2014 UKSim-AMSS 8th European Modelling Symposium

Performance Analysis of Sequential Monte Carlo MCMC and PHD Filters on Multi-target Tracking in Video

Abdullahi Daniyan *Member, IEEE*
School of Electronic, Electrical and Systems Engineering
Loughborough University
Leicestershire, LE11 3TU, U.K.
a.daniyan@lboro.ac.uk

***Abstract*—The Bayesian approach to target tracking has proven to be successful in the tracking of multiple targets in various application contexts. This paper applies sequential Monte Carlo (SMC) filtering techniques such as the Markov Chain Monte Carlo particle filter (MCMC PF) and the SMC probability hypothesis density (PHD) filter as suboptimal Bayesian solutions to multi-target tracking (MTT) in video. The MCMC PF by virtue of its information-centric property, can automatically explore the posterior distribution at each sampling step making it possible to track multiple targets. In doing so, it propagates the full multi-target posterior. The SMC PHD filter however propagates only the first order moment of the multi-target posterior density thereby making it computationally less intensive. A comparison of both filters was carried out in tracking multiple human targets in a video scene demonstrating superior performance by the SMC PHD filter in a realistic scenario. The SMC PHD filter was seen to have higher performance than the MCMC PF in terms of the number of particles, the processing speed, and the tracking performance for multiple targets.**

Keywords - MTT; SMC PHD filter; Multi-target tracking; MCMC PF.

I. Introduction

Object tracking is of key importance in computer vision and machine intelligence and has various applications in areas such as motion-based recognition, automated security and surveillance, medical imaging, traffic control, and human computer interaction [1], [2]. The problem of MTT entails correctly detecting, identifying and finding the location of targets of interest with the aid of noisy measurements obtained from various sensors. These sensors include laser-based tracking systems, depth sensors and video cameras. The latter provides diversity of information in recordings which can be more helpful in identifying, and differentiating different targets thereby making it a more preferred source for obtaining measurements in most tracking applications [3]. Video MTT in the context of people tracking is already available in the literature and has several applications. Such applications include anormaly detection, unusual events such as accidents or sudden movements in monitored areas [3], people counting for data analysis [4], gesture recognition, and client activity analysis in retail environments [5], [6].

In the literature, there are a number of proposals that have been used in applying individually, the MCMC PF and SMC PHD filter to multiple target tracking in vision. In the MCMC PF case, [7] and [8] implemented the MCMC PF to track multiple targets in video with emphasis on occlusion. Ref. [4], the focus was on target interaction without data association while [9] presented the MCMC PF within the context of intense clutter. However, in all of these application considerations, the full multi-target posterior is propagated. The SMC PHD filter has also been implemented in different application considerations in various vision based contexts such as in [10]–[15], and [16] with only the first order moment of the multi-target posterior being propagated. In all the above literatures, both filters were shown to achieve tracking.

The objective of the current research is to investigate side by side the performance of MCMC PF and SMC PHD filter in the tracking of multiple human targets in an enclosed environment to determine which filter gives superior performance. It is assumed that observations are already associated with their respective targets and thus data association is not considered. The rest of the paper is organized as follows. In Sec.II, firstly we highlight the main components of a video tracker; secondly we discuss the concept of background subtraction as used in our problem. In Sec.III, we present the dynamical models used in making inference about targets in tracking, introduce and explain the Bayesian approach to MTT, and finally discuss the description and implementation of both the MCMC PF and the SMC PHD filter. Next, Sec. IV presents, compares, analyses and discuss results obtained from the MATLAB implementation of the two filter in tracking four targets in video sequence from the CAVIAR datasets. Finally, conclusions are drawn in Sec.V.

II. Target Tracking in Video

In this section, the main components of a target tracker in video are highlighted and briefly explained.The background subtraction technique is also presented and its implementation explained as the target detection mechanism.

978-1-4799-7412-2/14 $31.00 © 2014 IEEE
DOI 10.1109/EMS.2014.65

A. Components in Video Tracking

In order to achieve tracking in video, the following have been used by researchers in the field [17]:

(i) *target representation*: In order to be able to successfully track a target of interest, it is imperative that the target in question be described so the tracker knows what to look for. Targets are usually described through *feature extraction* i.e. selecting those features that are unique to the target. Feature extraction helps to distinguish targets of interest from others. Once a target has been described, it can then be represented. Targets can be represented by their characteristics such as shape and appearance. The target representation approach we used was the *point* representation method.

(ii) *target detection*: Once the tracker knows what particular feature(s) to look for and how to represent them when seen, the next stage is for the tracker to search for and represent those feature(s) when seen. This is referred to as target detection. We implemented the background subtraction method as the target detection mechanism.

(iii) *target tracking*: The next stage after detecting the target of interest is to track the target so as to obtain some information about the target such as target location, position and trajectory over time. The *point* tracking method was used in our approach. More details on target representation, detection and tracking methods can be found in [2], [17].

B. Background Subtraction

Target detection can be achieved by building a representation of the scene and then finding deviations from the model for each incoming frame. This scene representation is called the background model. Any significant change within an image region from the background model denotes a moving target. The pixels constituting the regions undergoing change are marked for further processing. This process is known to as the background subtraction [2].

