Multiple Object Tracking Using Evolutionary MCMC-Based Particle Algorithms

François Septier, Avishy Carmi, Sze Kim Pang, Simon Godsill

Multiple Object Tracking Using Evolutionary MCMC-Based Particle Algorithms

F. Septier, A. Carmi, S. K. Pang and S. J. Godsill

Signal Processing Laboratory, University of Cambridge, UK
(e-mail : {fjms2,ac599,skp31,sjg30}@cam.ac.uk)

Abstract: Algorithms are presented for detection and tracking of multiple clusters of coordinated targets. Based on a Markov chain Monte Carlo sampling mechanization, the new algorithms maintain a discrete approximation of the filtering density of the clusters' state. The filters' tracking efficiency is enhanced by incorporating various sampling improvement strategies into the basic Metropolis-Hastings scheme. Thus, an evolutionary stage consisting of two primary steps is introduced: 1) producing a population of different chain realizations, and 2) exchanging genetic material between samples in this population. The performance of the resulting evolutionary filtering algorithms is demonstrated in two different settings. In the first, both group and target properties are estimated whereas in the second, which consists of a very large number of targets, only the clustering structure is maintained.

Keywords: Monte Carlo method; Genetic algorithms; Estimation algorithms; Recursive estimation; Multitarget tracking

1. INTRODUCTION

The purpose of multiple object tracking algorithms is to detect, track and identify targets and/or group of targets from sequences of noisy observations provided by one or more sensors. The difficulty of this problem has increased as sensor systems in the modern battlefield are required to detect and track objects in very low probability of detection and in hostile environments with heavy clutter. A common assumption in the target tracking literature is that each target moves independently of all others. However, in practice, this is not always true as targets may move in a common formation; for example, a group of aircraft moving in a tight formation or a convoy of vehicles moving along a road. If the dependencies of the group objects can be exploited, it can potentially lead to better detection and tracking performances, especially in hostile environments with high noise and low detection probabilities.

In recent years, sequential Monte Carlo (SMC) methods were applied for various nonlinear filtering problems. These methods otherwise known as particle filters (PF) exploit numerical representation techniques for approximating the filtering probability density function of inherently nonlinear non-Gaussian systems. Using these methods, the obtained estimates can be set arbitrarily close to the optimal solution (in the Bayesian sense) at the expense of computational complexity. Due to their sampling mechanization, PFs tend to be inefficient when applied to high-dimensional problems such as multi-target tracking.

Markov chain Monte Carlo (MCMC) methods are generally more effective than PFs in high-dimensional spaces. Their traditional formulation, however, allows sampling from probability distributions in a non-sequential fashion. Recently, sequential MCMC schemes were proposed by Berzuini et al. [1997], Khan et al. [2005], Golightly and Wilkinson [2006], Pang et al. [2008]. In Berzuini et al. [1997], a sequential MCMC algorithm was designed to do inference in dynamical models using a series of Metropolis-Hastings-within-Gibbs. A similar idea was applied in Golightly and Wilkinson [2006] for imputing missing data from nonlinear diffusion. In Khan et al. [2005], a MCMC-Particles algorithm was proposed using a numerical integration of the predictive density but unfortunately its computational demand can become excessive as the number of particles increases owing to its direct Monte Carlo calculation of the filtering density at each time step. In Pang et al. [2008], a MCMC particles algorithm was designed for tracking multiple coordinated target groups. The approach adopted in Pang et al. [2008] is distinct from the Resample-Move scheme in Gilks and Berzuini [2001] where MCMC steps are used to rejuvenate degenerate samples.

The algorithms presented here are partially based on the method in Pang et al. [2008]. In this work, however, the efficiency of the MCMC particles algorithm is enhanced by incorporating various sampling improvement strategies into the basic Metropolis-Hastings scheme. In particular, notions from genetic algorithms and simulated annealing are considered. The performance of the newly derived algorithms is demonstrated in two complex multi-target scenarios.

2. MCMC-BASED PARTICLE FILTERING

In a Bayesian framework, we are aimed at computing the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{Z}_{0:t})$ recursively by

$$p(\mathbf{x}_{0:t}|\mathbf{Z}_{0:t}) \propto p(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{Z}_{0:t-1}) \quad (1)$$

Unfortunately in many applications, this distribution is analytically intractable. If, however, we can somehow simulate samples from $p(\mathbf{x}_{0:t-1}|\mathbf{Z}_{0:t-1})$ then we can write down the following empirical estimate

$$\widehat{p}(\mathbf{x}_{0:t-1}|\mathbf{Z}_{0:t-1}) = \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}_{0:t-1}^{j}) \tag{2}$$

Now, both (1) and (2) facilitate the generation of candidate samples from the posterior distribution at time t. These samples are then accepted using an appropriate Metropolis-Hastings (MH) step of which the converged output forms the desired approximation $\widehat{p}(\mathbf{x}_{0:t}|\mathbf{Z}_{0:t})$.

