05$, and the survival probability $e_{k|k-1} = 0.995$. The new-born particles are spread around the detections with $\sigma_{b,y^{(1)}} = \sigma_{b,y^{(2)}} = \sigma_{b,w} = \sigma_{b,h} = 0.02$ and $\sigma_{b,\dot{y}^{(1)}} = \sigma_{b,\dot{y}^{(2)}} = 0.05$. The resampling strategy uses $N_s = 7$ stages. The number of resampled particles for GMM clustering is $\rho_{GM} = 500$ per target. Clusters with weight lower than $S = 10^{-3}$ are discarded, while $T_M = 0.5$ is used to accept the cluster centers as real targets. For data association, the depth of the graph is $W = 50$ and means that the algorithm is capable of resolving occlusions for a maximum of 2 seconds with a 25Hz frame rate. The parameters of the gain function of (23) are: $\sigma_{g,y^{(1)}} = \sigma_{g,y^{(2)}} = 0.075$, $\sigma_{g,\dot{y}^{(1)}} = \sigma_{g,\dot{y}^{(2)}} = 0.09$, $\sigma_{g,w} = \sigma_{g,h} = 0.15$, $P_m = 0.5$ for the change detector and $0.9$ for the face detector. The higher value of $P_m$ used in the

TABLE III
COMPARATIVE RESULTS ON SDV DYNAMIC AND OBSERVATION MODELS ON THE TWO TESTING SCENARIOS BW (BROADWAY CHURCH) AND QW (QUEENSWAY) FROM THE CLEAR-2007 DATASET.

| | | **B**W | | **Q**W | |
|---|---|---|---|---|---|
| | | **S**DV | **L**inear | **S**DV | **L**inear |
| **M**ODP | Avg | 0.537 | 0.530 | 0.382 | 0.377 |
| | Significance | 5.55E-09 | | 1.54E-02 | |
| **M**ODA | Avg | 0.444 | 0.429 | 0.211 | 0.153 |
| | Significance | 2.63E-04 | | 7.33E-06 | |
| **M**OTP | Avg | 0.544 | 0.536 | 0.388 | 0.381 |
| | Significance | 9.75E-08 | | 3.37E-03 | |
| **M**OTA | Avg | 0.436 | 0.415 | 0.194 | 0.128 |
| | Significance | 2.11E-06 | | 1.28E-06 | |

face detector lowers the penalty on edges modeling missing detections and occlusions. As we will see in the following this facilitates track continuity when a face is occluded by the other objects in the scene.

The main body of the tests is conducted on the CLEAR-2007 dataset using a *change detector*. The dataset contains $25$ sequences from two different surveillance scenarios, Broadway Church (BW) and Queensway (QW).The videos have a frame size of $720\times480$ pixels with a frame rate of 25Hz. The ground-truth annotation is available for $121354$ frames (approximately 1 hour and 21 minutes of video), divided into $50$ evaluation segments.

The detector used is a color statistical change detector [29], followed by morphological filtering and connected component analysis. To facilitate the reproducibility of the experiments, the files containing the detector output $Z_k$ are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/PHD-MT.html.

The objective performance evaluation follows the VACE-CLEAR protocol [33], which uses four scores, namely Multiple Object Detection Accuracy (MODA), Multiple Object Detection Precision (MODP), Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). Unless otherwise stated the MOTP and MOTA values for each scenario are the average over the evaluation segments weighted by the segment frame span.

Table III shows the performance comparison between the SDV dynamic and observation models described in Sec. II-E, and linear models with fixed variances. Fixing the variances is equivalent to removing from (20) and (21) all references to target width, $w$, and height, $h$. The fixed values of the standard deviations are chosen as a compromise between large and small targets ($\sigma_{n^{(1)}} = \sigma_{n^{(2)}} = \sigma_{n^{(w)}} = \sigma_{n^{(h)}} = 3$ and $\sigma_{v^{(w)}} = \sigma_{v^{(h)}} = 5$). The tracker with SDV models is better in terms of both precision and accuracy. Also, the significance of the performance difference is always below the $5\%$ validation threshold. The compromise selected for the standard deviation values is not appropriate near the extrema of the target scale range. When a large object (i.e., $200$ pixels wide) is partially detected, the error associated with the observation $z_k$ may be several times larger than the standard deviation. Similarly, while an acceleration of $3$ pixels per frame may be appropriate for a middle–size target, this value is large compared to the typical motion of a pedestrian located in the camera's far-field.

