

# Assessment of multi-hypothesis tracking performance for low signal-to-noise ratio targets

Madeleine G. Sabordo<sup>1</sup>

Samuel J. Davey<sup>2</sup>

(Received 20 January 2010; revised 29 April 2010)

## Abstract

The Multi-Hypothesis Tracker (MHT) is generally considered to be the best performing conventional tracker. It assesses the feasible association of sequences of measurements, calculates the probabilities of the association hypotheses and has track initiation capability. Whilst conventional tracking systems use a detection algorithm to extract measurements from the sensor data, Track-Before-Detect techniques remove the detection algorithm and supply all information received from the signal processing system as measurements to be associated and filtered by the tracker. We compare the performance of the MHT with that of a grid-based Hidden Markov Model Track-Before-Detect algorithm for low signal-to-noise ratio targets. The performances of the MHT and grid Hidden Markov Model algorithms are quantified using six measures: root-mean-square position error, overall detection probability, instantaneous detection probability, false track count, false track length, and computation resource. The grid Hidden Markov

Model algorithm is found to have better detection and significantly better false track performance at the cost of computation resource.

# Contents

<b>1</b>	<b>Introduction</b>	<b>C248</b>
<b>2</b>	<b>Problem definition</b>	<b>C249</b>
2.1	Target state model . . . . .	C250
2.2	Measurement model . . . . .	C250
<b>3</b>	<b>Algorithms</b>	<b>C251</b>
3.1	Multi-hypothesis tracker . . . . .	C251
3.2	Track-before-detect . . . . .	C253
3.3	HMM algorithm . . . . .	C254
<b>4</b>	<b>The experiments</b>	<b>C256</b>
4.1	The scenarios . . . . .	C256
4.2	Measures of performance . . . . .	C256
4.3	Simulation results . . . . .	C257
<b>5</b>	<b>Conclusion</b>	<b>C262</b>
	<b>References</b>	<b>C262</b>

## 1 Introduction

Conventional target tracking systems apply a sequential collection of algorithms to the sensor data, culminating with a detector and a tracker. The detector takes a single frame of data and extracts point measurements from

data features likely to correspond to targets, whilst the tracker uses a dynamic model to associate measurements over time. The Multi-Hypothesis Tracker (MHT) [1] is generally considered to be the best performing conventional tracker. However, it is complex to implement and may carry a high computation load.

Track-Before-Detect (TkBD) is a paradigm that removes the detector and provides sensor data directly to the tracker [2]. This allows the hard-decision detection process to be deferred until after the temporal correlation of the data and improves performance for low Signal to Noise Ratio (SNR) targets. Although TkBD algorithms have been used for thirty years [3, e.g.] there has not been a thorough comparison of the various algorithms for TkBD with each other or with the MHT. Recent work compared TkBD with Probabilistic Data Association (PDA) [4] and showed that PDA was far inferior. However, MHT would be expected to give better performance than PDA. To date the only comparison of TkBD with MHT used an unusual TkBD approach [1, 5]. This article extends the research of Davey et al. [4] to include MHT.

## 2 Problem definition

This article aims to determine whether the MHT provides detection and tracking performance comparable to, or better than, the discrete state space Hidden Markov Model (HMM) algorithm solution to the TkBD problem [6]. From here onwards, the latter will be referred to as the grid HMM method.

Performance was quantified by the ability of the algorithms to detect targets, the estimation accuracy once targets were detected, and the computation resource required by each. There may be a trade-off between some of these measures for both the MHT and the HMM techniques.