2.1 Metropolis Hastings Step

The MH algorithm generates samples from an aperiodic and irreducible Markov chain with a predetermined (possibly unnormalized) stationary distribution. This is essentially a constructive method which specifies the Markov transition kernel by means of acceptance probabilities based on the preceding time outcome. As part of this, a proposal density is used for drawing new samples. In our case, setting the stationary density as the posterior density, a new set of samples from this distribution can be obtained after the MH burn in period.

2.2 Evolutionary Algorithms

The basic MH scheme can be used to produce several chain realizations each starting from a different (random) state. In that case, the entire population of the converged MH outputs (i.e., subsequent to the burn-in period) approximates the stationary distribution. Using a population of chains enjoys several benefits compared to a single-chain scheme. The multiple-chain approach can dramatically improve the diversity of the produced samples as different chains explore various regions that may not be reached in a reasonable time when using a single chain realization. Furthermore, having a population of chains facilitates the implementation of interaction operators that manipulate information from different realizations for improving the next generation of samples.

Population-based MCMC was originally developed by Geyer [1991]. Further advances came with an evolutionary Monte Carlo algorithm in Liang and Wong [2000] who attempted to produce genetic algorithm (GA) type moves to improve the mixing of the Markov chain. It works by simulating a population of M Markov chains in parallel, where a different (or not) temperature is attached to each chain. The population is updated by mutation (Metropolis update in one single chain), crossover (partial states swapping between different chains), and exchange operators (full state swapping between different chains). Recently, a combination of population-based MCMC with SMC methodology has been proposed in Bhaskar et al. [2008]. The proposed evolutionary MCMC-based particle algorithm aimed at approximating the following target distribution :

$$\pi_*(\mathbf{x}_{0:t}) = \prod_{c=1}^{M}\pi_c(\mathbf{x}_{0:t}) \tag{3}$$

where we have $\pi_c(\mathbf{x}_{0:t}) = p(\mathbf{x}_{0:t}|\mathbf{Z}_{0:t})$ for at least one chain $c = 1, ..., M$. Thus, the output MCMC samples from the chains of target distribution $p(\mathbf{x}_{0:t}|\mathbf{Z}_{0:t})$ are kept as particle approximation (2) for the next time step. At this stage an improved generation of samples from $\pi_*(\mathbf{x}_{0:t})$ is produced using several successive genetic operations.

Crossover Operator The crossover works by switching genetic material between two parent samples from two different chains for producing offspring. The two parents $\mathbf{x}_t^{c_1,m}$ and $\mathbf{x}_t^{c_2,m}$ are selected uniformly from the current population at the m^{th} iteration of the MCMC. The chromosomes A and B corresponding to the chosen parents are then manipulated as follows. For any i, the bits A_i and B_i are swapped with probability β. The resulting offspring chromosomes are then encoded to produce two new candidates $\mathbf{x}_t^{c_1,*}$ and $\mathbf{x}_t^{c_2,*}$. At this point an additional MH step is performed for deciding whether the new offspring will be a part of the improved population. This step is crucial for maintaining an adequate approximation of the target distribution. In order to ensure that the resulting chain is reversible, on acceptance both new candidates should replace their parents, otherwise both parents should be retained.

Following the above argument, it can be shown that the acceptance probability of both offspring is (Liang and Wong [2000])

$$\min\left(1, \left(\frac{1-\beta}{\beta}\right)^{\alpha}\frac{\pi_{c_1}(\mathbf{x}_{0:t-1}^{c_1,m}, \mathbf{x}_t^{c_2,*})\pi_{c_2}(\mathbf{x}_{0:t-1}^{c_2,m}, \mathbf{x}_t^{c_1,*})}{\pi_{c_1}(\mathbf{x}_{0:t}^{c_1,m})\pi_{c_2}(\mathbf{x}_{0:t}^{c_2,m})}\right) \tag{4}$$

where α denotes the number of swapped bits.

Exchange Operator This operation is similar to the one used in parallel tempering (Geyer [1991]). Given the current population, we exchange the state of two different chains, $\mathbf{x}_{0:t}^{c_1,m}$ and $\mathbf{x}_{0:t}^{c_2,m}$. The new moves are accepted with probability

$$\min\left(1, \frac{\pi_{c_1}(\mathbf{x}_{0:t}^{c_2,m})\pi_{c_2}(\mathbf{x}_{0:t}^{c_1,m})}{\pi_{c_1}(\mathbf{x}_{0:t}^{c_1,m})\pi_{c_2}(\mathbf{x}_{0:t}^{c_2,m})}\right) \tag{5}$$

We now describe the two major problems considered in this paper: 1) target cluster tracking, and 2) coordinated group tracking.

