A study of factors affecting multiple target tracking with a pixelized sensor

Mark R. Morelande, Christopher M. Kreucher and Keith Kastella*

ABSTRACT

Factors affecting the performance of an algorithm for tracking multiple targets observed using a pixelized sensor are studied. A pixelized sensor divides the surveillance region into a grid of cells with targets generating returns on the grid according to some known probabilistic model. In previous work an efficient particle filtering algorithm was developed for multiple target tracking using such a sensor. This algorithm is the focus of the study. The performance of the algorithm is affected by several considerations. The pixelized sensor model can be used with either thresholded or non-thresholded measurements. While it is known that information is lost when measurements are thresholded, quantitative results have not been established. The development of a tractable algorithm requires that closely-spaced targets are processed jointly while targets which are far apart are processed separately. Selection of the clustering distance involves a trade-off between performance and computational expense. A final issue concerns the computation of the proposal density used in the particle filter. Variations in a certain parameter enable a trade-off between performance and computational expense. The various issues are studied using a mixture of theoretical results and Monte Carlo simulations.

1. INTRODUCTION

In active radar the presence of targets in a region of interest, referred to as the surveillance region, is determined by transmitting radio waves and assessing the strength of signals incident at the receiver. Many target tracking algorithms assume a measurement model in which these return signals are thresholded to give noisy point measurements of possible target locations. Due to the presence of random point-scatterers, referred to as clutter, a high energy return does not necessarily mean the presence of a target. A tracking algorithm must therefore include provisions for resolving this uncertainty regarding the measurement origin. This is done in an optimal manner by data association in which the probability of each possible measurement origin hypothesis is evaluated. Since the number of possible hypotheses increases exponentially as more data is acquired, techniques are required for managing the hypotheses.[1, 2]

The reduction of the receiver intensities into a set of detections via thresholding involves a loss of information. This limits the performance of tracking algorithms which use such data, particularly in low signal-to-noise ratio (SNR) environments. Some improvement can be achieved by incorporating the measured intensities of the detected measurements.[3] An even better approach is to use the original amplitude information across the entire surveillance region. The pixelized sensor model used in this paper takes this approach.

A pixelized sensor model divides the surveillance region into cells with the measurements consisting of the intensity of the return signal in each cell. Tracking with this sensor model is often referred to as track-before-detect.[4] Despite its potential for providing improved performance, the use of the pixelized sensor model has not become widespread. This can probably be attributed to the difficulty in developing practically useful and computationally tractable tracking algorithms under this model. Earlier techniques, based on maximum likelihood, have had to make restrictive assumptions regarding the target motion.[5, 6] A Bayesian framework is more promising as it would allow considerably more flexibility in the modeling of target motion.

The Bayesian approach adopted here is predicated on computation of the joint multi-target probability density (JMPD).[7] The JMPD captures the available probabilistic information regarding the number of targets present and their states. Similar ideas have been proposed in.[8–10] The main barrier to use of the JMPD, and this applies

*M.R. Morelande is with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010, Australia (email: m.morelande@ee.unimelb.edu.au). C.M. Kreucher and K. Kastella are with General Dynamics- Advanced Information Systems, Ypsilanti, Michigan (emails: [christopher.kreucher][keith.kastella]@gd-ais.com).

Signal and Data Processing of Small Targets 2005, edited by Oliver E. Drummond,
Proc. of SPIE Vol. 5913, 59130L, (2005) · 0277-786X/05/$15 · doi: 10.1117/12.617978

equally well to related Bayesian techniques, is that it cannot be computed exactly in any situation of practical interest. Further, the high dimensionality of the JMPD makes computationally tractable approximation difficult.

The advent of sequential Monte Carlo techniques, or particle filters (PFs), has provided a promising method for approximating the JMPD. PFs are a class of techniques which enable approximately optimal, in the Bayesian sense, recursive state estimation in general stochastic dynamic systems.[11] The posterior distribution of the state is represented by a weighted set of particles. A sequential importance sampling procedure is used to propagate the particles and weights as data is acquired.

In previous work the authors have proposed a PF for approximating the JMPD.[12, 13] The performance of the proposed PF is promising with simulation results demonstrating the ability of the algorithm to detect and track ten targets with a relatively small number of particles. The purpose of this paper is to study several interesting facets of the filter performance which remain unexamined. The issues under consideration here are:

- The algorithm can be applied with either thresholded or non-thresholded measurements. Although it is known that thresholding removes information it is of interest to determine how much, and under which conditions, this removal of information affects performance. This issue is studied for a ten target scenario using Monte Carlo simulations.
- There are two thresholds used in the development of the algorithm. The selection of these thresholds involves a complicated trade-off between performance and computational expense. This trade-off will be studied with a view to determining thresholds which provide a balance between the requirements of good performance and low computational expense. The variance of the weights is used as a criteria of goodness.

The focus here will be on track maintenance, i.e., the number of targets will be assumed known. The algorithm under consideration has been extended to enable initiation as well as maintenance of tracks,[12] but this facet of its performance will not be studied here due to space restrictions.

The paper is organized as follows. The tracking model is defined in Section 2. The algorithm for approximate computation of the JMPD, along with a brief review of the JMPD, is presented in Section 3. Appropriate selection of thresholds is discussed in Section 4. Simulation results presented in Section 5 examine the benefits of non-thresholded vs. thresholded measurements for several SNRs. Conclusions are given in Section 6.

2. NOTATION AND MODELING

Assume a collection of targets moving in s dimensions. Let r^k denote the number of targets present at time kT, where T is the sampling period. The state of the ith target, denoted as $\mathbf{x}_i^k = (\boldsymbol{\rho}_i^{k\prime}, \mathbf{v}_i^{k\prime})' \in \mathbb{R}^{2s}$, $i = 1, \ldots, r^k$, contains target position $\boldsymbol{\rho}_i^k$ and velocity $\mathbf{v}_i^k$ in each direction. The individual target states are collected into the multi-target state $\mathbf{X}^k = (\mathbf{x}_1^{k\prime}, \ldots, \mathbf{x}_{r^k}^{k\prime})'$. Individual target states evolve independently according to

$$\mathbf{x}_i^k | \mathbf{x}_i^{k-1} \sim N\left(\mathbf{F}\mathbf{x}_i^{k-1}, \mathbf{Q}_i^k\right), \tag{1}$$

where $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{Q}_i^k$ is the process noise covariance matrix for the ith target at time kT and

$$\mathbf{F} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \otimes \mathbf{I}_s \tag{2}$$

with ⊗ the Kronecker product and $\mathbf{I}_s$ the $s \times s$ identity matrix. The process noise covariance matrix partitions as

$$\mathbf{Q}_i^k = \begin{pmatrix} \mathbf{Q}_{i,\rho}^k & \mathbf{G} \\ \mathbf{G}' & \mathbf{Q}_{i,v}^k \end{pmatrix} \tag{3}$$