## 2.1 Target state model

Assuming a discrete time model, the target state  $\mathbf{x}_k$  at scan  $k$  consists of the  $x$ -position,  $x$ -velocity,  $y$ -position and  $y$ -velocity; that is

$$\mathbf{x}_k = [ \mathbf{u}_k \quad \dot{\mathbf{u}}_k \quad \mathbf{v}_k \quad \dot{\mathbf{v}}_k ]^T. \quad (1)$$

The system is modelled by the linear stochastic process

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \quad (2)$$

where  $\mathbf{F}$  is the state transition matrix, and  $\mathbf{w}_k$  is the system noise, which is assumed to be described by a Gaussian distribution with zero mean and known covariance  $\mathbf{Q}$ . These matrices are

$$\mathbf{F} = \begin{bmatrix} 1 & \mathsf{T} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathsf{T} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \mathbf{q} \begin{bmatrix} \mathsf{T}^3/3 & \mathsf{T}^2/2 & 0 & 0 \\ \mathsf{T}^2/2 & \mathsf{T} & 0 & 0 \\ 0 & 0 & \mathsf{T}^3/3 & \mathsf{T}^2/2 \\ 0 & 0 & \mathsf{T}^2/2 & \mathsf{T} \end{bmatrix},$$

where  $\mathsf{T} = 1$  is the scan period, assumed uniform for simplicity (although this is not required) and  $\mathbf{q}$  represents the variance of the acceleration perturbations and was chosen to be  $\mathbf{q} = 0.5$ .

## 2.2 Measurement model

The MHT and HMM use different measurement models. For MHT, each measurement is a two dimensional vector consisting of observations of the  $x$ -position and the  $y$ -position. For HMM, the measurements at each scan  $k$  represent the received energy in the pixels of a two dimensional image taken at that time. If the target is present, then the intensity in pixel  $\mathbf{m}$  is assumed to be Ricean distributed [4] with pdf

$$\Pr(z_k^{\mathbf{m}} | \mathbf{x}_k) = \frac{2z_k^{\mathbf{m}}}{\sigma^2} \exp \left[ \frac{-(z_k^{\mathbf{m}})^2 - (\mathbf{h}^{\mathbf{m}}(\mathbf{x}_k))^2}{\sigma^2} \right] \mathsf{I}_0 \left[ \frac{2z_k^{\mathbf{m}}\mathbf{h}^{\mathbf{m}}(\mathbf{x}_k)}{\sigma^2} \right], \quad (3)$$

where  $h^m(x_k)$  is the contribution in the pixel from the target,  $I_0(\cdot)$  is the modified Bessel function of the first kind with order zero and  $\sigma^2$  is the variance of the measurement noise. Otherwise, the intensity in pixel  $m$  is assumed to be Rayleigh distributed [4] with pdf

$$\Pr(z_k^m \mid \text{notarget}) = \frac{2z_k^m}{\sigma^2} \exp \left[ -\frac{(z_k^m)^2}{\sigma^2} \right]. \quad (4)$$

The set of measurements at scan  $k$  is  $z_k = \{z_k^1, z_k^2, \dots, z_k^n\}$ , where  $n$  is the number of pixels in the image taken at that scan. The set of measurements for all scans is denoted by  $Z_k = \{z_i \mid i = 1, 2, \dots, k\}$ .

For the simulations, intensity images were generated and were sent directly to the HMM tracker for association and filtering. For the MHT tracker, the intensity images were fed into a peak detector that generated a set of peaks [7]. The peaks were the input for the MHT tracker. The set of peaks at each scan  $k$  is  $y(k) = \{y_n(k)\}_{n=1}^{m(k)}$  where  $m(k)$  is the number of peaks generated in scan  $k$ .

## 3 Algorithms

### 3.1 Multi-hypothesis tracker

The MHT is a technique that assesses the feasible association of sequences of measurements and calculates the probabilities of the association hypotheses. Suppose at time  $k$  we have a set of measurements  $y(k) = \{y_n(k)\}_{n=1}^{m(k)}$  and the set of association hypotheses  $\theta^{k-1}$ . We generate the set  $\theta^k$  by associating  $\theta^{k-1}$  first with  $y_1(k)$ , then the resulting set is associated again with  $y_2(k)$ . This process is performed repeatedly up to  $y_{m(k)}(k)$ . On the first scan, each  $y_n(1)$  can be taken as a false alarm or as a new track. The feasible association rules for all other  $y_n(k)$  are

- the measurement is a false alarm,

- the measurement is the continuation of a previous track, or
- the measurement initiates a new track.

