Multiple Model Based Point Targets Tracking Using Particle Filtering in InfraRed Image Sequence

Mukesh A. Zaveri S. N. Merchant Uday B. Desai
SPANN Lab, Electrical Engineering Dept., IIT Bombay - 400076.
Emails: [mazaveri,merchant,ubdesai]@ee.iitb.ac.in

Abstract

Particle filtering is being investigated extensively due to its important feature of target tracking based on nonlinear and non-Gaussian model. It tracks a trajectory with a known model at a given time. It means that particle filter tracks an arbitrary trajectory only if the time instant when a trajectory switches from one model to another model is known apriori. *Because of this reason particle filter is not able to track any arbitrary trajectory where a transition instant from one model to another model is not known. For real world application, a trajectory is always random in nature and may follow more than one model. In this paper we propose a novel method, which overcomes both the above problems. In the proposed method a multiple model based approach is used along with the particle filtering, which automates the model selection process for tracking an arbitrary trajectory. In the proposed approach, there is no need to have apriori information about the exact model that a target may follow. For data association, the uncertainty about the origin of an observation is overcome by using a centroid of measurements to evaluate weights for particles as well as to calculate the likelihood of a model.*

1. Introduction

The standard Kalman filter gives an optimal estimate with a linear and Gaussian model assumption. For nonlinear case, one typically uses the extended Kalman filter. The unscented Kalman filter [17] is being used to preserve the nonlinearity of the model by properly chosen sigma points. It also assumes the state to be Gaussian distributed. In recent times, for nonlinear and non-Gaussian models, particle filtering has been proposed as an alternative to the extended Kalman filter [12, 8, 10, 11]. Comparison of particle filter with other nonlinear filters can be found in [1, 20]. Particle filtering has been extended to multiple target tracking and different methods have been proposed for this problem [6, 4, 18, 5]. In [6] a stochastic simulation Bayesian method has been proposed for multitarget tracking but simulation and results are depicted for single target only. In [4] the data association problem is treated as an incomplete data problem [7]. It treats an observation to track assignment as missing data and Gibbs sampler method is used to estimate the assignment probability. Particles are sampled from the probability density function (pdf) representing the combined state of all the targets in [4]. In multiple target scenario the number of state parameters varies from target to target. Moreover, the computational complexity of this method increases exponentially as number of measurements increase, and number of targets to be tracked increase. Estimating the joint probability distribution of the state of all targets makes the problem intractable in practice. Particle filter, of course, needs the knowledge of the model to track a target; but more importantly, it needs to know the time instant when a trajectory switches from one model to another model. Now, if the target movement is random, then the trajectory formed by the target is arbitrary and there is no apriori knowledge about which model to use at a given time and when to switch. In such a situation, a particle filter suffers from the degeneracy problem, and the pdf of the state collapses. To track an arbitrary trajectory, it is incumbent to use a multiple model based approach, namely, interacting multiple model (IMM) filtering.

In this paper we propose a novel method that works with multiple nonlinear or non-Gaussian state space models to track arbitrary trajectories. *It is important to note that the proposed approach does not require any apriori information about the exact models which targets may follow for particle filtering. We performed the simulations where trajectories are generated using B-spline function and tracked successfully using the proposed method.* In the proposed method an IMM [21, 19] based approach is used along with particle filtering, which automates the model selection process. To incorporate multiple models for a given target the likelihood of an observation given a target state is modelled as a mixture pdf. Another problem with multiple target tracking in the presence of clutter is data association. Various data association methods like nearest neighbor (NN), joint probabilistic data association filter (JPDA) and multiple hypothesis tracking (MHT) have been described in the literature [22]. The performance of NN method degrades in the presence of dense clutter, whereas JPDA and MHT methods are computationally expensive. So in the proposed method probabilistic MHT (PMHT) based approach is used for data association. PMHT algorithm [14, 13] has been proposed to avoid the uncertainty about the origin of a measurement. It uses a centroid of measurements to evaluate state vector of a target. In the proposed approach, we have used a centroid of the measurements to evaluate likelihood of the model, which is required for mixing state vectors from different models in IMM based approach. The centroid is also used to update the particles.

