Proceedings of the American Control Conference
Arlington, VA June 25-27, 2001

# A particle filter for track-before-detect

D J Salmond* and H Birch†
Defence Evaluation and Research Agency,
Farnborough, Hants, UK

## Abstract

A Bayesian track-before-detect particle filter is proposed. 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 a target is present.

## 1 Introduction

In the usual approach to target tracking, one or a small number of position 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 (S/N) ratio is high, so that a good probability of detecting the target can be achieved while maintaining a low false alarm rate . However, for small S/N ratios information loss may be significant, so that 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 and is typically solved via dynamic programming or maximum likelihood techniques.

In this study, the track-before-detect problem has been approached via a Bayesian particle filter. Until the advent of particle filtering methods, it was not computationally feasible to implement a full Bayesian track-before-detect scheme (also see Ballantyne et al [1]). This approach has a number of potential advantages relative to many previous methods:

(i) The method provides a probability distribution for the target state (i.e. a measure of uncertainty) rather than only a point estimate.

(ii) The possibility of a target appearing in the sensor field-of-view (FOV) is explicitly modelled, so that the probability of a target being present is available from the filter.

(iii) The solution is valid for structured and / or non-Gaussian background noise, although the distribution of this structured noise must be known (and, admittedly, in many practical cases this is unlikely to be available).

(iv) An extended target and/or the effects of a point spread function can be accommodated.

(v) The method is not restricted to constant velocity trajectories - some stochastic target manoeuvre can be accommodated.

The track-before-detect particle filter has been demonstrated for the special case of a manoeuvring point target with high levels of background noise (when the appearance and movement of the target cannot be detected by a human observer).

## 2 Problem formulation

A staring EO sensor observers a region of the x-y plane. Each pixel or resolution cell of the sensor corresponds to a square region of dimension $\Delta \times \Delta$, and the sensor array consists of $N \times M$ pixels. It is assumed that at time step k, the output of all NM resolution cells are read co-incidently and the measured intensity of pixel (i,j) is denoted $z_{ij}(k)$. Thus, following the notation of Tonissen and Bar-Shalom [2], the complete sensor measurement at time step k is denoted:

$$Z(k) = \{ z_{ij}(k) : i = 1, ..., N, j = 1, ..., M \}.$$

If a target is present and its centroid is at position (x, y), it may contribute to the pixels in that vicinity. The contribution of the target to pixel (i,j) is denoted $f_S(i,j,x,y)$ and this is assumed to be known. This could be due to the extent of the target or to the sensor point spread function. If the target is a point object, the target intensity will be distributed over the surrounding resolution cells according to the sensor's point spread function. A truncated 2-D Gaussian density with circular symmetry is a common model for such a function. Thus for a point target of intensity I at position (x,y), the contribution to pixel (i,j) may be approximated by

---

* E-mail address: djsalmond@dera.gov.uk
† H Birch is now with the Research and Development Department of the British Broadcasting Corporation



0-7803-6495-3/01/$10.00 © 2001 AACC

$$f_s(i,j,x,y) = \begin{cases} \dfrac{\Delta^2 I}{2\pi s^2} \exp\left[ -\dfrac{(i\Delta - x)^2}{2s^2} - \dfrac{(j\Delta - y)^2}{2s^2} \right] & \text{for } |i\Delta - x| < 3 \text{ and } |j\Delta - y| < 3 \\ 0 & \text{otherwise.} \end{cases}$$

Here the parameter s represents the extent of blurring. Note that this expression is a somewhat crude approximation to the integral of the point spread function over a pixel, and smearing due to target motion is not represented. Also the approximation is only useful for $s \gg \Delta$ - see [2] for a more complete discussion. Note that as $s/\Delta \rightarrow 0$ and the point spread function tends to a delta function, a point target will only contribute to the resolution cell in which it falls.

It is assumed that the sensor pixels are corrupted by a background noise of known distribution. This noise may be structured - for example, the noise may have a non-zero mean which varies between pixels. However, the distribution of the noise for each pixel is assumed to be known; also the noise process is assumed independent between pixels and from frame to frame. Thus, although our solution relaxes the usual assumption that background noise is identically distributed amongst pixels, the distribution of the background clutter is assumed to be available.