This process leads to an exponential growth in the number of hypotheses as time advances. In order to manage these hypotheses, the total number of hypotheses is controlled by

- clustering of hypotheses with incompatible tracks, that is, tracks sharing common association history or measurements,
- pruning of hypotheses with low probabilities,
- merging/combining hypotheses with similar effects.

In general, there are two approaches to implementing the MHT [1]. The first approach originated from the work of Reid [8] and is referred to as the hypothesis oriented MHT. The second approach is known as the track oriented MHT.

The hypothesis oriented MHT maintains an array of hypotheses from scan to scan. These hypotheses continually grow as new measurements are received and are then pruned using empirically derived rules that vary with a particular study. At each scan, the set of association hypotheses  $\theta^{k-1}$  is carried over from the previous scan. Each hypothesis contains tracks that do not share common association history (measurements) with other tracks in the hypothesis. The set  $\theta^{k-1}$  is then expanded to the set  $\theta^k$  as new measurements are processed and associated according to the feasible association rules mentioned earlier. The hypotheses are evaluated by calculating their probabilities. These probabilities are then used to prune the hypotheses. The tracks contained in the hypotheses that passed the pruning process are then filtered.

The track oriented MHT does not maintain an array of hypotheses from scan to scan. Instead, the tracks generated on each scan are assigned into hypotheses and those tracks contained in the hypotheses that passed the pruning process are then predicted to the next scan.

The MHT implemented in this article employs the hypothesis oriented approach. Management of hypotheses is controlled by pruning the hypotheses with low probabilities and merging the hypotheses with similar effects.

The tracking filter used in this study was the Kalman filter (KF). It estimated the state of the discrete time dynamic process in a way that minimised the mean of the squared error [9]. The KF assumed a linear state space model and Gaussian dynamic noise in the system model equation and Gaussian measurement noise in the measurement equation, as described in Section 2.1.

## 3.2 Track-before-detect

Traditional tracking systems employ a detection threshold to extract point-measurements from sensor intensity images. The use of a detection threshold may lead to irretrievable loss of data: if the threshold is too high, then low SNR targets are completely ignored. A low threshold, on the other hand, may lead to an excessive number of false alarms.

The Track-Before-Detect (TkBD) approach uses all information received from the signal processing system as measurements to be associated and filtered by the tracker. The detection decision is deferred until after temporal correlation.

The TkBD paradigm models the sensor image as a function of an unknown target state. The likelihood of a sequence of images is a highly non-linear and non-Gaussian function of the target state. Since the measurement process is non-linear and non-Gaussian, we must use a numerical approach to solve it. The grid HMM algorithm approximates the state space with a constant spacing grid in position and velocity. This transforms the problem into a discrete state one whereby discrete state algorithms such as the HMM algorithm and Viterbi can be brought to bear. This is costly because the state space is very large and we must estimate the probability of the target state for each location in the space.

The likelihood ratio for pixel  $m$  is defined as the ratio of the likelihood given a target (3), and the likelihood given no target (4). This function is the main driver for many TkBD methods and is

$$L(z_k^m | x_k) = \frac{\Pr(z_k^m | x_k)}{\Pr(z_k^m | \text{notarget})} = \exp \left[ \frac{-h^m(x_k)^2}{\sigma^2} \right] I_0 \left[ \frac{2z_k^m h^m(x_k)}{\sigma^2} \right].$$

There are a number of approaches to TkBD. Comparative studies [4, 10, 11] were conducted among the H-PMHT, Viterbi, PDAF, HMM estimator and the Particle Filter. These demonstrated similar detection performance between the algorithms but varied with regard to the required computation time.

The TkBD algorithm used in this study is the grid HMM estimator, described in detail by Davey et al. [4, 10], an extract follows.