3. TARGET CLUSTER TRACKING

In this part of the work we consider a tracking scenario in which a very large number of coordinated targets evolve and interact. The number of targets may be greater than the number of samples used by a multi-target tracking particles algorithm. Obviously, in this case it is impractical to track individual targets and thus we are interested in capturing the clustering structure formed by the targets. The clusters act as extended objects which may split or merge, appear or disappear and may as well change their spatial shape over time.

3.1 Problem Statement

Assume that at time k there are l_k clusters, or targets at unknown locations. Each cluster may produce more than one observation yielding the measurement set realization $z_k = \{z_k(i)\}_{i=1}^{m_k}$, where typically $m_k >> l_k$. At this point we assume that the observation concentrations (clusters)

can be adequately represented by a parametric statistical model $p(z_k(i) \mid \theta_k)$.

Letting $z_{1:k} = \{z_1, \ldots, z_k\}$ be the measurements history up to time k , the cluster tracking problem may be defined as follows. We are concerned with estimating the posterior distribution of the random set of unknown parameters, i.e. $p(\theta_k \mid z_{1:k})$, from which point estimates for θ_k and posterior confidence intervals can be extracted.

The evaluation of the various possible estimates requires the knowledge of the filtering pdf $p_{\theta_k|z_{1:k}}$. For reasons of convenience we consider an equivalent formulation of this pdf that is based on existence variables. Thus, following the approach adopted in Pang et al. [2008] the random set θ_k is replaced by a fixed dimension vector coupled to a set of indicator variables $e_k = \{e_{k,i}\}$ showing the activity status of elements (i.e., $e_{k,i} = 1$ indicates the existence of the ith element). To avoid possible confusion, in what follows we maintain the same notation for the descriptive parameter set θ_k which is now of fixed dimension.

3.2 Bayesian Modeling

Following the Bayesian filtering approach, the filtering pdf is completely specified given some prior $p(\theta_0, e_0)$, a transition kernel $p(\theta_k, e_k \mid \theta_{k-1}, e_{k-1})$ and a likelihood pdf $p(z_k \mid \theta_k, e_k)$. These are derived next for the cluster tracking problem.

Likelihood Derivation Recalling that a single observation $z_k(i)$ is conditionally independent given (θ_k, e_k) yields

$$p(z_k \mid \theta_k, e_k) = \prod_{i=1}^{m_k} p(z_k(i) \mid \theta_k, e_k) \tag{6}$$

In the above equation the pdf $p(z_k(i) \mid \theta_k, e_k)$ describes the statistical relation between a single observation and the cluster parameter sets. An explicit expression for this pdf is given in Gilholm et al. [2005] assuming a spatial Poisson distribution for the number of observations m_k. In this work we restrict ourselves to clusters in which the shape can be modeled via a Gaussian pdf. Following this only the first two moments, namely the mean and covariance, need to be specified for each cluster. Note however, that our approach does not rely on the Gaussian assumption and other parameterized density functions could equally be adopted in our framework. Thus, $\theta_{k,j} = \{\mu_{k,j}, \Sigma_{k,j}\}$, $\theta_k = \{\theta_{k,j}\}_{j=1}^n$, and Gilholm et al. [2005]

$$p(z_k \mid \theta_k, e_k, m_k) = \prod_{i=1}^{m_k} \left[\sum_{j=0}^{n} \mathbf{1}_{\{e_{k,j}=1\}} \mathcal{N}\left(z_k(i) - \mu_{k,j}, \Sigma_{k,j}\right) \right] \tag{7}$$

where $j = 0$ and $\mathbf{1}_{\{e_{k,j}=1\}}$ are the clutter group index and the existence variable of the jth cluster, respectively.

Modeling Clusters' Evolution The overall clustering structure may exhibit a highly complex behavior resulting, amongst other things, from group interactions between different clusters. This in turn may bring about shape deformations and may also affect the number of clusters involved in the formation (i.e., splitting and merging of clusters). In this work, in order to maintain a generic modelling approach, the filtering algorithm assumes no such interactions which thereby yields the following independent cluster evolution model

$$p(\theta_k, e_k \mid \theta_{k-1}, e_{k-1}) = \prod_{i=1}^{n} p(\mu_{k,i} \mid \mu_{k-1,i}) p(\Sigma_{k,i} \mid \Sigma_{k-1,i}) \prod_{j=1}^{n} p(e_{k,j} \mid e_{k-1,j}) \tag{8}$$

where

$$\mu_{k,i} = \mu_{k-1,i} + \zeta, \quad \zeta \sim \mathcal{N}(0, Q_\zeta) \tag{9}$$

Covariance Propagation The following proposition suggests a simple propagation scheme of the covariance $\Sigma_{k,i}$ that is analogous to a random-walk :

$$\Sigma_{k,i} \sim \mathcal{W}(\frac{\Sigma_{k-1,i}}{n_\Sigma}, n_\Sigma) \tag{10}$$

where $\mathcal{W}(V, n_\Sigma)$ denotes a Wishart distribution with a scaling matrix V and a number of degrees of freedom n_Σ.