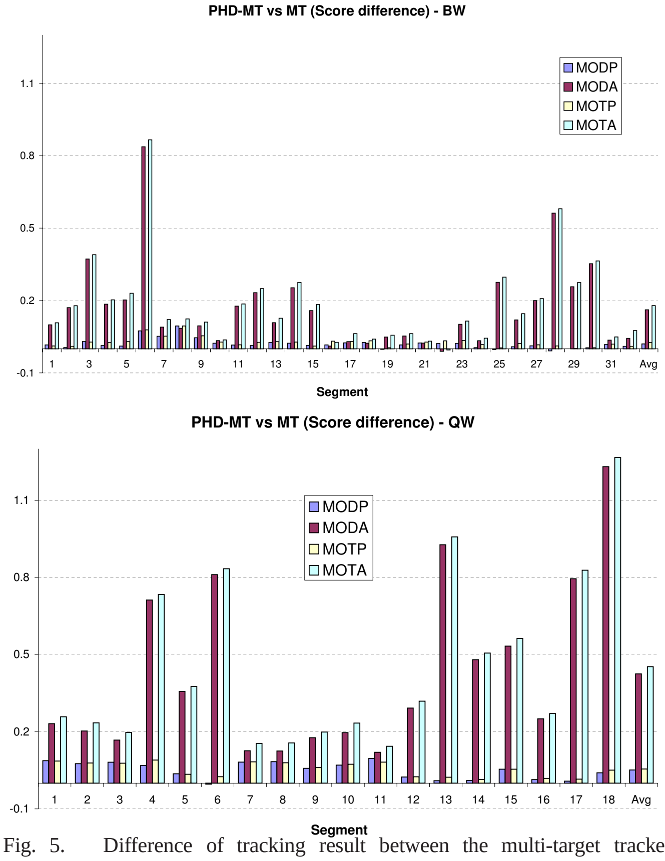
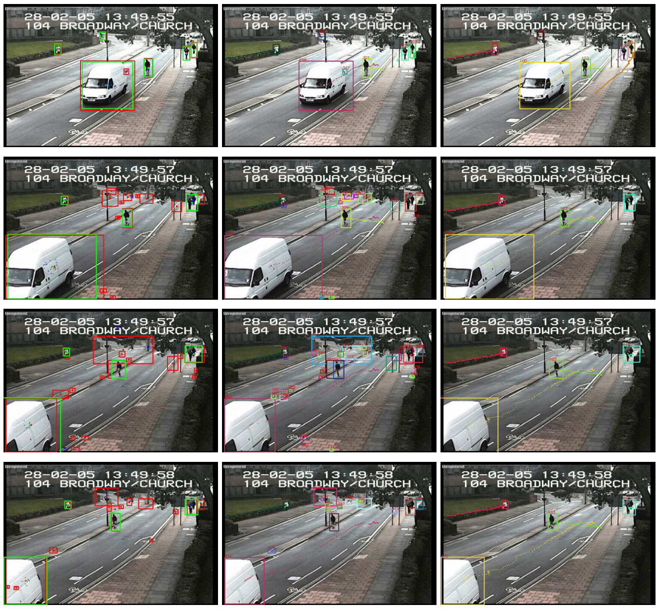


Fig. 5. Difference of tracking result between the multi-target tracker with (PHD-MT) and without (MT) PHD filter. The bar plots show the evaluation score difference for all the evaluation segments in the two scenarios of the CLEAR-2007 dataset. Top: BW (Broadway Church); bottom: QW (Queensway). The last set of four bars shows the average difference over the segments of each scenario. Positive values correspond to performance improvements achieved with the PHD filter.