At most one target at a time may be present in the sensor's scanned region. Initially, at time step k=0, 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 is assumed to be known (for example, uniform over the field-of-view). Following its appearance, the target then proceeds on a trajectory until it disappears (i.e. no signal is present) or passes out of the scanned region. Following common practice, the birth / death of a target is modelled as a Markov process with parameter $\lambda$. $\lambda=1$ indicates that a target is present, otherwise $\lambda=0$. A transition from $\lambda=0$ to $\lambda=1$ occurs with probability $P_B=\Pr\{\lambda_{k+1}=1 \mid \lambda_k=0\}$ for all $k\geq 0$. For a target within the scanned region, a transition from $\lambda=1$ to $\lambda=0$ occurs with probability $P_L=\Pr\{\lambda_{k+1}=0 \mid \lambda_k=1\}$ for all k>0. If a target passes out of the scanned region, the target is assumed to die and $\lambda$ becomes 0 with probability 1. $P_B$ and $P_L$ define the birth / death behaviour of targets within the observation region and are assumed to be known. In particular, the average lifetime of a target within the observation region is $1/(1-P_L)$ time steps and the average period between the death of one target and the birth of another is $1/(1-P_B)$ time steps.

When a target is present, it is assumed to move according to a known dynamics model of the form $\underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{w}_k)$ where $\underline{w}_k$ is an independent noise process of known pdf and $\underline{x}_k$ is the target state vector. This is equivalent to knowledge of the transition density $p(\underline{x}_{k+1} \mid \underline{x}_k)$. The filter described in this paper is applicable to any Markov model of this form - there is no restriction to linear-Gaussian dynamics.

## 3 Formal Bayesian solution

From a Bayesian perspective, a complete solution of the above problem is given by the posterior pdf of $(\underline{x}_k, \lambda_k)$: $p(\underline{x}_k, \lambda_k \mid Z'(k))$, where $Z'(k)$ denotes the complete set of all past images, $Z'(k) = \{ Z(k), Z(k-1), ..., Z(1)\}$.

The construction of this posterior pdf depends on the transition density between time steps and the measurement likelihood. First consider the transition between time steps:

$$p(\underline{x}_{k+1}, \lambda_{k+1} \mid \underline{x}_k, \lambda_k) =$$

$$p(\underline{x}_{k+1} \mid \underline{x}_k, \lambda_k, \lambda_{k+1})\, p(\lambda_{k+1} \mid \underline{x}_k, \lambda_k) .$$

Apart from the possibility of an existing target passing out of the sensor field-of-view, the transition of $\lambda$ is independent of $\underline{x}_k$ and is defined by the birth/death Markov model, i.e. $P_B$ and $P_L$. If $\lambda_{k+1}=0$, the target is not present and $\underline{x}_{k+1}$ is undefined, otherwise the pdf of $\underline{x}_{k+1}$ conditional $\underline{x}_k$ and $\lambda_k$ is given by:

$$p\big(\underline{x}_{k+1} \mid \underline{x}_k, \lambda_k, (\lambda_{k+1}=1)\big) = \begin{cases} p(\underline{x}_{k+1} \mid \underline{x}_k) & \text{for } \lambda_k = 1 \\ p_B(\underline{x}_{k+1}) & \text{for } \lambda_k = 0 \end{cases} ,$$

where the transition density $p(\underline{x}_{k+1} \mid \underline{x}_k)$ is defined by the target dynamics model and $p_B(.)$ is the initial pdf of a target on its appearance.

The likelihood $p(Z(k) \mid \underline{x}_k, \lambda_k)$ of the state given the pixel measurements is given by (omitting the time index k):

$$p(Z \mid \underline{x}, \lambda) = \begin{cases} \prod_{i,j} p_{S+N}(z_{ij} \mid x, y) & \text{for } \lambda = 1 \\ \prod_{i,j} p_N(z_{ij}) & \text{for } \lambda = 0 . \end{cases}$$

