Data Association for the PHD Filter

# Daniel E. Clark [1], Judith Bell [2]

Electrical and Computer Engineering, Heriot-Watt University,
Riccarton, Edinburgh. $\{^1$*dec1*,^{2}*j.bell*$\}$*@hw.ac.uk*

Abstract

The Probability Hypothesis Density (PHD) filter was developed as a suboptimal method for tracking a time varying number of targets. The first order statistical moment of the multiple target posterior distribution, called the Probability Hypothesis Density, gives the expected locations of the targets. This property is used instead of the full multi-target posterior distribution as it requires significantly less computation. Particle filter implementations have demonstrated the potential of the algorithm for real-time tracking applications.

One of the main criticisms of the PHD filter is that there is no means of associating the same target between frames. Whilst this may be of advantage if the main concern is where the targets are, it is a major drawback if it is necessary to identify the trajectories of the different targets. Novel techniques for solving the problem of track continuity are presented here and demonstrated on simulated data.

1. Introduction

The multiple target tracking problem is to estimate the positions of an unknown number of targets, based on observations of the targets corrupted by noise, with the possibility that observations may be false alarms due to clutter. The usual method for solving this problem is to assign a single target stochastic filter such as a Kalman filter or extended Kalman filter to each target and use a data association technique to assign the correct measurement to each filter [1]. These filters have various drawbacks, however, as they rely on either assuming that the dynamic and observation models are linear in the case of the Kalman filter or can be linearised in the case of the extended Kalman filter. The extended Kalman filter also requires derivation of Jacobian matrices which in many cases is non-trivial. Another approach to non-linear filtering is the Unscented Kalman filter [2], which provides an interesting alternative to the extended Kalman filter by linearising the filter equation instead of the model.

Particle filtering methods have received a great deal of attention for target tracking recently; instead of trying to compute the analytic solution to a Stochastic Differential Equation, they represent the probability distribution by a set of discrete samples and use Sequential Monte Carlo methods for Bayesian inference to predict and update target estimates based on the observations [3].

Particle filter approaches to multiple target tracking have continued to use data association techniques but they often rely on knowing the number of targets *a-priori* [4] [5]. Schulz *et al.*[6] used a sample-based JPDAF (Joint Probabilistic Data Association Filter) for people tracking, which is similar to the Kalman filter technique with the JPDAF commonly used except that the Kalman filter for each target is replaced by a particle filter. It also provides an estimate of the target state but is computationally expensive.

Bayesian approaches can be used for representing multiple target posteriors, but the computational time required grows exponentially with the number of targets. A practical alternative to Bayesian multiple target tracking was devised by Mahler [7], by propagating the first-order statistical moment, or Probability Hypothesis Density (PHD), instead of the multiple target posterior itself.

Particle filter methods for the PHD-filter have been devised by Vo *et al.* [8] and Zajic *et al.* [9]. Practical applications of the filter include tracking vehicles in different terrains [10], tracking targets in passive radar located on an ellipse [11] and tracking a variable number of targets in forward scan sonar [12].

The advantage of the PHD filter is that it has the ability to track a variable number of targets, estimating both the number of targets and their locations. It avoids the need for data association techniques as the identities of the individual targets are not required. Whilst this may be of advantage if the main concern is where the targets are, it is a major drawback if it is necessary to identify the trajectories of the different targets.

A couple of different methods for associating the targets between frames have been reported in the literature. The first of these, by Panta *et al.* [13], used the PHD filter for pre-filtering the data as input to a with Multiple Hypothesis Tracker, which is known to have a high time and space complexity. The other technique, by Lin [14], associated the estimated targets between frames. Novel techniques are presented in this paper and demonstrated on simulated data.

2. Multiple Target Filtering

The multiple target filtering problem is to estimate the positions of an unknown varying number of targets, based on observations which may also include false alarms due to erroneous measurements. In standard tracking applications, this is usually solved by assigning a single target stochastic filter, such as a Kalman filter, to each target and managing the different tracks with a data association technique.

We now present the multiple tracking model used in the paper. Define two random-set stochastic processes $X = \{X_t :$

0-7803-9399-6/05/$20.00 © 2005 IEEE

$t \in \mathbb{N}\}$ and $Y = \{Y_t : t \in \mathbb{N}\backslash\{0\}\}$, where process X is called the *state* process and process Y is called the *observation* process.

