A particle filter for track-before-detect of a target with unknown amplitude viewed against a structured scene

By Malcolm Rollason & David Salmond

QinetiQ Limited, Cody Technology Park, Ively Road, Farnborough, GU14 0LX

Abstract

Track-before-detect methods operate directly upon raw sensor signals without a separate, explicit detection stage. An efficient implementation of a Bayesian track-before-detect particle filter is described for tracking of a single target in a sequence of images. The filter provides a sample based approximation to the distribution of the target state directly from pixel array data. The filter also provides a measure of the probability that the target is present. Spatial differentiation of the pixel array data allows objects to be tracked when viewed against a general scene with additive noise. Simulated results illustrate that a dim point target of unknown amplitude, which has become spatially blurred, may be tracked through a sequence of structured images. Detection sensitivity is established using simulated results.

1. Introduction

In the usual approach to target tracking, measurements are extracted via sensor signal processing and are then passed to the tracking function. Measurement extraction is usually via some thresholding process which inevitably results in a loss of information. This is of little consequence if the target signal-to-noise ratio (SNR) is high, so that a good probability of detecting the target can be achieved while maintaining a low false alarm rate. However, for small SNRs information loss may be significant.

In principle it would be better for the tracking function to operate directly on the raw sensor signal. For an electro-optical (EO) staring array, this means that the grey-scale levels from every pixel should be available to the tracking function. This approach of avoiding an explicit detection stage is known as track-before-detect.

Spatial differentiation of the pixel array data allows objects to be tracked when viewed against a general scene with additive noise [1]. Simulated results illustrate that the resulting algorithm may be applied when the target is viewed against a background of non-Gaussian noise (e.g. an EO image of a general scene), has unknown amplitude, and has spatial extent through sensor blur.

The outputs of the algorithm include an assessment of the probability that a target is present, and when a target is present, the target's estimated state (position, velocity, intensity).

Whilst particle filters may be used to solve non-linear, non-Gaussian filtering problems, care must be taken with the efficiency of implementation. We describe an efficient implementation of the algorithm, achieved via the following two refinements to a standard particle filter [1]:

- A standard filter [2] [3] represents the probability that a target is present by the ratio of 'active' to 'inactive' particles – denoted by the discrete state λ_i (i denotes the i^{th} particle). This requires propagation of 'inactive' particles $\lambda_\mathrm{i} = 0$ which do not carry information about the target. The 'inactive' particles are computationally cheaper to propagate than the 'active' ones. However, once the target has been found then most particles become active with $\lambda_\mathrm{i} = 1$. Thus the computational load increases for the easier part of the problem (tracking the target once found). A scheme for analytic calculation of probability that the target is present has been implemented, which does not use 'inactive' particles.

- A Marginalised Particle Filter (MPF) [4] has been implemented to exploit linear sub-structure in the problem. This enables the problem to be partitioned into two elements, using a particle filter to interact with the images, and applying one Kalman filter per particle. This provides more 'informative' particles so substantially reducing the number required relative to the standard particle filter. For this application, this approach reduces the number of state vector components represented by the particle filter from 5 to 3. Marginalised particle filters have been applied to the track-before-detect problem [1] [5], where the probability of the target being present is represented as a ratio of 'active' and 'inactive' particles.

This paper is structured as follows. The problem is defined in section 2. The Bayesian solution is described in section 3. Section 4 describes the implementation of a baseline recursive track-before-detect algorithm, including the prediction and update stages. (The target appearance model is described in the prediction stage). Improvements to the baseline implementation are described in section 5, together with an algorithm 'recipe'. Simulation results are in section 6, followed by conclusions in section 7.

2. Problem Statement

A staring EO sensor observes a region of the x-y plane. Each pixel or resolution cell of the sensor corresponds to a square region of dimension $s \times s$, and the sensor array consists of $N \times M$ pixels. It is assumed that at time step t, the output of all NM resolution cells are read coincidently and the measured intensity of pixel (i, j) is denoted $z^{ij}(t)$. The complete sensor measurement set at time step t is denoted:

$$z_t = \left\{ z_t^{ij} : i = 1, \cdots, N;\ j = 1 \cdots, M \right\} \tag{2.1}$$

If a target is present and its centroid is at position (x, y), it may contribute to the pixels (i, j), in that vicinity. Given the target position, the contribution to each pixel is assumed known and for point objects is due to the sensor point spread function (psf), $\gamma_{ij:xy}$, represented by a 2D Gaussian function with standard deviation σ_{psf}:

$$\gamma_{ij:xy} = \frac{s^2}{2\pi\sigma_{psf}^2} exp\left\{ -\frac{(x-is)^2}{2\sigma_{psf}^2} - \frac{(y-js)^2}{2\sigma_{psf}^2} \right\} \tag{2.2}$$

In addition to the intensity attributable to the target (when it is present) the sensor may also detect a background. It is assumed that the sensor pixels are corrupted by noise.

