Needle Picking: A Sampling Based Track-Before-Detection Method for Small Targets

Fatih Porikli

Mitsubishi Electric Research Labs, Cambridge, MA, USA

ABSTRACT

We present a computationally efficient track-before-detect algorithm that achieves more than 50% true detection at 10^{-6} false alarm rate for pixel sized unknown number of multiple targets when the signal-to-noise ratio is less than 7dB. Without making any assumptions on the distribution functions, we select a small number of cells, so called as needles, and generate motion hypotheses using the target state transition model. We accumulate cell likelihoods along each hypothesis in the temporal window and append the accumulated values to the corresponding queues of the cell positions in the most recent image. We assign a target in case the queue maximum is greater than a threshold that produces the specified false alarm rate.

Keywords: Automatic target detection, track-before-detect, radar, ATR, TBD

1. INTRODUCTION

Detection of very small targets in noisy measurements is a challenging task with critical applications including high-end radar systems. Typically, the returning echoes of a radar emission are amplified then filtered to extract a 2D image data where each cell corresponds to the received power in a particular spatial location on range (bins) and azimuth (beams). In addition to potential target reflection signature, the image data also comprises the receiver noise, electromagnetic interference and scene clutter.

Most simple schemes[1,2] apply a threshold to the input image and label the cells exceeding this threshold value as candidate targets. If this threshold is too low, then more targets will be detected at the expense of increased numbers of false alarms. Conversely, if the threshold is too high, then fewer targets will be detected, but the number of false alarms will also be low. Often the threshold is set in order to achieve a constant false alarm rate (CFAR) by adaptively estimating the level of the noise floor around the cell using the background statistics. This is acceptable as long as the signal-to-noise (SNR) and signal-to-clutter (SCR) ratios are sufficiently high. However, for lower signal ratios where targets cannot be easily distinguished from the cluttered noisy background, such cell thresholding approaches give high rates of false detections.

Instead of making a decision solely based on the current image, detectors can be supplied with a temporal window of previous measurements to allow the detection of low signal ratio targets. Evidence of being a target is accumulated by integrating individual cell likelihoods over time in the temporal window. In other words, hypothetical targets are tracked even before they are detected. This class of algorithms is so often called as track-before-detect[3] (TBD).

Ideally, the evidence accumulation should be done by evaluating all possible states of the dynamic and intrinsic evolution of the target. Here a target state may correspond, for instance, to the position and velocity of the target in the image and the intensity of the underlying cell. For simplicity, the state evolution is usually modeled by a linear process especially when the temporal window duration is short. However, the observed image is a stochastically sampled process and has only a nonlinear relation with the target state albeit the target distribution characteristics are assumed to be available. Besides, high intensity cell responses only weakly correlated to the target locations. As a result, analytically intractable number of states can be spawned for most basic specifications.

One way to render this problem feasible is to quantize the state space and use discrete valued target models. Several grid methods have been developed to estimate the evidence in discrete space including Bayesian MAP

Send correspondence to Fatih Porikli, e-mail: fatih@merl.com

estimator,[4] ML estimator[5] (e.g. Viterbi[6]), and statistical graph networks (e.g. hidden Markov models[7]). The Bayesian estimator is an approximation to the posterior distribution of the target state based. On a uniformly spaced set of states, which is augmented with a null state to indicate the possibility of no target case, it applies the Bayes rule by imposing certain heuristics on the state transition probability and marginal likelihoods. Among those, the parameters of the probability of target existence and probability of target discontinuation control the detection performance and can be tuned to optimize detection performance. The selection of the quantization steps is a trade off between estimation accuracy, which improves with finer resolution, and computational requirements. Bayesian estimator selects the state with the highest probability by recursively defining the probability of the target occupying a particular location by the superposition of all of the possible paths to that position. If the accumulated probability is higher than the null state probability, then a detection is reported. Rather than accumulating the probability from alternate paths, an ML estimator selects the single best path. Within the quantized state space Viterbi algorithm is designed to find the most likely sequence of states by maximizing the joint posterior probability of the sequence of states. Propagating likelihoods through dynamic models on the most likely state sequence leading into a current state and backtracking the Viterbi algorithm has linear complexity in time. One advantage of this is that it always produces an estimate consistent with the dynamic model.

