

# Robust Real-Time Multiple Target Tracking

Nicolai von Hoyningen-Huene and Michael Beetz

Intelligent Autonomous Systems Group,  
Technische Universität München,  
Boltzmannstr. 3,  
85748 Garching, Germany  
{hoyninge,beetz}@cs.tum.edu

**Abstract.** We propose a novel efficient algorithm for robust tracking of a fixed number of targets in real-time with low failure rate. The method is an instance of Sequential Importance Resampling filters approximating the posterior of complete target configurations as a mixture of Gaussians. Using predicted target positions by Kalman filters, data associations are sampled for each measurement sweep according to their likelihood allowing to constrain the number of associations per target. Updated target configurations are weighted for resampling pursuant to their explanatory power for former positions and measurements. Fixed-lag of the resulting positions increases the tracking quality while smart resampling and memoization decrease the computational demand. We present both, qualitative and quantitative experimental results on two demanding real-world applications with occluded and highly confusable targets, demonstrating the robustness and real-time performance of our approach outperforming current state-of-the-art.

## 1 Introduction

Low cost and high availability of digital cameras offer opportunities in traditional surveillance tasks and open up new fields like automatic sports analysis or studies of social insect behavior. While the available video footage grows constantly, these data need to be examined. Robust automatic tracking of multiple targets in video is a key feature to assist or avoid expensive and tedious human interactions.

We present a multiple target tracking algorithm that can follow more than twenty similar targets robustly over long sequences in real-time. The proposed method constitutes a Rao-Blackwellized Resampling Particle filter with fixed-lag estimates. The posterior of target positions given the observed measurements is approximated as a mixture of Gaussians approving the use of Kalman filters. The data association problem is solved by sampling according to the likelihood of an assignment for one measurement sweep. Multiple measurements can be assigned to the same target following a Poisson distribution. While smart resampling and memoization allows for real-time capability, fixed time delay for the estimates offers an increase in robustness.

We applied our method to demanding sequences in the soccer and insects domain as they contain a high volume of similar and confusable targets with

natural motion. Our algorithm exhibits higher quality in tracking at less computational time than the current state-of-the-art multiple target tracking method by Khan et al. [1]. Lower failure rate can be achieved by disallowing merged measurements and restricting multiple measurements to a Poisson distribution, which better matches reality. Performance gain is due to the direct sampling of associations instead of running a Markov chain, better exploiting recyclability and avoiding uninformative computations like burn-in steps.

After a survey of related work in the field of multiple target tracking, we detail our new Rao-Blackwellized Resampling Particle filter in section 3. Section 4 depicts the experimental results for soccer and ant tracking with comparison to the state-of-the-art. We finish in section 5 with our conclusions.

## 2 Related Work

The problem of tracking is to recursively estimate an unknown state based on limited observations. Tracking algorithms follow predominantly a Bayesian approach approximating the posterior probability density function (pdf) of the target states given all measurements up to that time, an initial target distribution and the process of how positions evolve over time (motion model) and how measurements inform about target states (sensor model).

In single target tracking two approaches are widely used. The Kalman filter [2] constrains the target state distribution to a Gaussian, consists of a predict and an update step and has shown to be the optimal estimator for linear motion and sensor model. Several suboptimal extensions as e.g. the Extended and the Unscented Kalman filter have been proposed for nonlinear motion and/or sensor models and additional constraints. The second approach known as Particle filter or sequential Monte Carlo method (SMC) approximates arbitrary probability density functions (pdf) on discrete points (particles) only (see [3, 4]) yielding a fast tracking method also for nonlinear motion and sensor models.

Multiple-target tracking algorithms differ from single target tracking by the problem of associating each measurement with an appropriate target which is known as data association. Multiple target tracking approaches can be categorized by their handling of the data associations problem.

The Nearest-Neighbor Data Association (NN) assigns each measurement to the closest target mostly based on the Mahalanobis distance (e.g. [5]). The Joint Probabilistic Data Association filter (JPDAF) forms a sub optimal Bayesian algorithm that approximates the posterior distributions of the targets as separate Gaussians for each target, that is assigned to all measurements with weights depending on the predicted association probability (see [6, 2]).

Multiple hypothesis tracking (MHT) [6] builds a (mostly pruned) tree of all possible association sequences of each measurement with close targets. The restriction to single associations only as well as the use of Kalman filters and the Hungarian method to find the  $k$  best global associations allow computation in polynomial time, but inhibit to handle multiple or merged associations. The Probabilistic MHT (PMHT) [7] does not attempt to enumerate all possible

combinations of feasible data association links, but uses a probabilistic structure derived using expectation-maximization.

