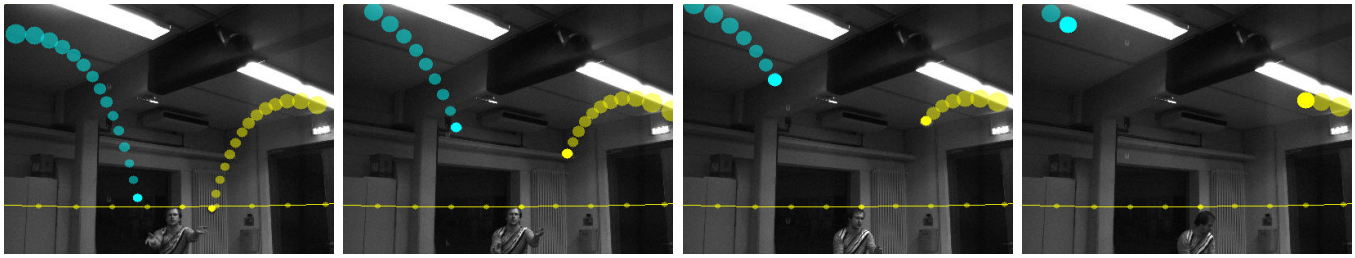
# Estimation and Prediction of Multiple Flying Balls Using Probability Hypothesis Density Filtering

Oliver Birbach Udo Frese



***Abstract*— We describe a method for estimating position and velocity of multiple flying balls for the purpose of robotic ball catching. For this a multi-target recursive Bayes filter, the Gaussian Mixture Probability Hypothesis Density filter (GM-PHD), fed by a circle detector is used. This recently developed filter avoids the need to enumerate all possible data association decisions, making them computationally efficient. Over time, a mixture of Gaussians is propagated as tracks, predicted into the future and then sent to the robot. By learning a prior from training data we are focusing on detections that are likely to lead to a catchable trajectory which increases robustness. We evaluate the tracker's performance by comparing it with ground truth data, assessing tracking performance as well as the prediction precision of single tracks. Reasonable prediction performance is acquired right from the start, leading to a good overall catching rate.**

## I. Introduction

Target tracking from image sequences has the goal to estimate the states of an unknown number of targets by integrating detections. Often, this includes dealing with false-alarms, missed and noisy detections, and target birth and death. In a classical way, this problem is solved using Multiple Hypothesis Tracking (MHT) [1], [2], [3], which hypothesizes associations between measurements and targets and propagates a set of these, where the one with the highest posterior probability is considered to be the most probable association. Optimally, all hypotheses should be propagated but due to combinatorial intractability when the number of targets and measurements is increased, only the most probables are kept [4].

An emerging technique for multi-target tracking is the Random Finite Set approach presented in [5], [6]. Here, the fundamental idea is to model states and measurements as random variables that take random sets as values. This allows a direct Bayesian formulation of the multi-target tracking problem, instead to the explicit modeling of data associations between targets and measurements as in MHT. For most practical applications, the computational intractability of the multi-target integrals prohibit using the formulation directly. For this, a first moment approximation known as the Probability Hypothesis Density (PHD) Filter was proposed which propagates a posterior intensity recursively. This intensity is similar to a distribution in state space, *i.e.* with peaks denoting targets, except that its integral is not 1 but the expected number of targets. Implementations of the PHD recursion exist modeling the intensity as particles, known as the Sequential Monte Carlo (SMC) PHD filter [5], or as a Gaussian Mixture (GM), known as the GM-PHD filter [7], [5]. These filters still enumerate between measurements and targets but do not consider different combinations of associating these such as MHT does. This makes them computational attractive.

In this paper, we make use of the GM-PHD filter to address the problem of estimating and predicting multiple balls pitched towards a humanoid robot (see image sequence from the robot's cameras on top) with the goal to catch each ball with one arm. This delicate task is a challenging tracking problem: Due to the short flight time, the trajectory of pitched balls must be detected as a track as early as possible so the ensuing planning stage has enough time to find a valid arm posture. Also, tracking accuracy is crucial, especially at the end of the trajectory. A special problem are systematic false-alarms for instance created by people's heads. The tracker initially creates a track from such a measurement, but soon discards it because it does not follow the parabolic flight of a ball.

The paper presents three main contributions. First, we present the foundations to successfully track multiple balls pitched to the observer using the PHD filter with non-linear Gaussian models. In particular, we feed circle measurements to the filter with the need to handle birth and death of targets,

O. Birbach and U. Frese are with German Center for Artificial Intelligence (DFKI). 28359 Bremen, Germany oliver.birbach@dfki.de

false-alarms and skipped tracks. We try intuitive explanations despite a complex mathematical topic.

Second, we learn a prior from training data on position and velocity of pitched balls. This reduces the number of tracks initially created from systematic false alarms. This is realized in a special way, which prevents linearization problems that occur with nonlinear models and established techniques for track creation in the PHD filter.

Our third contribution is the evaluation of catching experiments on a real humanoid robot (namely DLR's humanoid *Rollin' Justin* [8]). We evaluate the prediction accuracy metrically using an external reference tracking system and asses the multiple-target tracking performance while comparing it with MHT. In fact, this paper extends our previous work [9], [10], [11] on a perception system for catching two balls with a mobile humanoid. Here, we introduce, adapt and successfully use a conceptually different tracking paradigm as an alternative to the previous tracking backend (MHT).

The tracking system has shown its practicality during demonstrations of the robot in different locations. Although PHD filtering is usually utilized in the tracking community, this work is the first, to our knowledge, to employ PHD filtering in a computer vision application for robotics.

