The Gaussian mixture MCMC particle algorithm for dynamic cluster tracking

Avishy Carmi, François Septier, Simon Godsill

To cite this version:

HAL Id: hal-00566634
https://hal-imt.archives-ouvertes.fr/hal-00566634
Submitted on 15 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
The Gaussian Mixture MCMC Particle Algorithm for Dynamic Cluster Tracking

Avishy Carmi, François Septier and Simon J. Godsill
Signal Processing and Communications Laboratory
Department of Engineering
University of Cambridge, U.K.

Abstract – We present a new filtering algorithm for tracking multiple clusters of coordinated targets. Based on a Markov Chain Monte Carlo (MCMC) mechanism, the new algorithm propagates a discrete approximation of the underlying filtering density. A dynamic Gaussian mixture model is utilized for representing the time-varying clustering structure. This involves point process formulations of typical behavioral moves such as birth and death of clusters as well as merging and splitting. Following our previous work, we adopt here two strategies for increasing the sampling efficiency of the basic MCMC scheme: an evolutionary stage which allows improved exploration of the sample space, and an EM-based method for making optimized proposals based on the frame likelihood. The algorithm’s performance is assessed and demonstrated in both synthetic and real tracking scenarios.

Keywords: Multiple cluster tracking, Markov chain Monte Carlo filtering, Evolutionary MCMC, EM algorithm

1 Overview

Multi-Target tracking (MTT) poses major challenges for researchers in the fields of estimation and information fusion. The extensive studies that have been conducted over recent decades have yielded various tracking techniques which can be divided informally into two classes: non-statistical and statistical. Non-statistical methods typically rely on both image differencing techniques and heuristic smoothing algorithms for identifying targets’ trajectories [1]. The non-statistical schemes are considered to be fast and viable and have been extensively deployed for tracking in various applications. Nevertheless, these methods are incapable of adequately handling complex tracking scenarios such as those studied here, in which there is significant statistical ambiguity and dynamical structure in the models used. Hence statistical inference approaches are now preferred in many cases.

Owing to the complex nature of MTT problems, statistical tracking methods usually involve smart implementation of tightly coupled data association and filtering schemes [2]. This in turn may result in computationally intensive algorithms such as the multiple hypothesis tracker (MHT) in [3]. Another major difficulty imposed by a typical MTT scenario is related to the mathematical modeling of complex interactions between entities. This consists mainly of birth and death of targets as well as coordinated behavioural patterns which arise in group motions.

The optimal filtering scheme involves the propagation of the joint probability density of target states conditioned on the data. Following the conventional approach, in which all states are concatenated to form an augmented vector, leads to a problematic statistical representation owing to the fact that the targets themselves are unlabeled and thus can switch positions within the resulting joint state vector. Furthermore, targets may appear and disappear thereby yielding inconsistencies in the joint state dimension. These problems can be circumvented by adopting one of the following approaches: 1) introducing some sort of labeling mechanism which identifies existing targets within the augmented vector [4], or 2) considering the joint state as a random finite set. The latter approach provides an elegant and natural way to make statistical inference in MTT scenarios. Nevertheless, its practical implementation as well as its mathematical subtleties need to be carefully considered [5].

Random sets can be thought of as a generalization of random vectors. The elements of a set may have arbitrary dimensions, and as opposed to vectors, the ordering of their elements is insignificant. These properties impose difficulties in constructing probability measures over the space of sets. This has led some researchers to develop new concepts based on belief mass functions such as the set derivative and set integral for embedding notions from measure theoretic probability within random set theory (e.g., Bayes rule). As part of this,
point process statistics are commonly used for deriving probabilistic quantities [6]. The PHD filter presented in [7] is the first attempt to implement finite set statistics concepts for MTT. This algorithm uses a Poisson point process formulation to derive a semi-closed-form recursion for propagating the first moment of the random set’s intensity (i.e., the set’s cardinality). A brief summary of the PHD algorithm can be found in [5].

In recent years, sequential Monte Carlo (SMC) methods were applied for MTT. 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. An extensive survey and application of SMC methods is given in [8].

The MTT PF algorithms in the works [9–13] are intended to work with a fixed number of targets. These PFs exploit point process formulations for properly assigning measurements to their originating targets. As part of this, smart procedures are used to eliminate non-probable association hypotheses.