2. Multiple Model based Particle Filtering

It is important to note that the proposed method does not need to have any apriori information about the exact dynamic models which target may follow at a given time. This approach is completely different compared to conventional particle filtering. The conventional particle filters needs the knowledge of the model to track a target. In this section, the problem is described in the multimodel framework to track both maneuvering and non-maneuvering targets. For each model, the pdf is approximated by a set of samples, called particles. Each particle is assigned a weight, known as importance weight. For every model, particle weights [2, 15, 3] are evaluated at each time instant independently. If the trajectory does not follow any model at a given time instant its pdf may collapse or all importance weights may have negligible value for respective particles. At this time instant, particles are initialized using mix state vector given by the IMM filtering method and hence, it is possible to follow an arbitrary trajectory. IMM filtering mixes the state vector from different models using model probabilities. When a trajectory switches from one model to another, particle weights have marginal values if it matches a model and hence, it is reflected in model probability. Mix state vector takes care of the likelihood of a model for a given trajectory. Model probability is calculated using the centroid of observations. Inclusion of IMM based approach allows us to track an arbitrary trajectory with different models. For observation to track association, PMHT based approach is used which is described in [14].

For particle filter, a set of weighted particles are drawn from the posterior pdf of the state. The pdf can be approximated using discrete sums in place of integrals as follows:

$$p(x_t|Y_{1:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_t^i}(dx_t) \tag{1}$$

where $Y_{1:t} = \{y_1, y_2, \ldots, y_t\}$ is a set of measurements up to time t and y_t is a measurement available at time t. x_t^i $(1 \leq i \leq N)$ represents i^{th} sample drawn from pdf at time t. Here, N is the total number of samples used to represent pdf and $\delta_{x_t^i}$ is the Dirac delta function. Based on this approximation, any moment can be evaluated [9]. It can be written as

$$E(g_t(x_t)) = \int g_t(x_t)p(x_t|Y_{1:t})dx_t \approx \frac{1}{N}\sum_{i=1}^{N} g_t(x_t^i) \tag{2}$$

The particles x_t^i are assumed to be independent and identically distributed. As $N \rightarrow \infty$ an estimation converges to its true value [15]. Generally, it is difficult to sample from the posterior pdf. But it is easy to sample from the proposal distribution function $q(x_t|Y_{1:t})$. There are various method for sampling from the proposal function. Sequential importance sampling (SIS) is one of these techniques. Each particle is weighted by an importance weight and it is given by,

$$w_t(x_t) = \frac{p(y_t|x_t)p(x_t|Y_{1:t-1})}{q(x_t|Y_{1:t})} = \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|Y_{1:t})} \tag{3}$$

The proposal function should be chosen to minimize the variance of the importance weights [2]. The most popular choice for proposal function [12] is

$$q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}) \tag{4}$$

The problem with the above choice is that the most recent measurement is not incorporated but it is very easy to implement. This simplifies the evaluation of weights $w_t(x_t)$ and written as

$$w_t(x_t) = p(y_t|x_t) \tag{5}$$

and it can be shown that expectation in (2) can be written as,

$$E(g_t(x_t)) = \sum_{i=1}^{N} g_t(x_t^i)\tilde{w}_t(x_t^i) \tag{6}$$

where $\tilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^{N} w_t^j}$ is normalized weight. The major problem with the above technique is that the variance of the importance weights increases over time. It results into degeneracy phenomenon. To overcome this degeneracy problem a resampling is performed to eliminate the particles with low weights and multiply particles with high weights. There are number of resampling methods: sampling importance resampling (SIR), residual resampling and minimum variance sampling. In our proposed method, residual resampling method is used because it is computationally less expensive and the variance is smaller than that given by SIR method.