The problem with using a discrete state space is that it leads to high computation and memory requirements. An alternative is to use a sequential analogue of the Markov chain Monte Carlo (MCMC) batch method such as particle filter[8,9] to accumulate the evidence within the Bayesian framework. It is a numerical approximation technique that uses randomly placed samples instead of fixed grid. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this characterization becomes an equivalent representation to the usual functional description of the posterior pdf, and the particle filter approaches the optimal Bayesian estimate. Despite the particle filtering may achieve similar estimation performance for lower cost by using less sampling points than would be required for a discrete grid, it usually requires a considerable amount of particles to effectively approximate the continuous probabilistic distributions, thus, computational burden for high dimensional state spaces, e.g. where acceleration and non-linear motion are parameterized, becomes an issue.

Instead of using a numerical model for the target distribution, the multiple-hypothesis tracker (MHT) imposes a parametric representation to reduce the computational load. It allows a hypothesis to be updated by more than one consecutive state at each update, spawning multiple possible hypotheses. With each received image, all existing hypotheses are updated and unlikely ones are removed to upper bound the computational complexity. The probabilistic MHT[10] (PMHT) uses a recursive expectation maximization such as Kalman filter to compute in an optimal way the associations between the measurements and targets instead of measurement-to-hypothesis assignment. The probability that each measurement belongs to each hypothesis is estimated using a maximum a posteriori method. In other words, the PMHT uses soft posterior probability associations between measurements and targets. These soft associations can be considered as mapping the problem from discrete (i.e. of combinatorial complexity) to continuous (i.e. amenable to iterative algorithms). In histogram PMHT (H-PMHT) version[3] the received energy in each cell is quantized, and the resulting integer is treated as a count of the number of measurements that fell within that cell. The sum over all of the cells is the total number of measurements taken. The probability mass function for these discrete measurements is modeled as a multinomial distribution where the probability mass for each cell is the superposition of target and noise contributions. Rather than using the whole image, maximum likelihood joint probabilistic data association[11] (JPDA) reduces the threshold to a low level and then applies a grid-based state model for estimation to avoid track coalescence. Another approach to detect targets in a TBD fashion is to apply a state parameter mapping, so called as Hough transform,[12] after quantizing the parameters.

In addition to being computationally expensive, above approaches assume the signal, clutter, and noise distribution functions to be known beforehand due to their dependency on the likelihood ratio function itself. Furthermore, they impose single-stage Markovian updates (as particle filters) for the computation of the cell likelihoods even though a larger portion of the previous measurements is often available.

Here, we present a computationally efficient, grid based TBD algorithm that can utilize all available measure-

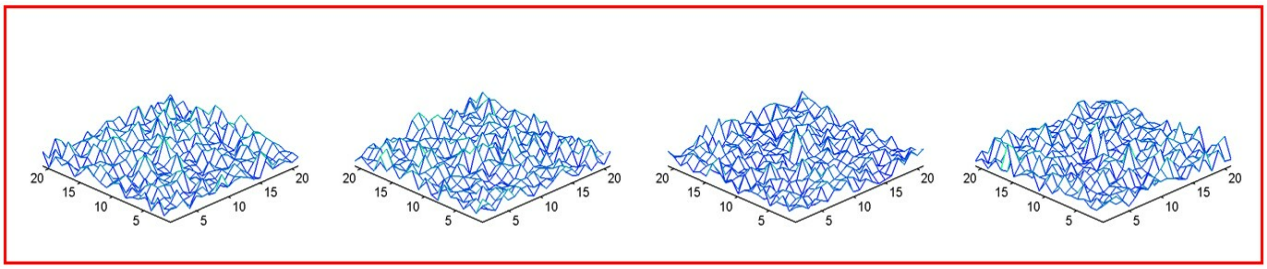

Figure 1. Four sample snapshots of the measurements when SNR=20dB and SCR=7dB. There is a single target in the center of each image. Even for these very small 21×21 images the target signal cannot be easily identified easily by eye. Typically image size is 1000×100, which means targets will be even less distinguishable from the background.