Birth and Death Moves The existence indicators $e_{k,i}$, $i = 1, \ldots, n$ are assumed to evolve according to a Markov chain. Denote γ_j the probability of staying in state $j \in \{0, 1\}$, then

$$p(e_{k,i} \mid e_{k-1,i} = j) = \begin{cases} \gamma_j, & \text{if } e_{k,i} = j \\ 1 - \gamma_j, & \text{otherwise} \end{cases} \tag{11}$$

Merging and Splitting of Clusters As previously mentioned, two additional types of moves, merging and splitting, are considered for adequate representation of typical clustering behaviour. The transition kernels for these moves follow the Markov chain formulation (11) with the only difference being a state dependent probability γ. The idea here is that the probability of either merging or splitting is related to the clusters' spatial location. This allows smooth and reasonable transitions, essentially discouraging 'artificial' jumps to some physically unlikely clustering structure.

Let $\bar{e}_{k,i}$ be the ith existence variable obtained by using (11). Then the merging kernel is given by

$$p(e_{k,i}, e_{k,j} \mid \bar{e}_{k,i} + \bar{e}_{k,j} = 2, \theta_{k,i}, \theta_{k,j}) = \begin{cases} \gamma_{ij}, & \text{if } e_{k,i} + e_{k,j} = 1 \\ 1 - \gamma_{ij}, & \text{if } e_{k,i} + e_{k,j} = 2 \end{cases} \tag{12}$$

for $i \neq j$ where the merging probability γ_{ij} is

$$\gamma_{ij} = \gamma^m \mathbf{1}_{\{\|\mu_{k,i} - \mu_{k,j}\|_2 \leq d_{\min}\}} \tag{13}$$

for some $\gamma^m \in (0, 1)$ and $d_{\min} > 0$. Here, $\mathbf{1}_A$ denotes the indicator function for the event A. Similarly, the splitting kernel is specified by

$$p(e_{k,i}, e_{k,j} \mid \bar{e}_{k,i} + \bar{e}_{k,j} = 1, \theta_{k,i}, \theta_{k,j}) = \begin{cases} \gamma^s, & \text{if } e_{k,i} + e_{k,j} = 2 \\ 1 - \gamma^s, & \text{if } e_{k,i} + e_{k,j} = 1 \end{cases} \tag{14}$$

where the splitting probability $\gamma^s \in (0, 1)$. In this work, both merging and splitting kernels are applied for all possible combinations (i, j), $i \neq j$.

The parameters $\theta_{k,i}$, $\theta_{k,j}$ of either splitting or merging clusters should be updated properly. This consists of finding a single cluster representation $\theta_{k,+} = \{\mu_{k,+}, \Sigma_{k,+}\}$ which forms the outcome of the pair $\theta_{k,i}$, $\theta_{k,j}$. One way to accomplish this is by matching the first and second moments of the Gaussian, that is

$$\int g^{+}(x)\mathcal{N}\left(x-\mu_{k,+},\Sigma_{k,+}\right)dx = \xi\int g^{i}(x)\mathcal{N}\left(x-\mu_{k,i},\Sigma_{k,i}\right)dx + (1-\xi)\int g^{j}(x)\mathcal{N}\left(x-\mu_{k,j},\Sigma_{k,j}\right)dx \quad (15)$$

where $\xi \in (0,1)$ is a weighting parameter, and $g^{a}(x)$ may be either x or $(x-\mu_{k,a})(x-\mu_{k,a})^{T}$ corresponding to the first two statistical moments. When merging clusters, we set the weighting parameter as $\xi = 1/2$ and solve for both $\mu_{k,+}$ and $\Sigma_{k,+}$. Thus,

$$\mu_{k,+} = \xi\mu_{k,i} + (1-\xi)\mu_{k,j} \quad (16a)$$

$$\Sigma_{k}^{+} = \xi\Sigma_{k,i} + (1-\xi)\Sigma_{k,j} + \xi(1-\xi)\left[\mu_{k,j}(\mu_{k,j})^{T} + \mu_{k,i}(\mu_{k,i})^{T} - 2\mu_{k,j}(\mu_{k,i})^{T}\right] \quad (16b)$$