Markov Chain Monte Carlo (MCMC) methods sample the data associations based on an importance density by starting with an initial association and proposing local modifications of it (e.g. associate, dissociate or swap) that are accepted with a special acceptance ratio. The sampling performs a Markov Chain over associations in a Bayesian graph where the transition probabilities are chosen so that the stationary distribution of the chain converges to the density of the data associations. The Markov Chain is usually run for a burn-in time for initialization of real sampling, the relative frequency of the sampled associations form the desired pdf. Khan et al. proposed in [1, 8] a real-time Rao-Blackwellized MCMC-based particle filter allowing also sampling of split and merged measurement associations. Counterintuitively for a particle filter the MCMC approach can not easily be parallelized maintaining the correct sampling behavior, although work has been published recently by [9].

The Rao-Blackwellized Monte Carlo data association (RBMCD) approach by Särkkä [10, 11] sequentially samples one association after another estimating target positions as a mixture of Gaussians and handling dependencies between assignments of each single measurement by data association priors. The assumption of independence of the order of data associations in one sweep is made. RBRPF [12] is an extension of RBMCD introducing smart resampling and memoization, that lead to real-time tracking in the first place, and relaxation of the association independence assumption.

### 3 Rao-Blackwellized Resampling Particle Filter with Fixed-Lag

Following the Bayesian approach our tracking method approximates the posterior probability density function (pdf)  $p(x_k|z_{1:k})$  of the target positions  $x_k$  at time  $k$  given all measurements  $z_{1:k}$  seen so far. A particle filter for complete player configurations constitutes the base of our algorithm. The pdf is approximated only at  $S$  discrete points  $x_k^i$  with weights  $w_k^i$  called weighted particles:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^S w_k^i \delta(x_k - x_k^i). \quad (1)$$

Each particle consists of Gaussians for all  $N$  target states with mean  $m$  and covariance  $V$ :

$$x_k^i = \{\mathcal{N}(x_{j,k}^i; m_{j,k}^i, V_{j,k})\} j = 1, \dots, N. \quad (2)$$

The main loop of the algorithm is depicted in fig. 1 following the Sample Importance Resampling (SIR) framework described in [4] with a combined step of an early resampling and the drawing of new particles. These steps are merged due to the discrete (but still exponential) number of possible data associations

which change the nature of sampling. To save computational time, every particle is predicted once with a given motion model  $f$  (also called system model)

$$\hat{x}_k^i = f_k(x_{k-1}^i, \Gamma_{k-1}) \quad (3)$$

with i.i.d. process noise  $\Gamma_{k-1}$ . Each target state is predicted individually under the assumption that their motion is independent. The prediction is solved analytically which is known as Rao-Blackwellization, which is possible owing to the Gaussian nature of the target states. The most probable associations given predicted positions and measurements are sampled several times according to the weight of their former particle plus a constant minimum number  $o$ . An assignment  $J_{k,r}^i(j, l)$  of a specific target  $j$  to a measurement  $l$  is drawn using the importance density

$$p(J_{k,r}^i(j, l)) = \frac{p(z_{l,k} | \hat{x}_{j,k}^i)}{p(J_{k,r}^i(l, \emptyset)) + \sum_j p(z_{l,k} | \hat{x}_{j,k}^i)} \quad (4)$$

with  $J_{k,r}^i(l, \emptyset)$  denoting the measurement to be clutter.

Measurements and target states are linked by the sensor model  $h_k^r$  (also called measurement model)

$$z_k = h_k^r(\hat{x}_k^i, R_k), \quad (5)$$

with i.i.d. measurement noise  $R_k$ . The function  $h_k^r$  depends on  $J_{k,r}^i$  and relates assigned target states to the measurements  $z_k$ . If  $h_k^r$  is a linear function (written as a matrix  $H_k^r$ ) and target state and measurements are Gaussian, individual assignment probabilities can be evaluated analytically as

$$p(z_{l,k} | \hat{x}_{j,k}^i) \sim \mathcal{N}\left(z_{l,k}; H_{j,k}^r \hat{x}_{j,k}^i, H_{j,k}^r V_{j,k}^i H_{j,k}^{rT} + R_{l,k}\right) \quad (6)$$