The core track-before-detect algorithm is predicated upon there being at most one target at a time in the sensor's scanned region. (Multi target tracking could be via multiple 'core' track-before-detect algorithms running on different (non-overlapping) parts of the image). Initially, at time step $t = 0$, it is assumed that no target is present so that the pixel grey levels are solely due to background + noise. A target may appear at any time step and at any point in the scanned region. The initial distribution of the target state vector when it first appears is assumed known (for example, uniform over the field-of-view). Following its appearance, the target then proceeds on a trajectory according to a known dynamics model until it disappears or passes out of the scanned region. Following common practice, the birth / death of a target is modelled as a Markov process with parameter λ, where $\lambda = 1$ indicates that a target is present, otherwise $\lambda = 0$. Further detail on the birth / death process is provided in section 4.

3. Solution

The filter state vector consists of the velocity, position and intensity of the target and is augmented with a discrete state, the target present / absent flag, λ. The state vector, excluding the discrete state, is:

$$x_t = [\dot{x}\ \dot{y}\ x\ y\ I]^T \tag{3.1}$$

From a Bayesian perspective, a complete solution of the above problem is given by the posterior probability density function (pdf) $p(x_t, \lambda_t \mid z_{1:t})$ where $z_{1:t}$ denotes the complete set of all past images.

$$p(x_t, \lambda_t \mid z_t, z_{1:t-1}) = \frac{p(z_t \mid x_t, \lambda_t, [z_{1:t-1}])\, p(x_t, \lambda_t \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})} \tag{3.2}$$

The construction of this posterior pdf depends on the measurement likelihood $p(z_t \mid x_t, \lambda_t)$, which does not depend upon the previous measurements, and the prediction between time steps of the posterior from the previous step, $p(x_{t-1}, \lambda_{t-1} \mid z_{1:t-1})$. The prediction depends upon the transition density $p(x_t, \lambda_t \mid x_{t-1}, \lambda_{t-1}, z_{1:t-1})$ according to the following relation:

$$\begin{aligned} p(x_t, \lambda_t \mid z_{1:t-1}) &= \sum_{\lambda_{t-1}} \int p(x_t, \lambda_t, x_{t-1}, \lambda_{t-1} \mid z_{1:t-1})\, dx_{t-1} \\ &= \textstyle\sum_{\lambda_{t-1}} \int p(x_t, \lambda_t \mid x_{t-1}, \lambda_{t-1}, [z_{1:t-1}])\, p(x_{t-1}, \lambda_{t-1} \mid z_{1:t-1})\, dx_{t-1} \end{aligned} \tag{3.3}$$

The transition density comprises the target's physical dynamic model and that of the appearance model:

$$p(x_t, \lambda_t \mid x_{t-1}, \lambda_{t-1}, [z_{1:t-1}]) = p(x_t \mid \lambda_t, x_{t-1}, \lambda_{t-1})\, p(\lambda_t \mid x_{t-1}, \lambda_{t-1}) \tag{3.4}$$

Apart from the possibility of an existing target passing out of the sensor field-of-view, the transition of λ is independent of x_{t-1} and is defined by the birth/death Markov model. If $\lambda_t = 0$, the target is not present and x_t is undefined, otherwise the pdf of x_t conditional x_{t-1} and λ_{t-1} is given by:

$$p(x_t \mid \lambda_t = 1, x_{t-1}, \lambda_{t-1}) = \begin{cases} p(x_t \mid x_{t-1}) & for\, \lambda_{t-1} = 1 \\ p_B(x_t) & for\, \lambda_{t-1} = 0 \end{cases} \tag{3.5}$$

where the transition density $p(x_t \mid x_{t-1})$ is defined by the target dynamics model and $p_B(x_t)$ is the initial pdf of a target on its appearance.

The likelihood $p(z_t \mid x_t, \lambda_t)$ of the state given the pixel measurements is given by:

$$p(z_t \mid x_t, \lambda_t) = \begin{cases} \prod_{i,j} p_{S+N}(z_t^{ij} \mid x, y, I) & for\, \lambda = 1 \\ \prod_{i,j} p_N(z_t^{ij}) & for\, \lambda = 0 \end{cases} \tag{3.6}$$

Here, $p_N(z_t^{ij})$ is the pdf of the background + noise in pixel (i, j) and $p_{S+N}(z_t^{ij} \mid x, y, I)$ is the pdf of the target signal+background+noise in pixel (i, j) given that the target of intensity I is located at (x, y).