The multiple target state at time t is represented by random-set $X_t = \{x_{t,1},...,x_{t,T_t}\}$, where $x_{t,i}$ represents the state of an individual target and T_t is the number of targets at time t. The multiple target measurement at time t is given by $Z_t = \{z_{t,1},...,x_{z,m_t}\}$, where $z_{t,j}$ represents a single target measurement or false alarm and m_t is the number of observations at time t.

The filtering problem is then to estimate the unobserved signal process $X_{0:t} = \{X_0,...,X_t\}$ based on observations $Z_{1:t} = \{Z_1,...,Z_t\}$, i.e. to obtain $\hat{X}_t = \{\hat{x}_{t,1},...,\hat{x}_{t,\hat{T}_t}\}$, where $\hat{x}_{t,i}$ are the individual target estimates and $\hat{T}_t$ is the estimate of the number of targets at time t.

Analogous equations to single-target Bayesian filtering can be found for the multiple-target model by using Finite Set Statistics [7].

3. The PHD Filter

The Probability Hypothesis Density (PHD) is the first moment of the multiple target posterior distribution [7]. The PHD represents the expectation, the integral of which in any region of the state space is the expected number of objects in that region. The PHD is estimated instead of the multiple target posterior distribution as it is much less computationally expensive to do so. The time required for calculating joint multi-target likelihoods grows exponentially with the number of targets and is thus not very practical for sequential target estimation as this may need to be undertaken in real time. The model used here only calculates single target likelihoods and so is a significant improvement on explicitly calculating joint multi-target likelihoods[15].

The estimated object states can be detected as peaks of the PHD distribution. The derivation for the PHD equations is provided by Mahler [7]. The prediction and update equations are given by:

$$\begin{aligned} &D_{t|t-1}(x|Z_{1:t-1}) \\ &= \gamma_t(x) + \int \phi_{t|t-1}(x,x_{t-1}) D_{t-1|t-1}(x_{t-1}|Z_{1:t-1}) dx_{t-1}, \end{aligned} \tag{1}$$

$$\begin{aligned} &D_{t|t}(x|Z_{1:t}) \\ &= \left[\nu(x) + \textstyle\sum_{z\in Z_t} \frac{\psi_{t,z}(x)}{\kappa_t(z)+\langle D_{t|t-1},\psi_{t,z}\rangle}\right] D_{t|t-1}(x|Z_{1:t-1}), \end{aligned} \tag{2}$$

where $\phi_{t|t-1}(x,x_{t-1}) = P_S(x_{t-1}) f_{t|t-1}(x|x_{t-1}) + b_{t|t-1}(x|x_{t-1}), \nu(x) = 1 - P_D(x)$, $\kappa_t(z) = \lambda_t c_t(z)$ and $\psi_{t,z} = P_D(x) g(z|x)$.

In the prediction equation, γ_t is the PHD for spontaneous birth of a new target at time t, $b_{t|t-1}$ is the PHD of a spawned target, P_S is the probability of target survival and $f_{t|t-1}(x_t|x_{t-1})$ is the single target motion distribution. In the data update equation, g is the single target likelihood function, P_D is the probability of detection, λ_t is the Poisson parameter specifying the expected number of false alarms and c_t is the probability distribution over the state space of clutter points. The $\langle .,. \rangle$ notation is defined as the inner product $\langle D_{t|t}, \varphi\rangle = \int D_{t|t}(x_t|Z_{1:t})\varphi(x_t)dx_t$.

A. PHD Particle Filter Algorithm

The implementation of the PHD Particle filter used is an adaptation of the method described by Vo *et al* [8] based on a sequential Monte Carlo algorithm for multi-target tracking. The algorithm can be informally described by the following stages: In the initialisation stage, particles are uniformly distributed across the field of view. The particles are propagated in the prediction stage using the dynamic model with added process noise and, in addition, particles are added to allow for incoming targets. When the measurements are received, weights are calculated for the particles based on their likelihoods, which are determined by the distance of the particles to the set of observations, the sum of the weights gives the estimated number of targets. Particles are then resampled from the weighted particle set to give an unweighted representation of the PHD.