## II. Related Work

Many approaches have been published on detection-based multiple-target tracking. Usually, these can be classified into two strategies: the ones that associate detections incrementally in a frame-by-frame or windowed way, while others perform global data association over all frames.

Sequential Monte Carlo methods, also known as particle filters, have been introduced for visual tracking [12] and extended by [13] for reliable multiple target tracking. Further extensions make use of an interleaved AdaBoost/particle filter [14] and a MCMC-based particle filter [15] for robust handling of data association. However, a huge number of particles is needed when the state space uncertainty decreases by a large factor during tracking. Parametric approaches, *e.g.* based on Kalman filters (KF), are usually better suited for real-time high-accuracy applications. Such an approach is kernel-based Bayesian filtering [16]. Here, a mixture of Gaussians is used to represent the posterior and likelihood using approximation and interpolation, reducing the number of samples required for robust tracking.

Classical methods, such as JPDAF [2], [3] and MHT [1], [2], [4], mostly used in conjunction with Kalman filters, optimize multiple trajectories at the same time in a frame-by-frame or windowed fashion. Unfortunately, computational complexity increases exponentially with number of targets and measurements. Nevertheless, MHT paired with an UKF was successfully used to track multiple balls thrown by people to each other [17].

Global approaches include modeling the data association problem in a flow network and looking for minimal cost [18], as a multi-path searching problem solved by Linear Programming [19], as the optimal concatenation of tracklets containing true positives [20] or using Quadratic Boolean Programming to couple object detection and space-time trajectory estimation [21]. Global and local data association techniques have been effectively coupled, where a particle filter generates reliable tracklets which are globally optimized using the Hungarian algorithm [22]. Despite the good performance of such approaches regarding ambiguities caused by occlusions and detection errors, their use in an online scenario is computationally prohibitive.

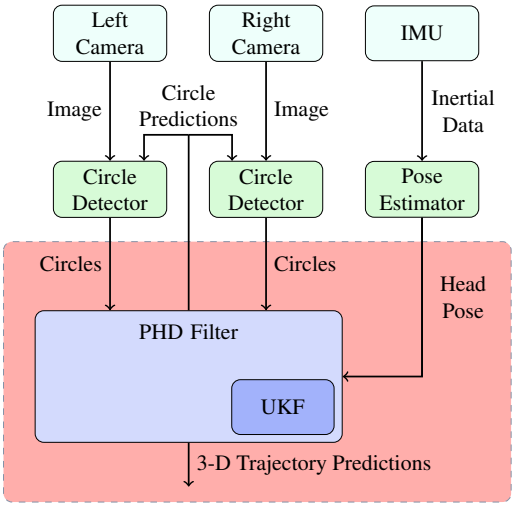



Fig. 1. Data flow of the system. Sensors are shown in yellow, sensor processing in green, and tracking in blue. The components enclosed by the red rectangle are subject of this paper.

Actually, most of tracking systems for robotic catching do not consider the multiple-target case. Existing ones [23], [24], [25], [26], [27] only actuate one arm, tracking a single ball. This simplifies the problem to the single target case, which is achieved by using an Extended Kalman Filter (EKF) [25], [27] or a simple parabolic fit [23], [26]. The authors' previous work [9] employed a MHT filter for handling the multiple-target case, which will be compared to the proposed approach in Sec. VI.

Previous usage of the PHD filter in computer vision is scarce. The only approaches so far employ PHD filtering for tracking feature points [28], human groups [29] or faces, people and vehicles [30]. All three use SMC-PHD to track in image space. This is contrary to our work, since we make use of the GM-PHD filter to track in 3-D. Recently, a way of initializing tracks from detections for tracking in 3-D using GM-PHD was proposed [31]. Our approach goes one step further by also defining a spatial prior, which encodes how the ball is typically thrown and merges this into new tracks. This greatly increases robustness.

## III. Overview

Our sensor-setup consists of a pair of forward-looking cameras and an Inertial Measurement Unit (IMU) mounted at the head of a humanoid robot on wheels (DLR's *Rollin' Justin* [8]). This paper focuses on integration of circle measurements from both cameras, the head pose is provided by an IMU-based dead-reckoning, considered black-box in

this paper. The data flow of these three sensors is shown in Fig. 1.

For detecting the ball as circles we use the circle-detection scheme introduced in [17]. Here, the average fraction of image energy that is a radial gradient is computed. To achieve real-time operation this is done in a multi-scale fashion passing the best $N$ circles to the tracking algorithm.

The IMU is used to provide pose information required in the camera's measurement model. This information could be acquired from the robot's kinematic chain, but since the robot's torso has elasticity and hysteresis, the wheels have dampers and slip as well as loose contact to the ground, the IMU is used. Structure from motion is an alternative, but relies on sufficient and near background. We discarded it for the moment with the outlook of using both in future.

The PHD filter, in the focus of this paper, then estimates the state, *i.e*. position and velocity of all flying balls from this input. For this, it uses multiple Unscented Kalman Filters (UKF) as the underlying single ball models. The motion and measurement model are taken from [17] with gravity, air-drag, and a radial distorted pin-hole camera. The PHD is executed on the left and on the right image. Thereby it implicitly both performs stereo matching and tracking over time. Finally, the position estimates from the UKF are predicted into the future and sent to the planning algorithm for finding the best arm posture to catch the ball.

## IV. PHD FILTERING

Using random finite set (RFS) theory, a multi-target Bayes filter can be derived as a generalization of the well-known single target Bayes filter, which uses the classical notion of a state vector. Both cases need approximations for a tractable solution. The (E/U) KF propagates (approximatly) the single-target posterior as a Gaussian over time, *i.e*. up to second order statistics.