3.1 Measurement Model – structured scene content

Equation 3.6 implies that the intensity distributions in each pixel are known and independent. If the background scene against which a target is viewed contains structure, then this is certainly not the case.

One way of dealing with this problem is to create a gradient image Δ by spatial differentiation of the intensity image z_t. This greatly reduces the inter-pixel dependency and it allows the background structure to be represented via a general, image independent, model.

For images where the scene contains structure, a gradient image Δ has been produced by convolution of the intensity image z_t with a Laplacian operator L which represents a numerical approximation to the second differential of intensity with respect to pixel position:

$$\mathrm{L} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.7}$$

If only the structured background is present, the distribution of gradient intensities can be reasonably modelled by a Lorentzian pdf [6].

$$p\left(\delta^{ij}\right) = \frac{\alpha}{\pi\left(\alpha^2+\delta^{ij^2}\right)} \tag{3.8}$$

where δ^{ij} are the gradient intensities in the new image Δ. The parameter α, represents the scale and general sharpness of the background structure. The validity of this assertion is demonstrated in Figure 1 where an image of Lena's face (a) is shown with a histogram (b) of the distribution of intensities in the image. It is hard to describe the shape of this histogram with a simple parametric model. An image of gradient intensities produced following spatial differentiation is presented in (c), and (d) shows the corresponding histogram with the minimum Mean Square Error (MSE) fitted Lorentzian curve superimposed.

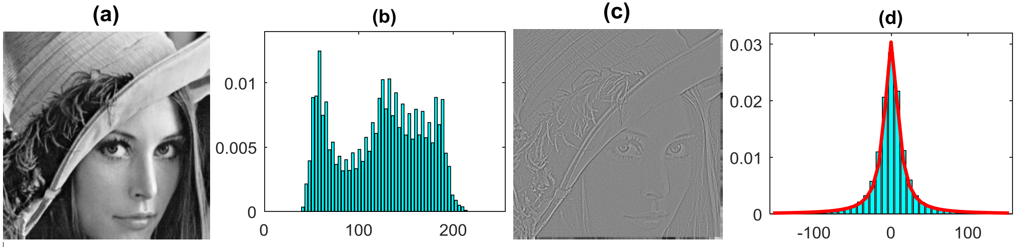

FIGURE 1. (a) image containing structure. (b) distribution of intensities in original. (c) image of gradient intensities. (d) Lorentzian distributed gradient intensities and superimposed minimum MSE Lorentzian curve, $\alpha = 10.47$

If the structured background image were also corrupted by additive Gaussian noise, the distribution of the differentiated structure + noise image would be a convolution of the Lorentzian pdf with a Gaussian pdf. This convolution is intractable and so has been approximated via a scaled t-distribution with two degrees of freedom:

$$p_N\left(\delta^{ij}\right) = \frac{\alpha^2}{2\sqrt{2}\left(\alpha^2+\delta^{ij^2}/2\right)^{3/2}} \tag{3.9}$$

where α is the scaling parameter. Thus we have a model for the combined background and noise distribution $p_N\left(\delta_t^{ij}\right)$ in the image of gradient intensities Δ.

If the measurement is taken to be the gradient image Δ, rather than the image of absolute intensities z_t, then the likelihood in Equation 3.6 is calculated by using Δ instead of z_t, and correspondingly, z_t^{ij} becomes δ_t^{ij}.

To complete the definition of the likelihood in Equation 3.6 we also need a description for the pdf $p_{S+N}\left(z_t^{ij} \mid x, y, I\right)$, the distribution of gradient intensity in pixel (i, j) given that the target of intensity I is located at (x, y). The effect of a target is to contribute intensity that adds to the noise and background defined values in all pixels affected by the target. Since our measurement model operates upon the differentiated image Δ, the intensity contribution to pixels surrounding a point target is not simply the sensor's point spread function $\gamma_{i,j;\,x,y}$ but rather its differential. However, the background and noise in each pixel is unaffected by the target. Thus in pixels with an intensity contribution from the target, the mean of the background distribution $p_N\left(z^{ij}\right)$ becomes displaced according to the contribution from the target, so defining the pdf $p_{S+N}\left(z_t^{ij} \mid x, y, I\right)$ as:

$$p_{S+N}\left(z_{t}^{ij}\,|\,x,y,I\right)=\frac{\alpha^2}{2\sqrt{2}\left(\alpha^2+\left(z^{ij}+I\,L\left(\gamma_{ij:xy}\right)\right)^2/2\right)^{3/2}} \tag{3.10}$$

where $\mathrm{L}(\gamma_{\mathrm{ij:xy}})$ denotes the Laplacian differential operator applied to the intensity produced by the sensor's point spread function centred at the target position (x,y).