In the context of our problem, it was assumed that the camera available for video recording is stationary. With this assumption, a relatively constant background is guaranteed. This assumption made the background subtraction method desirable for in problem.

Now we discuss how the background subtraction method was implemented. Given that **I** is a video sequence containing K images as frames:

$$\mathbf{I} = \left\{IMAG^1_{RGB}, IMAG^2_{RGB}, ..., IMAG^K_{RGB}\right\} \quad (1)$$

where $(\cdot)_{RGB}$ denotes an image with its RGB components, the background model was taken to be the mean of the first few frames in **I** and computed as:

$$\mathbf{BG} = \frac{1}{n}\sum_{k=1}^{n} IMAG^i_g \quad (2)$$

where $(\cdot)_g$ denotes the gray scale component of an image and $n < K$. The gray component was used so as to ignore the effect colour distribution and intensity in the video sequence and shift focus to image contrast instead. This approach was employed so as to tackle the lighting level problem.

Once the background model was obtained, it was compared to each frame in the video sequence through frame differencing (or background subtraction) to obtain the foreground image **FG**. The foreground image is the frame showing the target without the background. The foreground images were obtained using the expression below:

$$\mathbf{FG}_k = \mathbf{BG} - IMAG^k_g \qquad for\ all\ k = 1, 2, ..., K. \quad (3)$$

In order to mitigate the effect of clutter and other false

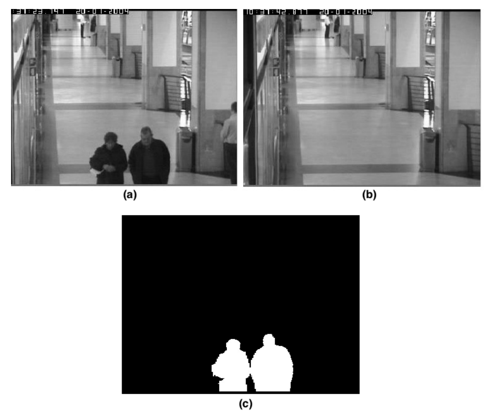

Figure 1. Background subtracted scene.

detections, a Gaussian low-pass filter was used to filter out high frequency components within the foreground image and then compared with a threshold value at each background subtraction step. This threshold value distinguishes detected targets from other detections. Next, the centroids of targets are obtained and stored. These stored detections were then used with the two filters to track the targets. Figure 1 shows results from a typical background subtraction technique. Figure 1 (a) is the foreground and background image showing two targets, Figure 1 (b) is the background model in gray scale and Figure 1 (c) is the foreground image after background subtraction and Gaussian low-pass filtering.

III. Multi-target Tracking Problem Formulation

This section discusses the dynamical models used in making inference about target, the Bayesian approach to MTT, the MCMC PF description and implementation, and the SMC PHD filter description and implementation.

A. Dynamical Systems

A system whose physical process can be mathematically modelled as it changes or evolves over time is known as a dynamical system. In making inference for such a system, two models will be considered, the system model and the measurement model. The problem of tracking in video can be related to dynamical systems.

Consider the non-linear system governed by the state evolution model:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \tag{4}$$

$$\boldsymbol{X}_k = \left\{\mathbf{x}_k^1, \mathbf{x}_k^2, ...\mathbf{x}_k^M\right\} \subset E_S \tag{5}$$

where k = 1,2,... is the time instant of the discrete model, $\mathbf{f}_k$: $\Re^{n_x} \times \Re^{n_v} \longrightarrow \Re^{n_x}$ is a function of $\mathbf{x}_{k-1}$, $\{\mathbf{v}_k, k \in \mathbb{N}\}$ is an independent and identically distributed (i.i.d.) process noise vector, n_x, n_v are dimensions of the state and process noise vectors respectively, with $\mathbb{N}$ being a set of natural numbers. $\boldsymbol{X}_k$ is the joint state vector as a concatenation of the state vectors of all targets, M is the number of targets present at time k and E_S is the state-space [18].

The non-linear system's measurement model is given as:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{6}$$

$$\mathbf{Z}_k = \left\{\mathbf{z}_k^1, \mathbf{z}_k^2, ...\mathbf{z}_k^N\right\} \subset E_O \tag{7}$$

where $\mathbf{h}_k$: $\Re^{n_x} \times \Re^{n_v} \longrightarrow \Re^{n_z}$ is a function, $\{\mathbf{n}_k, k \in \mathbb{N}\}$ is an i.i.d. process noise vector, n_x, n_v are dimensions of the state and process noise vectors respectively, with $\mathbb{N}$ being a set of natural numbers. $\boldsymbol{X}_k$ is the joint measurement vector is denoted as the concatenation of the measurement vectors of all targets, N is the number of measurements received at time k and $\mathbf{Z}_k$ is a subset of the observation space E_O [18].

B. The Bayesian Tracker

The Bayesian approach to target tracking is the process of recursively computing the posterior distribution using $prdiction$ and $update$ stages. The prediction stage uses the system model to predict the state probability density function (pdf) forward from one measurement time to the next. While the update operation uses the latest measurement to modify the prediction pdf. It may be recalled from Bayes' theorem that given the likelihood and prior, the posterior can be computed. The tracking problem from a Bayesian perspective is to calculate recursively some degree of belief in the state $\boldsymbol{X}_k$ at time k. The construction of a pdf $p(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k})$ is thus required. This is also known as the posterior of the pdf given measurements up to time k. The initial state $p(\boldsymbol{X}_0|\mathbf{Z}_0) \equiv p(\boldsymbol{X}_0)$ is assumed to be the prior. Then, the posterior $p(\boldsymbol{X}_k|\mathbf{Z}_{1:k})$ can be obtained recursively using the $prediction$ and $update$ stages.