ments in the given temporal window without imposing any assumptions on the probability distributions. Our goal is to achieve at least 50% true detection rate at only 10^{-6} false alarm rate for the pixel sized multiple targets when the overall signal-to-noise and signal-to-clutter ratios are less than 7dB. A snapshot of sample images under these settings is given in Fig. 1. In addition, the number of multiple targets is unknown and should be estimated from the measurements. By picking a small number of cells, so called as needles due to the mutually independent nature of individual cell measurements for targets smaller than the physical cell coverage, our method collects potential footprints of targets in each image. Using the state transition models, it generates a set of hypotheses and aggregates the cell likelihoods along each hypothesis. In the streaming mode, needle selection only applies to the current image by updating the hypotheses for the needles in the temporal window.

In the following sections we discuss the details of the needle picking algorithm and present quantitative performance evaluation results.

2. NEEDLE PICKING

The observed image I_t comprises noise n_t and clutter c_t in addition to target signal z_t

$$I_t : z_t + n_t + c_t \tag{1}$$

where clutter refers to echoes returned from background objects such as ground, sea, atmospheric conditions (including rain, snow, hail, sand storms, clouds, and turbulence), man-made objects such as buildings, etc. These constituents are assumed to be independent random variables. Noise is typically present in the communication channel and appears as random variations superimposed on the desired echo signal received in the radar receiver.

Let s_t be the target state at time t consisting of position and velocity in the image

$$s_t = [x_t \quad \dot{x}_t \quad y_t \quad \dot{y}_t]' \tag{2}$$

and the state transition is modeled by linear motion

$$s_t = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} s_{t-1}, \qquad M = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \tag{3}$$

between consecutive images where the time is uniformly sampled at $k = 1$. This model does not contain the target signal z_t and its noise process, which is mainly due to the swelling case assumptions of the target as the formulation here will not require its estimation. Note that, each target has its own speed $[\dot{x}_t \quad \dot{y}_t]'$, which is unknown to the algorithm, in the range of $[\dot{x}_{max}, \dot{y}_{max}]$.

The number of the images within the temporal window is T such that $t=1$ corresponds to the most recent frame and I_T to the earliest. N_r and N_b are the number of cells along the range and beam axes respectively. N_n stands for the total number of needles.

In the streaming (moving temporal window) mode, the needle picking algorithm proceeds by sorting the cells $q_i^1 : (x_i, y_i)^1$ in the new image I_1 according to their intensity values $I_1(q_i^1)$. Note that $I_1(q_i^1) \geq I_1(q_{i+1}^1)$. N_n/T