For our algorithm, $\mathcal{Y}$ and $\mathcal{X}$ denote the observation process and the state process respectively. Y^t is a set of all observation set for time $t \geq 1$, where t is current time. $Y(t)$ and $X(t)$ represent the realization of observation process and state process at time t, respectively. At time t, a vector of measurements is received,

$Y(t) = (y_t(1), \ldots . y_t(m_t))$, where m_t represents the number of measurements received. Similarly, $X(t) = (x_t(1), \ldots, x_t(N_t))$. Here, N_t is the total number of targets at time instant t and $x_t(s)$ $(1 \leq s \leq N_t)$ represents the combined state estimate for target s. $x_t^m(s)$ is the state estimate of target s due to model m at time t, where $1 \leq m \leq M$. M is the total number of models used to track a particular target.

To overcome the uncertainty about the measurement origin, an assignment process $\mathcal{K}$ is used and K^t is a set of all its realization for time $t \geq 1$. Its realization at time t is denoted by, $K(t) = (k_t(1), \ldots, k_t(m_t))$ where $K(t)$ is an assignment vector and each element of vector $k_t(j) = s$ indicates that target s produces measurement j at time t. The measurement to track assignment probability Π at time t is given by, $\Pi(t) = (\pi_t(1), \ldots, \pi_t(N_t))$. Here, $\pi_t(s)$ indicates the probability that a measurement originates from the target s. This probability is independent of the measurement, i.e.,

$$\pi_t(s) = p(k_t(j) = s), \quad \forall j = 1, \ldots, m_t \tag{7}$$

It is assumed that one measurement originates from one target or clutter, which leads to the following constraint on an assignment probabilities, $\sum_{s=1}^{N_t} \pi_t(s) = 1$. Each element of assignment vector $K(t)$ is assumed to be independent of each other. The sequence of the steps for the proposed method are described in a Figure 1. With this problem formulation, the proposed algorithm, automated model selection based tracking using particle filter is described as follows:

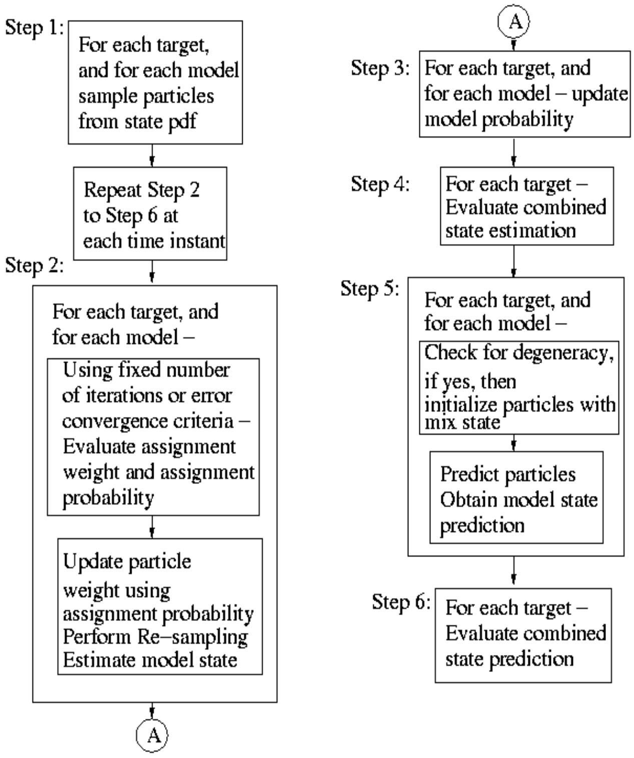


Figure 1: Flow chart for the proposed method

1. Initialize particles for each model m $(1 \leq m \leq M)$, by drawing samples x^i $(i = 1, \ldots, N)$ from the prior $p_m(x_0)$ for each target s $(1 \leq s \leq N_t)$. Initialize model probability (for example $M = 2$ then)
$$\mu = \{0.5 \;\; 0.5\}$$
and transition probability
$$[\xi] = \begin{bmatrix} 0.998 & 0.002 \\ 0.002 & 0.998 \end{bmatrix}.$$
Here, N_t represents the total number of targets at time t.