4. Implementation of baseline solution

The baseline implementation of the Bayesian solution to this problem is via a particle filter technique (see [2] [7]). This is a means of implementing a general Bayesian recursive filter without the usual linear-Gaussian restrictions. The recursive Bayesian 'predict-update' relations (Equations 3.5 and 3.6) are implemented to obtain the posterior estimate $p(x_t, \lambda_t \mid z_{1:t})$.

For the baseline implementation, the target state vector is augmented with the target present / absent flag: $[\mathrm{x_t(i)}, \lambda_\mathrm{t}(\mathrm{i})]$ (where, as previously noted, $\lambda = 1$ indicates that a target is present, otherwise $\lambda = 0$).

4.1 Measurement update – structured scene content

The update stage of the filter incorporating measurement information is achieved via weighted resampling - the weight for a particle being proportional to its likelihood. Thus for the i^{th} particle, $[\mathrm{x_t(i)}, \lambda_\mathrm{t}(\mathrm{i})]$, the resampling weight $\mathrm{q(i)} \propto \mathrm{p}\left(\mathrm{z_t} \mid \mathrm{x_t(i)}, \lambda_1(\mathrm{i})\right)$ is defined by Equation 3.6.

If the presence of the target only affects a (small) clump of pixels $\mathrm{C(x)}$ in the vicinity of (x,y), the resample weights may be rewritten so that the weight of each particle for $\lambda = 1$ only depends on the product of likelihood ratios in the vicinity of the particle, rather than on all M×N pixel likelihoods in the image. Thus the resampling weight may be written:

$$q(i) \propto \begin{cases} \displaystyle\prod_{i,j\in C(x(i))} l\left(z_t^{ij} \mid x(i), y(i), I(i)\right) & for\, \lambda(i) = 1 \\ 1 & for\, \lambda(i) = 0 \end{cases} \tag{4.1}$$

where the likelihood ratio in pixel (i,j) for a target of intensity I located at pixel (x,y) is:

$$l\left(z_t^{ij} \mid x,y,I\right)=\frac{p_{S+N}\left(z_t^{ij} \mid x,y,I\right)}{p_N\left(z_t^{ij}\right)} \tag{4.2}$$

This simple trick greatly reduces the computational requirement of the particle filter implementation.

4.2 Prediction step – baseline birth/death model

The prediction phase of the filter from $\mathrm{t}-1$ to t is fairly standard. The particle set is divided into two parts; those for which $\lambda_{\mathrm{t}-1} = 1$ and those for which $\lambda_{\mathrm{t}-1} = 0$, corresponding to the target being present, or absent, at the previous time step. The Markov process defining the birth/death model acts on the two sets of particles in the following manner.

In the transition from $\mathrm{t}-1$ to t, $\lambda_{\mathrm{t}-1} = 1$ particles either remain at $\lambda_\mathrm{t} = 1$ with probability p_{11} or transition to $\lambda_\mathrm{t} = 0$ (target dies) with a probability $1 - \mathrm{p}_{11}$. The process is similar for $\lambda_{\mathrm{t}-1} = 0$ particles (where the probability of target birth, $\mathrm{p_{Birth}} = 1 - \mathrm{p}_{00}$ replaces the probability of death, $1 - \mathrm{p}_{11}$).

For particle i, if $\lambda_t(i) = 0$ then the target does not exist and $x_t(i)$ is undefined. If $\lambda_t(i) = 1$ and $\lambda_{t-1}(i) = 1$, then $x_t(i)$ is given by the dynamics model:

$$x_t(i) = f\big(x_{t-1}(i),\ w_{t-1}(i)\big) \tag{4.3}$$

where $w_{t-1}(i)$ is a random sample drawn from the known pdf of the system driving noise.

Note that if this prediction sends the target out of the sensor field-of-view then the target is assumed to have died and $\lambda_t(i)$ is set to zero. If $\lambda_t(i) = 1$ and $\lambda_{t-1}(i) = 0$, then for this particle the target was born during the transition $t - 1 \rightarrow t$ and so $x_t(i)$ is drawn from the pdf of the target birth distribution.