To quantify the change in performance when adding the PHD filter to the tracking pipeline, we compare the multi-target tracker based on the particle PHD filter (PHD-MT) with the multi-target tracker (MT) where the data association described in Sec. III is performed directly on $Z_k$. Figure 5 shows the difference in terms of evaluation scores between PHD-MT and MT. The last set of bars in the two plots shows the average results over the segments. It is possible to notice that the filtering of clutter and noise consistently improves both accuracy and precision for all the evaluation segments in both scenarios. In the video segments with higher levels of clutter and where tracking is more challenging, the performance improvement is larger. Similar considerations can be drawn by comparing the results of the two different scenarios. More false-positive detections are generated by the change detector on QW; by removing these false positives, the PHD-MT obtains larger improvements in terms of evaluation scores than on BW (Fig. 5).

Sample results of the PHD-MT used to process the output of the change detector are shown in Fig. 6. In this challenging situation generated by a sudden change in illumination, although the target size accuracy is not perfect, the heavy clutter is filtered by PHD-MT (Fig. 6, second, third and fourth row). Furthermore, in cases when a target generates noisy observations, the spatial smoothing produced by the PHD filter



(a) (b) (c)

Fig. 6. Comparison of tracking results between the multi-target tracker with (PHD-MT) and without (MT) PHD filter. (a) Detections (color-coded in red) and PHD output (color-coded in green). Several false detections are filtered by the PHD (second, third and fourth row). (b) MT results. (c) PHD-MT results. False tracks due to clutter are removed by PHD-MT.

facilitates data association preventing an identity switch on the same target (Fig. 6, third row, the pedestrian in the center of the scene).

Figure 7 shows the accuracy and the precision scores when we change the set-up of the PHD filter parameters. Each plot was obtained by changing with $\log_2$ scale one parameter at a time while fixing the rest to values defined at the beginning of this section. It is interesting to observe that large variations of tracking performance are associated with changes of the observation and dynamic model configurations (see Fig. 7 for $\sigma_{v(.)}$ and $\sigma_{n(.)}$). Too large or too small noise variances result in insufficient or excessive filtering and produce a drop of tracking accuracy. Also, decreasing $\rho$ (i.e, the number of particles per estimated target) reduces the quality of the filtering result as the approximation of the PHD propagation becomes less accurate. The PHD filter is less sensitive to variation of the other parameters. In the case of birth and clutter parameters ($\bar{s}$ and $\bar{r}$), low variability is associated with the fact that birth and clutter events are relatively sparse in the state and observation spaces. When varying $\bar{r}$, the average number of clutter points per scan, the result is stable until $\bar{r}$ is grossly overestimated. Similarly, only a small impact is associated with variations of missing detection ($P_M$) and survival ($e_{k|k-1}$) probabilities.

To demonstrate the flexibility and modularity of the proposed multi-target framework, we show the results obtained when substituting the change detector with a *face detector* [30]. The dataset used in this section is available at ftp://motinas.elec.qmul.ac.uk/pub/multi_face. Figure 8 shows a comparison of the results obtained with and without the use of the PHD filter on the detected faces. When false detections are processed, the mass of the PHD starts growing around them.

