

Evaluation of Multistatic Tree-Search Based Tracking on the SEABAR Dataset

Hossein Roufarshbaf and Jill K. Nelson*

Department of Electrical and Computer Engineering
George Mason University
Fairfax, VA, U.S.A.
hroufars,jnelson@gmu.edu

Abstract – *The focus of this paper is the extension of tree-search based tracking to multistatic tracking problems and the evaluation of the proposed algorithm on the SEABAR’07 sonar dataset. The tree-search based tracker, originally introduced in [1], is built upon the stack algorithm for convolutional decoding. To perform track estimation, the tracker navigates a search tree in which each path represents a sequence of states visited by the target. By exploring only a subset of the search tree, the stack-based tracker computes only likely regions of the posterior distribution at each update, thereby approximating the Bayesian inference solution to the tracking problem. In this work, the monostatic stack-based tracker is extended to multistatic tracking. The structure of the tree-search approach facilitates the incorporation of information from multiple source-receiver pairs with minimal complexity increase. The performance of the multistatic stack-based tracker on the SEABAR’07 dataset shows that the tracker is able to maintain track through highly nonlinear target maneuvers and in the presence of heavy clutter.*

Keywords: multistatic active sonar, target tracking, Bayesian inference, tree search

1 Introduction

The challenge of tracking an object (target) moving through a region of interest presents itself in a wide variety of applications, including radar, sonar, and video surveillance. The aim of the tracker is to estimate the location (or full state) of the object of interest based on a sequence of scans or images. In this paper, we present and evaluate an approach to tracking that is built upon tree-search algorithms designed for solving sequential detection problems. While the tree search-based tracker we describe can be applied to tracking problems in any field, we focus here on its application to sonar target tracking, and specifically multistatic sonar scenarios.

At the most basic level, the target tracking problem can be defined as the challenge of estimating the state of a target based on noisy observations of some function(s) of its state variables. Typical state variables include target location and velocity, though others may arise as a function of application. In sonar tracking, our goal is to estimate target location based on noisy observations of target bearing and range.

When both the target motion and state measurement models are linear, and the process and observation noise are Gaussian, the optimal tracking solution can be obtained through the well-known and efficient Kalman filter [2]. The Kalman filter’s low computational complexity makes it particularly attractive in many tracking applications, even when the linear Gaussian model assumptions do not hold, e.g. for maneuvering targets or nonlinear state measurements. Kalman filter variants such as the Extended Kalman Filter (EKF), which operates on linearized motion and measurement models, have been developed to address these common nonlinear cases. While these approaches maintain low complexity and may perform satisfactorily for mildly nonlinear systems, their performance suffers dramatically for severely nonlinear systems, particularly when heavy clutter is present [1, 3].

An alternative solution to the tracking problem is via the Bayesian inference approach, which provides the optimal solution with respect to the maximum a posteriori (MAP) criterion. The tree-search based tracking algorithm employed in this work can be viewed as a low-complexity approximation to Bayesian inference. In the Bayesian inference approach, the posterior distribution of the target state conditioned on observation data is computed each time a new set of observations is obtained [4]. Letting \mathbf{x} denote the target state and \mathbf{y} denote the observations, the MAP state estimate can be determined from the Bayesian inference solution by maximizing over the state, e.g. $\hat{\mathbf{x}}_{\text{MAP}} = \text{argmax}_{\mathbf{x}} \mathbf{P}(\mathbf{x}|\mathbf{y})$. While the Bayesian inference approach yields strong tracking performance, its com-

*This research was funded by a grant from the United States Office of Naval Research Code 321US.

plexity may be prohibitively high. For most motion and measurement models, no closed-form expression for the posterior distribution exists, and hence the posterior must be computed numerically and the continuous function approximated by evaluation on a grid.