### 3.3 HMM algorithm

The posterior pdf of the target state is recursively determined using the well-known Bayesian relationship

$$\Pr(x_k | Z_k) \propto \Pr(z_k | x_k) \int \Pr(x_k | x_{k-1}) \Pr(x_{k-1} | Z_{k-1}) dx_{k-1} \quad (5)$$

The Bayesian estimator in this article is a direct approximation to (5) based on a discretisation of the state space. Choose a uniformly spaced set of states,  $\mathcal{X}$  (which is not necessarily related to the discrete measurement function). Equation (5) is then approximated by

$$\Pr(x_k | Z_k) \approx K \bar{\mathcal{L}}(z_k | x_k) \sum_{x_{k-1} \in \mathcal{X}} \Pr(x_k | x_{k-1}) \Pr(x_{k-1} | Z_{k-1}) \quad (6)$$

where  $K$  is a normalising constant. The approximation is exact in the limit as  $\mathcal{X}$  approaches  $\mathbb{R}^4$ . The first term in (6) is the intensity independent

marginal likelihood, defined by

$$\mathcal{L}(z_k | \mathbf{x}_k) = \int_0^\infty \mathcal{L}(z_k | \mathbf{x}_k) \Pr(I_k) dI_k. \quad (7)$$

The discrete state space is augmented with a null state,  $\emptyset$ , to indicate the possibility that there is no target. Denote the probability of target death as  $P_d$ , and the probability of target birth as  $P_b$ . Then the evolution probability in (6) is

$$\Pr(\mathbf{x}_k | \mathbf{x}_{k-1}) = \begin{cases} 1 - P_b, & \mathbf{x}_k = \emptyset, \mathbf{x}_{k-1} = \emptyset, \\ P_d, & \mathbf{x}_k = \emptyset, \mathbf{x}_{k-1} \neq \emptyset, \\ P_b/|\mathcal{X}|, & \mathbf{x}_k \neq \emptyset, \mathbf{x}_{k-1} = \emptyset, \\ (1 - P_d) \Pr(\mathbf{v}_k = \mathbf{x}_k - \mathbf{F}\mathbf{x}_{k-1}), & \mathbf{x}_k \neq \emptyset, \mathbf{x}_{k-1} \neq \emptyset, \end{cases}$$

where  $|\mathcal{X}|$  is the number of discrete states in  $\mathcal{X}$ .

The parameters  $P_b$  and  $P_d$  control the detection performance and can be tuned to optimise detection performance. The selection of the state space,  $\mathcal{X}$ , is a tradeoff between estimation accuracy, which improves with finer resolution, and computation requirement, which increases with  $|\mathcal{X}|$ . The process noise *pmf* also affects estimation accuracy, as well as providing some capacity to handle model mismatch between the assumed target model and the true target motion. The algorithm is initialised with  $\Pr(\mathbf{x}_0 = \emptyset) = 1$  and  $\Pr(\mathbf{x}_0) = 0$  for all  $\mathbf{x}_0 \neq \emptyset$ .

Once the pdf of the state has been evaluated, a state estimate is obtained by selecting the state with the highest probability. In the event that this state is the null state, then the algorithm reports that there is no target. To account for the case where the pdf has a peak that is spread over several grid cells, the implementation used finds the highest probability non-null state and accumulates the probability in the adjacent cells. If the accumulated probability is higher than the null-state probability, then a detection is reported.

Given that the relative performance of the grid HMM is known compared with other TkBD approaches, it is possible to extrapolate the comparison here to those approaches. Davey et al. [4] demonstrated that there was little difference in performance between the various TkBD algorithms tested, so a comparison of MHT with other TkBD algorithms would be expected to give similar results to the comparison contained in this article.