The sequential Monte Carlo implementation of the PHD filter is given here. The algorithm is initialised in Step 0 and then iterates through Steps 1 to 4.

Step 0:*Initialisation at t=0*
The filter is initialised with N_0 particles drawn from a prior distribution. The number of particles is adapted at each stage so that it is proportional to the number of targets. Let N be the number of particles per target. The mass associated to each particle is $\hat{T}_0/N$, where $\hat{T}_0$ is the expected initial number of targets (this will be updated after an iteration of the algorithm).
•$\forall i = 1,\dots,N_0$ sample $x_0^{(i)}$ from $D_{0|0}$ and set $t = 1$.

Step 1:*Prediction Step, for $t \geq 0$*
In the prediction step, samples are obtained by importance sampling two proposal densities, q_t and p_t:
•$\forall i = 1,..,N_{t-1}$, sample $\tilde{x}_t^{(i)}$ from a proposal density $q_t(.|x_{t-1}^{(i)}, Z_t)$.
•$\forall i = 1,..,N_{t-1}$, evaluate the predicted weights $\tilde{\omega}_{t|t-1}^{(i)}$:

$$\tilde{\omega}_{t|t-1}^{(i)} = \frac{\phi_{t|t-1}(\tilde{x}_t^{(i)}, x_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, Z_t)} \omega_{t-1}^{(i)}. \tag{3}$$

M new-born particles are also introduced from the spontaneous birth model to detect new targets entering the state space.
•$\forall i = N_{t-1}+1,..,N_{t-1}+M$, sample $\tilde{x}_t^{(i)}$from another proposal density $p_t(.|Z_t)$.
•$\forall i = N_{t-1}+1,..,N_{t-1}+M$, compute the weights of new born particles $\tilde{\omega}_{t|t-1}^{(i)}$:

$$\tilde{\omega}_{t|t-1}^{(i)} = \frac{1}{M} \frac{\gamma_t(\tilde{x}_t^{(i)})}{p_t(\tilde{x}_t^{(i)}|Z_t)}. \tag{4}$$

Step 2:*Update Step, for $t \geq 0$*
After the new measurements are obtained, the weights are recalculated using the likelihood function $g(.|.)$ to update the distribution based on new information:

• Let $R_t = N_{t-1} + M$. $\forall z \in Z_t$, compute:

$$\langle \tilde{\omega}_{t|t-1}, \psi_{t,z} \rangle = \sum_{i=1}^{R_t} \psi_{t,z}(\tilde{x}_t^{(i)}) \tilde{\omega}_{t|t-1}^{(i)}. \tag{5}$$

•$\forall i = 1, \ldots, R_t$, update weights:

$$\tilde{\omega}_t^{(i)} = \left[\nu(\tilde{x}_t^{(i)}) + \sum_{z \in Z_t} \frac{\psi_{t,z}(\tilde{x}_t^{(i)}))}{\kappa_t(z) + \langle \tilde{\omega}_{t|t-1}, \psi_{t,z} \rangle} \right] \tilde{\omega}_{t|t-1}^{(i)}. \tag{6}$$

Step 3:*Resampling Step*

The particles are resampled to obtain an unweighted representation of $D_{t|t}$.

• Compute mass of particles:

$$\hat{T}_t = \sum_{i=1}^{R_t} \tilde{\omega}_t^{(i)}, \tag{7}$$

and set $N_t = N.int(\hat{T}_t)$ (where $int(\hat{T}_t)$ is the nearest integer to $\hat{T}_t$).

• Resample $\left\{ \frac{\tilde{\omega}_t^{(i)}}{\hat{T}_t}, \tilde{x}_t^{(i)} \right\}_{i=1}^{R_t}$ to get $\left\{ \frac{\omega_t^{(i)}}{\hat{T}_t}, x_t^{(i)} \right\}_{i=1}^{N_t}$.

The particles each have weight $\hat{T}_t / N_t$ after resampling.

Step 4:*Target Extraction*

Target state estimates are obtained from the PHD posterior by extracting peaks from the particle distribution.