with  $R_{l,k}$  denoting the covariance of the measurement  $z_{l,k}$ . The probability for a measurement to be clutter depends on the application, but can mostly be approximated to be uniformly distributed over the sensor space  $\mathcal{M}$

$$p(J_{k,r}^i(l, \emptyset)) \sim |\mathcal{M}|^{-1}. \quad (7)$$

The probability for an assignment can also be influenced by additional constraints like the matching of untracked properties (e.g. color and appearance) and the probability for multiple assignments in one sweep. Multiple measurements for one target are sampled according to a Poisson distribution with  $\lambda = p_{sd}$ . If a target is tossed not to have an additional assignment, the probability for that assignment is zero and it is therefore not taken into account in the sum of the denominator in eq. 4 for that iteration. Before each sampling from one specific particle all measurements are shuffled randomly to prevent an unwanted prior due to the order of measurements (especially for multiple assignments).

During the sampling, data associations for one particle are checked for identity to skip unnecessary weight and update computations. Each drawn data

association  $J_{k,r}^i$  for one particle  $i$  results in a new particle  $u$  with the target states updated according to a weighted sum of predicted and observed states by their uncertainties:

$$m_{j,k}^u = \hat{m}_{j,k}^i + V_{j,k}^u (h_k^r)^{-1} \left( \sum_{J_{k,r}^i(l,j)} R_{l,k}^{-1} (z_{l,k} - h_k^r(\hat{m}_{j,k}^i)) \right) \quad (8)$$

and covariances updated as

$$V_{j,k}^u = \left( \hat{V}_{j,k}^{-1} + (h_k^r)^{-1} \left( \sum_{J_{k,r}^i(l,j)} R_l^{-1} \right) \right)^{-1}. \quad (9)$$

The inverse of the linear sensor model  $h_k^r$  is usually not linear for measurements providing partial information about the state only, but can be solved by copying the missing information from the target state into the measurement. For measurements providing only positional data but target states containing velocity information, eq. 5 can be written as

$$(x_z, y_z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} (x, y, \dot{x}, \dot{y})^T \text{ or } (x_z, y_z, \dot{x}, \dot{y}) = I (x, y, \dot{x}, \dot{y})^T \quad (10)$$

with the second (augmented) form being invertible.

The weights of the newly sampled particles  $u$  are evaluated as the probability for the former prediction and all assigned measurements given the newly sampled positions including a detection probability  $p_d$  for all  $a$  assigned targets

$$w_k^u \propto p(\hat{x}_k^i | x_k^u) p_d^a (1 - p_d)^{N-a} \prod_{J_{k,r}^i(l,j)} p(z_{l,k} | x_{j,k}^u) \prod_{J_{k,r}^i(l,\emptyset)} p(J_{k,r}^i(l,\emptyset)). \quad (11)$$

The weight is multiplied by the number of times this data association was sampled.

This approach differs from [11], where Särkkä et al. set the weights according to the probability of the associations that were used for sampling, but is similar to [1] where residuals between updated positions and predictions as well as measurements are used. It helps to avoid the hospitality problem where multiple measurements are preferred over only one measurement of high accuracy because the weight for such an association is higher due to smaller resulting covariances in the denominator of the Gaussians.

An estimate of the target states  $x_k$  is found by selecting the particle with maximum weight. In the case of fixed-lag estimation, target states are evaluated as particles which descendant is the particle with maximum weight for the fixed time distance  $\delta$  in the future resulting in higher robustness due to the exploitation of more informations by the cost of a delay.

Fig. 1: Main Loop of the Proposed Rao-Blackwellized Resampling Particle Filter

```

program RBRPF
  input
    { $x_{k-1}^i, w_{k-1}^i$ } $_{i=1}^{N_{k-1}}$  particles for time  $k-1$ 
     $z_k$  measurements at time  $k$ 
  output
    { $x_k^j, w_k^j$ } $_{j=1}^{N_k}$  particles for time  $k$ 
BEGIN
  FOR  $i = 1 : N_{k-1}$ 
     $x_k^j \sim \text{DRAW-N-RESAMPLE}[z_k, x_{k-1}^i]$ 
    Calculate  $w_k^j$  according to 11
  END FOR
  Normalize weights:  $w_k^i = w_k^i \left( \sum_{i=1}^{N_s} w_k^i \right)^{-1}$ 
END