The surveillance region is divided into J cells with the measurement vector $\mathbf{z}^k = (z_1^k, \ldots, z_J^k)'$ containing the intensity measurements obtained in each cell. Measurements are made independently in each cell with the distribution of the measurement in the jth cell depending on the number of targets in the cell. The individual target positions are collected into $\mathbf{P}^k = (\boldsymbol{\rho}_1^{k\prime}, \ldots, \boldsymbol{\rho}_{r_k}^{k\prime})'$ and $o_j(\mathbf{P}^k)$, $j = 1, \ldots, J$ denotes the number of targets occupying the jth cell for a given collection $\mathbf{P}^k$ of target positions. Then the measurement likelihood is

$$p(\mathbf{z}^k|\mathbf{P}^k) = \prod_{j=1}^{J} l_{o_j(\mathbf{P}^k)}(z_j^k). \tag{4}$$

where l_m is the single-cell measurement likelihood. Note that measurements are generated independently of $\mathbf{v}_1^k, \ldots, \mathbf{v}_{r_k}^k$. The development of the tracking algorithm does not require a particular form for the single-cell measurement likelihood l_m. The simulations in this paper use Rayleigh-distributed measurements. In the case of non-thresholded measurements, $z_j^k \in \mathbb{R}^+$ and the cell measurement likelihood is, for $m = 0, 1, 2, \ldots$, $z > 0$,

$$l_m(z) = \frac{z}{1 + \lambda m} \exp\left(-\frac{z^2}{2(1 + \lambda m)}\right), \tag{5}$$

where λ is referred to as the signal-to-noise ratio (SNR). For thresholded measurements, $z_j^k \in \{0, 1\}$ with $z_j^k = 1$ corresponding to a target detection in the jth cell. The threshold is set so that $\mathsf{Pr}(z_j^k = 1|o_j(\mathbf{P}^k) = 0) = P_{FA}$ where P_{FA} is the desired false alarm probability. Then, for $m = 0, 1, 2, \ldots$,

$$l_m(z) = \begin{cases} P_{FA}^{1/(1+m\lambda)}, & z = 1, \\ 1 - P_{FA}^{1/(1+m\lambda)}, & z = 0. \end{cases} \tag{6}$$

3. APPROXIMATE COMPUTATION OF THE JMPD

MTT involves recursive computation of the joint multi-target probability density (JMPD) $p(\boldsymbol{\xi}^k|\mathbf{Z}^k)$ where $\boldsymbol{\xi}^k = (\mathbf{X}^{k\prime}, r^k)'$ is the extended multi-target state, composed of the target number r^k and the multi-target state $\mathbf{X}^k$, and $\mathbf{Z}^k = \{\mathbf{z}^1, \ldots \mathbf{z}^k\}$ is the measurement history up to time kT.[7] The JMPD provides a probabilistic description of the number of targets present and their states from which quantities of interest can be estimated. For instance, the posterior probability that r targets are present in region $\Omega \subset \mathbb{R}^{2s}$ of the state space is

$$\int_{\Omega^r} p(\mathbf{X}^k, r^k = r|\mathbf{Z}^k)\, d\mathbf{X}^k. \tag{7}$$

Assuming $p(\boldsymbol{\xi}^{k-1}|\mathbf{Z}^{k-1})$ is available the JMPD at time k is updated via the recursion

$$p(\mathbf{X}^k, r^k|\mathbf{Z}^k) = \frac{p(\mathbf{z}^k|\boldsymbol{\xi}^k)}{p(\mathbf{z}^k|\mathbf{Z}^{k-1})} \sum_{r^{k-1}=0}^{\infty} \int p(\boldsymbol{\xi}^k|\boldsymbol{\xi}^{k-1}) p(\boldsymbol{\xi}^{k-1}|\mathbf{Z}^{k-1})\, d\mathbf{X}^{k-1}. \tag{8}$$

Closed-form solution of (8) is generally not possible so it is necessary to use an approximation. In this paper a particle filtering approximation is used.

In particle filtering using sequential importance sampling a collection of n weighted samples from the JMPD is obtained by drawing from an importance density. The manner in which particles are sampled, i.e., the choice of importance density q, has a great effect on performance for a given sample size n. For the multiple target tracking problem considered here this is achieved through exploiting the approximate factorization of the measurement likelihood for well-separated clusters of targets, the use of joint measurement-directed proposals for each target cluster and Rao-Blackwellization. A brief description will be given below. In the following it is assumed that the number of targets is known although the procedure has been extended to handle an unknown number of targets.[13] The number of targets will be denoted as r.

The JMPD of the trajectory $\mathbf{X}^0, \ldots, \mathbf{X}^k$ can be decomposed as

$$p(\mathbf{X}^0, \ldots, \mathbf{X}^k|\mathbf{Z}^k) = p(\mathbf{P}^0, \ldots, \mathbf{P}^k|\mathbf{Z}^k) \prod_{i=1}^{r} p(\mathbf{v}_i^0, \ldots, \mathbf{v}_i^k|\boldsymbol{\rho}_i^0, \ldots, \boldsymbol{\rho}_i^k, \mathbf{Z}^k), \tag{9}$$

where the densities $p(\mathbf{v}_i^0,\ldots,\mathbf{v}_i^k|\boldsymbol{\rho}_i^0,\ldots,\boldsymbol{\rho}_i^k,\mathbf{Z}^k)$, $i=1,\ldots,r$ are Gaussian and can be computed using the Kalman filter (KF) and $p(\mathbf{P}^0,\ldots,\mathbf{P}^k|\mathbf{Z}^k)$ can be approximated using a PF. Computation of the posterior density of $\mathbf{v}_i^k$ can be performed as described in.[15] The details are omitted for the sake of brevity. The posterior mean and covariance matrix of the velocity elements of the ith target at time kT are denoted as $\mathbf{v}_i^{k|k}$ and $\boldsymbol{\Sigma}_i^{k|k}$, respectively.