4. Data Association

The data association problem in multiple target tracking usually involves ensuring that the correct measurement is given to each stochastic filter so that the trajectories of each target can be accurately estimated, this is referred to as measurement to track association. The three main approaches in the literature are the Nearest Neighbour Standard Filter (NNSF), the Joint Probabilistic Data Association Filter (JPDAF) and the Multiple Hypothesis Filter (MHF) [1]. Before describing these, the required terminology is briefly outlined:

Each track i has an associated error covariance $S_{t,i}$ which defines a validation region

$$V_{t,i}(\gamma) := \{z : [z - \hat{z}_{t,i}]^T (S_{t,i})^{-1} [z - \hat{z}_{t,i}] \leq \gamma\}. \tag{8}$$

The predicted measurement $\hat{z}_{t|t-1,i} = H\hat{x}_{t|t-1,i}$ is obtained by projecting the previous estimate using the motion model: $\hat{x}_{t|t-1,i} = F\hat{x}_{t-1,i}$ and then using the observation function H. The difference between the new observation and the predicted measurement is called the innovation: $\nu_{ji} = |z_{t,j} - \hat{z}_{t,i}|$. The set of targets at time t is $X_t = \{x_{t,1}, ..., x_{t,T_t}\}$, and the set of measurements is $Z_t = \{z_{t,1}, ..., z_{t,m_t}\}$.

The Nearest Neighbour Standard Filter simply takes the nearest validated measurement to the predicted measurement for updating each of the target states. This can result in problems as the nearest validated measurement may be the same for two different targets.

The Joint Probabilistic Data Association Filter computes the joint probabilities for all the pairings between the predicted measurements and estimated target states. This technique also has to consider the false alarms from spurious measurements.

The ideal Multiple Hypothesis Filter maintains probabilities of all possible associations at each time step. Unlike the NNSF and JPDAF this does not just consider the probabilities from the previous time step which allows for backtracking. It also allows for track initiation. In practise, it is not feasible to keep track of all possible hypotheses as the time and computational complexity grows exponentially. Techniques for reducing the complexity include gating (ignore measurements outside validated regions), pruning (eliminating low probability hypotheses) and merging (combining hypotheses into a single track).

There have been two previous approaches to data association for the PHD Filter. Panta *et al.* [13] used the PHD filter to remove unlikely measurements before inputting the remaining data to a Multiple Hypothesis Tracker (MHT) filter. The implementation of the MHT filter is complicated and requires a lot of memory to consider all possible tracks and methods for pruning and gating measurements. Lin [14] described a data association technique for doing peak-track association. The peaks extracted as target states in the PHD filter algorithm were matched with tracks by using a global optimisation technique to minimise the cost of association and used a Kalman filter to estimate the state of the target in the next frame. A validation region was used to validate the peaks for the PHD filter as targets. Whilst this is good to increase the signal to noise ratio, this means that the number of targets needs to be known and that the introduction of new targets may not be detected.

The techniques for data association which we will consider here are based on peak-track association. In the PHD filter, estimates of the target locations are given at each time step but track continuity is not maintained. We present two methods to enable identification of the same targets between frames based on the target estimates provided by the PHD filter. These techniques take advantage of the ability of the PHD filter to estimate the number and locations of targets and filter out the clutter. The complexity of these techniques is less than the MHT and JPDA filters because the target states are provided by the PHD filter algorithm, we assume that the false alarms are filtered out by the PHD filter and there is no backtracking.

A. Method 1

The first method for data association is based on the observation that in the discrete representation of the multimodal density, the particles representing one of the modes will tend to track that mode if the motion of the particles is modelled well. In previous implementations of the Particle PHD filter, clustering techniques have been employed to extract the peaks of the PHD distribution. Here we extend this idea to assign labels to the particles based on a partitioning of the data by using a clustering technique.

The method can be informally explained as follows: At each iteration, partition the particle data and give each particle in the same partition the same label. In subsequent iterations, when resampling, give the children of a particle the same label as its parent. After resampling, repartition the data and if the majority of the particles in one partition have the same label then associate these partitions.

The example state model for the particles used here will be the two-dimensional position and velocity vector. If the data is not well partitioned in the position domain then it can be partitioned in the velocity domain, this will help to keep track of targets which cross each other.