Assuming that the required pdf at time $k-1$, $p(\boldsymbol{X}_{k-1}|\mathbf{Z}_{1:k-1})$ is available, using the system model in (4), the $prediction$ stage requires computing the $prior$ pdf at time k using the Chapman-Kolmogorov equation [19]

$$p(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1}) = \int p(\boldsymbol{X}_k|\boldsymbol{X}_{k-1})p(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{1:k-1})d\boldsymbol{X}_{k-1} \tag{8}$$

The fact that $p(\boldsymbol{X}_k|\boldsymbol{X}_{k-1}, \boldsymbol{Z}_{1:k-1}) = p(\boldsymbol{X}_k|\boldsymbol{X}_{k-1})$ has been made use of in (8) above since (4) describes a Markov process of order one.

The $update$ stage is where the posterior, $p(\boldsymbol{X}_k|\mathbf{Z}_{1:k})$ is computed. This requires updating the $prior$ at time k when a measurement $\mathbf{Z}_k$ become available. Applying Bayes' rule,

$$p(\boldsymbol{X}_k|\mathbf{Z}_{1:k}) = \frac{p\mathbf{Z}_k|\boldsymbol{X}_k)p(\boldsymbol{X}_k|\mathbf{Z}_{k-1})}{p(\mathbf{Z}_k|\mathbf{Z}_{k-1})} \tag{9}$$

where

$$p(\mathbf{Z}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{Z}_k|\boldsymbol{X}_k)p(\boldsymbol{X}_k|\mathbf{Z}_{k-1})d\boldsymbol{X}_{k-1} \tag{10}$$

is the normalizing constant, $p(\mathbf{Z}_k|\boldsymbol{X}_k)$ is the likelihood function defined by (6).

The approximation to the reoccurring relation of (8) and (9) gives rise to the $suboptimal\ Bayesian\ solution.$

C. Markov Chain Monte Carlo Particle Filter

MCMC methods have been applied by researchers to the tracking problem due to the limitations of importance sampling in high dimensional state spaces. MCMC methods work by defining a Markov Chain over a space of set $\boldsymbol{X}_k$, such that the stationary distribution $\pi(\boldsymbol{X}_k)$ of the chain is equal to the sought posterior $p(\boldsymbol{X}_k|\mathbf{Z}_{1:k})$. Typically, MCMC methods have been applied as a search technique to obtain a maximum a posteriori estimate, or as a way to diversify samples inside the framework of traditional particle filters. Maximum a posteriori estimates can be obtained by taking the most likely sample generated from the Markov Chain. More generally, MCMC is a method intended to produce a sample of a distribution, and no guarantees exist about it yielding good point estimates [20].

1) MCMC PF Description: Alg. 1 shows the pseudo code of the MCMC PF implemented in our problem. This algorithm uses the $Metropolis-Hastings$ sampling method in the update stage of the filter. The algorithm was adapted from [20], [21]. The algorithm is summarized as follows:

- At each time instant k, initialize the Markov chain. Propagate the particles $\{\boldsymbol{X}_{k-1}\}_{i=1}^{N_s}$ and randomly choose one propagated particle as the initial state to start the chain.
- Repeat the following for $i = 2, 3, ...,B{+}MN_s$ times, where B is the burn in period and M is the thinning interval.
- Given the current state $\bar{\boldsymbol{X}}_k$, randomly pick the target to be perturbed, denoted as $\bar{\mathbf{x}}_k^j$, and generate a sample $\bar{\mathbf{x}}_k^j\prime$, by sampling from the proposal density $q(\bar{\mathbf{x}}_k\prime;\bar{\mathbf{x}}_k)$. The proposal state vector has the component corresponding to the target states of $\bar{\mathbf{x}}_k^j$ updated while other components unchanged.
- Calculate the acceptance rate α. If $1 \leq \alpha$, set the next state to be $\bar{\boldsymbol{X}}_k\prime$; otherwise set it as $\bar{\boldsymbol{X}}_k$.

- Discard the first B samples of the iterations and use the rest MN_s particles $\{\boldsymbol{X}_k\}_{i=1}^{N_s}$ as the discrete approximation to the posterior.