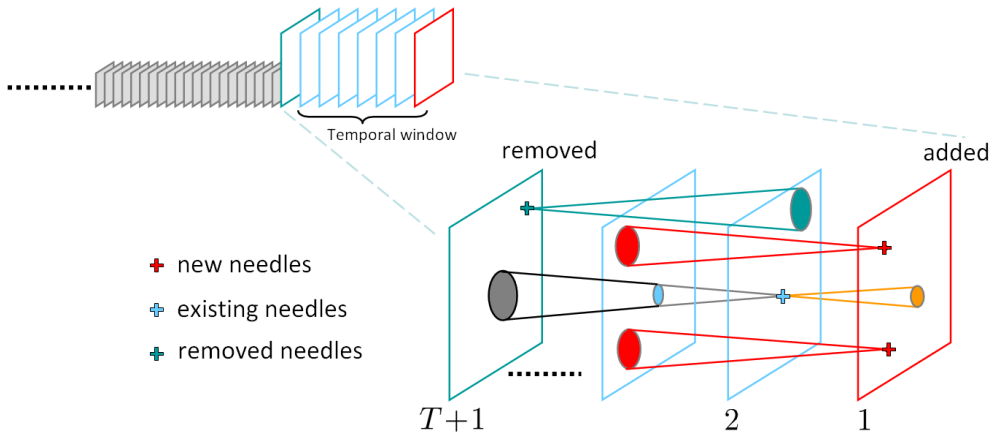

Figure 2. In the streaming mode, for the newly added needles (red dots) the new hypotheses are constructed (red lines). The coverage of these new hypotheses in the most earliest frame is shown in red ovals. The hypotheses (blue lines) of the existing needle (blue dot) are extended to the current image (yellow lines) and the part of the hypotheses now outside the temporal window (black lines) are subtracted. The hypotheses belongs to outside the temporal window (green lines) and the corresponding needle (green dot) are removed.

cells having the highest values are added to the current set of needles $A_1 : \{q_i^1 | i = 1, .., N_n/T\}$. For the newly added cells q_i^1, all possible hypotheses $H_{i,1}^1, ..., H_{i,J}^1$ are generated by the state transition model (Eqn. 3) and its maximum limits. This means that we apply a grid $(g_x, g_y)_{i,u,v,T}$ to the earliest image I_T in the window around the needle q_i^1 coordinates $(x_i, y_i)^1$ to determine the extend of the hypotheses, e.g. their coverages. The grid generates a fixed number J of hypotheses

$$(g_x, g_y)_{i,u,v,T} = \left(x_i - u\frac{\dot{x}_{max}}{U}T, y_i - v\frac{\dot{y}_{max}}{V}T \right) \tag{4}$$

where $-U \leq u \leq U$, $-V \leq v \leq V$ with J is the total number of grid locations that have $(\dot{x}_{max}^2 + \dot{y}_{max}^2)^{0.5}$ distance from the center of the grid. We choose U, V to obtain subpixel resolution. Note that, a hypothesis corresponds to a motion trajectory in the spatiotemporal window passing through the needle it belongs $H_{i,j} : \{q_i^1, .., (g_x, g_y)_{i,u,v,t}, .., (g_x, g_y)_{i,u,v,T}$. We accumulate the likelihood of each hypothesis $L(H_{i,j})$ by traversing backwards in time along the motion trajectory $L(H_{i,j}) = \sum_{t=1}^{T} (g_x, g_y)_{i,u,v,t}$. The accumulated likelihoods are either the underlying cell intensity values or their individual likelihood scores obtained from the likelihood ratio in case the distribution parameters are known. Finally, we append the likelihoods to the queue of likelihoods Q_l of the corresponding cell position in I_1. A queue stores the hypotheses reaching to that cell position and their likelihoods. Since the newly added hypotheses already in the same image, their location $l = q_i^1$ for them. The appending operation is only done for the cell locations at the current image I_1 as the final detection will be evaluated in this image.

For the existing needles q_i^t in the previous sets A_t, $1 < t \leq T$, we compute the forward motion on the corresponding hypotheses, we update the hypotheses and update their likelihoods. Due to this update, the final appending locations l may change, so we reassign such hypotheses to the correct queues. While updating the existing hypotheses, we subtract the contribution of the cells that are no longer in the current temporal window, in other words, the cells in I_{T+1} along the trajectory. The needles of the I_{T+1} are removed, and the hypotheses generated for them are deleted from the queues to keep a constant number of needles. Figure 2 illustrates the streaming mode updates.

After the above accumulation steps are applied, the queue of likelihoods Q_l, which indicate the target existence in I_1 are thresholded to detect the target locations. The threshold is set such that the detection performance achieves the specified false alarm rate.

In the batch mode, all N_n needles for the whole temporal window is selected at the same time either by

sorting all cells and choosing the highest valued ones or sorting cells within each image as above. All hypotheses are generated at the same time and the likelihoods and queues constructed accordingly.