## 4 The experiments

### 4.1 The scenarios

The MHT and grid HMM algorithms were compared using simulated scenarios of a manoeuvring target and a scenario with no target present. The scenarios of a manoeuvring target had the same speed but variable SNR. The target peak SNR quantifies the height of the peak of the target point spread function relative to the noise floor, and represents a measure of how easy it is to detect the target [4]. The peak SNR is defined as  $20 \log(I_k/\sigma^2)$  where  $I_k$  is the contribution to the pixel at scan  $k$  when the target is located exactly on the sample point for the pixel and  $\sigma^2$  is the variance of the measurement noise. The target SNR was varied from 3 dB to 12 dB and the speed was one pixel per frame for all scenarios. The no-target scenario ran over 2000 scans to quantify the number of false tracks formed.

### 4.2 Measures of performance

The metrics used to measure performance were as follows.

1. *False track count* The number of false tracks generated in the no-target scenario.

TABLE 1: False track performance and computation resource.

Algorithm	False track count	False track length	CPU resource	
			seconds	ratio
MHT	438	27	128000	1
Grid HMM	9	2	177000	1.4

2. *False track length* The average number of frames for which these false tracks existed.
3. *Computation resource* The total CPU time in seconds required to evaluate all of the scenarios.
4. *RMS position error* The error averaged over those times when the target was detected.
5. *Overall detection probability* The fraction of Monte Carlo runs for which the target was detected at any time.
6. *Instantaneous detection probability* The total fraction of times for which the target was detected.

### 4.3 Simulation results

The scenarios described in Section 4.1 were simulated to evaluate the performance of the trackers. 500 Monte Carlo trials were performed for each scenario. Table 1 shows the false track performance and computation resource requirements of MHT and grid HMM algorithms.

A plot of the RMS position error is shown in Figure 1. A plot of the overall detection probability is shown in Figure 2 and a plot of the instantaneous detection probability is shown in Figure 3. The metric in each figure is plotted as a function of SNR. Note that the standard deviations are shown as upper and lower range limits of error bars.

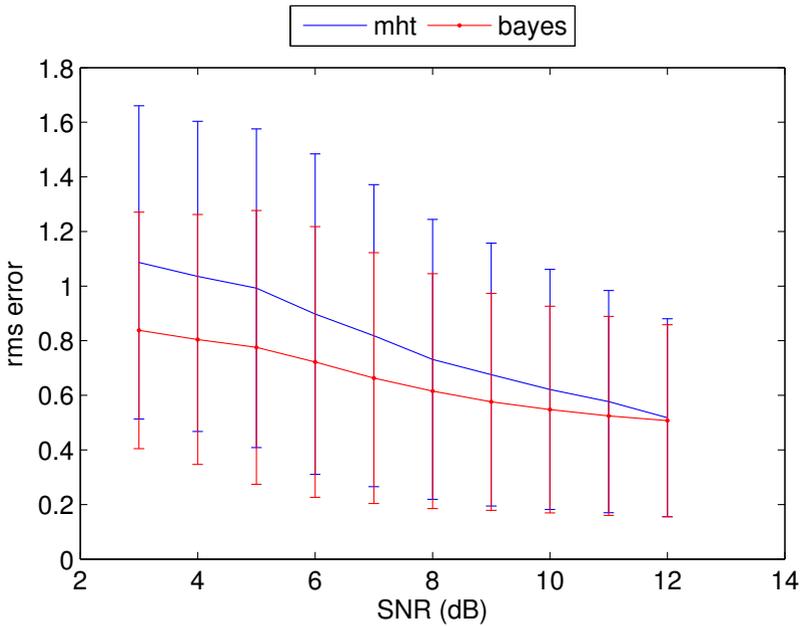


FIGURE 1: RMS position estimation error: “bayes” actually denotes the HMM algorithm; and “mht” denotes the multi-hypothesis tracker algorithm.