The JMPD of $\mathbf{P}^{k-1}$ is represented by the particle set $\{\mathbf{P}_t^{k-1},w_t^{k-1}\}_{t=1}^n$. Note that we must have $w_t^{k-1}=1/n$ for reasons which will be explained below. The targets are first separated into $h\le r$ clusters $C_1,\ldots,C_h$ such that $\bigcup_{l=1}^h C_l=\{1,\ldots,r\}$ and $\forall l\in\{1,\ldots,h\}$, $\forall i\in C_l$,

$$|\hat{\boldsymbol{\rho}}_i^{k|k-1}-\hat{\boldsymbol{\rho}}_j^k|\le\Xi\Rightarrow j\in C_l, \tag{10}$$

where $\hat{\boldsymbol{\rho}}_i^{k|k-1}$ is the filter's predicted estimate of $\boldsymbol{\rho}_i^k$ and Ξ is a threshold. The positions of the targets in the lth cluster are collected into $\mathbf{c}_l^k$, $l=1,\ldots,h$. Samples are drawn from an importance density of the form.

$$q(\mathbf{P}^k|\mathbf{Z}^k)=\prod_{l=1}^h\sum_{t=1}^n\psi_{l,t}\pi(\mathbf{c}_l^k|\mathbf{c}_{l,t}^0,\ldots,\mathbf{c}_{l,t}^{k-1},\mathbf{Z}^k). \tag{11}$$

The important features of the importance density in (11) are:

- particles for each cluster are drawn independently;
- particles can be composed of clusters from different particles, i.e., clusters in a particular particle can be sampled conditional on the state of the corresponding clusters from different particles;
- samples for individual target positions and maneuvering modes are drawn from the joint optimal importance density (JOID).

The independent sampling of target clusters is motivated by the approximate factorization of the measurement likelihood for well-separated targets. This is a well-known property of the likelihood which is used in many popular multi-target tracking algorithms.[2,16] Here it is used to reduce computational expense and improve performance, particularly for small sample sizes. This introduces a "bias" in the sampling procedure which must be counteracted by the application of appropriate weights. These second stage weights are discussed below. The use of joint measurement-directed sampling provides a more efficient dispersal of particles for targets which are in close proximity. The π notation used for the JOID in (11) indicates a density conditioned on the given cluster under the assumption that only this target cluster exists. The first-stage weights $\psi_{l,t}$, $l=1,\ldots,h$, $t=1,\ldots,n$ determine which particles provide clusters in the sampling procedure. These weights are selected as the usual OID weight update, the predictive likelihood,[17]

$$\psi_{l,t}\propto\pi(\mathbf{z}^k|\mathbf{c}_{l,t}^{k-1}),\qquad t=1,\ldots,n. \tag{12}$$

It will be seen that this choice of first-stage weighting results in a particularly appealing form for the second-stage weights.

Expressions will now be given for the JOID and the predictive likelihood required for computation of the first-stage weights. The following notation is used. Consider a cluster of q targets at time kT and let $\mathbf{c}^k=(\boldsymbol{\rho}_1^{k\prime},\ldots,\boldsymbol{\rho}_q^{k\prime})'$ denote the collection of target positions. Let M_i, $i=1,\ldots,q$ denote measurement cells in the neighborhood of the ith target and V_j, $j=1,\ldots,J$ denote the region of measurement space occupied by the jth measurement cell. For a given collection of cell indices $j_1,\ldots,j_q$, let $\bar{q}$ denote the number of distinct cell indices, $\bar{\jmath}_1,\ldots,\bar{\jmath}_{\bar{q}}$ denote the distinct cell indices and $m_1,\ldots,m_{\bar{q}}$ the multiplicities of the distinct cells. These quantities are used to define

$$\omega_{j_1,\ldots,j_q}=\prod_{u=1}^{\bar{q}}\frac{l_{m_u}(z_{\bar{\jmath}_u}^k)}{l_0(z_{\bar{\jmath}_u}^k)}. \tag{13}$$

The JOID can then be written as

$$\pi(\mathbf{c}^k|\mathbf{c}^0,\ldots,\mathbf{c}^{k-1},\mathbf{Z}^k)=\sum_{j_1\in M_1}\cdots\sum_{j_q\in M_q}\beta_{j_1,\ldots,j_q}\phi_{j_1,\ldots,j_q}(\mathbf{c}^k), \tag{14}$$

where

$$\beta_{j_1,\ldots,j_q} = \alpha_{j_1,\ldots,j_q} \Bigg/ \sum_{e_1 \in M_1} \cdots \sum_{e_q \in M_q} \alpha_{e_1,\ldots,e_q} \ , \tag{15}$$

$$\alpha_{j_1,\ldots,j_q} = \omega_{j_1,\ldots,j_q} \prod_{i=1}^{q} \eta_{j_i,i}, \tag{16}$$

$$\phi_{j_1,\ldots,j_q}(\mathbf{c}^k) = \prod_{i=1}^{q} \left\{ \chi_{V_{j_i}}(\boldsymbol{\rho}_i^k) N(\boldsymbol{\rho}_i^k; \hat{\boldsymbol{\rho}}_i^k, \boldsymbol{\Psi}_i^k)/\eta_{j_i,i} \right\}, \tag{17}$$

with $\chi_A(z) = 1$ if $z \in A$ and zero otherwise and, for $i = 1, \ldots, q$,

$$\hat{\boldsymbol{\rho}}_i^k = \boldsymbol{\rho}_i^{k-1} + T\hat{\mathbf{v}}_i^{k-1|k-1}, \tag{18}$$

$$\boldsymbol{\Psi}_i^k = T^2 \boldsymbol{\Sigma}_i^{k-1|k-1} + \mathbf{Q}_{i,\rho}^k, \tag{19}$$

$$\eta_{j,i} = \int_{V_j} N(\boldsymbol{\rho}_i^k; \hat{\boldsymbol{\rho}}_i^k, \boldsymbol{\Psi}_i^k)\, d\boldsymbol{\rho}_i^k, \qquad j = 1, \ldots, J. \tag{20}$$

The predictive likelihood, required for the first stage weights in (12), is given by

$$\pi(\mathbf{z}^k|\mathbf{c}^{k-1}) \propto \sum_{j_1 \in M_1} \cdots \sum_{j_q \in M_q} \alpha_{j_1,\ldots,j_q} \tag{21}$$

Thus target positions are drawn from a mixture of truncated Gaussian distributions with each mixture component a hypothesis on the cell locations of the q targets in the cluster. A draw from (14) can be made by selecting a mixture component using the probabilities $\beta_{j_1,\ldots,j_q}$ and then drawing each target position from the appropriate truncated Gaussian distribution. The collection of viable cell locations for the ith target is given by the neighborhood M_i, defined as

$$M_i = \{j \in \{1, \ldots, J\} : \eta_{j,i} > \Upsilon\} \tag{22}$$

where Υ is a small, pre-defined lower bound.