2. For time $t = 1, 2, \ldots$

   (a) Update particle weights:
   For each target s and for each filter model m, initialize the assignment probabilities $\pi_t(s)$ and repeat the following steps (i)-(v) during each iteration, till error converges to a fixed threshold value, i.e. $\|\hat{x}^{m(p-1)}(s) - \hat{x}^{m(p)}(s)\| < \epsilon$. Initially $\hat{x}^m = x^m$ where x^m is the predicted state vector for the model m at previous time.

      i. Evaluate the likelihood for an observation falling inside the validation region formed using the predicted state vector with respect to the model m, i.e. $p_m(y_t(j)|\hat{x}^{m(p)})$. Here, $x_t^{m(p)}$ represents the state vector of a model m at iteration p.

      ii. Calculate the assignment weights for each measurement $j = 1, \ldots, m_t$, and for each target $i = 1, \ldots, N_t$

$$\hat{z}_{t,j}^{(p+1)}(i) = \frac{\pi^{(p)}(i) p(y_t(j)|\hat{x}^{(p)}(i))}{\sum_{n=1}^{N_t} \pi^{(p)}(n) p(y_t(j)|\hat{x}^{(p)}(n))} \tag{8}$$

      where $p(y_t(j)|\hat{x}^{(p)}(n))$ is a mixture probability of an observation given the combined state estimate x of the target n, and it is given by

$$p(y_t(j)|\hat{x}^{(p)}(n)) = \sum_{m=1}^{M} \mu^m p_m(y_t(j)|\hat{x}^{m(p)})$$

      Here, μ^m is a model probability which is described later.

      iii. Calculate the assignment probabilities for target s

$$\pi_t^{(p+1)}(s) = \frac{1}{m_t}\sum_{j=1}^{m_t} \hat{z}_{t,j}^{(p+1)}(s) \quad (9)$$

iv. Calculate the centroid of measurements

$$y_t^{cm}(s) = \frac{1}{m_t \pi_t^{(p+1)}(s)} \sum_{j=1}^{m_t} \hat{z}_{t,j}^{(p+1)}(s) y_t(j) \quad (10)$$

Using a centroid y_t^{cm} evaluate the likelihood of model for target s

$$\mathcal{L}_s(m)^1 = p_m(y_t^{cm}|x_t^m)$$

v. Using an observation centroid and the state vector of a model particles weights are calculated

$$\hat{w}_t^i = \pi_t^{(p+1)}(s) p_m(y_t^{cm}|\hat{x}_t^{im(p)})$$

If $\sum_{j=1}^N \hat{w}_t^j$ equals zero then go to next model otherwise obtain the estimation $E(\hat{x}_t^m)$ for the model m for next iteration using (6).

vi. Particle weights are updated using an observation centroid and the state vector of a model ($\hat{x}^m = x^m$) which is evaluated at the end of iteration by,

$$w_t^i = \pi_t(s) p_m(y_t^{cm}|x_t^{im})$$

- Normalize the weights.
- Perform Residual Resampling to obtain N particles distributed according to $p_m(x_t^m|Y_{1:t})$.
- Obtain an estimation $E(x_t^m)$ for the model m (in our case mean of a state) using (2).

(b) Propagate particles: For each target s ($1 \leq s \leq N_t$),

i. Update the model probability $m = 1, \ldots, M$:

$$\mu_t^m = \frac{\mu_{t|t-1}^m \mathcal{L}^m}{\sum_i \mu_{t|t-1}^i \mathcal{L}^i}$$

where $\mu_{t|t-1}^m = \sum_{i=1}^M \xi_{im} \mu_{t-1}^i$

If $\mathcal{L}^m$ is negligible value (as it is calculated using the centroid only), initialize it with equal probability value with an assumption that the centroid has equal likelihood with each model m.

ii. Combined the state update for a target s:

$$x_t(s) = \sum_{m=1}^M x_{t|t}^m \mu_t^m$$

iii. Mix state initialization for a model m:
$x_s^{0m} = \sum_{i=1}^M x_{t|t}^i \mu^{i|m}$
where $\mu^{i|m} = \xi_{im}\mu_t^i/\mu_{t+1|t}^m$ and $\mu_{t+1|t}^m = \sum_{i=1}^M \xi_{im}\mu_t^i$
If $\sum_{j=1}^N w_t^j$ equals zero for a model m then initialize the particles with mix state x_s^{0m} otherwise go to next step.

iv. Predict particles:
For each model $m(1 \leq m \leq M)$; draw a process noise sample v_t^i from a pdf $p(v_t)$ and propagate particle x^i by
$x_{t+1}^i = f(x_t^i) + v_t^i \ (1 \leq i \leq N)$

v. Obtain the predicted state $E(x_{t+1}^m)$ for a model m using (2).

vi. Combined the state prediction $x_{t+1}(s)$ for a target s.