Here, $p_N(z_{ij})$ is the pdf of the background noise in pixel (i,j) and $p_{S+N}(z_{ij} \mid x,y)$ is the pdf of the target signal + noise in pixel (i,j) given that the target is located at (x,y). Note that given the target location, the S+N distributions in each pixel are independent (as are the pure noise pixels), hence the product of the pdfs in the above likelihood. If the presence of the target only affects a (small) clump of pixels in the vicinity of (x,y), then

$$p(Z \mid \underline{x}, \lambda=1) = \prod_{i,j} p_{S+N}(z_{ij} \mid x, y) =$$

$$\prod_{i,j \in \mathbb{C}(\underline{x})} p_{S+N}(z_{ij} \mid x,y) \prod_{i,j \notin \mathbb{C}(\underline{x})} p_N(z_{ij}) ,$$

where $\mathbb{C}(\underline{x})$ is the set of subscripts of pixels affected by the target (with state vector $\underline{x}$). It is assumed that $p_N(z_{ij})$ and $p_{S+N}(z_{ij} \mid x,y)$ are known for all i,j.

## 4 Implementation of Bayesian solution via particle filters

We propose to implement the Bayesian solution to this problem via a particle filter technique (see [3] and [4]). This is a means of implementing a general Bayesian recursive filter without the usual linear-Gaussian restrictions. 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 $p^{th}$ particle, $(\underline{x}^*(p)\ ,\ \lambda^*(p))$, the resampling weight $q(p) \propto p(Z \mid \underline{x}^*(p), \lambda^*(p))$ - where the asterisk indicates the prior value of the particle before resampling. Since the weights are only required up to proportionality, we may divide through by $\prod_{i,j} p_N(z_{ij})$ and set:

$$q(p) \propto \begin{cases} \prod_{i,j \in \mathbb{C}(\underline{x}^*(p))} \ell(z_{ij} \mid x^*(p), y^*(p)) & \text{for } \lambda^*(p) = 1 \\ 1 & \text{for } \lambda^*(p) = 0\ , \end{cases}$$

where $\ell(z_{ij} \mid x, y) = p_{S+N}(z_{ij} \mid x, y) / p_N(z_{ij})$, the likelihood ratio in pixel (i,j) for a target at (x,y). Thus the weight of each particle for $\lambda=1$ only depends on the product of likelihood ratios in the vicinity of the particle. This simple (and rather obvious) trick greatly reduces the computational requirement of the particle filter implementation. For example, for a 20×20 (N=M=20) array of pixels, the brute force approach of setting $q(p) = p(Z \mid \underline{x}^*(p)\ ,\ \lambda^*(p))$ would entail the evaluation of 400 pixel likelihoods rather than a few likelihood ratios in the vicinity of the particle - an important consideration for a filter employing a large number of particles.

Note that, in principle, the filter can accommodate structured noise / clutter (i.e. $p_N(z_{ij})$ may depend on i,j) provided its distribution is known over the pixel array. This could be useful, for example, if it were possible to calibrate the array to some fixed background clutter or detector response (including so-called "dead pixels"). For instance, for a "stationary" scene it would be possible to learn the background distribution by extended observation of the target-free scenario.

The prediction phase of the filter $k \rightarrow k+1$ is fairly standard. The particle set is divided into two parts; those for $\lambda=1$ and those for $\lambda=0$, corresponding to target present and target absent. In the transition from k to k+1, $\lambda_k = 1$ particles either transition to $\lambda^*_{k+1} = 0$ (target dies) with a probability $P_L$ or remain at $\lambda^*_{k+1} = 1$ with probability $1\text{-}P_L$ . The process is similar for $\lambda_k=0$ particles. The transition is effected for individual particles via a random number generator. If $\lambda^*_{k+1}(p)=0$, then the target does exist for particle p and $\underline{x}^*_{k+1}(p)$ is undefined. If $\lambda^*_{k+1}(p)=1$ and $\lambda_k(p)=1$, then $\underline{x}^*_{k+1}(p) = \underline{f}\ (\underline{x}_k(p)\ ,\ \underline{w}_k(p))$ where $\underline{w}_k(p)$ 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 $\lambda^*_{k+1}$ is set to zero. If $\lambda^*_{k+1}(p)=1$ and $\lambda_k(p)=0$, then for this particle the target was born during the transition $k \rightarrow k+1$ and so $\underline{x}^*_{k+1}(p)$ is drawn from the pdf of the target birth distribution $p_B(\underline{x})$.