It remains to compute the weight update for the reconstructed particles. Let $\mathbf{c}_{l,1}^k, \ldots, \mathbf{c}_{l,n}^k$ denote the collection of particles drawn for the lth target cluster. The weight of the tth particle can be found as

$$\tilde{w}_t^k \propto p(\mathbf{z}^k|\mathbf{c}_{1,t}^k, \ldots, \mathbf{c}_{h,t}^k) \Bigg/ \prod_{l=1}^{h} \pi(\mathbf{z}^k|\mathbf{c}_{l,t}^k) \tag{23}$$

Resampling based on $\tilde{w}_1^k, \ldots, \tilde{w}_n^k$ is performed in order to obtain an evenly weighted particle set, i.e., $w_t^k = 1/n$ for $t = 1, \ldots, n$. This resampling step is required because reconstructed particles contain target clusters originating from different particles so that the weight assigned to a particular particle at time $(k-1)T$ has no connection to the reconstructed particle at time kT. This is not a problem if the weights are uniform. For a sufficiently large Ξ,

$$p(\mathbf{z}^k|\mathbf{c}_{1,t}^k, \ldots, \mathbf{c}_{h,t}^k) \approx \prod_{l=1}^{h} \pi(\mathbf{z}^k|\mathbf{c}_{l,t}^k), \qquad t = 1, \ldots, n, \tag{24}$$

so that the weights $\tilde{w}_1^k, \ldots, \tilde{w}_n^k$ will be approximately uniform and the resampling procedure will select a large number of distinct particles. This is a desirable property of the second stage weights conferred by selecting the first stage weights as in (12).

For a given collection of target positions the computational expense of the sampling procedure can be controlled by varying the clustering threshold Ξ and/or the neighborhood threshold Υ. The computational expense of drawing a sample for a cluster of q targets is $O(m^q)$ where m is the geometric mean of the number of cells in the neighborhood of each target. The values of q and m are controlled by the clustering threshold Ξ and

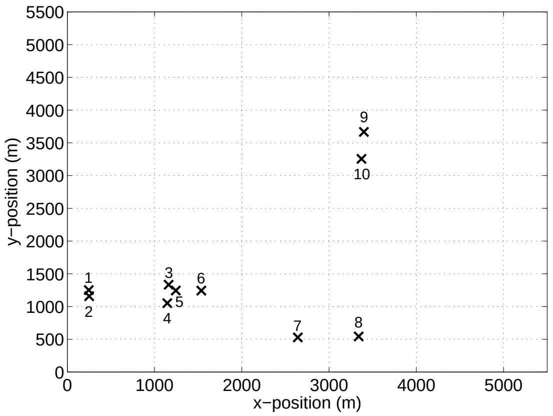


Figure 1. A ten target scenario for examining the effect of clustering on performance. This shows the target positions at the time of interest.

the neighborhood threshold Υ, respectively. The cluster size q is an increasing function of Ξ. This suggests choosing a small value for Ξ although selecting Ξ too small will reduce the benefits of joint sampling and worsen performance. An increase in Υ causes a decrease in m and therefore a decrease in computational expense. This decrease in computational expense is potentially accompanied by a decrease in accuracy as feasible target destinations may be ignored if Υ is too large. In the following section these trade-offs between performance and computational expense are examined through the variances of the first and second stage weights.

4. SELECTING THE THRESHOLDS

The performance of an importance sampling scheme is heavily influenced by the variance of the weights with a low variance required for good performance.[18] In the scheme described in Section 3 there are two sets of weights to consider: the first-stage weights defined in (12) and second-stage weights of (23). In this section the variances of these weights are examined with a view to balancing the trade-off between computational expense and performance involved in the selection of the clustering and neighborhood thresholds. The variances of the first and second stage weights are derived in the appendix.

The weights variances are examined for a scenario involving ten targets moving in a plane at a time at which there are several targets in close proximity. The target configuration at the time of interest is shown in Figure 1. Non-thresholded measurements are generated with an SNR of 6 dB and the grid size is 100m x 100m. Since the targets regularly perform maneuvers target motion is modeled with sufficient process noise for the filter to respond to these maneuvers.

Figure 2 shows the variances of the first and second stage weights and a measure of performance plotted against the clustering threshold for several neighborhood thresholds. The clustering threshold varies between 0 and 500 and the neighborhood threshold varies between 0.002 and 0.2. Tracking performance is measured by the mean number of targets in track averaged over 25 realizations and 1000 time steps. Roughly speaking, a target is deemed to be in track if the filter estimate of target position is close to the true target position.

Consider first the results as a function of clustering threshold for a fixed neighborhood threshold. The two sets of weights display the expected behavior. Initially, for $\Xi < 90$, no clustering is performed so that targets are processed individually. This is seen to result in a negligible first stage weight variance, because each cluster contains only one target, and a very large second stage weight variance, because the clusters violate the approximate factorization of the likelihood. As the clustering threshold increases, a closer approach to the exact likelihood factorization is achieved by the clusters and the second stage weight variance decreases. At the same time the cluster sizes increase causing an increase in the variance of the first stage weights. A final, large decrease in the variance of the second-stage weights is obtained for $\Xi > 240$. This decrease corresponds to the addition of target 6 to a cluster containing targets 3, 4 and 5. The remaining cluster at this point contains targets 1 and 2