An extension of the PF technique to varying number of targets is introduced in [5], [14] and [15]. In [5, 16] a PF implementation of the PHD filter is derived. This algorithm maintains a representation of the filtering belief mass function using random set realizations (i.e., particles of varying dimensions). The samples are propagated and updated based on a Bayesian recursion consisting of set integrals. Both works of [14] and [15] develop a Markov Chain Monte Carlo (MCMC) PF scheme for tracking varying numbers of interacting objects. The MCMC approach does posses a reported advantage over conventional PF due to its efficient sampling mechanism. Nevertheless, in its traditional non-sequential form it is inadequate for sequential estimation. The techniques used by [14] and [15] amend the MCMC for sequential filtering (see also [17]). The work in [15] copes with inconsistencies in state dimension by utilizing the reversible jump MCMC method introduced in [18], [14], on the other hand avoids the computation of the marginal filtering distribution as in [17] and operates on a fixed dimension state space through use of indicator variables for labeling of active target states (the two approaches being essentially equivalent, see [19]).

1.1 Cluster Tracking

In recent years there has been an increasing interest in tracking scenarios in which a very large number of coordinated objects evolve and interact. One could think of many fields in which such situation is frequently encountered: video surveillance, biomedicine, neuroscience and meteorology to mention only a few. Considering the nature of the cluster tracking problem an efficient approach would consist of estimating the clustering structure formed by object concentrations rather than tracking individual entities. This is the case both since the number of individual objects may be too large to track practically and indeed we cannot necessarily expect individual objects in a coordinated motion to be detected by the sensor in adjacent data frames.

It should be noted that clusters can be thought of as extended objects that produce a large number of observations. This approach yields the Poisson likelihood formulations in [20]. In recent work [21] merging and splitting objects are modeled using point processes. This is another fundamental issue characterizing cluster behavior that is given full consideration in this work.

1.2 Proposed Method

The algorithm proposed herein is based on the evolutionary MCMC mechanism derived in [22]. While assuming that target locations resemble independent samples from a Gaussian mixture we parameterize each cluster by its mean and covariance. A Bernoulli-Markov process is used for describing the evolution of the clustering structure over time (i.e., birth and death of clusters).

Similarly to [22], the MCMC filtering scheme here incorporates two enhancements that are aimed at increasing the efficiency of the Metropolis-Hastings sampler: a genetic manipulation stage in which members from possibly different chain realizations are combined for generation of new MCMC moves, and an optimized proposal generation scheme based on the EM algorithm. In contrast to the optimization scheme in [22], which relies on a variational Bayesian extension of the EM, the scheme here is much simpler to implement as it is exclusively based on the well-known closed form analytic solution of the EM for Gaussian mixture likelihoods.

1.3 Outline

This paper is organized as follows. Section 2 mathematically formulates the cluster tracking problem. The likelihood and time evolution models used by the filtering algorithm are presented in Section 3. Section 4 develops the MCMC particle filtering algorithm. Section 5 presents the results of a simulation study that has been conducted to assess the new algorithm’s tracking performance. Finally, conclusions are offered in the last section.

2 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 $z_k = \{y_k(i)\}_{i=1}^{m_k}$, 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 \( p(Y_k \mid \theta_k) \).

Letting \( Z_{1:k} = \{Z_1, \ldots, Z_k\} \) and \( z_{1:k} = \{z_1, \ldots, z_k\} \) be the measurements history up to time \( k \) and its realization, respectively, 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 \( \theta_k \) and posterior confidence intervals can be extracted.

### 2.1 Random Set Representation

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 [14] the random set \( \theta_k \) 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 \( i \)-th element). To avoid possible confusion, in what follows we maintain the same notation for the descriptive parameter set \( \theta_k \) which is now of fixed dimension.

### 3 Bayesian Inference

Following the Bayesian filtering approach while assuming that the observations are conditionally independent given \( (\theta_k, e_k) \) the density \( p(\theta_k, e_k \mid z_{1:k}) \) is obtained recursively using the conventional Bayesian recursion [20]. Thus, 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.