Steps 3 and 4 of the PHD filter algorithm given previously are replaced with the following:

Step 3:*Resampling*
The particles are resampled to obtain an unweighted representation of $D_{t|t}$.
- Compute mass of particles:

$$\hat{T}_t = \sum_{i=1}^{R_t} \tilde{\omega}_t^{(i)}, \tag{9}$$

and set $N_t = N.int(\hat{T}_t)$ (where $int(\hat{T}_t)$ is the nearest integer to $\hat{T}_t$).
- Resample $\left\{ \frac{\tilde{\omega}_t^{(i)}}{\hat{T}_t}, \tilde{x}_t^{(i)} \right\}_{i=1}^{R_t}$ to get $\left\{ \frac{\omega_t^{(i)}}{\hat{T}_t}, x_t^{(i)} \right\}_{i=1}^{N_t}$.

The particles each have weight $\hat{T}_t/N$ after resampling.

- At time $t > 1$, create new labels $\tilde{L}$:
If $x_t^{(j)} \in Child(\tilde{x}_t^{(i)})$ then $\tilde{L}(x_t^{(j)}) = L(\tilde{x}_t^{(i)})$.

Step 4:*Target Extraction and Data Association*
- A clustering technique is used to create $\hat{T}_t$ partitions based on the particle location vectors (x,y).

The target position estimates $\hat{x}_{t,1},..,\hat{x}_{t,\hat{T}_t}$ are taken to be the centroids of these partitions: $\{P_{t,1},...,P_{t,\hat{T}_t}\}$ with labels $\{L_{t,1},...,L_{t,\hat{T}_t}\}$ such that $L(x_t^{(i)}) = L_t^k$ if $x_t^{(i)} \in P_{t,k}$.
If the estimates of the target locations are too close then repartition based on the particle velocity vectors $(\dot{x},\dot{y})$.

1) Data Association: We have two sets of partitions of the particles at time t: $\{\tilde{P}_{t,1},\ldots,\tilde{P}_{t,\hat{T}_{t-1}}\}$ and $\{P_{t,1},\ldots,P_{t,\hat{T}_t}\}$, where the first is the partition related to the previous timestep and the second is the new partition. In addition to the particles propagated from the previous time step, we have M new particles to allow for new targets and so that we actually have $\hat{T}_{t-1}+1$ partitions in the first set $\{\tilde{P}_{t,1},\ldots,\tilde{P}_{t,\hat{T}_{t-1}+1}\}$, where the partition $\tilde{P}_{t,\hat{T}_{t-1}+1}$ will be treated separately. We therefore have two labels for each particle $x_t^{(i)}$: $\tilde{L}(x_t^{(i)})$ and $L(x_t^{(i)})$ and from these we create two matrices: $A_{g,h} = \#\{i : x_t^{(i)} \in \tilde{P}_{t,g} \cap P_{t,h}\}$ and $B_{g,h} = \#\{i : Child(\tilde{x}_t^{(i)}) \in \tilde{P}_{t,g} \cap P_{t,h}\}$ (where the notation # represents the number of particles in the set). Matrix A tells us how many particles in each partition at the current time step correspond with partitions in the previous time step. Matrix B is similar although it tells us how many resampled particles correspond.

If the partitioning worked well, then the number of particles around each target should be roughly N, the number allocated per target, so at time t: $\sum_{g=1}^{\hat{T}_{t-1}} A_{g,\beta} \approx N$ and $\sum_{h=1}^{\hat{T}_t} A_{\alpha,h} \approx N$. We should expect that if the target g at time $t-1$ is to be associated with target h at time t, then $A_{g,h} \approx N$. This is what happens if the conditions are ideal but it not always the case as we may also have target births, deaths, clutter and possible mispartioning.

If, for some target g from the previous timestep, $\sum_{h=1}^{\hat{T}_t} A_{g,h} \leq \varepsilon_1 N$, where ε_1 is a threshold defining when there is an insufficient number of particles to represent a target, then target g is destroyed.

In the partition in the first set which represents newly introduced particles, if there is a number of these particles above a given threshold, $\varepsilon_2 N$, which are also in a partition in the second set, then we declare a new born target and create a new column in matrix A with the corresponding number of particles which are in both partitions.

Sometimes, the particles around the new targets are sampled from the clouds of particles around already existing targets; in this instance, there may be two partitions at time t with particles from previous target g. The number of particles for these two targets in matrix A may be similar as the particles will have been resampled to represent two targets and so we need to look in matrix B, which tells us how many particles have been resampled, as the number of different particles resampled for the existing target should be greater than for the new target if the existing target is being tracked well. Conversely, the number of different particles for the new target is expected to be smaller and may be represented by only a few resampled particles, we then define this to be a new target and create a new column for this target in matrix A and the other to be for the already existing target.