The tree-search based tracker provides an alternative that elegantly incorporates nonlinear and non-Gaussian motion and measurement models while maintaining reasonable computational complexity. It is built upon the stack algorithm, which guides a depth-first search through a tree. For tracking, each node of the tree represents a possible target state at a given time, or scan. Hence, each possible sequence of states visited by the target is mapped to a path through the search tree. At each iteration of the algorithm, the path that appears most likely is extended to all possible next states. The stack-based tracker serves as an approximate version of Bayesian inference in that, by exploring only a subset of the paths through the tree, the tracker is computing only certain regions of the posterior distribution in each stage. Calculation of the posterior only for likely state values can dramatically reduce complexity.

Multistatic tracking has become a topic of recent interest in the active sonar community. In a multistatic scenario, multiple geographically distributed receivers are employed to observe the echo from a single source, and hence measurements are obtained from multiple source-receiver pairs. (The multistatic scenario may be extended to include multiple sources, as well, but we consider only a single source in this work.) The presence of multiple receivers provides the benefit of additional information for tracking, but it also presents the challenge of efficiently fusing multiple data sources. As will be shown, the stack-based tracker is particularly attractive for multistatic tracking, as its path metric can be adapted to exploit information from multiple sources with little to no increase in algorithm complexity.

The focus of this paper is the augmentation of tree-search based tracking to address multistatic problems and the evaluation of the resulting tracker on real sonar data. The remainder of the paper is organized as follows. In Section 2, we describe the multistatic system model under consideration. In Section 3, we briefly describe tree-search algorithms and present the stack-based approach to target tracking. We derive a metric proportional to the posterior density function and a state space discretization approach for the application of stack-based algorithms to multistatic tracking. Evaluation of the stack-based tracker on data from the SEABAR’07 experiment is presented in Section 4, and conclusions are drawn in Section 5.

2 System Model

Consider a single source and R receivers present in the region of interest. We assume that the source and receiver locations are known (at least approximately) to

the tracker. Let the vector \mathbf{x}_k denote the state of the target at time (or scan) index k . In the model under consideration, each scan produces a set of contacts at each of the R receivers. (A receiver may not be functioning at a particular scan, in which case the associated set of contacts will be empty.) One could assume an equivalent model in which the contacts at each receiver are viewed as different scans, and hence one ping from the source results in R scans of data. We assume a centralized fusion model in which contacts from all receivers are transmitted to a fusion center for tracking. Development of distributed approaches to tree-search based tracking will be addressed in future work.

Let the contacts observed by receiver r at time k be denoted by the length- $m_k^{(r)}$ vector $\mathbf{y}_k^{(r)}$, where $m_k^{(r)}$ denotes the number of contacts observed by receiver r in scan k . For purposes of deriving data association probabilities, $m_k^{(r)}$ is assumed to follow a Poisson distribution. A full contact vector for each scan is formed as $\mathbf{Y}_k = [\mathbf{y}_k^{(1)} \ \mathbf{y}_k^{(2)} \ \cdots \ \mathbf{y}_k^{(R)}]$. Target motion and measurement models are given by

$$\mathbf{x}_k = \mathcal{F}_k(\mathbf{x}_{k-1}) + \mathbf{G}\mathbf{v}_k \quad (1)$$

$$\mathbf{Y}_k = \mathcal{H}(\mathbf{x}_k) + \mathbf{w}_k, \quad (2)$$

where \mathcal{F}_k denotes the state transition function, \mathbf{v}_k denotes the state transition noise, and \mathbf{G} controls which state variables are perturbed by state transition noise. \mathcal{H} denotes the function defining the transformation from the target state to the observed quantity, and \mathbf{w}_k denotes the observation noise. \mathcal{H} is assumed to be known to the tracker, and the observation noise \mathbf{w}_k is assumed to be independent of the state transition noise \mathbf{v}_k . While the stack-based tracker we consider assumes knowledge of \mathcal{F}_k and \mathbf{G} , a model of the target motion will not be known to the tracker in most practical scenarios; in fact, such a model may not exist for targets whose motion involves maneuvering, etc. Hence, for our work, we incorporate a basic “non-informative” motion model in the tracker. Evaluation results indicate that performance is strong even without true motion model knowledge.