Similarly, the Probability Hypothesis Density (PHD) filter was proposed as an approximation to the multi-target Bayes filter by propagating the first-order moment statistic of the multi-target posterior. The posterior PHD, a so-called intensity, is characterized by the property that integration of it over any region in state space results in the *expected* number of targets in this region. This number is fractional, being the sum of integer target numbers weighted by probability. By contrast, in the classical Bayesian single-target case the integral of the probability density gives the probability that the target is in this region. This difference has important implications: A PHD can not represent that almost for sure there is a target in a region. In a probability distribution, as used by the MHT, this is expressed by an integral of almost one (say $0.999$), but in a PHD the same value also represents two targets with a probability of $0.4995$. This representation issue will have important consequences in Section IV-C.

Nevertheless, instead to the MHT which enumerates all possible measurement to track assignments, this algorithm unifies all measurements with all tracks in a composite-hypothesis fashion. This leads to a computationally efficient algorithm handling missing and false alarm detections, birth and death of tracks as well as noisy measurements.

### A. PHD recursion

Although used for derivation, the closed-form formulas of the PHD filter do not make use of RFS theory. Here, we provide a brief review of the PHD recursion [5], [6]. An alternative derivation is given by [32] using a quantized state-space model of infinitesimal bins. Subscripts $_{k+1|k}$ refer to quantities before, subscripts $_{k+1|k+1}$ after fusing detections at time $k$. A target state is called $x$, usually as a free variable because the PHD operates on intensities over $x$.

**Prediction.** Let $D_{k|k}$ be the posterior PHD intensity from time $k$. The predicted PHD intensity is then given by

$$D_{k+1|k}(x) = b_{k+1|k} + \int p_S(x') f_{k+1|k}(x|x') D_{k|k}(x') dx', \quad (1)$$

where $b_{k+1|k}$ is the intensity of spontaneous births of targets, $f_{k+1|k}(x|x')$ is the single target Markov transition density (motion model) and $p_S(x')$ is the probability of target survival. Similar to a Bayes filter, old posterior density and transition model are multiplied and the old state is marginalized out by integrating. The birth intensity is usually vague or even uniform. It has the same role as a Bayesian prior plus the integral indicating the average number of new targets per frame.

**Update.** The corresponding updated PHD is given by

$$D_{k+1|k+1}(x) = [1 - p_D(x)] D_{k+1|k}(x) + \sum_{i=1}^{m_{k+1}} \frac{s_{z^i}(x)}{s^*_{z^i} + \int s_{z^i}(x') dx'}$$
$$s_z(x) = L_z(x) p_D(x) D_{k+1|k}(x), \quad s^*_z = \lambda c(z), \quad (2)$$

with $z^{1..m_{k+1}}$ being detections, $p_D(x)$ for the probability that a target with state $x$ will be detected, $\lambda$ the Poisson-distributed number of false alarms, spatially distributed as $c(z)$, and $L_z(x)$ the single-target measurement likelihood.

Intuitively, the support $s_z(x)$ indicates how much detection $z$ supports state $x$. It is proportional to how much target there is at $x$ ($D_{k+1|k}(x)$), how likely it is detected ($p_D(x)$), and how well the actual detection $z$ fits to the state $x$ ($L_z(x)$). Similarly, $s^*_z$ indicates how much $z$ supports the clutter ($\lambda c(z)$). Overall, a single detection can support exactly $1$ target, with $1$ meaning cardinality, *i.e*. integral over $x$. Hence, the support $s_z$ is normalized to $1$ by dividing by the overall $s_z$, *i.e*. the integral over $x$ plus clutter.

This normalization in the end makes consistent detections create integral-1-peaks in the PHD: If a detection fits well to an existing peak in $D_{k+1|k}$, the support $s_z(x)$ of these states dominates the normalization, so almost $1$ is contributed to these states in $D_{k+1|k+1}$. If a detection supports no state $x$ significantly, the normalization is dominated by the support $s^*_z$ for clutter. So in (2) little is contributed to any state in $D_{k+1|k+1}$ and the integrated number of targets.

Please note, that the single-target Bayes equation is achieved when $p_D = 1$, $|Z| = 1$ and $\lambda = 0$. Also the computational components resemble a Bayes filter: Multiplication of distributions, marginalization, and integration.

### B. Gaussian Mixture implementation

From the general PHD recursion (1) and (2) an analytic implementation, the GM-PHD filter, has been derived [7] that represents the PHD as a weighted mixture of Gaussians.

The sum of Gaussians is expanded with the result that every Gaussian is predicted individually in (1) and updated, *i.e.* multiplied, with the likelihood of every detection in (2). These two operations correspond to a single-target Bayes filter and are implemented by (E/U) KFs for (non-) linear measurements. The denominator is independent of $x$, *i.e.* a weighting factor and easily computed by summing weights. So in the big picture, a PHD runs a set of KFs, each on a combination of detections fused into one target (a track) just as the MHT. But, while the MHT argues on top of that about combinations of tracks forming a hypothesis, the PHD has a much simpler weighting scheme based on the support.

**Prediction.** Suppose we have a prior GM-PHD

$$D_{k|k}(x|Z^k) = \sum_{i=1}^{n_{k|k}} w^i_{k|k} \cdot \mathcal{N}(x; x^i_{k|k}, P^i_{k|k}). \tag{3}$$

Then the predicted GM-PHD is the same mixture, but with each Gaussian predicted according to the motion model, weighted by survival probability $p_S$ and new (vague) Gaussians added according to the birth intensity. The result is

$$\begin{aligned} D_{k+1|k}(x) = &\sum_{i=1}^{a_k} \beta^i_k \cdot \mathcal{N}(x; b^i_{k+1|k}, B^i_{k+1|k}) \\ &+ \sum_{i=1}^{n_{k|k}} p_S \cdot w^i_{k|k} \cdot \mathcal{N}(x; x^i_{k+1|k}, P^i_{k+1|k}) \end{aligned} \tag{4}$$

where $a_k$, $\beta^i_k$, $b^i_{k+i|k}$, and $B^i_{k+1|k}$ define a Gaussian mixture birth intensity, usually just a single very large Gaussian. The predicted targets, $\mathcal{N}(x; x^i_{k+1|k}, P^i_{k+1|k})$, are computed using the prediction step of the single-target Kalman filter. In our case, this is an UKF using sigma point propagation.