## 5 Illustrative results

To illustrate the operation of the filter, we have simulated the particular case of a point target with no sensor blurring. Thus, the target only activates the pixel in which it falls:

$$f_s(i,j,x,y) = \begin{cases} I & \text{for } |i\Delta - x| < \Delta/2 \text{ and } |j\Delta - y| < \Delta/2 \\ 0 & \text{otherwise,} \end{cases}$$

where I is the known intensity of the target. The background noise for this example is assumed to be zero mean Gaussian with variance $\sigma^2$ for all pixels, i.e. $p_N(z_{ij}) = \mathbb{N}(z_{ij}\ ;\ 0, \sigma^2)$ for all i,j. If the point target is present in pixel (i,j), the pixel response is assumed to be the sum of the target intensity and the background noise, i.e. $p_{S+N}(z_{ij} \mid x,y) = \mathbb{N}(z_{ij}\ ;\ I, \sigma^2)$. Thus the likelihood ratio is given by

$$\ell(z_{ij} \mid x,y) = \begin{cases} \exp\left[\dfrac{-I(I-2z_{ij})}{2\sigma^2}\right] & \text{for } |i\Delta - x| < \Delta/2 \text{ and } |j\Delta - y| < \Delta/2 \\ 1 & \text{otherwise.} \end{cases}$$

Also, since the set $\mathbb{C}(\underline{x})$ consists solely of the pixel in which the point target falls, the particle weights are given by

$$q(p) \propto \begin{cases} \exp\left[\dfrac{-I(I-2z_{ij})}{2\sigma^2}\right] & \text{for } \lambda^*(p) = 1 \text{ and } |i\Delta - x^*(p)| < \Delta/2 \text{ and } |j\Delta - y^*(p)| < \Delta/2 \\ 1 & \text{otherwise.} \end{cases}$$

A sequence of 30 frames of data has been simulated. Each frame consists of an array of 20×20 pixels. The standard deviation of the background noise level in each pixel is $\sigma = 10$ units. A target was introduced in frame 6 and deleted in frame 22. The intensity level of the target was I=20, resulting in a signal-to-noise ratio of $(I/\sigma)^2 = 4 = 6\,\text{dB}$ . Thus, for the 400 pixels in a frame, one would expect about 9 noise pixels to exceed the target signal level. When present, the target motion was simulated according to a second order model of the form

$$\begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix}_k + \begin{pmatrix} 1 & 0 \\ 1/2 & 0 \\ 0 & 1 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} w_X \\ w_Y \end{pmatrix}_k .$$

The state vector here is the target position and velocity in units of pixels and frame rate, and k is the frame number. The driving noise is a zero mean Gaussian process of variance $q = E[\ w_X^2\ ] = E[\ w_Y^2\ ] = 0.05^2$.

Six frames of the data sequence (frame numbers 1, 5, 9, 13, 17 and 21) are shown in fig 1. Individual pixel intensity is shown as a grey scale in 64 (linear) levels, with white indicating the highest intensity. The 64 levels are a linear scale between the dimmest and the brightest pixel in the full 30 frame sequence. The position of the target, if present, is marked by a circle. It is clearly (visually) difficult to detect the appearance of the target at this signal-to-noise level.

The track-before-detect particle filter has been applied to this data set. The filter state vector consists of the position and velocity of the target augmented with the target present / absent flag: $(x, \dot{x}, y, \dot{y}, \lambda)_k$. The dynamics model employed by the filter is the same as the above target generation model, so the filter is perfectly matched to the data in this respect. The probability of the target appearing on a particular frame is set to $P_B = 0.1$ and the probability of the target dying is also $P_L = 0.1$. The initial pdf of the target's location, given that it has just appeared and prior to any measurements, is uniform over the sensor field-of-view. The initial prior pdf of its velocity is assumed to be uniform over [-1, 1] in the x and y directions, i.e.

$$p_B(x) = \mathrm{Un}(x; 0.5, 20.5) \qquad p_B(\dot{x}) = \mathrm{Un}(\dot{x}; -1, 1)$$
$$p_B(y) = \mathrm{Un}(y; 0.5, 20.5) \qquad p_B(\dot{y}) = \mathrm{Un}(\dot{y}; -1, 1).$$

The initial information at k=1 is that the target is not present, i.e. $p(\lambda_1=0)=1$, so all prior particle filter samples are initialised at $\lambda_1^*(p)=0$ for $p=1, ..., N_S$. The filter has been run with $N_S=30000$ samples. To alleviate the problem of sample degeneracy, a "roughening" procedure was used with K=0.2 (see [3]).