Note that, in general, a target may not exactly coincide to the underlying integer indexed cell locations on the uniformly quantized imaging grid because of the world to imaging plane mapping. For instance, a target at $(4.49, 3.51)$ will appear at the cell $(4, 4)$ due to quantization. Since all hypotheses pass through the integer valued needle locations in the above picking step, there is a chance that some hypotheses to contain a few inaccurate cells along their trajectories. This may become observable for extremely low false alarm rates and very large temporal window sizes. To prevent this, we assign multiple needles at subpixel proximity to the chosen cell instead of assigning a single needle to that cell.

A summary of the needle picking are given in the following steps.

Given $I_1, .., I_T$

Add new needles

- Sort cells $q_i^1 : I_1(q_i^1,) \geq I_1(q_{i+1}^1)$
- Choose needles $A_1 : \{q_i^1 | i = 1, .., N_n/T\}$
- Generate hypotheses $\forall\ q_i^1 \in A_1$, $H_{i,j} : \{q_i^1, .., (g_x, g_y)_{i,u,v,t}, .., (g_x, g_y)_{i,u,v,T}\}$ by Eqn. 4
- Update likelihoods $L(H_{i,j}) = \sum_{t=1}^{T} (g_x, g_y)_{i,u,v,t}$
- Append queues $Q_l \leftarrow \{H_{i,j}, L(H_{i,j})\}$

Update existing hypotheses

- Compute forward motion for $H_{i,j} \in q_{i,t}$, $1 < t \leq T$
- Update likelihoods $L(H_{i,j})$
- Update queues Q_l

Remove old needles

- Remove needles $q_i^{T+1} \in A_{T+1}$
- Remove hypotheses Hi, j of q_i^{T+1}

Update time

- $I_t \leftarrow I_{t-1}$, $A_t \leftarrow A_{t-1}$, $q_i^t \leftarrow q_i^{t-1}$

2.1 Number of Needles

The probability of target existence given the measurement intensity may have different correlation as demonstrated in figure 3. Consider a measurement scenario where cell intensity values I_t are not correlated with the probability of being a target. In other words, the conditional probability function of target existence given the cell intensity is uniformly distributed imposing all intensity values to have the same probability. Since there is no direct relation between the observed cell intensities and target probability, we can only select needles randomly at the current frame, hoping one would hit the target, and generate hypotheses based on those selections. An expected true detection rate for the N images can be derived as

$$P_{td}^T = 1 - \prod_{t=1}^{T} P_{miss}^1 = 1 - \left(1 - \frac{N_n}{N_b N_r}\right)^T \tag{5}$$

where P_{miss}^1 is the probability of miss at single image. For instance, to obtain an expected true detection rate 0.5 for the $N_r = 1000, N_b = 100$ and $T = 10$, we will need to pick $N_n = 6697$ needles at each frame.

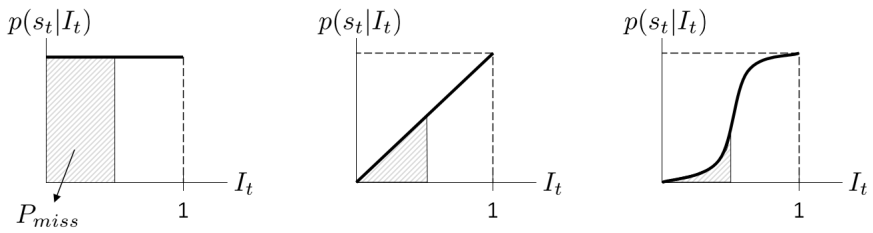

Figure 3. (Left) the target is not correlated with the intensity, (middle) linearly correlated, (right) higher intensity values much more likely to represent a target.