3. Simulation Results

Synthetic IR images were generated using real time temperature data[16]. For simulation, the generated frame size is 1024×256 and very high target movement of ± 20 pixels per frame. Maneuvering trajectories are generated using B-Spline function. It is important to note that these generated trajectories do not follow any specific model. In our simulations, we have used constant acceleration (CA) and Singers' maneuver model (SMM) for IMM. For all trajectories, filters are initialized using positions of the targets in the first two frames. For our simulations, the number of particles used to represent the target state pdf is set to 200. Figure 2 depicts the result of tracking using our proposed algorithm for ir44 clip with 0.01% clutter level. In the clip two targets are very closely spaced. Similar results for ir49 and ir50 clips with 0.03% clutter are shown in figures 3 and 4 respectively. In these figures true trajectories are represented by solid line and predicted trajectories are depicted by dotted line. In the figures clutter is represented using white dot. 0.03% clutter level represents the number of noisy pixels in an image frame. It gives on an average one clutter inside the validation gate. The model probability plots for the trajectories 1 and 2 for ir50 clip with 0.03% clutter level are shown in Figures 5-(a) and 5-(b) respectively. Table A depicts mean prediction error in position for each trajectory in different clips without clutter and with clutter. The key point in the proposed method is that during tracking the time instant when transition from one model to another model takes place is not known and is random in nature, and there

[1]Note: Likelihood of a model is calculated during first iteration only for given model and target.

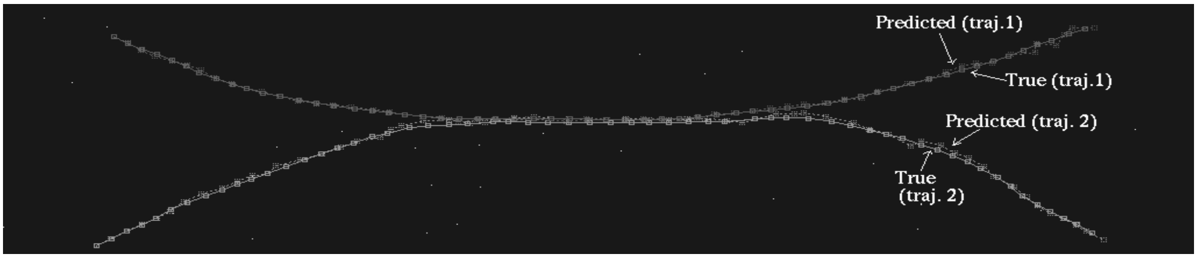


Figure 2: Tracked trajectories at frame number 57 - ir44 clip (0.01% clutter)

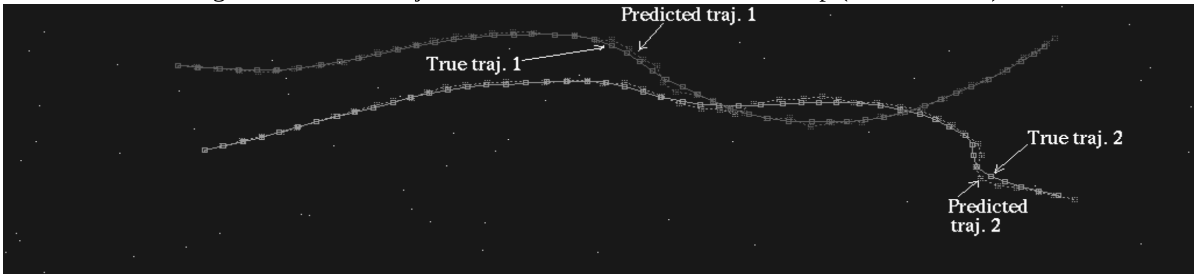


Figure 3: Tracked trajectories at frame number 48 - ir49 clip (0.03% clutter)