Subsets of target samples $\left(x_k(p), y_k(p) \mid \lambda_k(p) = 1, p \in S \subset \{1, ..., N_S\}\right)$ from six of the data frames are shown in fig 2. Note that initially samples appear to "swirl" randomly over the observation plane. Eventually, several time steps after the target has appeared, the particles begin to learn the target location and to cluster around it. When the track is well established, a tight clump of particles is formed. Note that the filter is able to exploit information on the shape of the pixels - see frame 21 of fig 2 where the straight edge in the target location distribution is due to the pixel boundary.

Fig 3 shows the "track" produced by the filter. This is simply the mean of the set of target samples. The target track is shown as a dashed line if the filter's assessment of the probability of the target being present is below 0.6. The sample mean in this case wanders near the centre of the field-of-view, as would be expected if no target were present. The continuous line shows the track when the probability is greater than 0.6. The continuous track follows the actual target path reasonably well. (Note that actual target trajectory is not a constant velocity path - although admittedly the manoeuvre is gentle.) The probability of the target being present is plotted in fig 4. It takes about seven frames following the appearance of the target before this probability rises above 0.6. The track is well established by frame 15. Subsequently, the probability remains above 0.9 until frame 25, after which the target disappears. Within two time steps the probability falls below 0.6 and the mean of the sample set reverts back to the centre of the field-of-view (see fig 3).

## 6 Conclusions

A particle filter has been used to implement a Bayesian approach to the track-before-detect problem. The operation of the filter has been demonstrated for the case of a low signal-to-noise point target against a background of Gaussian noise. However, the filter can also be applied to non-Gaussian background noise and extended targets (or point targets that have been blurred by the sensor). Also, the filter could be applied with any shape of pixel. Extensions of the method to multiple targets and unknown target intensity should be pursued.

## 7 References


1 D.J.Ballantyne, Y.Chan and M.A.Kouritzin, (2000), "A novel branching particle method for tracking", *SPIE Signal and data processing of small targets*, Vol 4048, April 2000.

2 S.M.Tonissen and Y.Bar-Shalom, (1998), "Maximum likelihood track-before-detect with fluctuating target amplitude", *IEEE Trans on Aerospace and Electronic Systems*, Vol 34, No 3, July 1998.

3 N.J Gordon, D.J. Salmond and A.F.M. Smith, (1993), "Novel approach to non-linear/non-Gaussian Bayesian state estimation problems", *IEE Proceedings on Radar, Sonar and Navigation*, Vol 140, No 2, pages 107-113, 1993.

4 A.Doucet, J.F.G. de Freitas and N.J.Gordon (editors), (2001), *Sequential Monte Carlo methods in practice*, Springer-Verlag, to be published, 2001.


## Acknowledgement


This research was sponsored by the UK MOD Corporate Research Programme, TG3.