**Update.** Rewrite the predicted GM-PHD as a flat mixture

$$D_{k+1|k}(x) = \sum_{i=1}^{n_{k+1|k}} w^i_{k+1|k} \cdot \mathcal{N}(x; x^i_{k+1|k}, P^i_{k+1|k}). \tag{5}$$

For a Gaussian $L_{z^j}(x) = \mathcal{N}(z^j - h(x); 0, R)$ in measurement space and one Gaussian $\mathcal{N}(x; x^i_{k+1|k}, P^i_{k+1|k})$ from the mixture the support $s_z(x)$ is a product of Gaussians which is again a *weighted* Gaussian [5, (D.1)]

$$\begin{aligned} &\mathcal{N}(z^j - h(x); 0, R) \cdot \mathcal{N}(x; x^i_{k+1|k}, P^i_{k+1|k}) \\ &\quad \overset{=}{\approx} q^{i,j} \mathcal{N}(x; x^{i,j}_{k+1|k+1}, P^{i,j}_{k+1|k+1}). \end{aligned} \tag{6}$$

Practically, this is simply a KF update with the resulting weight computed from Mahalanobis distance. For non-linear measurement models $h$ as ours, (6) is approximate and obtained by an UKF or EKF. This weight reflects how well the detection fits the state and makes improbable combinations become lowly weighted in the mixture.

The support normalization is computed from the sum of weights, since the integral of a Gaussian is simply 1. Hence, the updated GM-PHD is

$$\begin{aligned} D_{k+1|k+1}(x) = &\sum_{i=1}^{n_{k+1|k}} (1 - P_D) w^i_{k+1|k} \mathcal{N}(x; x^i_{k+1|k}, P^i_{k+1|k}) \\ &+ \sum_{j=1}^{m_{k+1}} \sum_{i=1}^{n_{k+1|k}} \frac{s^{i,j}}{s^{*,j} + \sum_{k=i}^{n_{k+1|k}} s^{k,j}} \mathcal{N}(x; x^{i,j}_{k+1|k+1}, P^{i,j}_{k+1|k+1}), \end{aligned} \tag{7}$$

The first sum contains Gaussians not fused with any detection, the second double sum is the result of fusing all Gaussians with all detections by a KF update (6). It is weighted by the normalized support $s^{i,j}$ which is the integrated contribution of Gaussian $i$ to $s_{z^i}(x)$ in (8). It is computed as

$$s^{i,j} = w_{k+1|k}\, p_D(x^i_{k+1|k})\, q^{i,j}, \quad s^{*,j} = \lambda c(z^j) \tag{8}$$

from the result of the KF update (6).

**Mixture Management.** In each iteration, many new Gaussians are formed out of each existing ones. This is better than the MHT which considers hypotheses formed of several targets on top of the Gaussians. Still, we need to limit this combinatorial explosion by gating. If the Mahalanobis distance, which governs the weight $q^{i,j}$, is above a threshold the Gaussian $i, j$ is discarded. Also, we merge Gaussians until their number falls below a fixed value [7]. Pairs with small weighted mutual Mahalanobis distance $\chi^2$ are merged first using the formula below which preserves mean and covariance of the mixture:

$$\begin{aligned} \chi^2 &= w^1_{k|k} w^2_{k|k} (x^1_{k|k} - x^2_{k|k})^T \left( {P^1_{k|k}}^{-1} + {P^2_{k|k}}^{-1} \right) (x^1_{k|k} - x^2_{k|k}) \\ \tilde{w}_{k|k} &= \sum_{i=1}^{2} w^i_{k|k}, \quad \tilde{x}_{k|k} = \tilde{w}^{-1}_{k|k} \sum_{i=1}^{2} w^i_{k|k} x^i, \\ \tilde{P}_{k|k} &= \tilde{w}^{-1}_{k|k} \sum_{i=1}^{2} w^i_{k|k} \left( P^i_{k|k} + (x^i_{k|k} - \tilde{x}_{k|k})(x^i_{k|k} - \tilde{x}_{k|k})^T \right) \end{aligned} \tag{9}$$

### C. Missed detection problem

A well-known weakness of the PHD filter compared to MHT is its response to missed detections [33], [34]. If a target is not detected, only the first addend in (2) contributes, effectively scaling the corresponding peak by $1 - P_D$. If a MHT is convinced there is a target, several missed detections are needed to change its mind. For the PHD a single missed detection suffices, assuming $P_D > \frac{1}{2}$.

This behavior can be understood from the intuition in the beginning of Sec. IV. The MHT can represent that almost for sure there is one target by a probability of $0.9999$. For the PHD the same intensity also includes two or no targets. If no detection is observed the latter is the natural conclusion.

A modified version of the PHD, the Cardinalized PHD [33], [34] filter tries to resolve this problem by propagating the entire probability distribution of the number of targets in addition to the first moment of the multi-target posterior. The filter still operates on the single target space, but is much more complex. We currently ignore this problem as with the next detection the track appears again including all information from past detections.