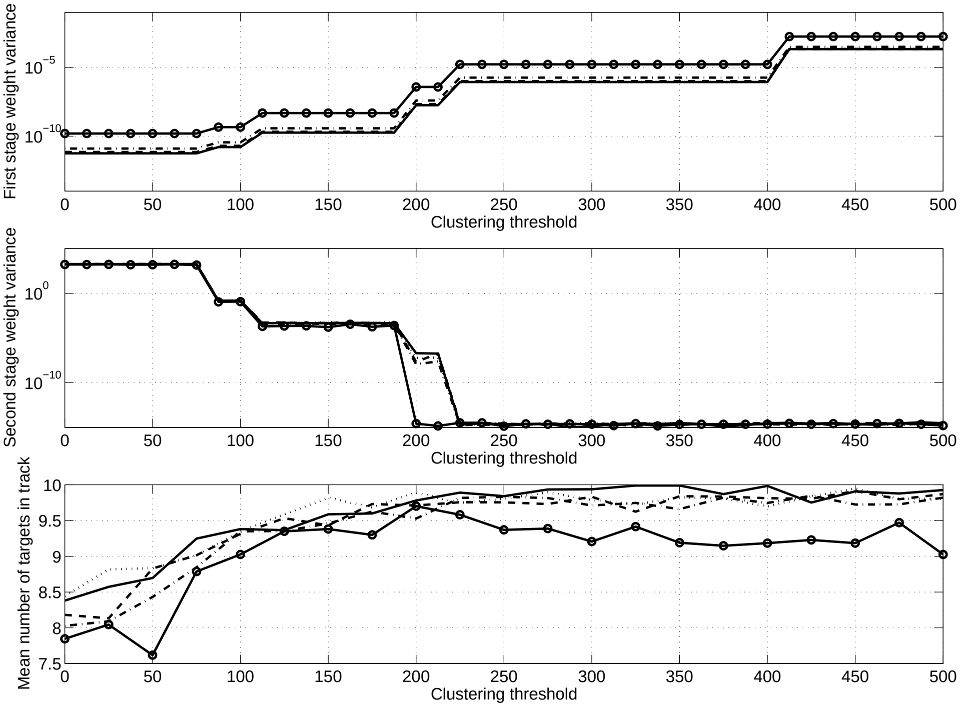


Figure 2. Weights variances and performance plotted against clustering threshold. Top: First stage weight variance plotted against clustering threshold, Middle: Second stage weight variance plotted against clustering threshold and Bottom: Mean number of targets in track plotted against clustering threshold. In each case results are given for neighborhood thresholds of $\Upsilon = 0.002$ (solid), 0.0063 (dotted), 0.02 (dashed), 0.063 (dash-dot) and 0.2 ($\circ$).

with the remaining targets processed individually. The performance results suggest that tracking performance is predominantly affected by the second stage weight variance although the first stage weights variance does have some effect on performance for large clustering thresholds. This can be seen by the drop in performance as the clustering threshold is increased beyond the value for which the second stage weights have negligible variance.

Now consider the results as a function of neighborhood threshold. It can be seen that the first stage weight variance is an increasing function of neighborhood threshold while the second stage weight variance is a decreasing function of neighborhood threshold. The former is due to an increasingly poor approximation to the JOID while the latter is due to an increasingly accurate approximation to approximate factorization of the likelihood. The best level of performance is obtained with the smallest clustering threshold although the drop in performance is quite modest for neighborhood thresholds up to 0.063. This is reflected in the weights variances which vary only slightly for neighborhood thresholds in the range 0.002 to 0.063. This suggests that, at least for the scenario considered here, larger neighborhood thresholds can be used to reduce computational expense without greatly sacrificing performance. The reduction in computational expense varies with the clustering threshold. For a clustering threshold of about 300, which seems to produce the best results, a 15% reduction in computational expense is obtained by moving from $\Upsilon = 0.002$ to $\Upsilon = 0.02$. The reduction increases to 85% for $\Xi = 500$, although this clustering threshold would not be recommended for this scenario.

The clustering threshold which results in the best performance varies for different neighborhood thresholds. This is particularly true of the largest neighborhood threshold, $\Upsilon = 0.2$, compared to the lower neighborhood thresholds. This can be attributed to a reduction in the clustering threshold required for negligible second stage weight variance. The performance obtained for this clustering threshold and $\Upsilon = 0.2$ is not much worse than that obtained for a higher clustering threshold with the lowest neighborhood threshold. This raises the interesting possibility of doubly reducing computational expense through simultaneously reducing the clustering threshold and increasing the neighborhood threshold. Further simulation results would be required to resolve this matter.

5. THRESHOLDED VS. NON-THRESHOLDED MEASUREMENTS

In this section the comparative effects of using thresholded and non-thresholded measurements are examined using Monte Carlo simulations. The scenario used here involves ten targets moving in a 5500m x 5500m surveillance region for 1000 time steps of 1s each. The target trajectories belong to real targets and were obtained from an exercise at the US Army's National Training Centre. A three-dimensional plot showing the target trajectories for the entire surveillance period is given in Figure 3. Time steps at which multiple targets are in close proximity are highlighted. The surveillance region is divided into 100m x 100m cells. Monte Carlo realizations are obtained by generating independent measurement sequences from the same set of target trajectories. The filter assumes constant velocity motion with a large diffusive component. The process noise covariance matrix is the same for each target and is given by

$$\mathbf{Q}_i^k = 20 \begin{pmatrix} 1 & 0 \\ 0 & 1/100 \end{pmatrix} \otimes \mathbf{I}_2. \tag{25}$$

This model attempts to accommodate the actual target behavior which contains long periods of constant velocity motion interspersed with sudden accelerations and move-stop-move behavior.

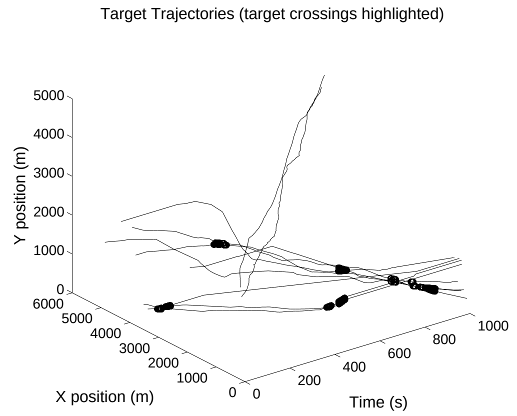
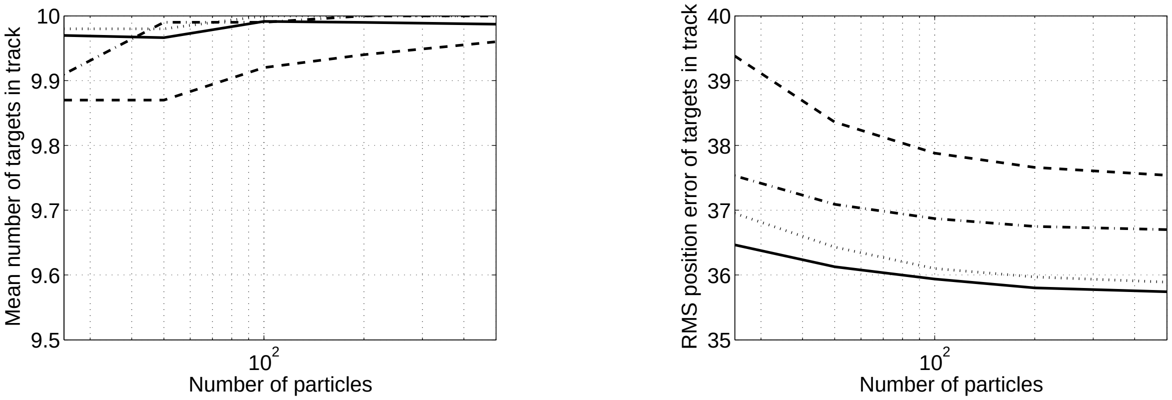


Figure 3. Target trajectories with periods in which multiple targets are in close proximity highlighted.