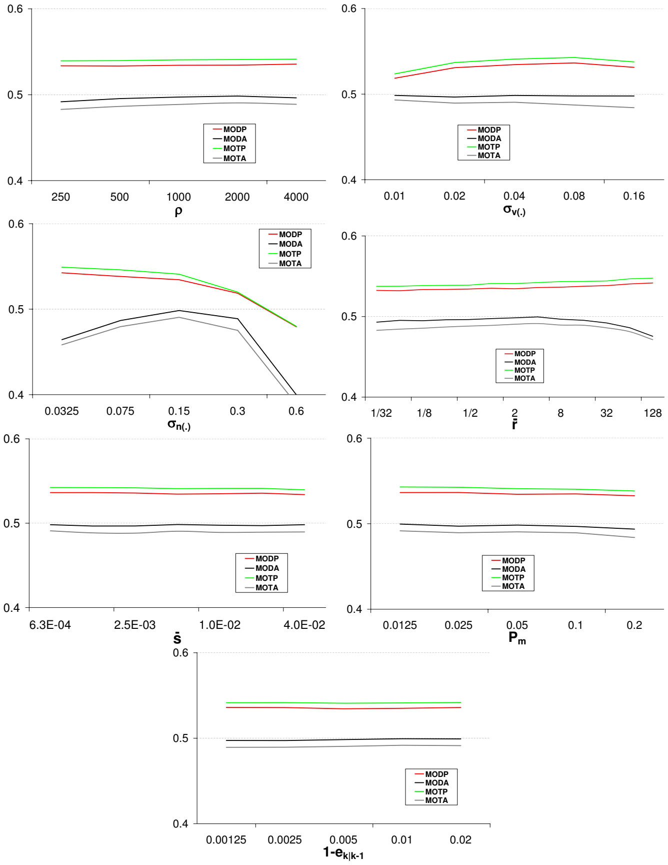
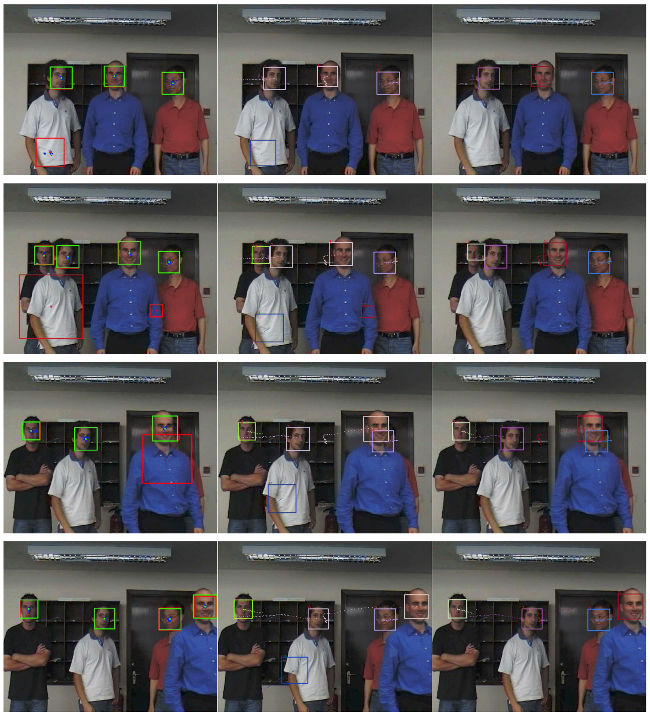


Fig. 7. Sensitivity analysis on the parameters of the PHD filter. Average evaluation scores on the Broadway Church (BW) scenario.

Multiple coherent and consecutive detections are necessary to increase the mass to a level greater than $T_M$. For this reason, when the clutter is not persistent, the PHD filter removes it. As mentioned earlier, due to the trade-off between clutter removal and response time, the drawback of this filtering is a slower response in accepting the birth of a new target.

In addition to the above, Fig. 8 shows how the combination of PHD filtering with the graph-based data association is able to recover the identity of faces after a total occlusion: in the third and fourth row, although a face is occluded by another person, data association successfully links the corresponding tracks. Finally, the results of PHD-MT compared with MT shows that two false tracks on the shirt of one of the targets are removed by the PHD-MT only.

To complete the analysis of the results, Fig. 9 shows two examples of *failure modalities* of the particle PHD filter. The close-up images in Fig. 9, top row, show a first failure modality. The change detector generates for the person in the far-field detections that are inconsistent over time. These detections are considered by the PHD filter as clutter and therefore eliminated. Figure 9, bottom row, shows a sample result when one of the assumptions of the PHD filter is violated (Sec. II-C), i.e., the targets generate dependent observations. As the targets overlap, the change detector merges the two blobs and produces one observation only. In this case the change of size is outside the range of changes modeled as noise. When the targets split, the delay introduced by



(a) (b) (c)

Fig. 8. Comparison of tracking results between the multi-target tracker with (PHD-MT) and without (MT) Particle PHD filtering. (a) Detections (color-coded in red) and PHD output (color-coded in green). Several false detections are filtered by the PHD (First, second and third row). (b) MT results. (c) PHD-MT results. The PHD-MT successfully recovers the faces after a total occlusion without generating false tracks.