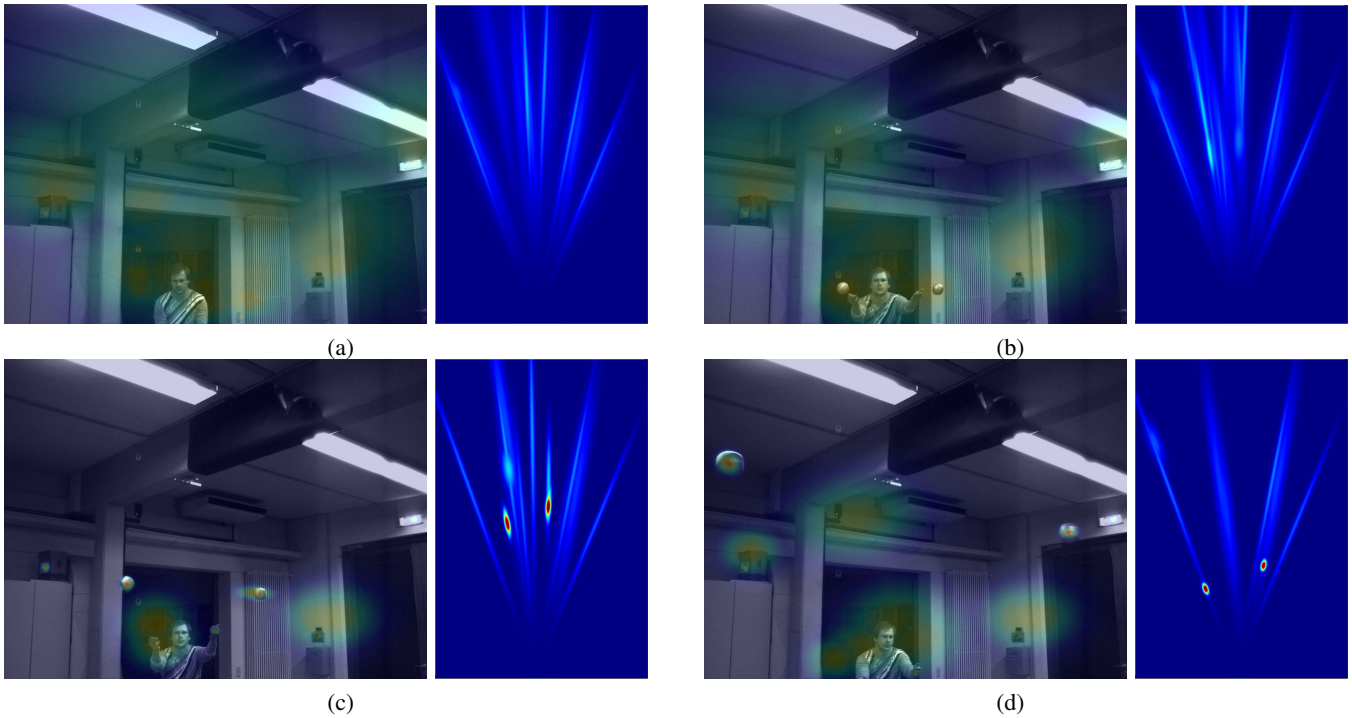


Fig. 2. Images from a ball throwing sequence showing the PHD (only position) projected into image space (left) and its projection into the floor of the (right) while using a learned prior. The projected space on the floor is 4 m in width and 6 m depth direction. It can be seen that when no ball is pitched only low weight components from false-alarm managements appear (a), indicating no target. When two balls are actually thrown, higher weighted Gaussians emerge (b) and become peaked as these are supported by ensuing measurements (c). Integrating close measurements leads to strong weighted and highly peaked Gaussians in the mixture (d). Please note how the covariance ellipse of (low weighted) initial Gaussians are directed towards one point (the cameras). This is due to the invariant application of the prior with regard to the horizontal direction of the incoming ball (see text).

## V. Prior-based target initialization

Using the classic way of starting new targets, one would integrate a Gaussian with mean at the standard throwing distance and a large position and velocity covariance for Eq. 4. This shows two problems, a qualitative and a technical one.

### A. Learning a Gaussian Prior

The first problem is every false-alarm causes computational load and systematic false-alarms, such as people's heads sometimes create unwanted tracks. We therefore propose starting tracks using a Gaussian prior learned from training-data. It defines with uncertainty from where a ball is typically pitched (position), in what trajectory it will roughly proceed (velocity), and that it is thrown towards the robot (correlation between both). If this prior is integrated as birth intensity, the PHD automatically discards many false alarms.

We perform non-linear least squares estimation on the circles of a set of trajectories to compute the initial state $x_0^i$ belonging to the first circle of each trajectory $i = 1 \dots n_s$. Using the whole trajectory as context improves precision of the ball's initial position and velocity. To such a set of samples a, 6-D Gaussian is fitted using the common sample-mean and sample-covariance formulas

$$b^1 = \frac{1}{n_s}\sum_{i=1}^{n_s} x_0^i,\ B^1 = \frac{1}{n_s - 1}\sum_{i=1}^{n_s}(x_0^i - b^1)(x_0^i - b^1)^T. \quad (10)$$

This Gaussian $\mathcal{N}(b^1, B^1)$ is used as birth intensity during the PHD prediction step. As we will see in the experiments this reduces the number of Gaussians needed in the Gaussian mixture.

### B. Special Update for the First Detection

The second problem is linearization: Because of the non-linear model, linearization has to be performed, in our case, through the UKF. This linearization is problematic, especially the mapping from the depth of a ball to a circle radius. The usual UKF update used in (7) linearizes the measurement model on the $1\sigma$-range of the Gaussian from the mixture and fuses the resulting linear measurement with the well-known KF formula. Now, in case of the prior Gaussian the $1\sigma$-range is very large, much larger than the $1\sigma$-range of the detection itself, so the measurement model is poorly approximated in that relevant region. The consequence is a systematic error in the initial state estimate.