We can then use these matrices to create a validation matrix V, such that $V_{g,h} = 1$ if $A_{g,h} \geq \varepsilon_1 N$ and zero otherwise. The data association then proceeds by finding the maximum sum of entries in matrix A for all valid associations.

B. Method 2

At each stage of the PHD filter algorithm, the target states are estimated by clustering the particle data and obtaining means of the clusters. The second method which we present here uses these estimated states and finds the best association between them and the predicted estimate, taken by projecting the previous estimates with the motion model.

The method proceeds as follows: In step 4 of the particle filter algorithm, the estimated target locations are found by clustering the data and taking the mean positions as the estimated random set of state vectors for the targets $\{\hat{x}_{t,1},..,\hat{x}_{t,\hat{T}_t}\}$. Let F be the transition function for the dynamic model which is used in the prediction model for the particles. Then before receiving any measurements, the predicted state vector for estimated target $\hat{x}_{t-1,j}$ at time t is $\hat{x}_{t|t-1,j} := F\hat{x}_{t-1,j}$ and the estimated random set is $\{\hat{x}_{t|t-1,1},..,\hat{x}_{t|t-1,\hat{T}_{t-1}}\}$.

The estimated target locations are known at each time step, the association stage is to identify these locations between time steps so that there is continuity of identity for each target. Thus the data association here does not involve the measurements, only the estimated positions. The advantage of this is firstly that they should give a better estimate than the measurements, secondly that the spurious measurements due to false alarms should have been filtered out and thirdly that the estimated state vectors have the unobservable dimensions such as velocity which allows for better discrimination when the measurement positions are close but velocities are different.

If the particles are distributed according to a multimodal Gaussian distribution, then the EM algorithm [16] can be used to cluster the data, as a Gaussian mixture model which gives not only the mean positions of the clusters but also their covariances matrices, $\{S_{t,1},..S_{t,\hat{T}_t}\}$, which can be used as the estimation error for the targets.

For each of the target estimates at time $t, \{\hat{x}_{t,1},..,\hat{x}_{t,\hat{T}_t}\}$, create a validation region:

$$V_{t,i}(\gamma) := \{x : [x-\hat{x}_{t,i}]^T (S_{t,i})^{-1} [x-\hat{x}_{t,i}] \leq \gamma\}. \qquad (10)$$

It is assumed here that the PHD filter has filtered out the false alarms so that all the estimates are treated as targets and there is only one association per target. The method will associate measurements from the previous timestep, decide that a target has died or declare a new target.

Let $\hat{B}_{t|t-1}$ denote the number of associations made between time steps t and $t-1$, then obviously $\hat{B}_{t|t-1} \leq \min(\hat{T}_{t-1}, \hat{T}_t)$ and the number of new targets is equal to $\hat{T}_t - \hat{B}_{t|t-1}$.

The best association θ is the set of pairs $(j,i) \in \{1,..,\hat{B}_{t|t-1}\} \times \{1,..,\hat{B}_{t|t-1}\}$ which maximises number of associations $\hat{B}_{t|t-1}$ and the sum of likelihoods $\sum\limits_{(j,i)\in\theta} g(\hat{x}_{t|t-1,j}|\hat{x}_{t,i})$ where the predicted estimates are within the validation regions defined above. The estimates for which no association has been made are defined to be new targets.

5. Simulated Examples

We shall now demonstrate the two methods proposed for data association and compare the results for the tracking. Simulated trajectories have been generated with added Gaussian noise to generate measurements. We have used a linear Gaussian model here for simplicity.

1) Example 1: In the first example, we test the two techniques on 4 simulated targets over 20 iterations with an average of 1 false alarm. Figures 1 and 3 show the results for methods 1 and 2 respectively on the same simulated data. Figure 2 shows a snapshot of the segmented particle cloud output, where different colours represent the different targets. The results of the two methods were provided on the same simulated measurements, so the differences in target estimate are due to the PHD filter. Both of the techniques correctly associate the same target between frames from the first iteration. In the early stages of the algorithm, there may be peaks detected from false alarms but these will become fewer over time as the peaks of the PHD track the correct targets.