Algorithm 1 MCMC PF filter

for $k = 1 : K$ **do**
  prediction
  initialize $p(\boldsymbol{X}_k|\boldsymbol{Z}_k) = \{\boldsymbol{X}_k\}_{i=1}^{N_s}$ with *unweighted* samples as:
  for $i = 1 : N_s$ **do**
    sample particles $\boldsymbol{X}_k^i \sim q(\boldsymbol{X}_k|\boldsymbol{X}_{k-1}^i, \boldsymbol{Z}_k)$ ▷ i.e. draw from the importance density
  end for
  randomly select a sample $\hat{\boldsymbol{X}}_k$ from $\{\boldsymbol{X}_k\}_{i=1}^{N_s}$
  move all targets $\hat{\mathbf{x}}_k^j$ in $\hat{\boldsymbol{X}}_k$ according to STT motion model, $f_{k|k-1}(\cdot|\mathbf{x}_{k-1}^i)$ to get $\bar{\boldsymbol{X}}_k$
  ▷ where $j = 1, 2, ...m$, m is number of targets in $\hat{\boldsymbol{X}}_k$. $\bar{\boldsymbol{X}}_k$ will be the initial state of the Markov Chain
  evaluate and cache likelihood $w_k^j = g(\boldsymbol{Z}_k|\bar{\boldsymbol{X}}_k)$ for each $\bar{\mathbf{x}}_k^j$ in $\bar{\boldsymbol{X}}_k$
  update
  for $i = 1 : B + MN_s$ **do**
    randomly pick a target $\bar{\mathbf{x}}_k^j$ from $\bar{\boldsymbol{X}}_k$ and propose a new state $\bar{\mathbf{x}}_{k'}^j$, to form $\bar{\boldsymbol{X}}_{k'}$ by sampling from $q(\bar{\mathbf{x}}_{k'}; \bar{\mathbf{x}}_k)$
    compute acceptance ratio $\alpha = \frac{proposed\ state\ likelihood}{randomly\ picked\ target\ likelihood}$
    if $\alpha \geq 1$ **then**
      accept proposal, $\bar{\boldsymbol{X}}_k = \bar{\boldsymbol{X}}_{k'}$
      update cached weight $w_k^j = g(\boldsymbol{Z}_k|\bar{\boldsymbol{X}}_k)$
    else
      leave $\bar{\boldsymbol{X}}_k$ unchanged
      $w_k^j = \alpha$
    end if
  end for
  discard all B iterations
  $\{\boldsymbol{X}_k\}_{i=1}^{N_s} = \{\bar{\boldsymbol{X}}_k\}_{i=1}^{N_s}$
end for

D. Probability Hypothesis Density

1) RFS and FISST: In MTT, varying number of targets are present. The targets can appear and disappear randomly in the state-space. $\boldsymbol{X}_k$ of (5) holds for all targets present in the state space E_S at time step k and $\mathbf{Z}_k$ in (7) holds for all measurements received in the observation space E_O at time k. However, some measurements $\mathbf{z}_{k,j} \in \mathbf{Z}_k$ may not necessarily originate from $\mathbf{x}_{k,i} \in \boldsymbol{X}_k$ and may be due to clutter. Such spurious measurements can be modelled using specific clutter models [18].

2) Random finite sets: A Random finite set (RFS) $\mathbf{X}$ is a finite-set valued random variable, which can be described by a discrete probability distribution and a family of joint probability densities [22]. RFS are models used to represent the uncertainty about the number of elements in multiple target state $\boldsymbol{X}_k$ and measurement state $\mathbf{Z}_k$ [23].

3) Finite-set statistics: Finite-set statistics (FISST) represent a mathematical framework which transforms multisensor-multitarget problems into single-sensor single-target problems mathematically by bundling all sensors into a single meta-sensor, all targets into a single meta-target and all observations into a single meta-observation [24].

The PHD

For a given RFS Ξ, its first order moment is its probability hypothesis density (PHD), D_Ξ and is given as [18]:

$$D_\Xi(\mathbf{x}) = \mathbf{E}\{\delta_\Xi(\mathbf{x})\} = \int \delta_\mathbf{X}\mathbf{x}P_\Xi(d\mathbf{X}) \tag{11}$$

where $\mathbf{E}\{\cdot\}$ is the expectation operator and $\delta_\Xi(\mathbf{x}) = \sum_{\mathbf{x}\in\Xi} \delta_\mathbf{x}$ is the random density representation of Ξ. P_Ξ is the probability distribution of the RFS. The PHD has the properties that [18],

- the integral of the PHD, $\int_S D_\Xi(\mathbf{x})\lambda(d\mathbf{x})$ is the expected number of targets in the measurable region S, and
- the peaks of the PHD function give the estimates of the target states.

The PHD filter

The PHD filter is a recursion of the PHD $D_{k|k}$ that is associated with the multiple target posterior density $p(\boldsymbol{X}_k|\mathbf{Z}_{1:k})$. Given that the Ξ is Poisson, the recursion for propagating $D_{k|k}$ is given as [24]:

$$D_{k|k} = \left[D_{k|k}(\mathbf{x}_k|\mathbf{Z}_{1:k}) \circ D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1})\right](D_{k-1|k-1}) \tag{12}$$

where $D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$ and $D_{k|k}(\mathbf{x}_k|\mathbf{Z}_{1:k})$ are prediction and update operators for the PHD respectively, $\circ$ denotes the composition function.

The prediction term is given as [25]:

$$D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \gamma_k(\mathbf{x}_k) + \int \phi_{k|k-1}(\mathbf{x}_k, \mathbf{x}_{k-1})D_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})d\mathbf{x}_{k-1}. \tag{13}$$

with the term

$$\phi_{k|k-1}(\mathbf{x}_k, \mathbf{x}_{k-1}) = p_S(\mathbf{x}_{k-1})f_{k|k-1}(\mathbf{x}_k, \mathbf{x}_{k-1}) + b_{k|k-1}(\mathbf{x}_k, \mathbf{x}_{k-1}) \tag{14}$$

where $\gamma_k(\cdot)$ is the PHD of the spontaneous birth, $p_S(\cdot)$ is the probability of the target survival, $f_{k|k-1}(\mathbf{x}_k, \mathbf{x}_{k-1})$ is the single target motion model, and $b_{k|k-1}(\mathbf{x}_k, \mathbf{x}_{k-1})$ is the PHD of spawned targets.