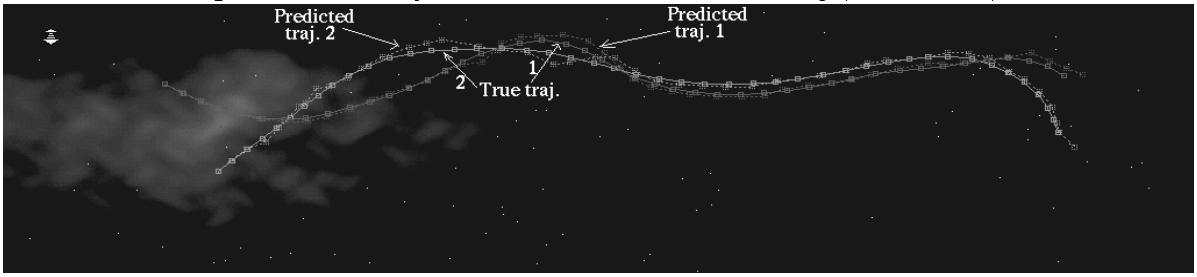
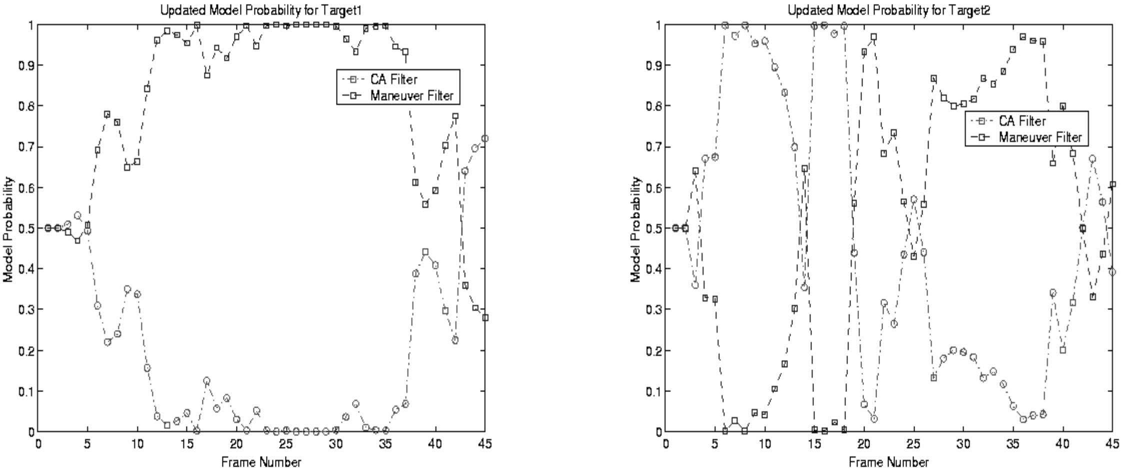


Figure 4: Tracked trajectories at frame number 44 - ir50 clip (0.03% clutter)



(a) (b)

Figure 5: Model Probability plot for target 1 (a) and for target 2 (b) in ir50 clip with 0.03% clutter

is no apriori information about the model that target obeys. From our extensive simulations, we have also noted that the application of a single model, either CA or SMM, with particle filtering for tracking fails.

Table A: Mean Prediction Error in Position.

Traj.	without clutter	with 0.03% clutter
ir44 clip		
1	0.7797	1.3119
2	0.7265	1.8121
ir49 clip		
1	0.8568	0.9441
2	1.0127	1.0073
ir50 clip		
1	0.6755	0.9609
2	0.6286	1.7737

4. Conclusion

From simulation results it is concluded that in absence of any apriori information about the exact dynamic models which targets may follow at a given time, our proposed method is able to track multiple arbitrary target trajectories in the presence of dense clutter. Only two filters, namely, CA and SMM filters were used in IMM mode to track random movement of targets.