#### 3.1 Likelihood Derivation

Recalling that a single observation \( y_k(i) \) is conditionally independent given \( (\theta_k, e_k) \) yields

\[
p(z_k \mid \theta_k, e_k) = \prod_{i=1}^{m_k} p(y_k(i) \mid \theta_k, e_k)
\]

(1)

In the above equation the pdf \( p(y_k(i) \mid \theta_k, e_k) \) describes the statistical relation between a single observation and the cluster parameter sets. An explicit expression for this pdf is given in [20] 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, \( \theta_k^i = \{\mu_k^i, \Sigma_k^i\}, \theta_k = \{\theta_k^i\}_{i=1}^n \), and [20]

\[
p(z_k \mid \theta_k, e_k) = \prod_{i=1}^{m_k} \sum_{j=0}^{n} 1_{\{e_k^i = 1\}} w_j \mathcal{N}(y_k(i) - \mu_k^i, \Sigma_k^i)
\]

(2)

where \( j = 0 \) and \( 1_{\{e_k^i = 1\}} w_j > 0 \) are the clutter group index and the intensity variable of the \( j \)-th cluster, respectively.

#### 3.2 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

\[
p(\theta_k, e_k \mid \theta_{k-1}, e_{k-1}) = \prod_{i=1}^{n} p(\mu_k^i \mid \mu_{k-1}^i)p(\Sigma_k^i \mid \Sigma_{k-1}^i) \prod_{j=1}^{n} p(e_k^i \mid e_{k-1}^i)
\]

(3)

where

\[
\mu_k^i = \mu_{k-1}^i + \zeta, \quad \zeta \sim \mathcal{N}(0, Q_{\zeta})
\]

(4)

#### 3.2.1 Covariance Propagation

The following proposition (given here without a proof) suggests a simple propagation scheme of the covariance \( \Sigma_k^i \) that is analogous to a random-walk.

**Proposition 1.** Let \( \Sigma_0^i \sim \mathcal{W}(V, n_1, n_2) \) where \( \mathcal{W}(V, n_1, n_2) \) denotes a Wishart distribution with a scaling matrix \( V \) and parameters \( n_1 \) and \( n_2 \). Let also

\[
\Sigma_k^i = \Sigma_{k-1}^i + W + CD^T + DCT
\]

(5)

where

\[
W \sim \mathcal{W}(V', n_1, n_2), \quad W = DDT', \quad \Sigma_{k-1}^i = CCT
\]

(6)

Then \( \Sigma_k^i \) is distributed according to \( \mathcal{W}(V + kV', n_1, n_2) \).

In the ensuing (5) is used to draw conditionally independent samples from \( p(\Sigma_k^i \mid \Sigma_{k-1}^i) \).

#### 3.2.2 Birth and Death Moves

The existence indicators \( e_k^i, i = 1, \ldots, n \) are assumed to evolve according to a Markov chain. Denote \( \gamma_j \) the probability of staying in state \( j \in [0, 1] \), then

\[
p(e_k^i \mid e_{k-1}^i = j) = \begin{cases} \gamma_j, & \text{if } e_k^i = j \\ 1 - \gamma_j, & \text{otherwise} \end{cases}
\]

(7)
3.2.3 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 point process formulation (7) with the only difference being a state dependent probability $\gamma$. 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 structures. Let $\bar{c}_k^i$ be the $i$th existence variable obtained by using (7). Then the merging kernel is given by

$$ p(e_k^i, e_k^j | \bar{c}_k^i + \bar{c}_k^j = 2, \theta_k^i, \theta_k^j) = 
\begin{cases} 
\gamma_{ij}, & \text{if } e_k^i + e_k^j = 1 \\
1 - \gamma_{ij}, & \text{if } e_k^i + e_k^j = 2 
\end{cases} 
\tag{8} $$

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

$$ \gamma_{ij} = \gamma^m 1_{\{\|\mu_k^i - \mu_k^j\|_2 \leq d_{\text{min}}\}} 
\tag{9} $$

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

$$ p(e_k^i, e_k^j | \bar{c}_k^i + \bar{c}_k^j = 1, \theta_k^i, \theta_k^j) = 
\begin{cases} 
\gamma^s, & \text{if } e_k^i + e_k^j = 2 \\
1 - \gamma^s, & \text{if } e_k^i + e_k^j = 1 
\end{cases} 
\tag{10} $$