We propose to flag the Gaussians that originate from the birth part of (4) and use a specialized update routine to fuse them with a detection according to (6). The result is a peaked Gaussian and can be further processed normally.

Now, in the special situation of the first detection, the detection itself is much more precise than the prior. Hence, linearization should be performed on the $1\sigma$-range of the detection not the prior. We implement this by converting the detection into a 3D position using the inverse measurement model $h^{-1}$. The covariance of this position is obtained by propagating the specified covariance of the detection's center and radius through $h^{-1}$ using $\sigma$-point propagation as in the

UKF. The result is a Gaussian in position which can be fused with the prior Gaussian (position and velocity) by a *linear* Kalman filter update. This procedure replaces the usual UKF-update for computing (6) in this special case.

Two remarks: First, the Mahalanobis distance of this update is also subject to gating, discarding highly improbable false-alarms from the beginning. Second, during learning balls may originate from a specific horizontal direction. We do not want the system to learn that. So, before we fuse the prior as described above, it is rotated such that the prior's mean points into the horizontal direction of the detection. This ensures that the system is invariant with regard to the horizontal direction of the incoming ball.

### C. Comparison to Initialization by Detection

In [31], a different method for initialization from the first detection is presented. The authors propose not to add birth intensity in (4) but instead add Gaussians created from the detections alone in (7). These would have a constant support defined by birth density and take part in the normalization, *i.e*. the denominator, as all support values do.

This procedure realizes an uniform prior, as the Gaussians are derived from the first detection alone, not from fusing it with a prior as our more general approach does. It can still be seen as an instance of our idea to have a special operation for fusing with the prior. When implementing an uniform prior, an uniform instead of a Gaussian intensity is added to $D_{k+1|k}$ in (4). This is no conceptual problem, just makes the mixture heterogeneous with some Gaussian and some uniform components. However, then a special operation must be implemented to multiply the uniform component with a Gaussian measurement likelihood in (7). The result is simply the Gaussian from the detection weighted with the uniform prior intensity. This happens in [31].

## VI. Experiments and Results

In a robotic catching system, the subsequent processes (*i.e*. planning and control) rely on properly working tracking for a satisfying catch rate (which is $\approx 80\%$). In this section, we demonstrate the performance of the prior-enhanced PHD filter by tracking balls pitched towards the robot.

We used two Gaussian priors for evaluation. First, we defined an *uninformed* prior with a sigma for velocity of $6m/s^2$ for each component, assuming a flying ball regardless of direction. Second, we learn prior from training data (see Sec. V). For this, 77 trajectories in which the robot reached for the ball were collected. To cover a broad range of throws, trajectories pitched by different people in different labs were used. Please note that not necessarily a new prior has to be learned when a different type of ball is used. Since the pitching behavior stays the same, only the air drag coefficient in the motion model and the ball diameter in the measurement model have to be adjusted.

Eleven sequences are tested in our experiments (15s duration, 8 pitched balls, mostly 2 at the same time, $\approx$ 1s flight duration of each ball). The circle detector [17] was instructed to return the $N = 25$ most circular looking objects, in which the ball was prevalently included. Circles that lie within circles were excluded (as commonly observed at balls). The multiple-target model (*e.g*. birth intensity *etc*.) is given in Tab. I (chosen by hand). Here, two probabilities depend on the state of the track. If its predicted measurement is outside of the image it is always a missing observation ($P_D = 0$). When the ground is hit, the track ends ($P_S = 0$).

TABLE I
Parameters of PHD and UKF, see Sec. IV–V

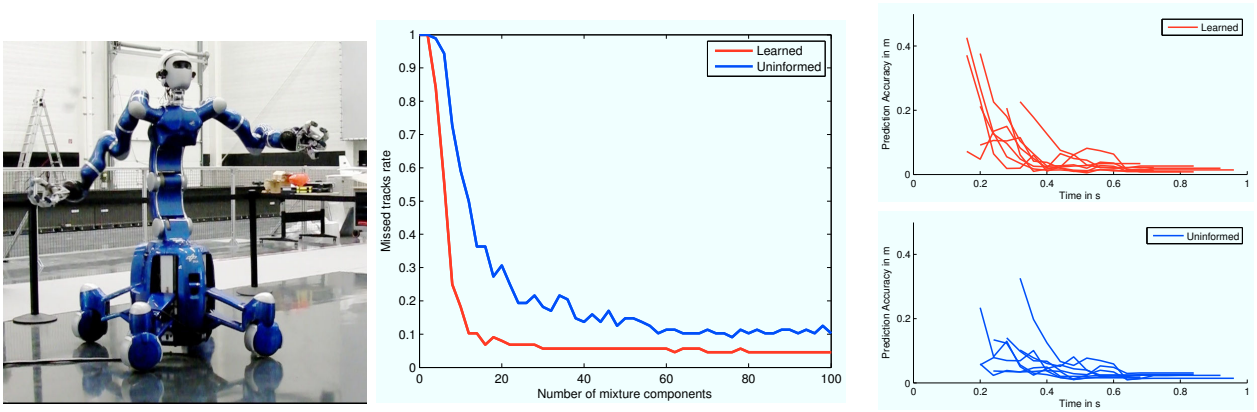
| PHD | | UKF | |
|---|---|---|---|
| $w_0$ | $1.46 \cdot 10^{-10}$ | $\sigma_{x,y}$ | 1.5px |
| $\lambda c(\cdot)$ | $1.08 \cdot 10^{-6}$ | $\sigma_r$ | $0.15\%r$ |
| $P_S(x)$ | 1 (0) | $\sigma_Q$ | 0.1 m/s$^2$ |
| $P_D(x)$ | 0.95 (0) | | |