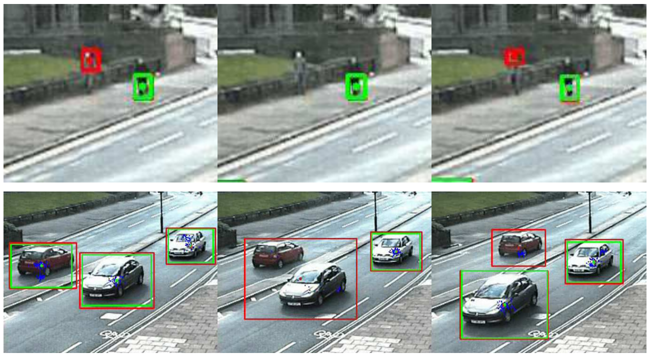


Fig. 9. Failure modalities of the Particle-PHD filter when using a change detector. The red boxes are the observations and the green boxes are the output of the PHD filter. (Top row) Inconsistent detections in the far field are interpreted by the PHD filter as clutter and therefore removed. (Bottom row) Interaction between targets (object merging) generates a bounding box for a group of objects.

the PHD filter generates a set of missing detections. While splitting could be partially handled by enabling spawning form targets (see (15)), merging of observations poses a problem as the PHD was originally designed to track using punctual observations just as for those generated in a radar scenario, where the target interaction is weak. These problems can be overcome by using a trained object detector (e.g., a vehicle detector), within the same framework.

The *computational cost* of the Particle PHD filter is comparable to that of the two object detectors (Fig. 10). The data association has low influence on the overall cost as the computation is based on positional information only. If more complex gain functions are used to weight the edges of the graph (for example by comparing target appearances using color histograms), then the data association would significantly contribute to the overall computational cost. The larger resource share claimed by the Particle PHD filter with the change detector, compared to the face tracking case, is mainly due to the larger average number of targets in the scene.

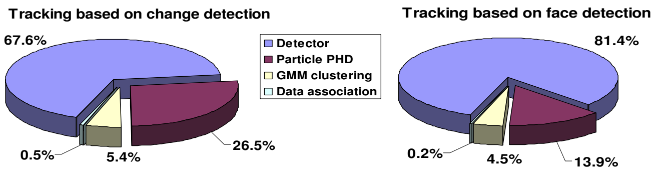


Fig. 10. Percentage of computational resources allocated to each of the tracker blocks. The PHD filter requires fewer resources than the detectors.

Figure 11 shows the processing time versus the number of targets estimated on the BW scenario. The processing time of the full tracker (PHD-MT) is compared with that of the recursive filtering step (PHD&GMM). The results are obtained with a non-optimized C++ implementation running on a Pentium IV 3.2GHz. As the number of particles grows linearly with the number of targets and the number of observations, the theoretical computational cost is also linear. The mild non-linearity of the curve PHD&GMM is due to the fact that with a low number of particles the processor performs most of the operations using the cache memory. When the number of targets increases, the filter propagates more particles and the curves become steeper as the cost is now associated with the use of off-chip memory. Also, a larger overhead of PHD-MT is due to the non-optimal implementation of the object detector (0.5 seconds/frame), and not to the filter itself. Furthermore, as most of the calculations necessary to propagate a particle depend on its previous state only, the Particle PHD is well suited for a parallel implementation. With an optimized implementation of the detector and a GPU (Graphics Processing Unit) or multi-core implementation of the PHD filter, the tracker could achieve real-time performance. It is of interest also to compare the computational time of PHD&GMM with the hypothetical results of a particle implementation that propagates the full multi-target posterior (FP). When one target only is visible, then the PHD and the FP resort to the same algorithm (that takes 40 milliseconds/frame). With multiple targets, because the dimensionality of the state space in FP grows, an exponential number of particles is necessary to achieve a constant density sampling. The computational time per frame of an FP implementation would then be: 1.5 seconds for two targets, 40 minutes for four targets and 187 years with 8 targets. In this case, the only feasible approach would be to use a more efficient sampling method in an MCMC fashion [9]. Unlike FP, the PHD filter limits the propagation of the particles to the single target state space and thus achieves linear complexity.