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

$$ \int g^+(x) \mathcal{N}(x - \mu_k^+, \Sigma_k^+) \, dx = 
\xi \int g^+(x) \mathcal{N}(x - \mu_k^i, \Sigma_k^i) \, dx 
+ (1 - \xi) \int g^+(x) \mathcal{N}(x - \mu_k^j, \Sigma_k^j) \, dx \tag{11} $$

where $\xi \in (0, 1)$ is a weighting parameter, and $g^+(x)$ may be either $x$ or $(x - \mu_k^2)(x - \mu_k^2)^T$ corresponding to the first two statistical moments. When merging clusters, we set the weighting parameter as $\xi = w_1/(w_1 + w_2)$ and solve for both $\mu^+$ and $\Sigma^+$. Thus,

$$ \mu^+ = \xi \mu_k^i + (1 - \xi) \mu_k^j, \tag{12a} $$

$$ \Sigma_k^+ = \xi \Sigma_k^i + (1 - \xi) \Sigma_k^j + \xi (1 - \xi) \left[ \mu_k^i (\mu_k^i)^T + \mu_k^j (\mu_k^j)^T - 2 \mu_k^i (\mu_k^j)^T \right] \tag{12b} $$

The same equations are used when splitting clusters. However, in this case one should properly determine both $\xi$ and 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 $\xi \sim U(0, 1)$, and $\mu_k^i = \mu_k^i + \zeta_\mu$, $\Sigma_k^i = \Sigma_k^i + \zeta_\Sigma I_{2 \times 2}$ where the random variables $\zeta_\mu$ and $\zeta_\Sigma$ represent spatial uncertainty.

4 MCMC Particle Algorithm

In practice the filtering pdf $p(\theta_k, e_k | z_{1:k})$ cannot be obtained analytically and approximations should be made instead. In this section we introduce a sequential MCMC particle filtering algorithm for approximating $p(\theta_k, e_k | z_{1:k})$. The new filter consists of a multiple-chain Metropolis Hastings (MH) sampler that uses both genetic operators and local optimization steps for producing improved proposals. Compared to a single-chain MH approach, this scheme increases the efficiency of the filtering algorithm mainly due to its ability to explore larger regions of the sample space in a reasonable time.

4.1 Basic Sampling Scheme

The following sequential scheme is partially based on the inference algorithm presented in [14]. Suppose that at time $k - 1$ there are $N$ samples $\{\theta_{k-1}(i), e_{k-1}(i)\}_{i=1}^N$ drawn approximately from the filtering density $p(\theta_{k-1}, e_{k-1} | z_{1:k-1})$ (i.e., the previous time target distribution). In order to obtain a new set of samples $\{\theta_k(i), e_k(i)\}_{i=1}^N$ for representing $p(\theta_k, e_k | z_{1:k})$, we first simulate the joint propagated pdf $p(\theta_k, e_k, \theta_{k-1}, e_{k-1} | z_{1:k-1})$ by drawing

$$ (\theta_k(i), e_k(i)) \sim p(\theta_k, e_k | \theta_{k-1}(i), e_{k-1}(i)) \tag{13} $$

where $(\theta_{k-1}(i), e_{k-1}(i))$ are uniformly drawn from the empirical approximation of $p(\theta_{k-1}, e_{k-1} | z_{1:k-1})$. These samples are then accepted or rejected using a proper MH step. The converged output of this scheme simulates the joint density $p(\theta_k, e_k, \theta_{k-1}, e_{k-1} | z_{1:k})$ of which the marginal is the desired filtering pdf.

4.1.1 Metropolis Hastings Step

The MH algorithm generates samples from an aperiodic and irreducible Markov chain with a predetermined (possibly unnormalized) stationary distribution. This is 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 joint filtering pdf $p(\theta_k, e_k, \theta_{k-1}, e_{k-1} | z_{1:k})$, a new set of samples from this distribution can be obtained after the MH burn-in period.
4.3 Optimizing Proposals Using EM

The evolutionary scheme in this work consists of an optimization step aimed at increasing the efficiency of the sampling algorithm. A natural approach in the case of MCMC sampling would be to obtain an optimized proposal pdf rather than a single deterministic solution. This insight is the basis for the variational optimization mechanism used in [22]. In this work we adopt a simplified approach for generating valid optimized proposals. In what follows we demonstrate how the EM algorithm can be used for generating such proposals.