3 Multistatic Stack Tracker

3.1 Tracking via Tree Search

The inspiration for a tree-search approach to target tracking is a set of tree-search algorithms designed for decoding convolutional codes in communications. Perhaps the most well-known of these is the stack algorithm [5], and it forms the basis for the tree-search based tracker. In communications applications, each possible transmitted data sequence is mapped to a path through a tree, and the stack algorithm searches the tree to identify the most likely path. In the stack algorithm, exploration of the search tree is governed at each stage by which branches appear most likely. Each

path within the tree has an associated likelihood metric. Paths that have been explored thus far are stored in a list, or stack, along with their metrics. At each stage of the stack algorithm, the path with the largest metric is extended to its children. The metrics of the extended paths are calculated, and these paths are placed in the stack. The process then repeats with extension of the most likely path in the newly-expanded stack. The algorithm terminates when the most likely path in the stack reaches a leaf of the tree, i.e., when a full-length path has the largest metric of any path in the stack.

In order to apply the stack algorithm to target tracking, each possible sequence of states visited by the target is mapped to a path through a search tree. The tracking problem is then framed as a generalized sequential detection problem in which the goal is to identify the most likely path through the tree, or equivalently the most likely sequence of target states. (Mapping target states to a search tree requires that the state space be discretized, as discussed in Section 3.3.) The metric associated with each sequence of target states is drawn from the posterior likelihood of the state sequence conditioned on observed data; its derivation is described in more detail in Section 3.2.

The stack-based tracker serves as an approximation to Bayesian inference tracking in that, like the Bayesian approach, it evaluates the posterior distribution of the target state conditioned on observations; however, it does so only for a subset of all possible state values. If the metric for the stack-based tracker is drawn from the posterior distribution of a single state, then the metric of each length- k path in the stack can be viewed as one point in the probability mass function (PMF) $\mathbf{P}(\mathbf{x}_k|\mathbf{Y}_1^k)$, where $\mathbf{P}(\mathbf{x}_k|\mathbf{Y}_1^k)$ is a discretized approximation to the continuous state posterior, and $\mathbf{Y}_1^k = [\mathbf{Y}_1 \mathbf{Y}_2 \cdots \mathbf{Y}_k]$. Since the stack algorithm explores only a small subset of the full search tree, there are many state values for which the posterior PMF is not evaluated. For any scan index k , the posterior PMF is evaluated only for states that appear likely based on contacts in that scan. As with the conventional stack algorithm, the stack-based tracker is not guaranteed to provide the optimal solution, since it is possible that the most likely path through the tree is never explored [6]. Despite sub-optimality, however, strong performance has been observed, and the complexity reduction achieved by computing only a subset of the points in the posterior PMF is critically important for many practical applications.

3.2 Multistatic Metric Computation

As in the original stack algorithm, the path metric for the stack-based tracker is derived from the posterior distribution on the state sequence at scan k given all observations through scan k . While the Bayesian inference approach considers the posterior for a single scan index, we exploit the ability of the tree-search

structure to retain state history information and consider a joint state posterior of the form $\mathbf{P}(\mathbf{x}_1^k|\mathbf{Y}_1^k) \propto \mathbf{P}(\mathbf{Y}_1^k|\mathbf{x}_1^k)\mathbf{P}(\mathbf{x}_1^k)$, where $\mathbf{x}_1^k = [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_k]$.

Note that, because the single most likely path in the stack is extended at each iteration of the stack algorithm, the paths stored in the stack at any given time will be of varying lengths. While $\mathbf{P}(\mathbf{Y}_1^k|\mathbf{x}_1^k)\mathbf{P}(\mathbf{x}_1^k)$ provides a measure of the posterior likelihood of any state sequence, it decreases as path length (k) increases and hence favors shorter paths. With such a metric, the stack algorithm will tend to explore much of the breadth of the tree before extending the depth of likely paths, which results in increased complexity. The conventional stack algorithm incorporates a length-based bias term of the form $\mathbf{P}(\mathbf{x}_{k+1}^K)$ (where K denotes the depth of the tree) in order to facilitate fair comparison across paths, but a similar bias term cannot be readily computed for the tracking problem. Instead, we use a windowing technique to normalize the tracker metric across path lengths. Specifically, we apply a forgetting factor $\lambda < 1$ that imposes an exponential window on the metric, giving the largest weight to the states most recently visited.