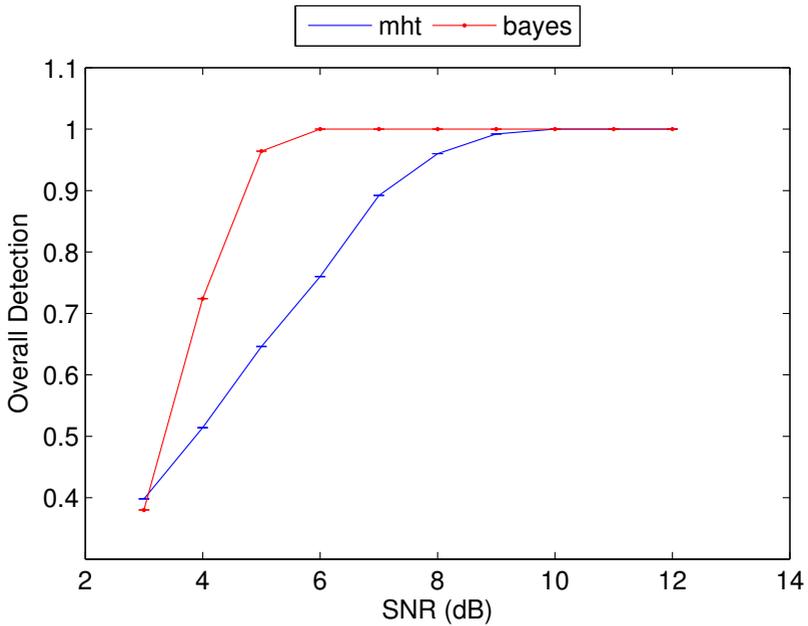


FIGURE 2: Overall detection probability: “bayes” actually denotes the HMM algorithm; and “mht” denotes the multi-hypothesis tracker algorithm.

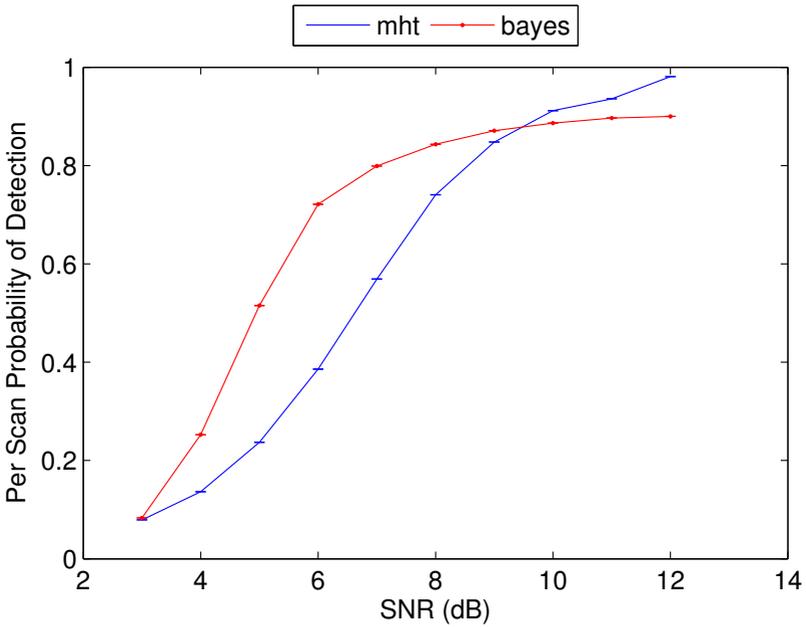


FIGURE 3: Average instantaneous detection probability: “bayes” actually denotes the HMM algorithm; and “mht” denotes the multi-hypothesis tracker algorithm.

From Table 1, the false track statistics of MHT exceeded those of the grid HMM algorithm by two orders of magnitude—an indication that the grid HMM technique was more effective and accurate in terms of false track performance.

In terms of computation resource, the MHT was about 1.4 times faster than the grid HMM algorithm. The MHT took around 35.5 hours to evaluate all scenarios over 500 Monte Carlo trials whereas the grid HMM algorithm took a little over 49 hours.