Both the reliability and tracking accuracy of the algorithm are measured in the performance analysis. Reliability is measured by the mean number of targets in track averaged over all time steps and all realizations. A target is said to be in track if the filter estimate of the target position is within a certain distance of the true target position. Tracking accuracy is measured by the RMS position error averaged over targets which are deemed to be in track at each time step.

The simulation results are shown in Figures 4-7 for SNRs of 10, 7.5, 5 and 2.5 dB. For each SNR, the mean number of targets in track and RMS position error for targets in track, averaged over 100 realizations, are plotted against sample size. Results are given for non-thresholded measurements and thresholded measurements with single target detection probabilities of 0.3, 0.5 and 0.7. Note that different scales are used on the vertical axes of each plot. The clustering threshold is set to 300m and the neighborhood threshold is 0.005. Consider first algorithm performance with non-thresholded measurements, indicated by the solid line in each plot. The results show that accurate and reliable tracking performance is achieved for all SNRs considered here with sample sizes of no more than 500 particles. As expected, performance deteriorates as SNR decreases although this effect becomes less pronounced as the sample size increases. This implies that the required sample size increases as the SNR decreases. For instance, little is gained by using more than 100 particles at an SNR of 10 dB while significant improvements are obtained by increasing the sample size to 500 particles at an SNR of 2.5 dB.

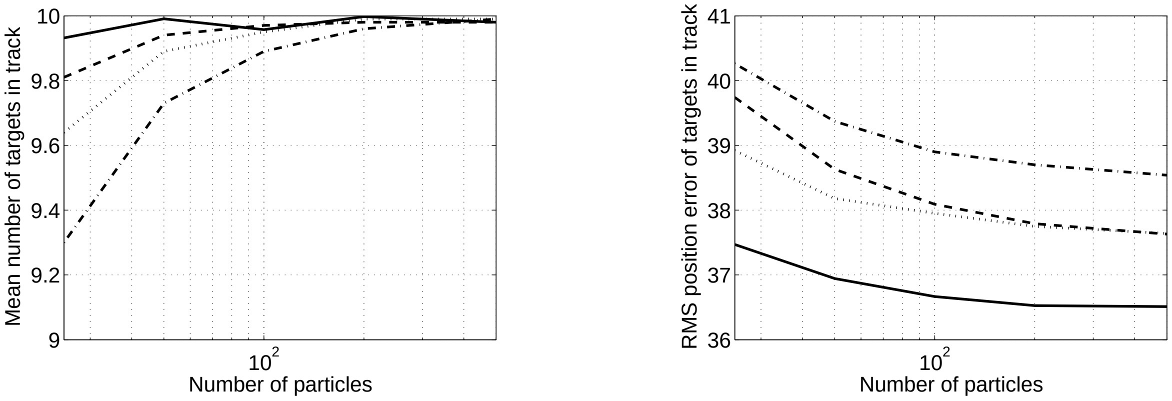
Tracking performance with non-thresholded measurements is now considered relative to performance with thresholded measurements. An important benefit of performing tracking prior to thresholding is the ability to achieve a given level of performance with a lower computational expense, since a smaller sample size can be used. For instance, to achieve a mean number of tracked targets of 9 at an SNR of 2.5 dB requires about 40 particles using non-thresholded measurements and 100 particles using thresholded measurements. Another benefit of using non-thresholded over thresholded measurements is the improvement in obtainable performance. This is particularly evident in the RMS position error of targets in track. Assuming that a sample size of 500 particles yields close to optimal performance for this scenario, which seems reasonable given the flattening in the performance curves between 200 and 500 particles, non-thresholded measurements produce an improvement in accuracy of about 10% over thresholded measurements at an SNR of 2.5 dB. Given the modesty of this improvement, and the fact that it is achieved at such a low SNR, this benefit of using non-thresholded measurements is perhaps of less significance than the reductions in required sample size discussed above.



(a)

(b)

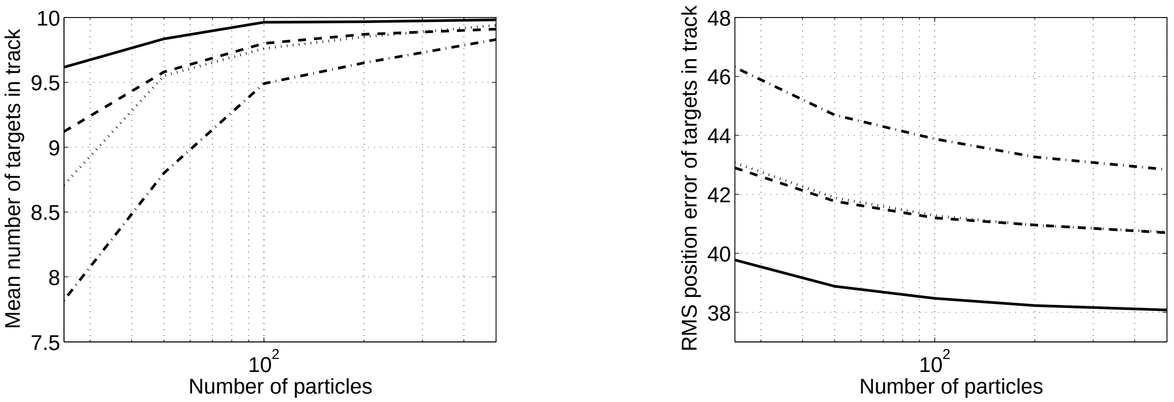
Figure 4. Performance results for tracking using non-thresholded (solid) and thresholded measurements with a single target detection probability of 0.3 (dashed), 0.5 (dotted) and 0.7 (dash-dot). The SNR is 10 dB: (a) Mean number of targets in track plotted against sample size; (b) RMS position error of targets in track plotted against sample size.