The same equations are used when splitting clusters. However, in this case one should properly determine either $\theta_{k,i}$ or $\theta_{k,j}$ for finding the missing parameters of the couple $\theta_{k,i}$, $\theta_{k,j}$. In this work splitting is carried out using $\mu_{k,i} = \mu_{k,j} + \zeta_{\mu}$, $\Sigma_{k,i} = \Sigma_{k,j} + \zeta_{\Sigma} I_{2\times 2}$ where the random variables ζ_{μ} and ζ_{Σ} represent spatial uncertainty.

3.3 Bayesian Solution

In practice the filtering pdf $p(\theta_k, e_k \mid z_{1:k})$ cannot be obtained analytically and approximations should be made instead. We propose to use the evolutionary MCMC-based particle filter, introduced in Section 2, for approximating $p(\theta_k, e_k \mid z_{1:k})$.

Metropolis Hastings Step For chain c, the MH algorithm generates samples from $\pi_c(\theta_k, e_k, \theta_{k-1}, e_{k-1})$. Let $(\theta_k^{c,m}, e_k^{c,m}, \theta_{k-1}^{c,m}, e_{k-1}^{c,m})$ be the current Markov chain state. Let also $(\theta_k^{c,*}, e_k^{c,*}, \theta_{k-1}^{c,*}, e_{k-1}^{c,*})$ be a candidate drawn from a proposal distribution $q(\theta_k, e_k, \theta_{k-1}, e_{k-1})$. Then the MH algorithm accepts the new candidate as the next realization from the chain with probability

$$\alpha = \min\left\{1, \frac{\pi_c(\theta_k^{c,*}, e_k^{c,*}, \theta_{k-1}^{c,*}, e_{k-1}^{c,*})}{\pi_c(\theta_k^{c,m}, e_k^{c,m}, \theta_{k-1}^{c,m}, e_{k-1}^{c,m})} \times \frac{q(\theta_k^{c,m}, e_k^{c,m}, \theta_{k-1}^{c,m}, e_{k-1}^{c,m})}{q(\theta_k^{c,*}, e_k^{c,*}, \theta_{k-1}^{c,*}, e_{k-1}^{c,*})}\right\} \quad (17)$$

In this work, we use the joint propagated pdf as our proposal. More precisely, $(\theta_{k-1}^{c,*}, e_{k-1}^{c,*})$ are first drawn from the empirical approximation of $p(\theta_{k-1}, e_{k-1} \mid z_{1:k-1})$. Then, $(\theta_k^{c,*}, e_k^{c,*})$ are obtained as follows :

$$(\theta_k^{c,*}, e_k^{c,*}) \sim p(\theta_k, e_k \mid \theta_{k-1}^{c,*}, e_{k-1}^{c,*}) \quad (18)$$

The evolutionary MCMC-based particle algorithm at the mth MCMC iteration is summarized in Algorithm 1.

Algorithm 1 Single-Chain MCMC

for c=1, ..., M **do**
  Propose $(\theta_{k-1}^{c,*}, e_{k-1}^{c,*}) \sim \hat{p}(\theta_{k-1}, e_{k-1} \mid z_{1:k-1})$
  Propose $(\theta_k^{c,*}, \bar{e}_k^{c,*}) \sim p(\theta_k, e_k \mid \theta_{k-1}^{c,*}, e_{k-1}^{c,*})$ using (8-11)
  For any pair $(\theta_{k,j}^{c,*}, \bar{e}_{k,j}^{c,*})$, $(\theta_{k,i}^{c,*}, \bar{e}_{k,i}^{c,*})$, $j \neq i$ perform either merging or splitting as described in Section 3.2.5.
  Compute the MH acceptance probability α of $(\theta_k^{c,*}, e_k^{c,*}, \theta_{k-1}^{c,*}, e_{k-1}^{c,*})$ using (17).
  Draw $u \sim U[0,1]$
  Set $(\theta_k^{c,m+1}, e_k^{c,m+1}) = (\theta_k^{c,*}, e_k^{c,*})$ if $u < \alpha$, otherwise set $(\theta_k^{c,m+1}, e_k^{c,m+1}) = (\theta_k^{c,m}, e_k^{c,m})$
end for
Draw $u \sim U[0,1]$
if $u < u_{crossover}$ **then**
  Perform the crossover operator,
  Accept the move with prob. of (4)
end if
Draw $u \sim U[0,1]$
if $u < u_{exchange}$ **then**
  Perform the exchange operator,
  Accept the move with prob. of (5)
end if

The evolutionary MCMC-based particle algorithm at the mth MCMC iteration is summarized in Algorithm 1.