4.3. Target appearance model (Birth/Death model)

The birth/death (Markov) model is the mechanism by which the track-before-detect algorithm searches for a target. It models the process by which a target may appear (and begin to be tracked) as a transition event which may occur with a specified probability. The following four transitions are possible in each step: (i) Target is born $\lambda_{t-1} = 0$, $\lambda_t = 1$; (ii) Target stays alive $\lambda_{t-1} = \lambda_t = 1$; (iii) Target dies $\lambda_{t-1} = 1$, $\lambda_t = 0$; (iv) Target stays dead $\lambda_{t-1} = \lambda_t = 0$.

When targets are born, they may appear anywhere within the sensor Field of View (FOV) (according to a random uniform distribution over the FOV, and with a zero mean velocity). The probability that the target stays alive is denoted p_{11}, and the probability that it stays dead is p_{00}. The probabilities of the four possible transitions are expressed in terns of p_{11}, and p_{00} in Table 1. (A value of $p_{00} = p_{11} = 0.9$ has been used.)

	$\lambda_{t-1} = 0$	$\lambda_{t-1} = 1$
$\lambda_t = 0$	p_{00}	$1 - p_{11}$
$\lambda_t = 1$	$1 - p_{00}$	p_{11}

TABLE 1. Transition probabilities in the birth/death model for target appearance.

5. Improvements to baseline implementation

When the probability that a target is being tracked is low, most of the particles are inactive (by definition, since only a small fraction of the particles have $\lambda_t(i) = 1$). In this case (i.e. the algorithm is searching for a target) the 'new born' particles are used to propose new candidate target positions and intensity in each step.

According to the birth/death model, the fraction of the 'inactive' particle set that becomes 'new born' is defined according to the probability that a target is born, $p_B = 1 - p_{00}$, for which we have used 0.1. In this case 10% of the 'inactive' particles are potential sources for a target, at the following step, whilst the other 90% of the particles remain 'inactive'. Thus a maximum of 10% of the particles are used to search for a target in each step.

The 'inactive' particles are computationally cheaper to propagate than the 'active' ones (the likelihood ratio of 'inactive' particles is 1, by Equation 3.6). However, once the target has been found then most particles become active with $\lambda_t(i) = 1$. Thus the computational load increases for the easier part of the problem (tracking the target once it has been found).

This wasteful procedure is avoided by an analytic scheme for calculating the probability that the target is present $p(\lambda_t \mid z_{1:t})$, without propagating 'inactive' particles.

5.1. Scheme for calculating the probability that the target is present $p(\lambda_t \mid z_{1:t})$

A scheme has been implemented which calculates the probability that the target is present $p(\lambda_t \mid z_{1:t})$ analytically, without propagating 'inactive' particles, and substantially improves efficiency. The target present / absent flag, $\lambda_t(i)$, is no longer required in the particle state vector, so the number of state dimensions represented by the particle filter is reduced by one. Furthermore, when the probability that a target is already being tracked is calculated to be (near) zero, the analytic scheme uses (practically) all of the particles to search for the target. (Whereas the baseline scheme would use just the fraction p_B of the total number of particles – making the new scheme 10 x more efficient for the assumed value of $p_B = 0.1$.) Space restrictions do not permit a full description of this new method, which will be published separately.

5.2. Marginalised Particle Filter

A further improvement in efficiency of implementation exploits linear sub-structure in the problem and enables the problem to be partitioned into two elements, $x_t = [x_t^l \;\; x_t^n]^T$. This partitioning is possible because the dynamic model is a linear function of the state vector elements x_t^l and the measurement model is a non-linear function of the elements x_t^n. Accordingly, the 'non-linear' states x_t^n represent the target's position in image axes, and its intensity, whereas the 'linear' states x_t^l represent target velocity components. The 'informal labelling' of parts of the state vector as 'linear', 'non-linear', is adopted from [4], which provides a good description of this partitioned approach, known as the Marginalised Particle Filter.

The Marginalised Particle Filter represents x_t^n with particles, and applies one Kalman filter per particle, that provides the conditional distribution for the 'linear states' x_t^l, conditioned upon the 'non-linear' states. For this application, this approach reduces the number of state vector components represented by the particle filter from 5 → 3. This substantially reduces the number of particles required, relative to the standard particle filter, to populate the state space at an equivalent density. (Equivalent to a factor of 100 reduction in the number of particles required, e.g. 100,000 particles over 5 states is 10 per axis, so for the same density, 10^3 = 1000. It is recognised that more effort is required for the Kalman Filter update of each particle, than for propagation of standard particles.)