```

## 4 Experimental Results

We conducted two experiments on real world tracking problems with a fixed number of about twenty targets. Target states have been modeled as 2-dimensional position and velocity  $m_{j,k}^i = (px, py, vx, vy)^T$ . The motion model was chosen as the discretized Wiener velocity model  $A_{\Delta t}$  (see [2]) for time difference  $\Delta t$  between  $k-1$  and  $k$  as a linear motion model:

$$\hat{m}_{j,k}^i = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} m_{j,k}^i, \quad V_k' = A_{\Delta t} V_{k-1} A_{\Delta t}^T + \begin{pmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{pmatrix} \tilde{q} \quad (12)$$

with power spectral density  $\tilde{q}$  as a constant factor.

We used the additional number of samplings  $o = 10$  through all experiments. Initial positions were given manually.

### 4.1 Tracking Soccer Players

Identity tracking in sports is an interesting and demanding testing bed for multiple target tracking algorithms due to frequent occlusions of similar targets. The tracker was evaluated as part of the ASPOGAMOSystem [13]. We provide a video sequence of the beginning of the 2006 world championship's final consisting of 1262 frames shot with 25Hz. We tracked all soccer players and the main referee (23 targets) captured by a nonstatic pan-tilt-zoom camera used for TV broadcasting. An example image is depicted in fig. 2(a). Homographies for each frame have been computed using [14]. Groundtruth was collected by manually marking each target position in the video image and transforming it to world

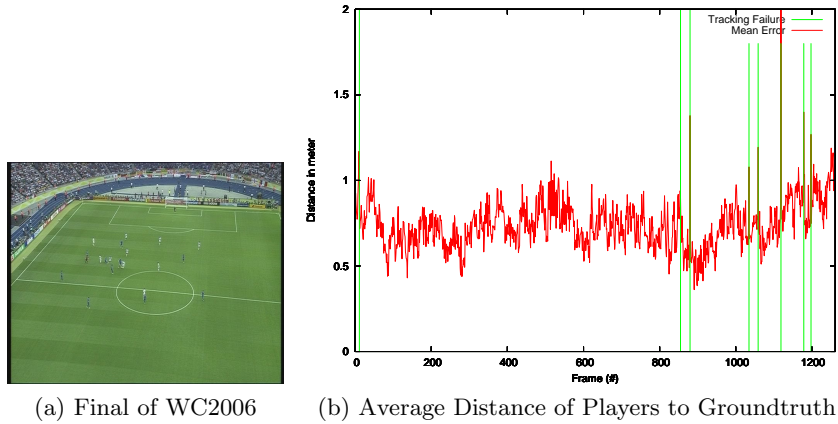


Fig. 2: The Soccer Groundtruth Sequence with 1262 Frames

coordinates for the whole scene. Mostly all players are visible in the sequence, but goalkeepers and wing players are sometimes not shown due to zooming. The sequence constitutes a demanding test for any multiple target tracking method because of uncertain, missed and occluded targets captured by a moving camera.

The automatic player detection was done following [15] by segmenting possible player regions via thresholding the local variance of the grayscale video image while skipping the field lines. These regions were matched by color templates to suppress outliers and to robustly estimate the players center of gravity. The center point was assumed to be at 0.9m above the ground to allow the computation of real-world coordinates by transforming the projected point with the inverse homography. One sweep corresponds to measurements at one frame of the video. The measurement covariances depend on the estimation of the camera parameters and differ also inside one frame depending on the distance to the camera. They are in the range of  $[0.17, 12.5]$  in goal and  $[0.15, 2.53]$  in sideline direction.

We used the following parameters for the constant velocity model:  $\Delta t = 0.04$  (due to 25fps),  $\tilde{q} = 0.0008$  (due to max acceleration of humans) and set  $p_{sd} = 0.3$ ,  $p_d = 0.3$ . Covariances were initied with  $V_0 = 0.001I_{4N}$ . Tracking was done with  $S = 50$  in real-world coordinates; all positions and covariances are specified in meters.

Failures were counted when a target deviated from the ground truth position by more than 5.0 meters. After a failure, only the failed target was reinitialized to the ground truth position and tracking was resumed. Our method without delay failed 8 times on the soccer sequence with 1262 frames. A delay could not reduce the number of failures further. The mean distance to the groundtruth is depicted in fig. 2(b). The smart resampling resulted in  $38.91 \pm 13.92$  effectively used particles. Tracking without player detection needed  $15.05ms \pm 5.077$  per frame resulting in a mean frame rate of 66.4 fps.