4. COORDINATED GROUP TRACKING

In this section, we address the problem of detection and tracking of group and individual targets. In particular, we focus on a group model with a virtual leader which models the bulk or group parameter, proposed in Pang et al. [2008]. This formulation leads to a simple analytic solution owing to the linear Gaussian structure of the stochastic differential equation.

Concerning the observation model, an association free approach,popularly known as Track-Before-Detect (TBD), is taken Ristic et al. [2004], Kreucher et al. [2005]. More specifically, for the synthetic data simulation, we will specify the observation model as a simplified ground moving target indicator (GMTI) radar with position only Rayleigh-distributed measurements Kreucher et al. [2005]. We will also use thresholded measurement that returns 1 or 0 for each pixel.

To detect and track targets within groups, as well as infer both the correct group structure and the number of targets over time, a (single chain) MCMC-based particle algorithm has been proposed in Septier et al. [2009]. Here, we propose to extend this algorithm by using the evolutionary strategy described in Section 2. For sake of space, readers are referred to Septier et al. [2009] for details about the models and the single chain scheme. The evolutionary method at the m^{th} MCMC iteration is summarized in Algorithm 2.

Algorithm 2 Evolutionary MCMC for Group Tracking

for c=1, ..., M **do**
  Perform Algorithm 1 and 2 described in Septier et al. [2009]
end for
Draw $u \sim U[0,1]$
if $u < u_{crossover}$ **then**
  Perform the crossover operator,
  Accept the move with prob. of (4)
end if
Draw $u \sim U[0,1]$
if $u < u_{exchange}$ **then**
  Perform the exchange operator,
  Accept the move with prob. of (5)
end if

5. NUMERICAL RESULTS

5.1 Target Cluster Tracking

The evolutionary MCMC scheme is implemented using $N = 1500$ particles and $M = 5$ chain realizations in which the target distribution corresponds to the filtering posterior distribution. The chains' burn-in period is set to $N_{\text{Burn-in}} = 200$ based on tuning runs. The genetic operators are used with probability of $u_{crossover} = 0.1$ and $u_{exchange} = 0.1$.

The clusters trajectories and observations were generated using the models described in Carmi et al. [2009]. Both actual X and Y tracks over time are shown in Figs. 1a, 1b, 2a and 2b. These figures depict a typical scenario which involves splitting (at approximately $k = 20$) and merging (at $k = 60$) clusters. The densely cluttered observations are shown in the corresponding Figs. 1c, 1d, 2c and 2d. The performance of the MCMC filtering algorithm is demonstrated in the remaining figures, Figs. 1e, 1f, 2e and 2f. These figures show the level plots of the estimated Gaussian mixture model over time. Thus, it can be clearly seen that on the overall the filtering algorithm is capable of adequately tracking the varying clustering structure.

Using the particles approximation one can easily compute the probability hypothesis density (PHD) over the entire field of view. An empirical estimate of the PHD in this case is given by $N^{-1} \sum_{i=1}^{N} \sum_{j=1}^{n} e_{k,i}^{j}$. Notice, however, that this rather unusual PHD corresponds to number of clusters and not directly to target counts. The average PHD was computed based on 10 Monte Carlo runs and is depicted along with the actual average number of clusters in Fig. 3.

5.2 Coordinated Group Tracking

A single discretised sensor model is used which scans a fixed rectangular region of 100 by 100 pixels, where each pixel is 50m by 50m. Thresholded measurements are used with $P_{d,1} = 0.7$ and a false alarm probability for each pixel of $P_{fa} = 0.002$ (*i.e.* SNR= 17 dB). The sensor returns a set of observations every 5s. The tracks and observations were generated using the models described in Septier et al. [2009]. This scenario consists of 2 groups of 2 targets moving towards each other from time step 1 to 45, and then merged to form a combined group from time step 46 to 110.

The Evolutionary MCMC-based particle algorithm is used to detect and track targets in the scenario described above. All the particles are initialised as inactive in order to allow the algorithm to detect all targets unaided. At each time step, 3 chains of 2000 MCMC iterations are performed with burn-in of 500 iterations. In the first two chains, the target distribution is the posterior distribution but in the third chain, the likelihood is tempered by setting $P_{d,1} = 0.7$ and $P_{fa} = 0.005$ (*i.e.* SNR= 14 dB).

The tracking performances are shown in Fig. 4. The algorithm has successfully detected and tracked the 4 targets in a hostile environment with heavy clutter. The ellipse in Fig. 4c shows the mode of the group configuration and the number indicates the number of targets in the group. The proposed algorithm is clearly able to infer the correct group structure.