References

[1] A. Farina, B. Ristic and D. Benvenuti. Tracking a Ballistic Target: Comparison of Several Nonlinear Filters. *IEEE Transactions on Aerospace and Electronics Systems*, 38(3):854–867, July 2002.

[2] Arnaud Doucet et. al. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Technical Report CUED/F-INFENG/TR 310*, 1998.

[3] Arnaud Doucet, Nando de Freitas and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[4] C. Hue, J-P. Le Cadre and P. Pérez. Tracking Multiple Objects with Particle Filtering. *IEEE Transactions on Aerospace and Electronics Systems*, 38(3):791–812, July 2002.

[5] Cody Kwok, Dieter Fox and Marina Meila. Real-time Particle Filters. *Advances in Neural Information Processing System (NIPS)*, 2002.

[6] D. Avitzour. Stochastic simulation Bayesian approach to multitarget tracking. *IEE Proc. - Radar, Sonar Navig.*, 142(2):41–44, April 1995.

[7] H. Gauvrit, J. P Le Cadre and C. Jauffret. A Formulation of Multitarget Tracking as an Incomplete Data Problem. *IEEE Trans. on Aerospace and Electronic Systems*, 33(4):1242–1257, Oct. 1997.

[8] J. Carpenter, P. Clifford and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proc. - Radar, Sonar Navig.*, 146(1):2–7, Feb. 1999.

[9] J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-sate nonlinear filtering. *International Journal of Control*, 9(5):547–559, 1969.

[10] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.

[11] Neil Gordon. A Hybrid Bootstrap Filter for Target Tracking in Clutter. *IEEE Trans. on Aerospace and Electronics Systems*, 33(1):353–358, Jan. 1997.

[12] N.J. Gordon, D.J. Salmond and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.

[13] Peter Willett, Yanhua Ruan and Roy Streit. The PMHT for Maneuvering Targets. In *Proc. of SPIE Signal and Data Processing of Small Targets*, volume 3373, pages 416–427, July 1998.

[14] Roy L. Streit. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proceeding of SPIE Signal and Data Processing of Small Targets*, volume 2235, pages 394–405, July 1994.

[15] Rudolph van der Merwe, Arnaud Doucet, et. al. THE UNSCENTED PARTICLE FILTER. *www.ece.ogi.edu/ rvdmerwe - Technical Report CUED/F-INFENG TR 380*, Aug. 2000.

[16] Shankar T. More, Avinash A. Pandit, S.N. Merchant and U.B. Desai. Synthetic IR Scene Simulation of Air-borne Targets. In *Proceedings of 3rd Conference ICVGIP 2002*, pages 108–113, Ahmedabad, India, Dec. 2002.

[17] Simon J. Julier. A Skewed Approach to Filtering. In *Proceeding of SPIE, Signal and Data Processing of Small Targets*, volume 3373, pages 271–282, April 1998.

[18] Simon Maskell, Malcom Rollason and Neil Gordon. Efficient Particle Filtering for Multiple Target Tracking with Application to Tracking in Structured Images. *SPIE - Signal and Data Processing of Small Targets*, 4728, April 2002.

[19] T. Kirubarajan, Murali Yeddanapudi, Yaakov Bar-Shalom and Krishna Pattipai. Comparison of IMMPDA and IMM-Assignment algorithms on Real Traffic Surveillance Data. In *Proc. of SPIE Signal and Data Processing of Small Targets*, volume 2759, pages 453–464, May 1996.

[20] X. Lin, T. Kirubarajan, Y. Bar-Shalom and Simon Maskell. Comparison of EKF, Pseudomeasurement and Particle Filters for a Bearing-only Target Tracking Problem. *http://www-sigproc.eng.cam.ac.uk/ sm224/comparison.pdf*, April 2002.

[21] X. Rong Li, Youmin Zhang. Numerically Roubst Implementation of Multiple-Model Algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):266–277, Jan. 2000.

[22] Y. Bar-shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1989.