Figure 3(b) shows the rate of detected pitched balls over the number of mixture components for both priors. A pitched ball is counted as successfully detected when its predicted accuracy is below 25cm, which is also the used criterion that the predicted trajectory will be sent to the next stage (*i.e*. the planner). It can be seen that using the learned prior outperforms using the uninformed one. Less components are needed for tracking the same number of targets and the learned prior is able to track more targets overall. This results suggest that a prior, trained to the expected data, considerably focuses on the relevant measurements for generating new components which are then continued much more effectively. Unfortunately, full detection rate is not achieved in Fig. 3(b), as a couple of balls were thrown too low or off the robot due to rapid throwing . The prior defines that the tracker should look for catchable balls. So these balls were discarded as is the intention of giving such a prior. Therefore, virtually perfect performance is achieved for the learned prior.

**Single-Target Prediction Accuracy.** From an actuation point of view, a catch will be smoother if the trajectory is known early. In Fig. 3(c) the tracker's single-target tracking performance is shown. Here, the predicted 3-D points of detected tracks (when they would hit the robot) of one sequence and the corresponding points extracted from an external 3-D tracking system (ground-truth) are compared over the time starting from the point where the ball left the hand. Using 50 mixtures for each, the detected tracks using the learned prior appear earlier (the best at 2 fr., *i.e*. 80ms) while being imprecise. Over time, both trackers approach each other, while reaching a final accuracy of $\approx 2.5$cm.

**Multiple-Target Performance.** It is worthwhile to compare the PHD performance to MHT [4], [9]. Both algorithm use the same underlying UKF (including parameters), the same multiple-target parameters (see Tab. I) and track starting using a Gaussian prior. We adjusted the MHT parameters for robust tracking by hand (likelihood ratio: $1e^{-11}$, $n$-scan back pruning: 10, $k$-best hypotheses: 10). The PHD filter used the learned prior while 25 mixtures were propagated. This number was experimentally found to be robust.

Figure 4(a) shows the cardinality statistics of both algorithms, *i.e*. the number of estimated targets. While the




(a) (b) (c)

Fig. 3. (a) DLR's humanoid *Rollin' Justin* on which the tracking system is integrated. (b) Missed detections rate of uninformed and learned prior over the number of components. (c) Prediction accuracy over time of uninformed and learned prior after the ball left the hand.

track starting behavior of both methods is almost the same, the PHD filter terminates a track much earlier than MHT in seven out of eight times. This is due to the discussed characteristics of the PHD filter when opposed missing detections, prematurely declaring an ending track. This is not a problem, since subsequent detections recover the correct target state (as around second 12). Unfortunately, missing detections at image's border are more common. This ends tracks before they would continue until reaching the ground outside the image. Again, this does not pose a problem as all possible information of the ball's flight has already been integrated and submitted to the planning algorithm.

In Fig. 4(b) the computation time over one sequence is given. Because of propagating a constant number of components, the PHD filter is always equally efficient. This is in contrast to the MHT which needs more time for data association while tracking. To be precise, GM-PHD only needs 4.3ms while MHT needs 8.2ms in the worst case, which is the relevant case for real-time applications. Thus GM-PHD only needs 53% of the time MHT requires. This is a substantial difference since in our case only 10ms are available for tracking. In fact, knowing the trajectory of the ball early is essential for successful catching. Thus the computational advantage outweighs the rarely occurring (and not harmful) disadvantages for our application.

## VII. Conclusion and Future Work

Multiple-target tracking is a challenging task which is required in many computer vision applications. In this paper, we introduced Probability Hypothesis Density (PHD) filtering, a recently emerged frame-by-frame multiple-target tracking approach for a 3-D real-time computer vision application. Instead of replicating PHD equations used with the commonly utilized multiple-target model, we proposed an alternative way composing the density, namely the notion of measurements supporting tracks, see Eq. 2. This contribution of a general way to look at PHD filtering makes the description more accessible.

In conjunction with the proposed target initialization approach through an offline-learned prior, a humanoid robot is able to catch two simultaneously pitched balls robustly and accurately. Although PHD filtering lacks robustness in certain detection situations, its simplicity (only Eq. 4 and 7 have to be implemented) and its computational behavior may convince researchers to employ this approach in the future. To quantify the compactness, the used MHT implementation has 3262 lines of code while our GM-PHD implementation only has 619 lines of code (all C++) excluding a minimum (and trivial) part of the MHT's interface. One might argue that separate single trackers might be a solution, but it should be considered if its worth to build an ad-hoc solution if the proposed GM-PHD provides a compact and sound solution.

For the future, approaches for improving filter behavior are desirable, where the CPHD filter has made a step in the right direction. On the other hand, it would be interesting to integrate a real-time robust global data association approach for the described multiple-target tracking scenario.

## VIII. Acknowledgments

This work was supported under DFG grant FR2620/1-1. We thank DLR's *Rollin' Justin* team for providing the robot.

## References

[1] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, vol. AC-24, no. 6, pp. 843–854, 1979.