## V. Conclusions

We have presented a multi-target visual tracker that employs Particle PHD filtering to remove clutter and missing detections

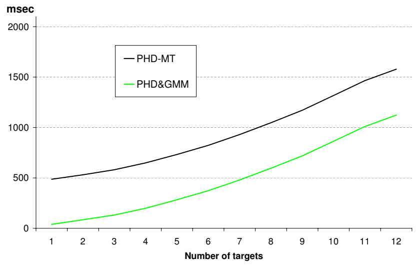


Fig. 11. Processing time in milliseconds versus estimated number of targets in the scene on a sequence from the CLEAR-2007 dataset. PHD-MT: full tracker; PHD&GMM: PHD filtering and GMM particle clustering steps.

from noisy observations. The motion of the targets and the noise on the observations are modeled using Gaussians with scale-dependent variances. To account for the different nature of the particles a multi-stage resampling strategy has been proposed. The resulting set of particles is clustered by a modified GMM adapted to the Particle PHD. To generate the final tracks, the centers of the clusters are processed by a data association algorithm based on graph matching. The proposed algorithm has the capability to remove non-persistent clutter, to filter missing detections, to smooth the tracks, and to overcome short-term occlusions. The approximation introduced by the PHD filter allows the reduction of computational cost from exponential (with the number of targets) to linear. Experimental results over a large dataset of real-world sequences show that the Particle PHD filter improves the robustness of the tracker against clutter by verifying the coherence of consecutive sets of detections.

As part of our current work, we are investigating data-driven methods to learn the parameters of the filter and models of track merging and splitting that combine the information within the PHD filter and the vertices of the graph. Future work also includes the integration of the proposed framework with an event detection algorithm to extract higher-level information from surveillance videos.

## References

[1] M. Isard and A. Blake, “Condensation – conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[2] ——, “ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework,” *Lecture Notes in Computer Science*, vol. 1406, pp. 893–908, 1998.

[3] E. Maggio and A. Cavallaro, “Hybrid particle filter and mean shift tracker with adaptive transition model,” in *Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, Philadelphia, USA, Mar. 2005, pp. 221–224.

[4] P. Hargrave, “A tutorial introduction to Kalman filtering,” in *IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments*, Feb. 1989, pp. 1/1–1/6.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[6] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Academic Press, 1988.

[7] K. Okuma, A. Taleghani, N. d. Freitas, J. J. Little, and D. G. Lowe, “A boosted particle filter: Multitarget detection and tracking,” in *Proc. of the European Conf. on Computer Vision*, 2004, pp. 28–39.

[8] M. Isard and J. MacCormick, “Bramble: A bayesian multiple-blob tracker.” in *Proc. of International Conf. on Computer Vision*, vol. 2, Vancouver, Canada, Jul. 2001, pp. 34–41.

[9] Z. Khan, T. Balch, and F. Dellaert, “An MCMC-based particle filter for tracking multiple interacting targets,” in *Proc. of the European Conf. on Computer Vision*, 2004, pp. 279–290.

[10] C. Hue, J.-P. Le Cadre, and P. Prez, “Tracking multiple objects with particle filtering,” *IEEE Trans. on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 791–812, Jul. 2002.

[11] A. Doucet, B. Vo, C. Andrieu, and M. Davy, “Particle filtering for multi-target tracking and sensor management,” in *Proc. of International Conf. on Information Fusion*, vol. 1, 2002, pp. 474–481.