The update term is given as:

$$D_{k|k}(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \left[\nu(\mathbf{x}_k) + \sum_{\mathbf{z}\in\mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}}(\mathbf{x}_k)}{\kappa_k(\mathbf{z}) + \langle D_{k|k-1}, \psi_{k,\mathbf{z}}\rangle}\right] \times D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) \tag{15}$$

with

$$\nu(\mathbf{x}_k) = 1 - p_D(\mathbf{x}_k), \tag{16}$$

$$\psi_{k,\mathbf{z}}(\mathbf{x}_k) = p_D(\mathbf{x}_k) g(\mathbf{z}|\mathbf{x}_k), \tag{17}$$

$$\kappa_k(\mathbf{z}) = \lambda_k c_k(\mathbf{z}) \tag{18}$$

where p_D is the probability of detection, $\nu(\mathbf{x}_k)$ is the probability of target non-detection of state $\mathbf{x}_k$, $g(\mathbf{z}|\mathbf{x}_k)$ is the measurement likelihood function for single target, $\kappa_k(\mathbf{z})$ is the clutter intensity, λ_k is average number of Poisson clutter points per scan, and $c_k(\mathbf{z})$ is the probability density over the state-space of clutter point; $\langle \cdot, \cdot \rangle$ denotes inner product and is computed as [25]

$$\langle D_{k|k-1}, \psi_{k,\mathbf{z}} \rangle = \int D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) \psi_{k,\mathbf{z}}(\mathbf{x}_k) d\mathbf{x}_k. \tag{19}$$

SMC PHD filter implementation

The PHD filter is computationally less intensive as it propagates only the first moment of the full multi target posterior in time instead of the whole multi target posterior. Utilizing the SMC approach to approximate the PHD reduces the computational complexity further [19]. The implementation presented here was adapted from [25]–[27]. It was assumed that all measurement data were associated with a target with zero clutter intensity. The pseudo code of the SMC PHD filter is illustrated in Alg. 2. The SMC PHD filter algorithm is summarized as follows:

- At time step $k = 0$, L_k particles $\mathbf{x}_k^i$ with associated weights w_k^i are initialized from a given prior PHD $D_{0|0}$ with $\hat{T}_0$ targets present.
- After the initialization process is complete, iteration over the measurement set received begins. For each time step $0 < k \leq K$, the PHD prediction, update and resampling steps are performed.
- The prediction step predicts the PHD represented by the particles states and weights. Particles that have survived from the previous time-step are predicted using the single-target motion model. Furthermore, J_k newborn particles are injected into the system. Like in the initialization step, if no assumption about the target distribution is made the newborn particles are drawn from a uniform density across the whole state-space.
- The update of the particle PHD takes the current measurement set $\mathbf{Z}_k$ into account. All predicted particle states are evaluated against all measurements within the RFS $\mathbf{Z}_k$ and reweighted accordingly. Non-detected targets are accounted for by adding the intensity of non-detected targets $\nu(\mathbf{x})$ to the measurements. To filter clutter, which might be present in the measurements, particle weights are influenced by the clutter PHD intensity $\kappa_k(\mathbf{z})$. The result of this step is a set of updated weights for all particles within the system. The updated particle weights approximate the present multi-target PHD. However, when no measurements are available, the update step of is skipped.
- The resampling step is performed to avoid degeneracy. Here, resampling is performed in every time-step. The resampling method used was systematic resampling. First, the cummulative distribution function (cdf) is built based on the current particle distribution in state-space. Next, a set of new particles is generated. Therefore, L times the number of estimated targets particles are randomly drawn based on their likelihood described by the cdf.

Algorithm 2 SMC PHD filter

$k = 0$;
Initialization:
the posterior PHD $D_{0|0}$ is represented by a set of particles with their associated weights $\{\mathbf{x}_0^i, w_0^i\}_{i=1}^{L_k}$
for $k = 1 : K$ **do**
  Prediction:
  for $i = 1 : L_{k-1}$ **do**
    sample $\tilde{\mathbf{x}}_k^i \sim q_k(\cdot|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
    compute predicted weights:
    $\tilde{w}_{k|k-1}^i = \frac{\phi_{k|k-1}(\tilde{\mathbf{x}}_k^i, \mathbf{x}_{k-1}^i)}{q_k(\cdot|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} \tilde{w}_{k-1}^i$
  end for
  for $i = 1 : L_{k-1} + 1, \ldots L_{k-1} + J_k$ **do**
    sample $\tilde{\mathbf{x}}_k^i \sim p_k(\cdot|\mathbf{z}_k)$
    compute newborn particles' weight:
    $\tilde{w}_{k|k-1}^i = \frac{\gamma_k(\tilde{\mathbf{x}}_k^i)}{J_k}$
  end for
  Update:
  For each $\mathbf{z} \in \mathbf{Z}_k$, compute:
  $C_k(\mathbf{z}) = \sum_{i=1}^{L_k+J_k} \psi_{k,\mathbf{z}}^i(\tilde{\mathbf{x}}_k^i) \tilde{w}_{k|k-1}^i$
  for $i = 1 : L_{k-1} + J_k$ **do**
    update weights:
    $\tilde{w}_k^i = \left[\nu(\tilde{\mathbf{x}}_k^i) + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}}^i(\tilde{\mathbf{x}}_k^i)}{\kappa_k(\mathbf{z}) + C_k(\mathbf{z})}\right] \tilde{w}_{k|k-1}^i$
  end for
  Resample:
  compute estimated no. of targets $\hat{T}_{k|k} = \sum_{i=1}^{L_k+J_k} \tilde{w}_k^i$
  resample
  $\left\{\tilde{\mathbf{x}}_k^i, \tilde{w}_k^i, \hat{T}_{k|k}\right\}_{i=1}^{L_{k-1}+J_k}$ to obtain $\left\{\mathbf{x}_k^i, w_k^i, \hat{T}_{k|k}\right\}_{i=1}^{L_k}$
end for