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

[3] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Int. Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[4] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[5] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[6] R. P. S. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

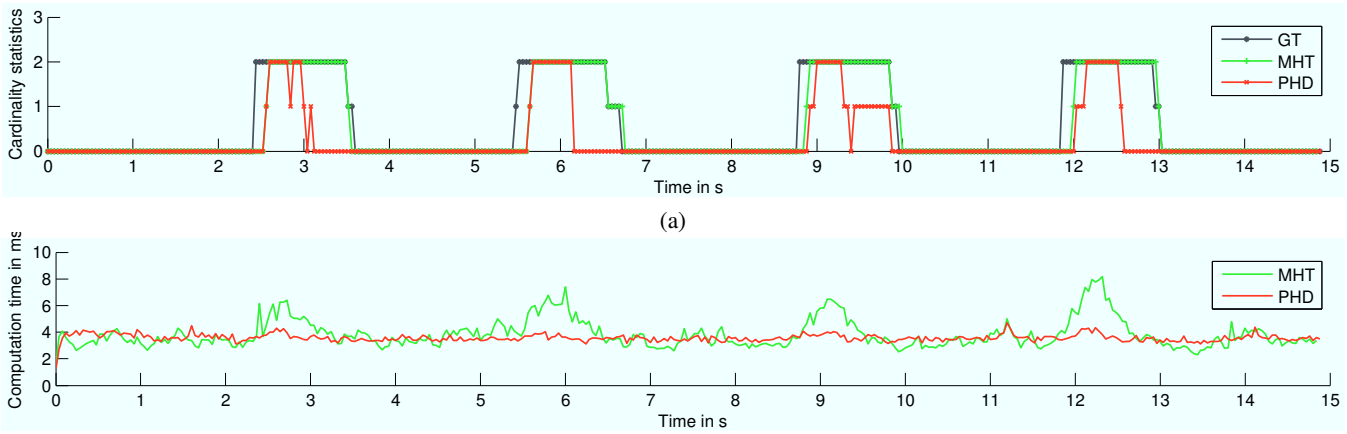



Fig. 4. Comparison of the PHD filter and MHT. (a) Cardinality statistics versus Ground-truth (GT) over time. A ground-truth track starts when it lefts the hand and ends when it reaches the ground. (b) Computation time of over time on a Intel Core$^{TM}$2 Quad Q9000 embedded system on the robot.


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

[8] C. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schäffer, and G. Hirzinger, "Rollin justin - mobile platform with variable base," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 1597–1598.

[9] O. Birbach, U. Frese, and B. Bäuml, "Realtime perception for catching a flying ball with a mobile humanoid," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 5955–5962.

[10] B. Bäuml, F. Schmidt, T. Wimböck, O. Birbach, A. Dietrich, M. Fuchs, W. Friedl, U. Frese, C. Borst, M. Grebenstein, O. Eiberger, and G. Hirzinger, "Catching flying balls and preparing coffee: Mobile humanoid Rollin Justin perfoms dynamic and sensitive tasks," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3443–3444.

[11] B. Bäuml, O. Birbach, T. Wimböck, U. Frese, A. Dietrich, and G. Hirzinger, "Catching flying balls with a mobile humanoid: System overview and design considerations," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robotics*, 2011, submitted.

[12] M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking," *Int. Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[13] J. Vermaak, A. Doucet, and P. Prez, "Maintaining multi-modality through mixture tracking," in *Proc. of the IEEE Int. Conf. on Computer Vision*, 2003.

[14] K. Okuma, A. Taleghani, N. de 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.

[15] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1918, 2005.

[16] B. Han, Y. Zhu, D. Comaniciu, and L. S. Davis, "Visual tracking by continuous density propagation in sequential bayesian filtering framework," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 919–930, 2009.

[17] O. Birbach and U. Frese, "A multiple hypothesis approach for a ball tracking system," in *Computer Vision Systems*, ser. LNCS, M. Fritz, B. Schiele, and J. H. Piater, Eds., vol. 5815, 2009, pp. 435–444.

[18] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[19] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[20] F. Yan, A. Kostin, W. Christmas, and J. Kittler, "A novel data association algorithm for object tracking in clutter with application to tennis video analysis," in *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[21] B. Leibe, K. Schindler, and L. V. Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[22] J. Xing, H. Ai, and S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," in *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[23] R. L. Andersson, "Dynamic sensing in a ping-pong playing robot," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 728–739, 1989.

[24] K. Nichiwaki, A. Ionno, K. Nagashima, M. Inaba, and H. Inoue, "The humanoid saika that catches a thrown ball," in *Proc. of the IEEE Int. Workshop on Robot and Human Communication*, 1997, pp. 94–99.

[25] C. Smith and H. I. Christensen, "Using COTS to construct a high performance robot arm," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2007.

[26] M. Riley and C. G. Atkeson, "Robot catching: Towards engaging human-humanoid interaction," *Autonomous Robots*, vol. 12, no. 1, pp. 119–128, 2002.

[27] U. Frese, B. Bäuml, S. Haidacher, G. Schreiber, I. Schaefer, M. Hähnle, and G. Hirzinger, "Off-the-shelf vision for a robotic ball catcher," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001.

[28] N. Ikoma, T. Uchiuo, and H. Maeda, "Tracking of feature points in image sequence by SMC implementation of PHD filter," in *SICE Annual Conf.*, 2004.

[29] Y.-D. Wang, J.-K. Wu, A. A. Kassim, and W.-M. Huang, "Tracking a variable number of human groups in video using probability hypothesis density," in *Proc. of the IEEE Int. Conf. on Pattern Recognition*, 2006.

[30] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, "Particle PHD filtering for multi-target visual tracking," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2007.

[31] J. Houssineau and D. Laneuville, "PHD filter with diffuse spatial prior on the birth process with applications to GM-PHD filter," in *13th Int. Conf. on Information Fusion*, 2010.

[32] O. Erdinc, P. Willett, and Y. Bar-Shalom, "The bin-occupancy filter and its connection to the PHD filters," *IEEE Trans. on Signal Processing*, vol. 57, no. 11, pp. 4232–4246, 2009.

[33] R. P. S. Mahler, "PHD filters of higher order in target number," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.

[34] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. on Signal Processing*, vol. 55, no. 7, pp. 3553–3567, 2007.