The target signal is additive to the noise and clutter, thus higher the intensity of a cell gets more likely the target becomes at that location. For the scenario of the intensity is linearly correlated with the probability of being target, the probability of miss at a single image is

$$P_{miss}^{1} = \left(1 - \frac{N_n}{N_b N_r}\right)^2 \tag{6}$$

When we choose the highest intensity valued cells, we will need $N_n = 3406$ needles (3% of all cells in the image) at each image to achieve the same P_{td}^{T} above. Yet, the correlation is much stronger in actual systems resembling Heaviside function $0.5+\pi^{-1}\arctan(\beta[I_t - I_\alpha])$ where the equilibrium point I_α is defined by the likelihood functions $L(h_0|I_\alpha) = L(h_1|I_\alpha)$ and $\beta \approx 0.01$. As an example, using the likelihood computation in the following section, we require only $N_n = 70$ needles (0.07% of all cells in the image).

2.2 Likelihood Ratio Computation

Our method does not require signal distributions to be known beforehand and a likelihood ratio to be employed. Still, certain systems assume noise n_t to be a Rayleigh distribution as it corresponds to the Euclidean distance between two orthogonal, normally distributed random variables, and clutter c_t to be a Weibull random variable. Except the SNR of the target signal often the other distribution parameters are not fully specified.

The likelihood ratio is the ratio of the null and target existence hypotheses functions. The null hypothesis h_0 represents the probability of no target exists in a given cell. This can be modeled as the distribution of the sum of Rayleigh and Weibull random variables corresponding to the clutter and channel noise

$$p(h_0) : p(c + n) \tag{7}$$

where we assumed random variables c_t and n_t maintain their distribution properties between the measurements, thus we drop the t index (clutter, or course, may change in time). Given two independent random variables c and n with densities $p(c)$ and $p(n)$, the probability distribution density for the random variable $r = c + n$ is defined as the convolution operator

$$p(r) = p(n) * p(c). \tag{8}$$

However, neither of the noise and clutter distribution parameters are known to the system and we cannot directly apply the convolution operator. It is possible to approximate this summed distribution as we are given relatively large amount of data (1000×100 cells) where possible targets may have only insignificant statistical contribution in it. Hence, an arbitrary shape envelope can be fit to the measurements exploiting the strong law of large numbers by kernel density estimation in terms of Gaussian kernels

$$p(r) = \frac{1}{K\sigma}\sum_{k=1}^{K}\psi\left(\frac{r - r_k}{\sigma}\right), \qquad \psi\left(\frac{r - r_k}{\sigma}\right) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(r-r_k)^2}{2\sigma^2}} \tag{9}$$

where ψ is a zero mean Gaussian kernel, and K is the total number of kernels used. This type of kernel fitting, in general, can model any nonlinear distributions.

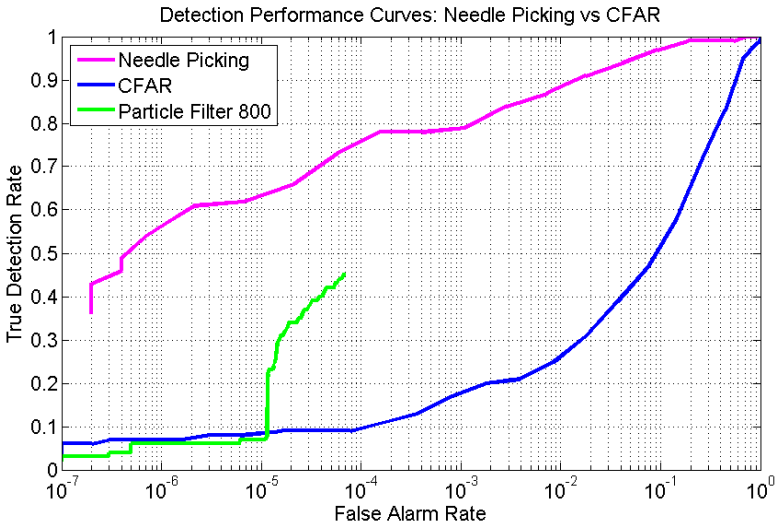

Figure 4. Single target detection curves for SNR + SCR less than 7 dB.