4.3.1 EM for Gaussian Mixtures

The EM algorithm maximizes an auxiliary function \( Q(\theta_k, w_j) \) which is a lower bound for the log-likelihood \( \log p(z_k \mid \theta_k, e_k) \). In detail, from (2) we have

\[
\log p(z_k \mid \theta_k, e_k) \geq \sum_{i=1}^{m_k} \sum_{j=0}^{n} 1_{(e'_{k,j})} w_j \log c_j
\]

\[
-\frac{1}{2} \left( y_k(i) - \mu^j \right)^T \left( \Sigma_k^{-1} \right) \left( y_k(i) - \mu^j \right) = Q(\theta_k, w_j)
\]  

where the optimum of \( Q(\theta_k, w_j) \) is obtained by iterating

\[
w^{t+1} = \arg\max_{w_j} Q(\theta_k, w_j)
\]

\[
\theta^{t+1} = \arg\max_{\theta_k} Q(\theta_k, w_j^{t+1})
\]

This recursion, which in our case has a closed form analytical expression, is guaranteed to converge to a local maximum. The issue of recovering \( e^j_k \) is resolved here by setting \( e^j_k = 1_{(w_j > w_{max})} \). In other words, the indicator variables are set to 1 if the corresponding weight is above some predetermined threshold value \( w_{th} \).

4.3.2 A Regularized Proposal

In this work we use the EM recursion (16) for constructing a valid proposal pdf. Let \( \{\theta_k^{0(i)}, w_j^{0(i)}\}_{i=1}^{N_{opt}} \) be a set of particles with which (16) is initialized for producing \( \{\theta_k^{j(i)}, w_j^{j(i)}\}_{i=1}^{N_{opt}} \) for \( t \) iterations (the initial set of particles can be taken from the propagated pdf in (13)). Following the above discussion the optimized set of particles is then taken as \( \{\theta_k^{j(i)}, e'_{k(i)}\}_{i=1}^{N_{opt}} \) where \( \theta_k^{j(i)} = \theta_k^{0(i)} \) and \( e'_{k(i)} = 1_{(w_j > w_{max})} \). This set is used for composing a regularized proposal pdf of the form

\[
q(\tilde{\theta}_k) \propto \sum_{i=1}^{N_{opt}} \sum_{j=0}^{n} c^{j(i)}_{k} K(\tilde{\theta}_k - \theta_k^{j(i)})
\]  

where \( K(\cdot) \) is some regularization kernel of infinite support (e.g., Gaussian). The above equation suggests a simple sampling scheme for producing candidates from \( q(\tilde{\theta}_k) \). Thus, we uniformly sample from \( \{\theta_k^{0(i)}, e'_{k(i)}\}_{i=1}^{N_{opt}} \) to get \( \{\theta_k, e_k\} \). We then use this sample for drawing a candidate \( \tilde{\theta}_k \sim K(\tilde{\theta}_k - \theta_k^{j(i)}) \), where \( j \in [1, n] \) \( e_1^j = 1 \), and \( e^j_k = e_k \). Finally, the acceptance probability of \((\theta_k, e_k)\) given the previously accepted sample \((\theta_k, e_k)\) is simply

\[
\alpha = \min \left\{ 1, \frac{p(\tilde{\theta}_k, e_k \mid z_{1:k}) q(\theta_k)}{p(\theta_k, e_k \mid z_{1:k}) q(\theta_k)} \right\}
\]  

4.4 Algorithm Summary

The basic MCMC filtering algorithm (without the evolutionary extension) is summarized in Algorithm 1.
Algorithm 1 Single-Chain MCMC