In the context of the track-before-detect problem, the partitioning applies as follows. The problem is to jointly estimate the probability that the target is present, λ, and if it is present then its position, intensity and velocity.

$$\begin{aligned} p(x_t^l, x_t^n, \lambda_t \mid z_{1:t}) &= p\left(x_t^l \mid x_t^n, \lambda_t, z_{1:t}\right) p(x_t^n, \lambda_t \mid z_{1:t}) \\ &= p\left(x_t^l \mid x_t^n, \lambda_t, z_{1:t}\right) p\left(x_t^n \mid \lambda_t, z_{1:t}\right) p(\lambda_t \mid z_{1:t}) \end{aligned} \tag{5.1}$$

Thus the desired posterior is available via the non-linear state update $p\left(x_t^n \mid \lambda_t, z_{1:t}\right)$, and then a linear state update (conditional on the non-linear update), i.e. $p\left(x_t^l \mid x_t^n, \lambda_t, z_{1:t}\right)$. These two updates are conditional upon the estimate of the probability that the target is present, $p(\lambda_t \mid z_{1:t})$.

5.3. Algorithm Recipe

- Calculate the likelihood ratio for each particle (depending upon whether it represents a new born particle or a 'stayed alive' particle);
- Update the target present probability: $P_{t-1} = p(\lambda_{t-1} = 1 \mid z_{1:t-1})$ to get P_t;
- Update the estimate of the target's state, $p\left(x_t^l, x_t^n \mid \lambda_t = 1, z_{1:t}\right)$. This is an **output**;
- Perform resampling of particles if needed (including 'jitter');
- Carry out the Birth/Death process;
- Prediction step (step 4b of the paper [8]);
- Kalman update of 'linear' states for all 'stayed-alive' particles (step 4c of the paper [8]).

6. Illustration

An illustration of the algorithm's operation, for a single run of the algorithm on synthetic data, is presented in subsection 6.1 for the case where the target is viewed against a structured background. In subsection 6.2 Monte Carlo simulations are used to illustrate the detection sensitivity for the case where the target is viewed against a structured background, and against a noise image (no structure).

6.1. Illustration of algorithm on synthetic data

To illustrate the operation of the filter we have simulated the case of a point target with sensor blurring by a Gaussian point spread function of standard deviation $\sigma_{psf} = 0.5$ pixels.

A sequence of 35 frames of data has been simulated. Each frame consists of an array of 60×40 pixels. There is a structured background scene in this illustration, to which white Gaussian noise of standard deviation 1 intensity unit has been added in each pixel. A target was introduced in frame 6 and deleted in frame 28. The intensity level of the target, prior to blurring, was I=13.

Example frames from the measurement sequence are presented in Figure 2. The target was present from step 6 to 27 inclusive, and so its signal is present in three of the four subplots.

The filter was run with 10,000 particles. The algorithm was applied to the measurement intensities following spatial differentiation (producing images such as those in Figure 3).

The green dots in Figure 3 show samples from the position distribution of the estimated target (i.e. 'stayed alive' particles) at four frames (three of which correspond to the measurements shown in Figure 2). The red dots show samples from the 'new-born' position distribution.

Note that initially samples appear to "swirl" randomly over the image. Several time steps after the target has appeared the particles learn the target location and cluster around it. When the track is well established a tight clump of particles is formed.

The purple ellipse is drawn to enclose the 0.989 probability of the green ('stayed alive') particles (it is a co-variance fit to the position distribution, which is clearly non-Gaussian – particularly before the target is found).

The probability of the target being present is plotted in Figure 4. It takes about ten frames following the appearance of the target before this probability rises above 0.9. The mean position estimate is also plotted in Figure 4 for those steps when the filter's assessment of the probability that a target is present exceeds 0.9.