2) Example 2: Figure 4 shows 4 linear targets, the estimated trajectories and the 2σ covariance rings obtained by fitting the Gaussian mixture model, which were used for validating the estimates and assigning the correct estimate to track. We can see in figure 4 that some of the covariance ring are much larger, this is at the start of the algorithm when the particles are distributed across the state space.

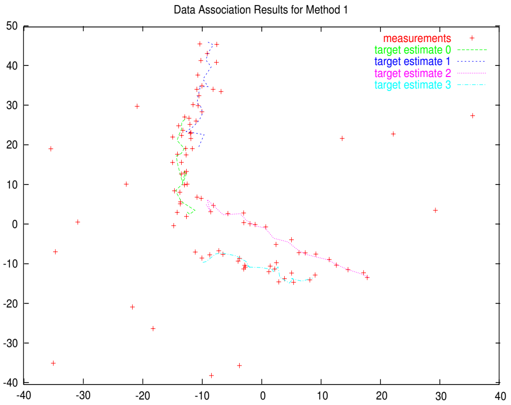

Fig. 1: Method 1: 4 Simulated Targets with Clutter

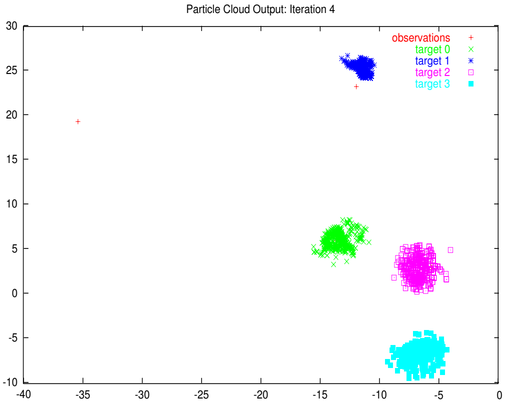

Fig. 2: Method 1: Particle Cloud Output

3) Example 3: Figure 5 uses method 2 and shows 3 targets over 30 iterations and a fourth target which enters in frame 10 and leaves in frame 20 (light blue trajectory), demonstrating the birth and death of a target. 100 extra particles have been added for the birth model at each iteration to allow for incoming targets.

6. Discussion

The PHD filter estimates the position and number of targets but does not associate the target tracks between frames. This paper has proposed two novel methods for incorporating data association into the PHD filter multiple target tracker. These methods are simpler in complexity than other reported techniques and have been shown on simulated data with clutter. The second method has been shown with birth and death of a target, which previous work on peak-track association was unable to provide.

The first of the techniques partitions the data at the target extraction stage into clusters around the individual targets and these partitions are used between the frames to enable track continuity. The second method uses the previous target state estimate and the motion model to estimate the target in the

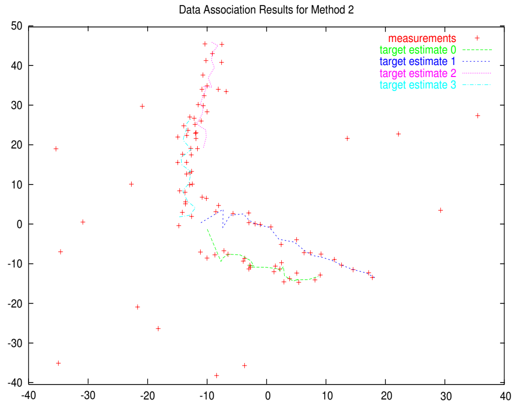

Fig. 3: Method 2: 4 Simulated Targets with Clutter

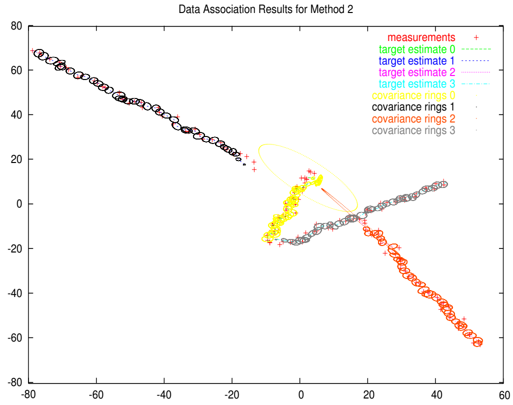

Fig. 4: Method 2: 4 Simulated Targets

next frame and uses a validation procedure taking advantage of the error covariances from the EM-algorithm used to cluster the data.