The second hypothesis, h_1, corresponds to the probability of observing a target

$$p(h_1) : p(s + c + n) = p(s + h_0) \tag{10}$$

which corresponds to the distribution of sum of two Rayleigh and one Weibull random variables in case the target random variable is Rayleigh distributed. Once the distribution function of the null hypothesis is obtained, it is numerically convoluted with the target density function where the value of this parameter comes from the working operating point. After both distributions of the null and target hypotheses are estimated, a log likelihood decision rule is applied to compute the likelihood ration for a cell.

3. DETECTION PERFORMANCE

We tested various scenarios to fully investigate the target detection performance of the needle picking algorithm. Note that, detection of the targets also provides target motion trajectories.

To generate the signal we adopted the following settings (none is known by the detection algorithm): target Swerling case is 1 (target response varies from image to image and along the range), both clutter and noise are apparent in the signal, clutter is Weibull distributed, noise is Rayleigh distributed, SNR is 20 dB, and SCR: 7 dB (overall signal to clutter plus noise ratio is much less than 7 dB). Targets move linearly at random velocities that is upper bounded to $[\dot{x}_{max}, \dot{y}_{max}]$, which is set to $[2, 2]$ cell/image. Target direction is randomly assigned.

In our experiments, the image size $[N_r \times N_b]$ to $[1000 \times 100]$. We used $N_s = 100 \sim 500$ different, non-overlapping temporal windows, each containing $T = 10$ images. In other words, the false alarm rate can go below 1 over $N_s \times N_r \times N_b$, i.e. to 10^{-7}. The number of targets is randomly assigned for each temporal window between $1 \sim 5$.

Note that, target speed, target direction, the number of targets, noise and clutter parameters, as well as their shapes are unknown to the needle picking method. We applied the same threshold all sets to obtain objective and practically applicable results. We generated the detection performance curves by changing the threshold value. For comparisons, we implemented an adaptive CFAR technique.

Figure 4 shows the performance graphs of the needle picking, the particle filter,[9] and CFAR when there is a single target in the measurements. As visible, the needle picking achieves 55% detection rate at 10^{-6} false alarm rate while the adaptive CFAR can only provide 8%. The particle filter has many parameters to fine tune. Its performance deteriorates for the low false alarm rates by constructing false trajectories in very low SNR measurements. Figure 5 gives the performance graphs of both algorithms for the multiple targets in the

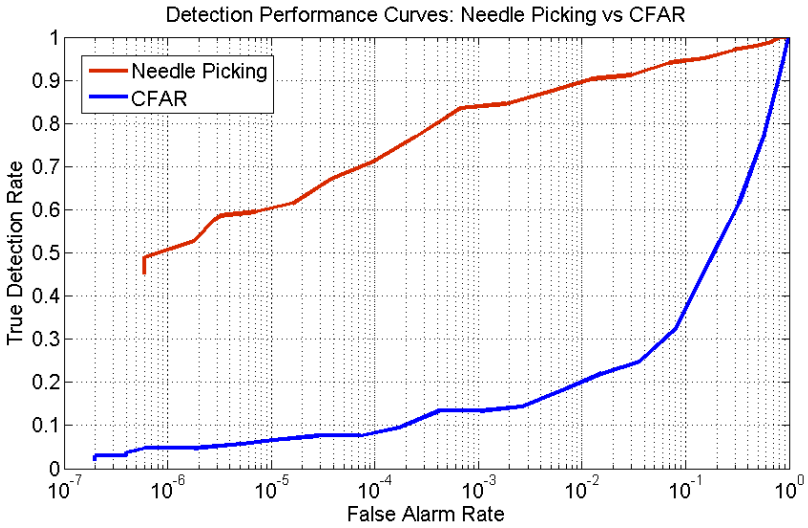

Figure 5. Multiple target detection curves for SNR + SCR less than 7 dB.

measurements. As visible, the needle picking achieves 51.5% detection rate at 10^{-6} false alarm rate while the adaptive CFAR 5%.