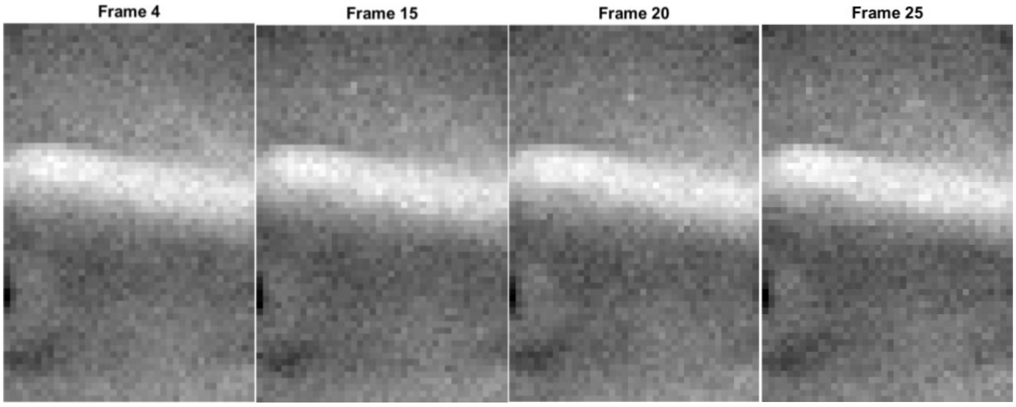

FIGURE 2. Example measurement frames showing structured background and dim target (frames 15, 20 & 25). Before blurring by the psf, the target's intensity was 13 times the standard deviation of Gaussian noise. Blurring psf standard deviation was 0.5 pixels.

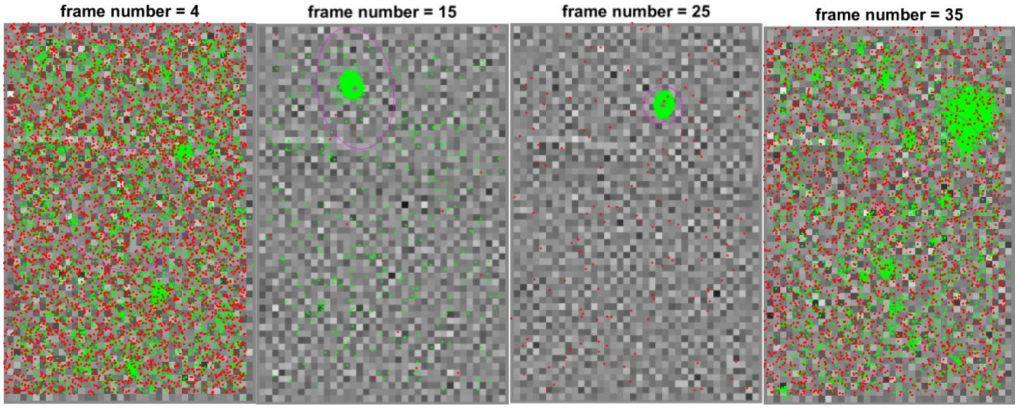

FIGURE 3. Illustration of particle position distribution. Track-before-detect algorithm output superimposed on spatially differentiated images (input to algorithm).

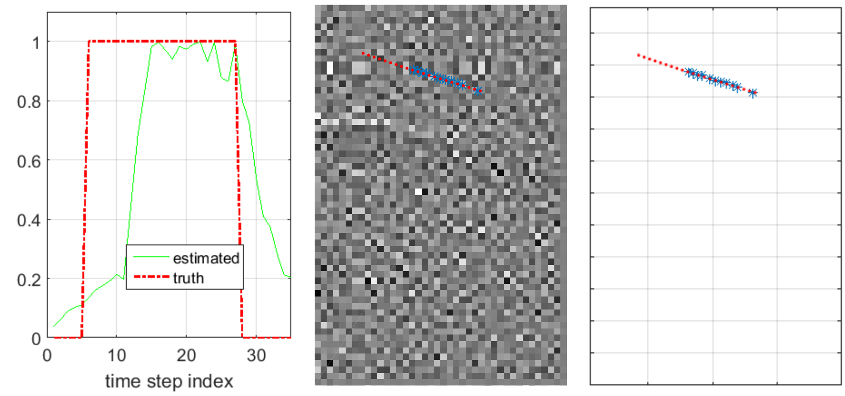

FIGURE 4. (left) Estimated probability that the target is present (green) compared with true target appearance (red) v.s. time step. (middle & right) Estimated mean target position (blue asterisk), plotted when the probability of the target being present > 0.9. (Truth shown by red dots.) Image shows the final frame of differentiated image.

6.2 Detection sensitivity via Monte Carlo simulations

Monte Carlo (MC) simulations using synthetic data are used to establish the detection sensitivity of the algorithm. The target's initial speed was randomly varied in each replication, and each true target path generated by a random (different in each rep) driving noise process (in the Discrete White Noise Acceleration model). Each frame of the random Gaussian noise is generated independently.