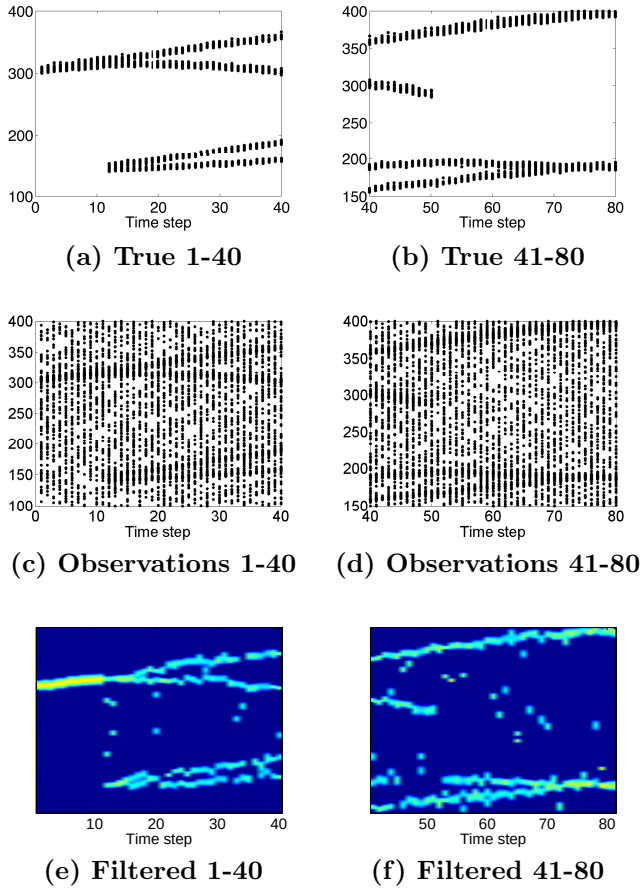


Fig. 1. Tracking performance. Showing X axis over time.

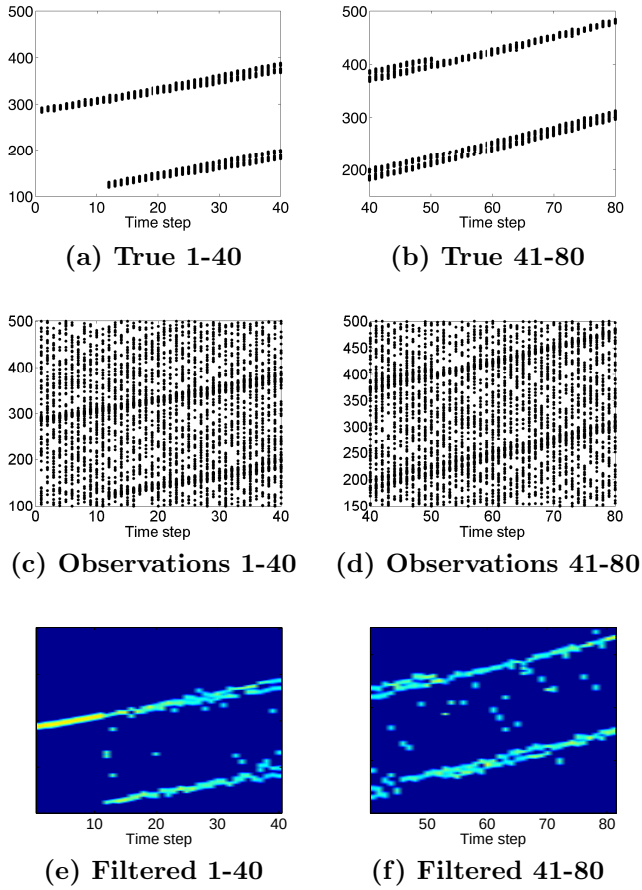
Finally, Fig. 4d shows the average number of targets, given by the existence variables, over the 40 Monte Carlo runs. From this figure, we can see that the proposed algorithm is able to detect consistently and rapidly that there are 4 targets in the observation scene.

6. CONCLUSION

A new Markov chain Monte Carlo filtering algorithm is derived for tracking multiple objects. This sequential approach incorporates several attractive features of genetic algorithms and simulated annealing into the framework of the MCMC-based particle scheme. This evolutionary strategy increases the efficiency of the filtering algorithm mainly due to its ability to explore larger regions of the sample space in a reasonable time. The new filter is tested in two difficult tracking scenarios. In either cases the algorithm exhibits a good tracking performance.