## 4.2 Tracking Ants

In [1] Khan et al. tested their proposed MCMC tracker on a challenging ground truth sequence of twenty ants in a small container. The image data and groundtruth are available online at <http://www.kinetrack.org>.

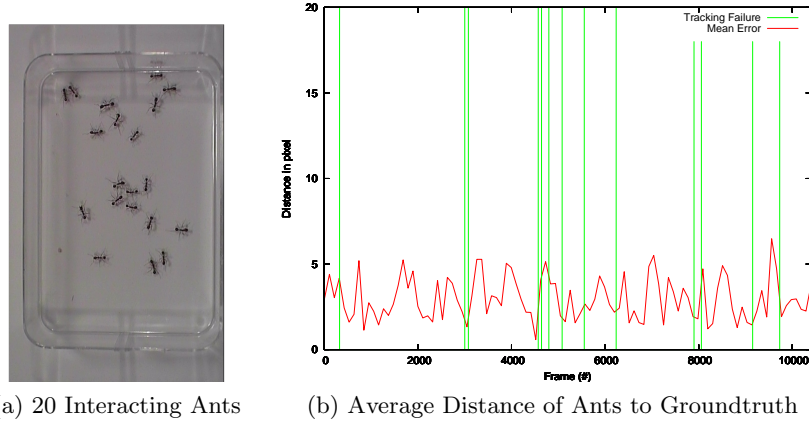


Fig. 3: Insects Domain with Frequent Interactions and Similar Target Appearance

The ants that should be tracked to gain insights in social behavior of insects are about 1cm long and move as quickly as 3cm per second frequently interacting with up to five or more ants in close proximity. The test sequence presents a substantial challenge for any multitarget tracking algorithm and was selected for comparison purpose. An image is depicted in fig. 3(a). The sequence consists of 10,400 frames recorded at a resolution of  $720 \times 480$  pixels at 30 Hz. We used the same simple thresholding procedure of the blurred and sownsampled video as [1] to obtain the measurements.

The number of failures detected on the ground truth sequence for the MCMC tracker with different number of particles and our tracker without and with smoothing are shown in table 1. Failures were counted when a target deviated from the ground truth position by more than 60 pixels. After a failure, all of the targets were reinitialized to the ground truth position and tracking was resumed. We used the same parameters as given in [1]. Measurements contain simple 2D positions  $z_l = [x, y]^T$  without velocity. Target motion was modeled using a constant velocity model as mentioned above with time step  $\Delta t = 0.033$  and  $\tilde{q} = 32$ . The initial covariance was set to  $V_0 = 32I_{4N}$  and the measurement noise was  $R = 32I_{4N}$ . All positions and covariances are specified in pixels.

We measured the run time as the average frame rate in frames per second (fps) including image processing time on a 2.2 GHz Dual-core PC and also on a Pentium 4-M 1.6 GHz for better comparability. With current standard hardware

our method is able to track the twenty ants faster than real-time (40 fps) with low failure rate. The smoothing reduces the number of failures even further while keeping the frame rate at 40 fps. Our algorithm exhibits higher quality in tracking needing about one half of the computational time than the current state-of-the-art tracker in [1]. Instead of [1] we do not allow merged measurements as these result mostly from the fore target occluding the back and may mislead the tracker. Also we restrict the number of detections of one target by a Poisson distribution with  $p_{sd}$ , yielding less (possibly wrong) associations. The speed-up is achieved as we directly sample the associations instead of running a Markov chain, allowing a better constriction to necessary computations by memoization without the need for uninformative burn-in steps. The average distance over all

Table 1: Experimental Results for Tracking Ants through 10,400 Frames

Algorithm	P4-M 1.6GHz	P4-M 3Ghz	Dual Core 2.5Ghz	Fail
MCMC [1] $S = 1$	-	$23.03 \pm 0.87$ fps	-	24
MCMC [1] $S = 6$	-	$8.75 \pm 0.55$ fps	-	21
RBRPF $S = 6$	$8.38 \pm 1.5$ fps	-	$40.68 \pm 1.0$ fps	19
RBRSPF $S = 6, \delta = 4$	$8.39 \pm 1.5$ fps	-	$40.76 \pm 1.0$ fps	13

ants is depicted in fig. 3(b). Analogous pictures have been published for the MCMC tracker in [8]. The distance for tracking without smoothing differs only minor to the ones with delay, which is based on the use of Kalman filters to predict and update target states in both approaches. The mean for the average tracking error is with 3.16 pixels low respecting a systematic error caused by the downsampling to forth of the original resolution only.