IV. Results and Comparisons

This section presents and discusses the results obtained from the MATLAB implementation of the filters. The relative performance of all two filter types is also discussed.

A. Simulation Setup

The experiment was conducted on a video sequence obtained from [28] in the context of a shopping mall as the enclosed environment. The video sequence with title

$EnterExitCrossingPaths2cor.mpg$ was used. The video sequence used had a total of 475 frames. In this sequence, a total of four targets was present but with a maximum of three at a time. The targets are seen walking along the corridor of a shopping mall.

The simulation of the target tracker in video was processed in the MATLAB environment on an Intel Core i5 laptop with 4GB Memory @ 1600MHz.

The targets present in a frame at time index k were represented as $\boldsymbol{X}_k$, were $\boldsymbol{X}_k$ is the joint state vector as a concatenation of the state vectors of all targets. This gives:

$$\boldsymbol{X}_k = \left\{\mathbf{x}_k^1, \mathbf{x}_k^2, ...\mathbf{x}_k^M\right\} \subset E_S$$

The system evolution model of each target $\mathbf{x}_k^i$ in $\boldsymbol{X}_k$ follows the system evolution model:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{v} \tag{20}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \delta t & 0 \\ 0 & 1 & 0 & \delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{v} = \mathcal{N}(\mathbf{0}, \sigma_x^2\mathbf{I})$$

where δt is velocity in either x or y direction, σ_x^2 is the process noise variance, and $\mathbf{I}$ is a 4x4 identity. A constant velocity model was adopted consequently making δt = 1. The process noise σ_x^2 used here was 2. The target state representation $\mathbf{x}_k$ is given as:

$$\mathbf{x}_k = [x, y, 1, 1]^T \tag{21}$$

The process noise σ_x^2 used here was 2.

The measurement model used was:

$$\boldsymbol{Z}_k = \left\{\mathbf{z}_k^1, \mathbf{z}_k^2, ...\mathbf{z}_k^N\right\} \subset E_O$$

where $\boldsymbol{Z}_k$ is a joint measurement vector that is a concatenation of the measurement vectors of all targets. The measurement vector $\mathbf{z}_k^i$ earlier described in (**??**) became

$$\mathbf{z}_k = \bar{\mathbf{z}}_k = [x, y, 1, 1]^T \tag{22}$$

This was as a result of not considering the effective target velocity.

For the purpose of MATLAB implementation, the following assumptions were made:

- Each measurement is associated with their respective targets,
- The number of measurements available at each time is equivalent to number of targets present, and
- No clutter is present in measurements because the background subtraction detections were low pass filtered with Gaussian filter.

The following are the filter specific parameters set:

MCMC PF

A B+MN MCMC sampling step was performed on each particle selected together with computing its acceptance ratio. B which is the number of burn-in samples was selected to be 50, and the thinning interval M = 0.5. The MCMC PF was used in tracking the target for varying number of particles N and measurement noise σ_z^2. An MCMC sampling was perform on each member $\mathbf{x}_k^j$ of selected set $\boldsymbol{X}_k^i$ for varying number of particles under different measurement noise conditions. The results obtained are presented in Tables I, II, and III. A plot showing the MCMC PF track of the true trajectory of all four targets is shown in Figures 3 (a) and (b). The plots were split into two for clarity.

SMC PHD filter

In the SMC PHD filter, the importance density was chosen to be the prior of individual targets. J_k which is the number of samples to be used for new born target was selected to be 10. The probability of target detection p_D and the probability of target survival p_S were both set as 1. The average number of Poisson clutter points per scan λ_k and clutter probability density c_k were both set as 0. The values for p_D, p_S, c_k, and λ_k were all time invariant. Simulation was run on the video sequence using the SMC PHD filter with varying number of particles L_k under varying measurement conditions and the results obtained are shown in Tables I, II, and III. A plot showing the SMC PHD filter track of the true trajectory of all four targets is shown in Figures 4 (a) and (b). The plots were split into two for clarity.