(a)

(b)

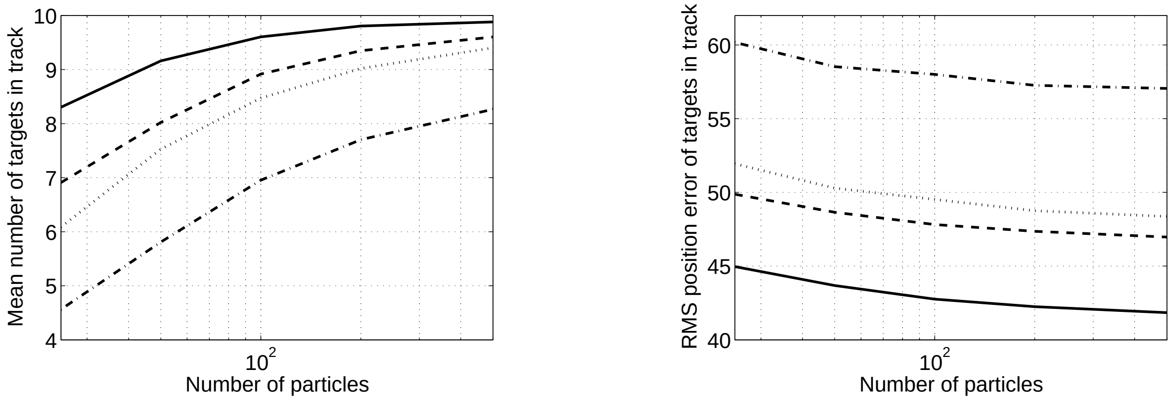
Figure 5. Performance results for tracking using non-thresholded (solid) and thresholded measurements with a single target detection probability of 0.3 (dashed), 0.5 (dotted) and 0.7 (dash-dot). The SNR is 7.5 dB: (a) Mean number of targets in track plotted against sample size; (b) RMS position error of targets in track plotted against sample size.



(a)

(b)

Figure 6. Performance results for tracking using non-thresholded (solid) and thresholded measurements with a single target detection probability of 0.3 (dashed), 0.5 (dotted) and 0.7 (dash-dot). The SNR is 5 dB: (a) Mean number of targets in track plotted against sample size; (b) RMS position error of targets in track plotted against sample size.



(a)

(b)

Figure 7. Performance results for tracking using non-thresholded (solid) and thresholded measurements with a single target detection probability of 0.3 (dashed), 0.5 (dotted) and 0.7 (dash-dot). The SNR is 2.5 dB: (a) Mean number of targets in track plotted against sample size; (b) RMS position error of targets in track plotted against sample size.

6. CONCLUSIONS

A study of the performance of a particle filtering algorithm for multi-target tracking with a pixelized sensor was presented. The algorithm uses several improvement strategies to enable accurate tracking of many targets with small sample sizes. The study focused on the appropriate selection of algorithm parameters and the use of thresholded vs. non-thresholded measurements. The parameters under consideration were two thresholds which trade-off performance and computational expense. A study of the variance of the importance sampling weights, correlated with tracking performance, showed how these weights may be selected for a given scenario. As expected, the use of non-thresholded measurements proved to be beneficial, with particularly large improvements obtained for smaller samples and low signal-to-noise ratios. The improvements obtained for large sample sizes were, however, quite modest. This suggests that the main motivation for using non-thresholded measurements would not be to improve the best possible performance but rather to reduce the sample size required to achieve a certain level of performance thereby reducing computational expense.

APPENDIX A. WEIGHT VARIANCE DERIVATIONS

A.1. First stage weights

The first stage weight for tth particle of the lth cluster is computed as

$$\psi_{l,t} = \pi(\mathbf{z}^k|\mathbf{c}_{l,t}^{k-1}) \bigg/ \sum_{e=1}^{n} \pi(\mathbf{z}^k|\mathbf{c}_{l,e}^{k-1}) \, . \tag{26}$$

This will be averaged over the particle index t to examine the variability in the weights induced by the use of large clusters. It is straightforward to see that the sample mean is $1/n$. The sample variance can be found as

$$\sigma_{\psi_l}^2 = 1/n \sum_{t=1}^{n} \pi(\mathbf{z}^k|\mathbf{c}_{l,t}^{k-1})^2 \bigg/ \left(\sum_{e=1}^{n} \pi(\mathbf{z}^k|\mathbf{c}_{l,e}^{k-1}) \right)^2 - 1/n^2, \tag{27}$$

where $\pi(\mathbf{z}^k|\mathbf{c}_{l,t}^{k-1})$ can be computed using (21).

In the analysis of Section 4 we consider the normalized product of first-stage weight variances $\prod_{l=1}^{s}(n^2\sigma_{\psi_l}^2)$. The normalization facilitates comparisons between the variances of the first and second stage weights.

A.2. Second stage weights

The unnormalized second stage weight at time kT is calculated as

$$\breve{w}^k = \frac{p(\mathbf{c}_1^k, d_1, \ldots, \mathbf{c}_h^k, d_h|\mathbf{Z}^k, \mathcal{P}^{k-1})}{q(\mathbf{c}_1^k, d_1, \ldots, \mathbf{c}_h^k, d_h|\mathbf{Z}^k, \mathcal{P}^{k-1})}. \tag{28}$$

where $d_l \in \{1, \ldots, n\}$, $l = 1, \ldots, h$ is the particle selected for the lth cluster and $\mathcal{P}^{k-1}$ is the particle set at time $(k-1)T$. The variance of the unnormalized second-stage weight conditional upon the measurements up time kT and the particle positions up to time $(k-1)T$ can be found as

$$\mathrm{var}(\breve{w}^k|\mathbf{Z}^k, \mathcal{P}^{k-1}) = \sum_{t_1=1}^{n} \cdots \sum_{t_h=1}^{n} \int \frac{p(\mathbf{c}_1^k, t_1, \ldots, \mathbf{c}_h^k, t_h|\mathbf{Z}^k, \mathcal{P}^{k-1})^2}{q(\mathbf{c}_1^k, t_1, \ldots, \mathbf{c}_h^k, t_h|\mathbf{Z}^k, \mathcal{P}^{k-1})} \, d\mathbf{P}^k - 1 \tag{29}$$