1: Given $N$ samples from $\hat{p}(\theta_{k-1}, e_{k-1}, \theta_{k-2}, e_{k-2} | z_{1:k-1})$ perform the following steps.
2: for $i=1, \ldots, N$ do
3: Uniformly draw $(\theta_{k-1}(i), e_{k-1}(i)) \sim \hat{p}(\theta_{k-1}, e_{k-1} | z_{1:k-1})$
4: for $j=1, \ldots, n$ do
5: Draw $\theta_k^j(i) \sim p(\theta_k^j | \theta_{k-1}(i))$ using both (4) and (5).
6: Draw $e_k^j(i) \sim p(e_k^j | e_{k-1}(i))$ using (7).
7: end for
8: For any pair $(\theta_k^j(i), e_k^j(i))$, $(\theta_k^l(i), e_k^l(i))$, $j \neq l$ perform either merging or splitting as described in Section 3.2.3.
9: end for
10: for $i=1, \ldots, N + N_{Burn-in}$ do
11: Draw $u \sim U[0,1]$
12: if $u < u_{EM}$ then
13: Propose a new candidate $\tilde{\theta}_k \sim q(\tilde{\theta}_k)$.
14: Compute the MH acceptance probability $\alpha$ of $(\tilde{\theta}_k, \tilde{e}_k)$ using (18).
15: else
16: Propose $(\tilde{\theta}_k, \tilde{e}_k, \tilde{\theta}_{k-1}, \tilde{e}_{k-1}) \sim \hat{p}(\theta_k, e_k, \theta_{k-1}, e_{k-1} | z_{1:k-1})$.
17: Compute the MH acceptance probability $\alpha$ of the new move using (14).
18: end if
19: Draw $u \sim U[0,1]$
20: if $u < \alpha$ then
21: Accept $s(i) = (\tilde{\theta}_k, \tilde{e}_k)$ as the next sample of the realized chain.
22: else
23: Retain $s(i) = s(i-1)$.
24: end if
25: end for

5 Simulation Study

The MCMC filtering algorithm is numerically tested in both synthetic and realistic tracking scenarios consisting of a varying number of clusters. The synthetic case is similar to the one in [22], however, with a total number of clusters not exceeding four. For conciseness we omit here the detailed description of the cluster dynamics and observations. The reader is referred to [22] for a brief summary of these models.

5.1 Algorithm Settings

The evolutionary MCMC scheme is implemented using $N = 1500$ particles and $l = 5$ chain realizations. The chains burn-in period is set to $N_{Burn-in} = 200$ based on tuning runs. During the MH step, an optimized move is sampled from the regularized pdf $q(\tilde{\theta}_k)$ with probability of $u_{EM} = 0.05$. The number of EM iterations used for composing $q(\cdot)$ do not exceed $t = 2$.

5.2 Synthetic Data

The clusters trajectories and observations were generated using the models described in [22]. 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 $N^{-1} \sum_{i=1}^N \sum_{j=1}^n c_k^j(i)$. 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.
5.3 Crowd Tracking Example

The MCMC filtering algorithm was applied for tracking a crowd of people in a video sequence. This scenario consists of people walking in a corridor as seen from above. The video was preprocessed using a corner detector for yielding the set of point observations to be used by the filtering algorithm. The obtained set of points is characterized by a relatively dense concentration near each person’s head which in some cases extends towards the shoulders. Other points corresponding to non-moving objects are considered here as clutter. The dynamical model assumed by the filter is similar to the one used in the synthetic data case (see [22]).

The actual scenario, the preprocessed observations and the filtering performance are shown in Fig. 4 at two distinct time points. Thus, the estimated using a corner detector for yielding the set of point observations to be used by the filtering algorithm. The obtained set of points is characterized by a relatively dense concentration near each person’s head which in some cases extends towards the shoulders. Other points corresponding to non-moving objects are considered here as clutter. The dynamical model assumed by the filter is similar to the one used in the synthetic data case (see [22]).

The actual scenario, the preprocessed observations and the filtering performance are shown in Fig. 4 at two distinct time points. Thus, the estimated Gaussian means are marked by pluses in both Figs. 4a and 4c whereas the corresponding covariance ellipses are shown along with the point observations in Figs. 4b and 4d. These figures demonstrate the viability of the MCMC filtering algorithm in this case as it identifies the exact number of walking people in the corridor.

6 Conclusions

A new Markov chain Monte Carlo filtering algorithm is derived for tracking multiple clusters. The clustering structure is represented using a dynamic Gaussian mixture model. The new filter is tested in both synthetic and realistic scenarios. In either case the algorithm exhibits a good tracking performance as it captures the essence of the clusters behavior. The realistic example clearly demonstrates the viability of the new algorithm in such scenarios where the spatial Poisson assumption is violated.

7 Acknowledgments

This work was sponsored by the Data and Information Fusion Defense Technology Centre, UK, under the Tracking Cluster. The authors thank these parties for funding this work.

References


Figure 4. Crowd tracking example.