The second method uses the covariance matrices provided by the EM clustering technique to provide error covariance matrices. One of the problems with using the covariance matrices to create the validation regions is that the cluster of particles around a target may be small due to particle impoverishment. This may be solved by comparing the size of the determinant of the covariance matrix with that of the measurement covariance matrix, this should not be significantly smaller. This clustering technique has a time complexity which is quadratic in the number of targets which may be impractical for a large number of targets.

Further investigation needs to be done to evaluate the ability of the first method for introduction of new targets. New born targets are located using the additional particles provided by the birth-model. To ensure that this technique will work, it will be need to be shown that the new-born particles are in the partition with the new target.

Error metrics for reliably evaluating the two algorithms are currently being investigated. Future work will involve demonstrating the technique on real data.

Acknowledgements

The authors wish to thank QinetiQ for sponsoring this work.

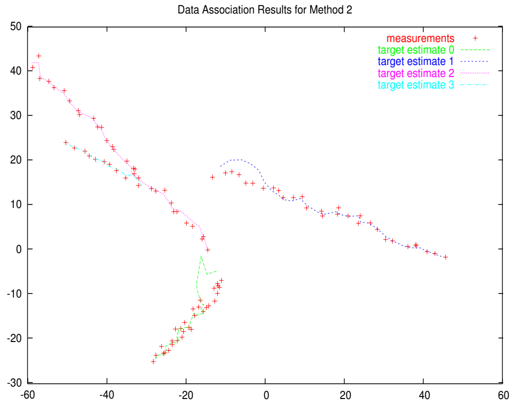

Fig. 5: Method 2: Targets with Birth and Death

References

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

[2] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. *In Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL.*, 1997.

[3] A. Doucet, N. de Freitas, and N. Gordon. Sequential Monte Carlo Methods in Practice. 2001.

[4] C. Hue, J.P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *IEEE Trans. on Aerospace and Electronic Systems*, 38(3):791–812, 2002.

[5] R. Karlsson and F. Gustafsson. Monte Carlo data association for multiple target tracking. *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, 1:13/1– 13/5, 2001.

[6] D. Schulz, W Burgard, D. Fox, and A. B. Cremers. People tracking with a mobile robot using sample-based Joint Probabilistic Data Association Filters. *International Journal of Robotics Research*, pages 99–116, 2003.

[7] R. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39, No.4:1152–1178, 2003.

[8] B.N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo Implementation of the PHD filter for Multi-target Tracking. *Proc. FUSION 2003*, pages 792–799, 2003.

[9] T. Zajic and R. Mahler. A particle-systems implementation of the PHD multitarget tracking filter. *SPIE Vol. 5096 Signal Processing, Sensor Fusion and Target Recognition*, pages 291–299, 2003.

[10] H. Sidenbladh. Multi-target particle filtering for the Probability Hypothesis Density. *International Conference on Information Fusion*, pages 800–806, 2003.

[11] M. Tobias and A.D. Lanterman. A Probability Hypothesis Density-based multitarget tracker using multiple bistatic range and velocity measurements. *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on , March 14-16, 2004*, pages 205–209, 2004.

[12] D.E. Clark and J. Bell. Bayesian Multiple Target Tracking in Forward Scan Sonar Images Using the PHD Filter. *to appear: IEE Radar, Sonar and Navigation*, 2005.

[13] K. Panta, B. Vo, S. Singh, and A. Doucet. Probability hypothesis density filter versus multiple hypothesis tracking. *Proceedings of SPIE – Volume 5429 Signal Processing, Sensor Fusion, and Target Recognition XIII, Ivan Kadar, Editor, August 2004, pp. 284-295.*

[14] Lin Lin. Parameter estimation and data association for multitarget tracking. *PhD Thesis, The University of Connecticut*, 2004.

[15] D.L. Hall and J. Llinas, editors. *Handbook of Multisensor Data Fusion*, chapter 7. CRC Press, 2001.

[16] C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/˜bouman, April 1997.