Figure 1 confirmed our anticipated result for the RMS position errors. Although the mean position errors of the MHT algorithm for SNR values were slightly higher than those of the grid HMM algorithm, the standard deviations of the two graphs overlapped for all scenarios. Owing to the overlapping of RMS position errors when standard deviations were taken into account, it is inferred that the RMS position errors of the algorithms were comparable. Overall, the position errors of both algorithms were very small. The trend of the curves indicates that as the SNR increased, the error decreased and the standard deviations became smaller. This was understandable because at low SNR, tracks were formed relatively far away from the true positions of the target. On the other hand, at high SNR values, tracks closely followed the true trajectory of the target, and standard deviations were small.

Figure 2 shows that the grid HMM algorithm outperformed the MHT when measured in terms of the overall detection probability metric. For SNRs of 10 dB and higher, both algorithms were able to detect the target for all Monte Carlo trials. For SNRs less than 10 dB, the grid HMM algorithm proved its superior performance. The overall trend was that, as the amplitude increased, overall detection probability also increased.

Figure 3 showed a similar pattern: instantaneous detection performance improved with increasing SNR. The MHT outperformed the grid HMM at SNRs of 10 dB and higher. For other SNR values, the HMM outperformed the MHT. This implies that when MHT detects the track, it detects it for longer. This is because the MHT reports the whole history of the track, not just the track points after the track has been deemed good. In contrast, the grid algorithm

is causal. For many applications, real time performance is required and the revision of history is of little interest. Hence, the slight advantage of MHT at high SNR is not highly important.

In summary, the grid HMM algorithm provided better detection and false track performance than the MHT at the expense of computational load. However, the RMS position errors of the algorithms were comparable.

## 5 Conclusion

The performance of MHT and grid HMM algorithms were evaluated based on six metrics: RMS position error, overall detection probability, instantaneous detection probability, false track count, false track length and computation resource. The grid HMM algorithm had better detection and significantly better false track performance at the cost of computation resource. The MHT was tuned to have a detection performance that was almost as good as that of the grid HMM. The false track statistics show that the former was two orders of magnitude worse than the latter. RMS position errors of the algorithms were small and comparable. For both algorithms, detection probabilities increased with increasing SNR and RMS position errors decreased with increasing SNR.

## References

- [1] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999. C249, C252
- [2] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004. C249

- [3] R. M. Burczewski and N. C. Mohanty. Detection of Moving Optical Objects. In *Proceedings of International Telemetry Conference*, pages 325–330, 1978. [C249](#)
- [4] S. J. Davey, M. G. Rutten, and B. Cheung. Comparison of Detection Performance for Several Track-Before-Detect Algorithms. *EURASIP Journal on Advances in Signal Processing*, 2008. [C249](#), [C250](#), [C251](#), [C254](#), [C256](#)
- [5] C. A. Barlow and S. S. Blackman. New Bayesian Track-Before-Detect Design and Performance Study. In *Proceedings of SPIE 3373*, 1998. [C249](#)
- [6] L. D. Stone, C. A. Barlow, and T. L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999. [C249](#)
- [7] S. B. Colegrove, S. J. Davey, and B. Cheung. Clutter Rejection Using Peak Curvature. *IEEE Trans AES*, 42:1492–1496, 2006. [C251](#)
- [8] D. B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, AC-24:843–854, December 1979. [C252](#)
- [9] B. Ristic and N. Gordon. *Lecture notes in Multisensor and Data Fusion*. Adelaide University, 2007. [C253](#)
- [10] S. J. Davey, B. Cheung, and M. G. Rutten. Track-Before-Detect for Sensors with Complex Measurements. In *Fusion Conference*, 2009. [C254](#)
- [11] S. J. Davey and M. G. Rutten. A Comparison of Three Algorithms for Tracking Dim Targets. In *IDC Conference*, 2007. [C254](#)

## Author addresses

1. **Madeleine G. Sabordo**, Defence Science and Technology Organisation, AUSTRALIA.
2. **Samuel J. Davey**, Defence Science and Technology Organisation, AUSTRALIA.  
<mailto:Samuel.Davey@dsto.defence.gov.au>