The green lines in Figure 5 show the detection performance (i.e. the filter's assessment of the probability that the target is present) for each of 10 MC replications, for two different experiments. The plot on the left shows the detection performance for the case where a target is viewed against a structured scene (as per illustration in section 6.1) with Gaussian-blurred target (whose intensity prior to blurring was I = 13). The plot on the right shows the corresponding performance for the case without a structured background (so no spatial differentiation) and with target intensity of I = 6 (prior to blurring). Note that for the unstructured case, the background noise is Gaussian distributed, but is modelled as a scaled t-distribution with 2 degrees of freedom. If the model were perfectly matched to the data, slightly improved detection performance may result. Even so, a target of this intensity is not visible to the human eye in the measurement data, yet is reliably detected by the algorithm.

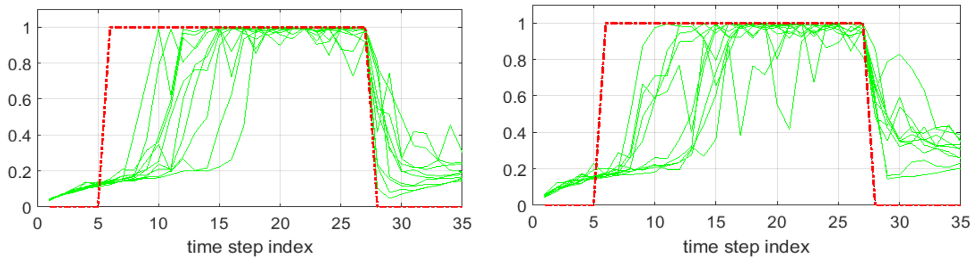

FIGURE 5. Estimated probability that the target is present (green) & true target appearance (red) v.s. time step for 10 Monte Carlo reps. (left) Target viewed against a structured scene (as per illustration in section 6.1). Target's (un-blurred) intensity I = 13; (right) Viewed against a background without structure, I = 6.

7 Summary & Conclusions

An efficient particle filter implementation of the track-before-detect algorithm has been developed, based upon the simple algorithm used in [1] [2]. The simple algorithm has been extended to use a Marginalised Particle Filter [4] to exploit linear sub-structure within the dynamic and measurement model. When combined with an analytic scheme for calculating the probability that a target is being tracked, the resulting reduction (by 3 dimensions) in the particle filter state vector (for this problem) provides a factor of 10→100 improvement in efficiency c.f. a basic particle filter.

Synthetic data has been used to establish the sensitivity of the algorithm. When the algorithm is used in a mode where input images contain substantial 'structure', its sensitivity is reduced by approximately a factor of 2 compared with that achieved without structure. An illustration of the algorithm for the case where a dim target moves over a structured scene shows detection and tracking of a target that is difficult for the human eye to identify in the measurement data. For the unstructured background, it is able to detect and track a target that is not visible to the human eye in the measurement data.

Acknowledgement

This work was funded by the Defence Science Technology Laboratory (Dstl) under contract number DSTLX-100010056. Dstl is part of the UK Ministry of Defence (MoD).

REFERENCES

[1] S. MASKELL, M. ROLLASON, D. SALMOND & N. GORDON, "Efficient particle filtering for multiple target tracking with application to tracking in structured images," *SPIE signal & data processing of small targets,* 2002.

[2] M. ROLLASON AND D. SALMOND, "A particle filter for track-before-detect of a target with unknown amplitude," *IEE Colloquium on Target Tracking: Algorithms and Applications, Twente,* 2001.

[3] Y. BOERS AND J. DRIESSEN, "Multitarget particle filter track before detect application," *IEE Proc. Radar Sonar Nav.,* vol. 151, no. 6, p. pp 351, 2004.

[4] F. GUSTAFSSON, Statistical Sensor Fusion, Studentlitteratur, ISBN 978-91-44-07732-1, 2012.

[5] C.-Y. LI AND H.-B. JI, "Marginalized Particle Filter based Track-Before-Detect Algorithm for Small Dim Infrared Target," in *Int. Conf. on Artificical Intelligence & Computational Intelligence*, 2009.

[6] A. H. LETTINGTON, M. P. ROLLASON AND S. TZIMOPOULOU, "Image restoration using a two-dimensional Lorentzian probability model," *Journal of Modern Optics,* vol. 40, no. 5, 2000.

[7] N. GORDON, D. SALMOND AND A. SMITH, "Novel approach to non-linear/non-Gaussian Bayesian state estimation problems," *IEE Proceedings on Radar, Sonar and Navigation,* vol. 140, p. 107, 1993.

[8] T. SCHON, F. GUSTAFSSON AND P. NORDLUND, "Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models," *IEEE Trans. Sig. Proc. ,* vol. Vol 53 , 2005.