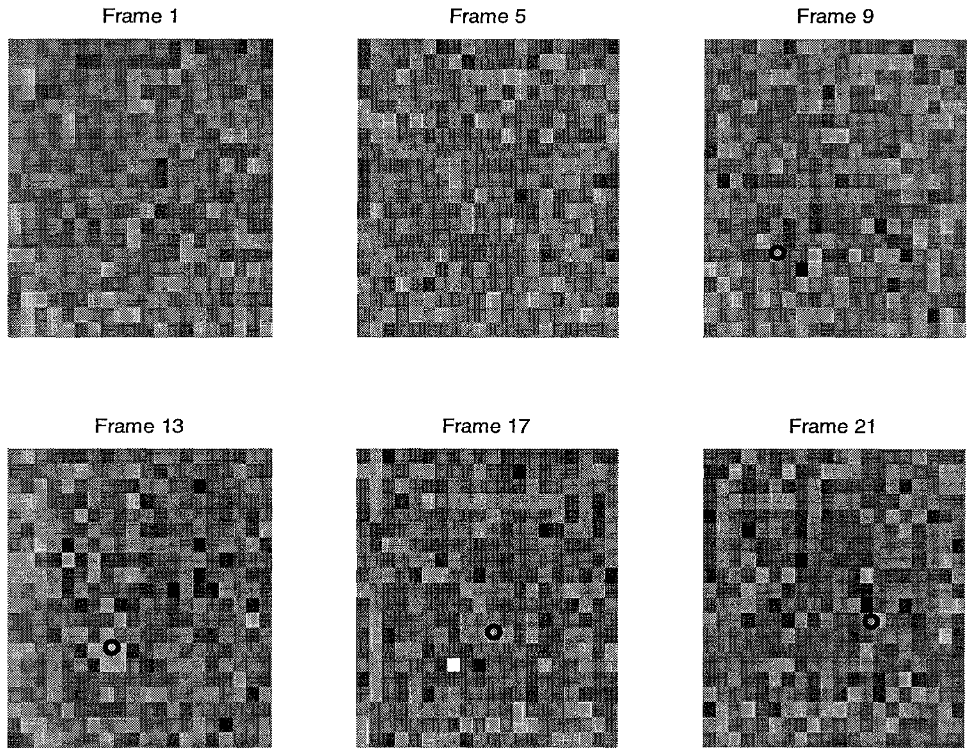


Fig 1: Six frames from the data sequence, pixel intensity indicated by grey scales

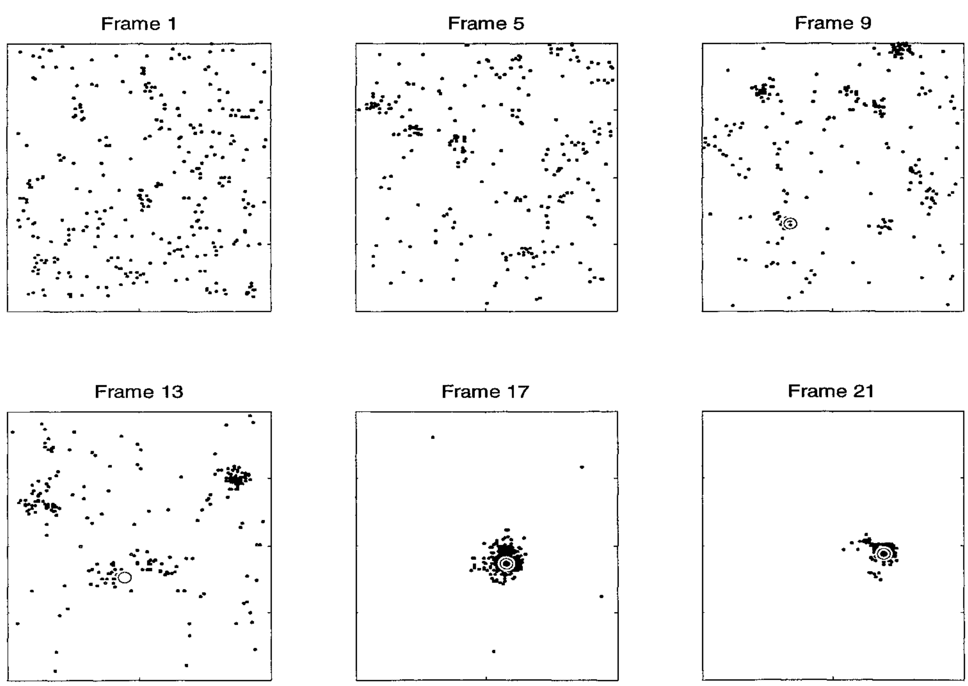


Fig 2: Subsets of target samples from the particle filter, circle indicates target position