B. Performance Analysis

Figure 2 shows screen shots of the targets being tracked the in video sequence. In the Figure, the targets are seen walking along the corridor of a shop. An is ellipse drawn around each target for visualization, blue colour representing the ground truth and red, the tracking. Figures 3 and 4 show

Figure 2. MTT: Screen grab showing targets being tracked.

the MCMC PF and SMC PHD filter trajectory trackings respectively of the four targets relative to the targets' ground

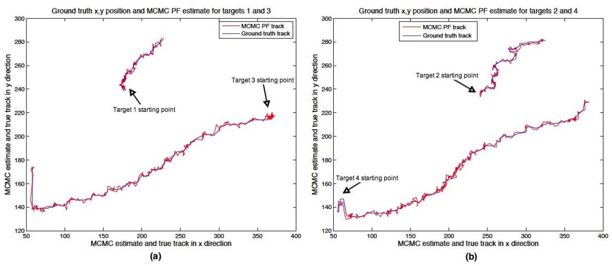

Figure 3. MTT: MCMC PF tracking trajectory of all 4 targets.

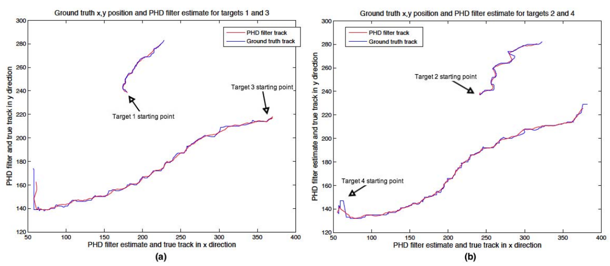

Figure 4. MTT: SMC PHD filter tracking trajectory of all 4 targets.

truths. On the one hand, it can be observed from Figure 3 that the MCMC PF track of the targets appears to be 'wild' even though the tracking was achieved. The SMC PHD filter track on the other hand appears to give a much smoother tracking of the target trajectories as observed from Figure 4. Table I shows the RMSE values of both filters for different number of particles and the computation time for all for targets. The RMSE values of the targets remained fairly constant for the MCMC PF as the number of particles were increased with the computation time increasing. With the SMC PHD filter however, the filter performance was observed to significantly increase as the number of particles increases. The computation time increases with number of particles as well but less when compared to the MCMC PF.

Table I
MTT: FILTER PERFORMANCE WITH MEASUREMENT NOISE, $\sigma_z^2 = 0.5$.

Filter	**Particles**	**RMSE**				**Time** (s)
		Target 1	Target 2	Target 3	Target 4	
MCMC PF	100	0.12	0.14	0.28	0.31	3.06
	500	0.12	0.13	0.27	0.30	9.25
	1000	0.11	0.13	0.27	0.30	16.90
	4000	0.11	0.12	0.27	0.30	63.40
PHD	100	0.01	0.02	0.72	0.20	2.00
	500	0.003	0.01	0.40	0.11	7.21
	1000	0.003	0.008	0.32	0.10	13.82
	4000	0.002	0.006	0.18	0.07	53.10

In Table II, similar observations can be made about both filters. Observing Table III, it is seen that under relatively high noise, the SMC PHD filter performs almost as well as it does under lower measurement noise conditions.

Overall, from observing Tables I, II, and III, it can be deduced that

- As measurement noise increases, the MCMC PF tracking performance decreases while the SMC PHD filter still performs optimally.
- Under a given measurement noise condition the MCMC PF saturates irrespective of the number of particles used.
- The SMC PHD filter gives best performance in tracking multiple targets when compared to the MCMC PF and is more robust to measurement noise.

Table II
MTT: FILTER PERFORMANCE WITH MEASUREMENT NOISE, $\sigma_z^2 = 1$.

Filter	**Particles**	**RMSE**				**Time** (s)
		Target 1	Target 2	Target 3	Target 4	
MCMC PF	100	0.21	0.28	0.55	0.59	3.14
	500	0.22	0.29	0.54	0.59	9.25
	1000	0.23	0.28	0.53	0.60	17.00
	4000	0.24	0.28	0.52	0.60	62.83
PHD	100	0.01	0.05	0.92	0.30	1.74
	500	0.01	0.03	0.58	0.18	7.32
	1000	0.008	0.02	0.40	0.14	13.51
	4000	0.008	0.02	0.27	0.13	53.37

Table III
MTT: FILTER PERFORMANCE WITH MEASUREMENT NOISE, $\sigma_z^2 = 2$.

Filter	**Particles**	**RMSE**				**Time** (s)
		Target 1	Target 2	Target 3	Target 4	
MCMC PF	100	0.51	0.52	1.12	1.28	3.13
	500	0.50	0.51	1.14	1.27	9.37
	1000	0.46	0.50	1.14	1.27	17.14
	4000	0.45	0.48	1.10	1.27	63.58
PHD	100	0.04	0.07	1.11	0.48	1.72
	500	0.03	0.06	0.68	0.33	7.10
	1000	0.03	0.06	0.48	0.30	13.80
	4000	0.03	0.06	0.40	0.28	53.30

It is the general belief with SMC methods that as the number of particles increases, so does the accuracy of the posterior density being approximated, with the cost of computational complexity. However, it has been shown here that:

i An SMC filter can reach saturation such that increasing number of particles won't matter. This was the case with MCMC PF.

ii In the SMC PHD filtering, performance was seen to increase as the number of particles was increased. This seemed to be the only case which agrees with the general belief on SMC methods.