We also conducted experiments on a second ant dataset of [1] where ants were moving on two glass layers. Khan et al. provide 16 video sequences that were preprocessed in the same way as above to extract measurements from video. The MCMC approach could track through 12 of the 16 demanding sequences successfully with parameters  $\Delta t = 0.1$ ,  $V_0 = 32I_{4N}$ ,  $\Gamma = 4I_{4N}$  and  $\Sigma_{ii} = 150I_{4N}$  but failed on sequences 5, 8, 12 and 14. Our approach could also handle 12 of the 16 sequences using  $p_d = 0.8$ ,  $p_{sd} = 0.14$  but failed on 3, 8, 12 and 16. With  $p_{sd} = 0.4$  our method also tracked through sequence 16 successfully. All sequences include longer partial or full occlusions or sudden changes in direction and velocity which makes it a hard task for every tracker assuming a constant velocity model. In average about 40fps could be achieved on the Core 2 Duo with 2.5 GHz emphasizing the real-time capability of RBRPF.

## 5 Conclusions

We presented the Rao-Blackwellized Resampling Particle filter with Fixed-Lag as a novel multiple target tracking algorithm. The method exhibits real-time performance by exploiting the properties of Gaussians through Rao-Blackwellization and the discreteness together with rareness of probable data associations through

smart resampling. Robustness of tracking is increased by retrieving target estimates after a fixed-lag and therefore utilizing more informations. Demanding real-world experiments with frequent interactions and highly similar targets demonstrate the capabilities of our approach, that outperformed the state-of-the-art MCMC method in robustness as well as computational time.

## References

1. Khan, Z., Balch, T., Dellaert, F.: MCMC Data Association and Sparse Factorization Updating for Real Time Multitarget Tracking with Merged and Multiple Measurements. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **28**(12) (2006) 1960–1972
2. Bar-Shalom, Y., Fortmann, T.: *Tracking and Data Association*. Academic Press (1988)
3. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision* **29**(1) (1998) 5–28
4. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Trans. on Signal Processing* **50**(2) (Feb 2002)
5. Cai, Y., de Freitas, N., Little, J.J.: Robust Visual Tracking for Multiple Targets. In: *Europ. Conf. on Computer Vision*. Volume 3954 of LNCS. (2006) 107–118
6. Bar-Shalom, Y., Li, X.R.: *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS (1995)
7. Streit, R.L.: The PMHT and related applications of mixture densities. In: *Proc. Intl. Conf. on Information Fusion*. (July 2006)
8. Khan, Z., Balch, T., Dellaert, F.: MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2005)
9. Bardet, F., Chateau, T., Ramadasan, D.: Real-Time Multi-Object Tracking with Few Particles. In: *VISAPP, INSTICC (2009)* 456–463
10. Särkkä, S., Vehtari, A., Lampinen, J.: Rao-Blackwellized Monte Carlo data association for multiple target tracking. In: *Proc. of Intl Conf. on Information Fusion*. Volume 7., Stockholm (June 2004)
11. Särkkä, S., Vehtari, A., Lampinen, J.: Rao-Blackwellized Particle Filter for Multiple Target Tracking. *Information Fusion Journal* **8** (2007) 2–15
12. v. Hoyningen-Huene, N., Beetz, M.: Rao-Blackwellized Resampling Particle Filter for Real-Time Player Tracking in Sports. In: *VISAPP, INSTICC (2009)* 464–471
13. Beetz, M., von Hoyningen-Huene, N., Kirchlechner, B., Gedikli, S., Siles, F., Durus, M., Lames, M.: ASPOGAMO: Automated Sports Games Analysis Models. *International Journal of Computer Science in Sport* **8**(1) (2009)
14. Gedikli, S.: *Continual and Robust Estimation of Camera Parameters in Broadcasted Sport Games*. PhD thesis, TU München (2008)
15. Beetz, M., Gedikli, S., Bandouch, J., Kirchlechner, B., von Hoyningen-Huene, N., Perzylo, A.: Visually Tracking Football Games Based on TV Broadcasts. In: *Proc. of Intl. Joint Conf. on Artificial Intelligence (IJCAI)*. (2007)