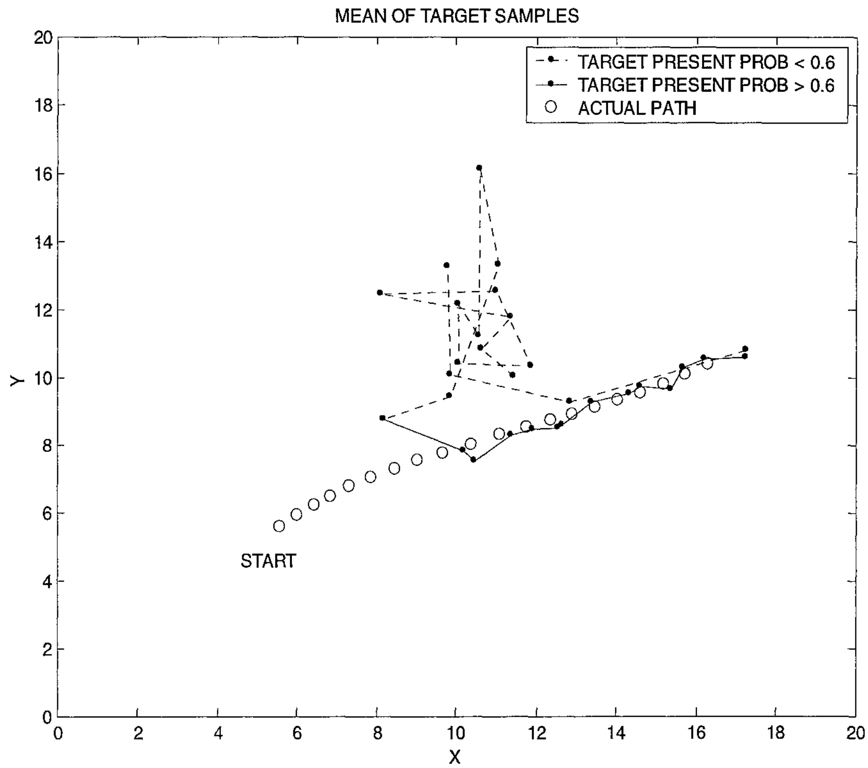


Fig 3: Target track: mean of target samples

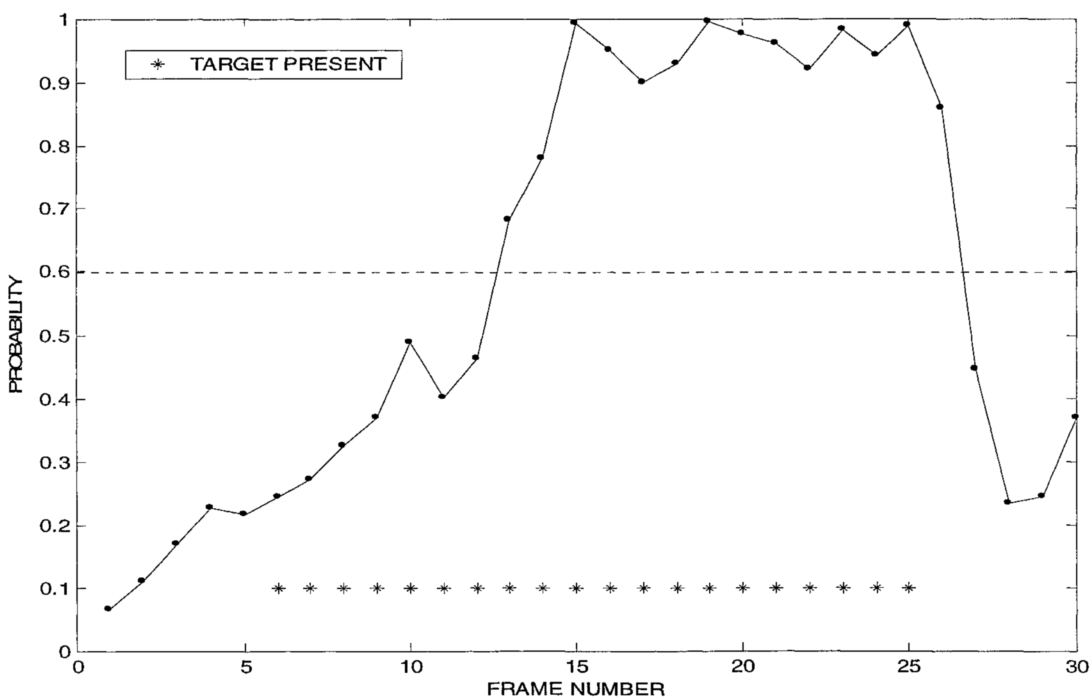


Fig 4: Probability that the target is present (proportion of target present samples)