[12] Y. Boers and J. Driessen, “Multitarget particle filter track before detect application,” *Radar, Sonar and Navigation, IEE Proceedings*, vol. 151, no. 6, pp. 351–357, 2004.

[13] J. Vermaak, S. Godsill, and P. Perez, “Monte Carlo filtering for multi-target tracking and data association,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, 2005.

[14] S. Sarkka, A. Vehtari, and J. Lampinen, “Rao-Blackwellized particle filter for multiple target tracking,” *Information Fusion*, vol. 8, no. 1, pp. 2–15, Jan. 2007.

[15] B. Vo, S. Singh, and A. Doucet, “Sequential monte carlo implementation of the PHD filter for multi-target tracking,” in *Proc. of International Conf. on Information Fusion*, 2003.

[16] K. Shafique and M. Shah, “A noniterative greedy algorithm for multi-frame point correspondence,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 1, pp. 51–65, 2005.

[17] R. Mahler, “A theoretical foundation for the stein-winter probability hypothesis density (PHD) multitarget tracking approach,” in *Proc. 2002 MSS Nat’l Symp. on Sensor and Data Fusion*, vol. 1, San Antonio, USA, Jun. 2000.

[18] D. Clark, K. Panta, and VoBa-Ngu, “The GM-PHD filter multiple target tracker,” in *Proc. of International Conf. on Information Fusion*, Quebec, CAN, Jul. 2006, pp. 1–8.

[19] D. E. Clark and J. Bell, “Data association for the PHD filter,” in *Proc. of Second International Conf. on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, AU, Dec. 2005, pp. 217–222.

[20] K. Panta, B.-N. Vo, and S. Singh, “Novel data association schemes for the probability hypothesis density filter,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 556–570, 2007.

[21] B.-N. Vo and W.-K. Ma, “The Gaussian mixture probability hypothesis density filter,” *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[22] B.-N. Vo, A. Pasha, and H. D. Tuan, “A Gaussian mixture PHD filter for nonlinear jump markov models,” in *Proc. of IEEE Conf. on Decision and Control*, 2006, pp. 3162–3167.

[23] A. Pasha, B. Vo, H. Tuan, and W.-K. Ma, “Closed form PHD filtering for linear jump markov models,” in *Proc. of International Conf. on Information Fusion*, 2006, pp. 1–8.

[24] H. Sidenbladh and S. Wirkander, “Tracking random sets of vehicles in terrain,” in *Proc. of IEEE Workshop on Multi-Object Tracking*, Madison, USA, Jun. 2003.

[25] D. Clark, I. Ruiz, Y. Petillot, and J. Bell, “Particle PHD filter multiple target tracking in sonar images,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 409–416, 2006.

[26] N. Ikoma, T. Uchino, and H. Maeda, “Tracking of feature points in image sequence by SMC implementation of PHD filter,” in *Proc. of SICE Annual Conf.*, vol. 2, Sapporo, JP, Aug. 2004, pp. 1696–1701.

[27] Y. Wang, J. Wu, A. Kassim, and W. Huang, “Tracking a variable number of human groups in video using probability hypothesis density,” in *Proc. of IEEE International Conf. on Pattern Recognition*, Hong Kong, CH, Aug. 2006.

[28] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, “Particle PHD filtering for multi-target visual tracking,” in *Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, 2007, pp. I–1101–I–1104.

[29] A. Cavallaro and T. Ebrahimi, “Interaction between high-level and low-level image analysis for semantic video object extraction,” *EURASIP Journal on Applied Signal Processing*, vol. 6, pp. 786–797, Jun. 2004.

[30] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[31] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. Automat. Contr.*, vol. 24, no. 6, pp. 843–854, 1979.

[32] J. Hopcroft and R. Karp, “An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs,” *SIAM J. Computing*, vol. 2, no. 4, pp. 225–230, Dec. 1973.

[33] R. Kasturi, *Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction*, Computer Science & Engineering University of South Florida, Jan. 2006.