ACKNOWLEDGEMENTS

This research was sponsored by the Data and Information Fusion Defence Technology Centre, UK, under the Tracking Cluster. The authors thank these parties for funding this work. We also would like to thank the Statistical and Applied Mathematical Sciences Institute - SMC program for providing a collaborative research environment that assisted with the development of our ideas.



(a) True 1-40 (b) True 41-80

(c) Observations 1-40 (d) Observations 41-80

(e) Filtered 1-40 (f) Filtered 41-80

Fig. 2. Tracking performance. Showing Y axis over time.

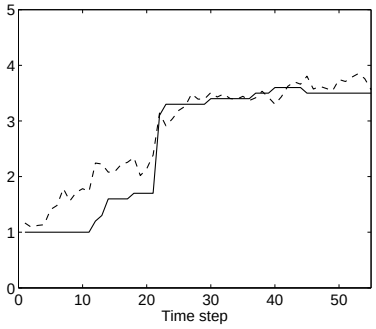
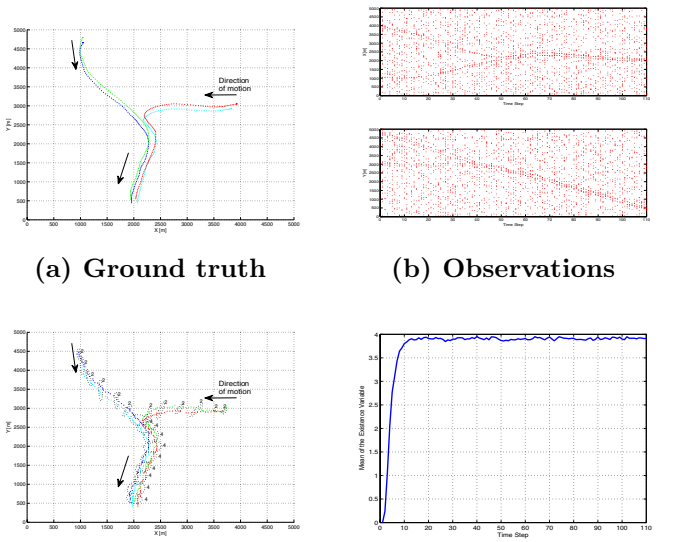


Fig. 3. Average number of clusters (solid line) and the mean PHD (dashed line) based on 10 Monte Carlo runs.



(a) Ground truth (b) Observations

(c) Estimated tracks (d) Average number of targets

Fig. 4. Tracking performance : group merging scenario.

REFERENCES

C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza. Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods. *Journal of the American Statistical Association*, 440:1403–1412, Dec. 1997.

H. Bhaskar, L. Mihaylova, and S. Maskell. Population-based Particle Filtering. In *of the IET Target Tracking and Data Fusion: Algorithms and Applications*, pages 31–38, Birmingham, UK, Apr. 2008.

A. Carmi, S. J. Godsill, and F. Septier. Evolutionary MCMC Particle Filtering for Target Cluster Tracking. Marco Island, Florida, 2009. Proceedings of the IEEE 13th DSP Workshop and the 5th SPE Workshop.

C. J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, volume 1, pages 156–163, 1991.

K. Gilholm, S.J. Godsill, S. Maskell, and D. Salmond. Poisson Models for Extended Target and Group Tracking. *SPIE Conference : Signal and Data Processing of Small Targets, Proceedings of*, Aug. 2005.

W. R. Gilks and C. Berzuini. Following a Moving Target-Monte Carlo Inference for Dynamic Bayesian Models. *Journal of the Royal Statisticsal Society. series B (Statistical Methodology)*, 63:127–146, 2001.

A. Golightly and D. J. Wilkinson. Bayesian Sequential Inference for Nonlinear Multivariate Diffusions. *Statistics and Computing*, pages 323–338, Aug. 2006.

Z. Khan, T. Blach, and F. Dellaert. MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1805–1819, Nov. 2005.

C. Kreucher, M. Morelande, K. Kastella, and A. O . H. III. Particle Filtering for Multitarget Detection and Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 41:1396–1414, Oct. 2005.

F. Liang and W. H. Wong. Evolutionary Monte Carlo: Applications to C_p Model Sampling and Change Point Problem. *Statistica Sinica*, 10:317–342, 2000.

S. K. Pang, J. Li, and S. J. Godsill. Models and Algorithms for Detection and Tracking of Coordinated Groups. In *IEEE Aerospace Conference*, pages 1–17, Mar. 2008.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter - Particle Filters for Tracking Applications*. Artech House, 685 Canton Street, Norwood, MA 02062, 2004.

F. Septier, S.K. Pang, S. J. Godsill, and A. Carmi. Tracking of Coordinated Groups using Marginalised MCMC-based Particle Algorithm. In *IEEE Aerospace Conference*, Mar. 2009.