We also observed that the performance may be higher for vertical (on range) and horizontal (on beam) target motions. Since we accumulate the hypotheses along the candidate trajectories, our candidate hypotheses do not change the trajectory points when the motion is vertical or horizontal regardless of the speed. However, for random directions, the chosen points may be slightly different than the round operator response (using bilinear interpolation this issue can be resolved). We use a quantized grid in to accumulate likelihoods, there is a chance that we may choose the hypotheses one or two points off from the actual trajectory, which may reflect on the performance curve. For random direction and speed, the hypothesis space covers all possible motions, thus, the grid resolution may be increased.

3.1 Computational Load

The Bayesian estimator is recursively defines the probability of the target occupying a particular location by the superposition of all of the possible paths to that position. The complexity of constructing the values for this space of motion trajectories is $O((N_r N_b)^T)$ for the brute force application and $O(N_r N_b K^T)$ for sparse solution, where M is the bins of the quantized grid $K \ll N_r N_b$ without considering the cost of finding the maxima in this space. Besides, imposing linear or other motion models is difficult in Bayesian estimator, and the algorithm may not fit into memory either. The dynamic programming is a batch processor that finds the most likely sequence of states by forward propagation through states followed by backward linkage. The cost is approximately $O(N_r N_b T M)$.

On the other hand, the needle picking requires $O(N_n T^2 K)$ computations.The load of the PMHT depends on the number of alive tracks (thousands), which is comparable to the needle picking. For $K = 25, N_n = 100$, Bayesian estimator involves $25^{10} \times 10^5$, the dynamic programming 500×10^5, and the needle picking 2.5×10^5 operations. In addition both Bayesian and dynamic programming are recursive, however to our advantage, the needle picking is parallelizable, thus, it enables GPU implementations. The algorithm runs approximately at 220 windows/second on a NVIDIA ATX using CUDA.

ACKNOWLEDGMENTS

We would like to thank Jun Nakamura, Yu Okada, and Yoshihisa Hara for their valuable feedback.

REFERENCES

[1] Jin, Y. and Friedlander, B., "A cfar adaptive subspace detector for second-order gaussian signals," *IEEE Transactions on Signal Processing* (2005).

[2] Tom, A. and Viswanathan, R., "Switched order statistics cfar test for target detection," *In Proceedings of IEEE Radar Conference* (2008).

[3] Davey, S., Rutten, M., and Cheung, B., "A comparison of detection performance for several track-before-detect algorithms," *EURASIP Journal on Advances in Signal Processing* (2008).

[4] Bruno, M., "Bayesian methods for multi aspect target tracking in image sequences," *IEEE Transactions on Signal Processing* (2004).

[5] Tonbsen, S. and Bar-Shalom, Y., "Maximum likelihood track-before-detect with fluctuating target amplitude," *IEEE Transactions on Aerospace and Electronic Systems* (1998).

[6] Barniv, Y., [*Dynamic programming algorithm for detecting dim moving targets*], Y. Bar-Shalom, Artech House, Norwood, Mass, USA (April 1990).

[7] Grossi, E. and Lops, M., "Sequential detection of markov targets with trajectory estimation," *IEEE Transactions on Information Theory* (2008).

[8] Driessen, H. and Boers, Y., "An efficient particle filter for nonlinear jump markov systems," *In Proceedings of IEE Seminar on Target Tracking: Algorithms and Applications, Sussex, UK* (2004).

[9] Rutten, M. G., Gordon, N., and Maskell, S., "Recursive track-before-detect with target amplitude fluctuations," *In Proceedings of IEE Radar, Sonar and Navigation* (2005).

[10] Hempel, C. and Pacheco, J., "Performance analysis of the probabilistic multi-hypothesis tracking algorithm on the seabar data sets," *In Proceedings of 12th International Conference on Information Fusion, FUSION'09* (2009).

[11] Bar-Shalom, Y., Daum, F., and Huang, J., "Probabilistic data association filter," *IEEE Control System Magazine* (2009).

[12] Liu, H., He, Z., and Zeng, J., "An improved radar detection algorithm based on hough transform," *International Journal of Sensing and Imaging* (2009).