V. CONCLUSIONS

Two SMC filtering techniques, the MCMC PF and SMC PHD filter have been presented and implemented to track multiple targets in an enclosed environment. Algorithms used to achieve tracking have also been shown. Overall, it was seen that as measurement noise increases, the MCMC PF tracking performance decreases while the SMC PHD filter still performs optimally. Under a given measurement noise condition the MCMC PF saturates irrespective of the number of particles used. The SMC PHD filter gives best performance in tracking multiple targets when compared to the MCMC PF even under relatively high measurement noise. There are various potential research avenues along which this work could be extended. Firstly, it would be of great practical interest to consider more sophisticated measurement models. Secondly, it would be interesting to use a non-Gaussian distribution having more pronounced tails as the importance distribution to accommodate non-Gaussian priors in a more challenging problem.

ACKNOWLEDGMENT

This work is supported by the National Information Technology Development Agency (NITDA), Nigeria, under its overseas Education Trust Fund scholarship scheme, NITDEF.

REFERENCES

[1] P. N. Trung, "Tracking of multiple objects using the PHD filter," Ph.D. dissertation, National University of Singapore, 2007.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM computing surveys (CSUR)*, vol. 38, no. 4, 2006.

[3] Ata-Ur-Rehman, "Bayesian-based techniques for tracking multiple humans in an enclosed environment," Ph.D. dissertation, Loughborough University, 2014.

[4] X. Zhao, E. Delleandrea, and L. Chen, "A people counting system based on face detection and tracking in a video," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. IEEE, 2009, pp. 67–72.

[5] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[6] N. Padoy and G. D. Hager, "Deformable Tracking of Textured Curvilinear Objects." in *BMVC*, 2012, pp. 1–11.

[7] T. Osawa, X. Wu, K. Sudo, K. Wakabayashi, H. Arai, and T. Yasuno, "MCMC based multi-body tracking using full 3D model of both target and environment," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, Sept 2007, pp. 224–229.

[8] K. Tanaka and E. Kondo, "Vision-based multi-person tracking by using MCMC-PF and RRF in office environments," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1, Sept 2004, pp. 637–642 vol.1.

[9] A.-T. Vu, B.-N. Vo, and R. Evans, "Particle Markov Chain Monte Carlo for Bayesian multi-target tracking," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, July 2011, pp. 1–8.

[10] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. Intl Conf. on Information Fusion*, 2003, pp. 792–799.

[11] Wood, T.M. and Yates, C.A and Wilkinson, D.A and Rosser, G., "Simplified Multitarget Tracking Using the PHD Filter for Microscopic Video Data," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 5, pp. 702–713, May 2012.

[12] T. Wood, C. Yates, D. Wilkinson, and G. Rosser, "Simplified Multitarget Tracking Using the PHD Filter for Microscopic Video Data," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 5, pp. 702–713, May 2012.

[13] W. Li, C. Han, X. Yan, and J. Liu, "Adaptive sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Information Fusion (FUSION), 2013 16th International Conference on*, July 2013, pp. 23–29.

[14] Y.-D. Wang, J.-K. Wu, A. Kassim, and W. Huang, "Data-driven probability hypothesis density filter for visual tracking," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 8, pp. 1085–1095, Aug 2008.

[15] N. Ikoma, H. Hasegawa, and Y. Haraguchi, "Multi-target tracking in video by SMC-PHD filter with elimination of other targets and state dependent multi-modal likelihoods," in *Information Fusion (FUSION), 2013 16th International Conference on*, July 2013, pp. 588–595.

[16] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, "Particle PHD Filtering for Multi-Target Visual Tracking," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, April 2007, pp. I–1101–I–1104.

[17] D. E. Maggio and D. A. Cavallaro, *Video Tracking: Theory and Practice*, 1st ed. Wiley Publishing, 2011.

[18] K. Panta, B.-N. Vo, S. Singh, and A. Doucet, "Probability hypothesis density filter versus multiple hypothesis tracking," in *Defense and Security*. International Society for Optics and Photonics, 2004, pp. 284–295.

[19] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[20] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 11, pp. 1805–1819, 2005.

[21] Z. Zhao and M. Kumar, "An MCMC-based particle filter for multiple target tracking," in *Information Fusion (FUSION), 2012 15th International Conference on*. IEEE, 2012, pp. 1676–1682.

[22] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4091–4104, 2006.

[23] H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density," in *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, vol. 2, July 2003, pp. 800–806.

[24] R. P. Mahler, "Statistics 101 for multisensor, multitarget data fusion," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 53–64, 2004.

[25] D. E. Clark and J. Bell, "Multi-target state estimation and track continuity for the particle PHD filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, no. 4, pp. 1441–1453, 2007.

[26] T. Zajic and R. P. Mahler, "Particle-systems implementation of the PHD multitarget-tracking filter," in *AeroSense 2003*. International Society for Optics and Photonics, 2003, pp. 291–299.

[27] M. Frohle, "Multi-target Tracking for UWB Channels Using PHD Filters," Master's thesis, Graz University of Technology, Austria, 2011.

[28] University of Edinburgh. (2003,2004) Context Aware Vision using Image-based Active Recognition, (CAVIAR) Test Case Scenarios. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/ Accessed 12/07/2014