Let $\mathbf{C}_{e_1,\ldots,e_h}^k$ denote a collection of target positions with the e_lth particle providing the positions for the lth cluster. Then, after substituting for the posterior density and importance density in (29) the second stage weight variance becomes

$$\frac{\displaystyle\prod_{l=1}^{h} \sum_{t=1}^{n} \pi(\mathbf{z}^k|\mathbf{c}_{l,t}^{k-1})}{\left(\displaystyle\sum_{e_1=1}^{n} \cdots \sum_{e_h=1}^{n} p(\mathbf{z}^k|\mathbf{C}_{e_1,\ldots,e_h}^{k-1}) \right)^2} \sum_{t_1=1}^{n} \cdots \sum_{t_h=1}^{n} \sum_{j_1 \in M_1^{(t_{\zeta(1)})}} \cdots \sum_{j_r \in M_r^{(t_{\zeta(r)})}} \frac{\omega_{j_1,\ldots,j_r}^2}{\displaystyle\prod_{l=1}^{h} \omega_{j_{i_{l,1}},\ldots,j_{i_{l,g_l}}}} \prod_{i=1}^{r} \eta_{j_i,i}^{(t_{\zeta(i)})} - 1, \tag{30}$$

where $i_{l,1}, \ldots, i_{l,q_l}$ are the indices of the targets in the lth cluster, $\zeta : \{1, \ldots, r\} \to \{1, \ldots, h\}$ maps each target index to a cluster, $M_i^{(t)}$ is the neighborhood of the ith target, as given in (22), for the tth particle and $\eta_{j,i}^{(t)}$ is the quantity of (20) computed for the tth particle.

Exact computation of the variance of the weights using (30) is infeasible as it requires evaluating a sum of n^h terms, with each term itself a sum of a number of terms which increases exponentially with the number of targets. We therefore resort to a Monte Carlo approximation in which a large number of combinations of the particle indices $t_1, \ldots, t_h$ are selected at random. A significant reduction in computational expense can be obtained by utilizing the approximate factorization of the likelihood in the weight variance computation. A valid approximation can be obtained by choosing the clustering threshold for the weight variance computation to be much larger than that used in the filter. Let $E_1, \ldots, E_f$ denote the clusters formed using a large threshold Λ and

$C_{l,1}, \ldots, C_{l,h_l}$, $l = 1, \ldots, f$ denote the clusters formed from the targets in E_l using the filter clustering threshold $\Xi < \Lambda$. The indices of the targets in E_l are denoted $i_{l,1}, \ldots, i_{l,p_l}$ while the indices of the targets in $C_{l,a}$ are denoted $i_{l,a,1}, \ldots, i_{l,a,q_{l,a}}$. An approximation to the variance of the second-stage weights is then

$$\frac{\sum_{b=1}^{B} \prod_{l=1}^{h} \pi(\mathbf{z}^k | \mathbf{c}^{k-1}_{l,t_{l,b}})}{\left(\sum_{e=1}^{B} \hat{p}(\mathbf{z}^k | \mathbf{C}^{k-1}_{t_{1,e},\ldots,t_{h,e}}) \right)^2} \sum_{b=1}^{B} \prod_{l=1}^{f} \sum_{j_1 \in M_{i_{l,1}}^{(t_{\zeta(i_{l,1})},b)}} \cdots \sum_{j_{p_l} \in M_{i_{l,p_l}}^{(t_{\zeta(i_{l,p_l})},b)}} \frac{\omega^2_{j_1,\ldots,j_{p_l}}}{\prod_{a=1}^{h_l} \omega_{j_{i_{l,a,1}},\ldots,j_{i_{l,a,q_{l,a}}}}} \prod_{a=1}^{p_l} \eta_{j_a, i_{l,a}}^{(t_{\zeta(i_{l,a})},b)} - 1, \quad (31)$$

where $t_{l,b} \sim \mathsf{U}_{\{1,\ldots,n\}}$, $l = 1, \ldots, h$, $b = 1, \ldots, B$ with B the number of Monte Carlo realizations, $\hat{p}(\mathbf{z}^k | \mathbf{C}^{k-1}_{t_{1,b},\ldots,t_{h,b}})$ is a factorized approximation of $p(\mathbf{z}^k | \mathbf{C}^{k-1}_{t_{1,b},\ldots,t_{h,b}})$. Since not all combinations of particle indices are being considered the normalizations are adjusted accordingly.

REFERENCES

1. S. Blackman, *Multiple-target Tracking with Radar Applications*, Artech House, 1986.
2. D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control* **24**(6), pp. 843–854, 1979.
3. D. Lerro and Y. Bar-Shalom, "Comparison of tracking/association methods for low SNR targets," in *Proceedings of OCEANS*, pp. 443–448, 1992.
4. B. Ristic, M. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.
5. Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *IEEE Transactions on Aerospace and Electronic Systems* **21**(1), pp. 144–156, 1985.
6. S. Tonissen and R. Evans, "Performance of dynamic programming techniques for track-before-detect," *IEEE Transactions on Aerospace and Electronic Systems* **32**(4), pp. 1440–1451, 1996.
7. K. Kastella, "Discrimination gain for sensor management in multitarget detection and tracking," in *Proceedings of the IMACS Conference on Computational Engineering in Systems Applications*, pp. 167–172, (Lille, France), 1996.
8. A. Srivastava, M. Miller, and U. Grenander, "Multiple target direction of arrival tracking," *IEEE Transactions on Signal Processing* **43**(5), pp. 1282–1285, 1995.
9. L. Stone, C. Barlow, and T. Corwin, *Bayesian Multiple Target Tracking*, Artech House, 1999.
10. R. Mahler, ""Statistics 101" for multisensor, multitarget fusion," *IEEE Aerospace and Electronic Systems Magazine* **19**(1), pp. 53–64, 2004.
11. A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.
12. M. Morelande, C. Kreucher, and K. Kastella, "Multiple target tracking with a pixelized sensor," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005.
13. M. Morelande, C. Kreucher, and K. Kastella, "A Bayesian approach to multiple target tracking," *IEEE Transactions on Signal Processing* (submitted, 2005).
14. N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings Part F* **140**(2), pp. 107–113, 1993.
15. M. Morelande and S. Challa, "Manoeuvring target tracking in clutter using particle filters," *IEEE Transactions on Aerospace and Electronic Systems* **41**(1), pp. 252–270, 2005.
16. T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering* **8**(3), pp. 173–183, 1983.
17. A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing* **10**, pp. 197–208, 2000.
18. M.-S. Oh, "Monte Carlo integration via importance sampling: dimensionality effect and an adaptive algorithm," *Contemporary Mathematics* **115**, pp. 165–187, 1991.
19. F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proceedings of the Aerospace Conference*, 2003.