In the log domain, the exponentially windowed path metric takes the form

$$\gamma(\mathbf{x}_1^k) = \frac{1-\lambda}{1-\lambda^k} \sum_{l=1}^k \lambda^{k-l} \times \log \left[\mathbf{P}(\mathbf{x}_l|\mathbf{x}_{l-1}) \sum_{j=0}^{m_l} \mathbf{P}(\mathbf{Y}_l|\mathbf{x}_l) \right], \quad (3)$$

where λ^{k-l} weights the posterior state likelihood for scans $l = 1$ through k . The normalization term $\frac{1-\lambda}{1-\lambda^k}$ serves to magnify the metric of paths that are shorter than the memory imposed by the exponential window. The state transition PMF on the discretized state space, $\mathbf{P}(\mathbf{x}_l|\mathbf{x}_{l-1})$, can be derived from the target motion model. A detailed description of the calculation of the path metric for monostatic tracking, as well as the data association approach applied, can be found in [7].

Calculation of the path metric requires evaluation of the observation likelihood conditioned on the target state $\mathbf{P}(\mathbf{Y}_l|\mathbf{x}_l)$ for $l = 1$ to k . When multistatic geometries are employed, calculation of $\mathbf{P}(\mathbf{Y}_l|\mathbf{x}_l)$ must be modified to reflect the fusion of data streams obtained from multiple source-receiver pairs. In this work, we adopt a simple approach to computing the multistatic observation likelihood that retains the same tree size (and hence the same algorithm complexity) as the monostatic tracker. Under the assumption that observations at any single receiver are conditionally independent of observations at all other receivers, we compute the multistatic observation likelihood as

$$\mathbf{P}(\mathbf{Y}_l|\mathbf{x}_l) = \prod_{r=1}^R \mathbf{P}(\mathbf{Y}_l^{(r)}|\mathbf{x}_l), \quad (4)$$

where $\mathbf{P}(\mathbf{Y}_l^{(r)}|\mathbf{x}_l)$ can be evaluated by expanding over all data association hypotheses [7]. In fact, this is a simplifying assumption that generally does not hold true in practice. While conditioning on the target state eliminates dependence among contacts generated by the target (since the only remaining random element is observation noise), contacts that are observed at different receivers but generated by the same clutter source will certainly be correlated. Hence, the chosen implementation is a simple first approximation to a more accurate approach to observation fusion. Performance evaluations thus far indicate that the tracker works well even with this simplifying assumption. Future work will explore the incorporation of clutter measurement dependence in likelihood fusion.

3.3 State Space Discretization

In order to map the target state space onto a tree, each state variable must be discretized to a finite number of possible values. State space discretization error necessarily introduces quantization noise, and while quantization noise can be reduced through finer discretization, the tree size (and hence computational complexity) increases as a result. The number of children at each node of the search tree is determined by the product of the number of quantization levels for each state variable, and hence coarse quantization is desirable from a complexity standpoint. The stack-based tracker employs a sampling-based approach to minimize quantization error while reducing the number of quantization levels required. Rather than choosing a fixed value to represent each point in the PMF, each quantization region of the state space is sampled on a finer scale, and the most likely sample is chosen to represent the region.

The state transition noise \mathbf{v}_k is quantized in our implementation, which in turn imposes quantization of position and velocity, the state variables. For each region of the state space, the likelihood of \mathbf{v}_k conditioned on the observations and previous state is computed on a fine grid. For the multistatic tracking problem, this likelihood takes the form

$$\begin{aligned} \mathbf{P}(\mathbf{v}_k^i|\mathbf{Y}_k, \mathbf{x}_{k-1}) &= \frac{\mathbf{P}(\mathbf{Y}_k|\mathbf{v}_k^i, \mathbf{x}_{k-1}) \mathbf{P}(\mathbf{v}_k^i)}{\mathbf{P}(\mathbf{Y}_k|\mathbf{x}_{k-1})} \quad (5) \\ &\propto \prod_{r=1}^R \mathbf{P}(\mathbf{Y}_k^{(r)}|\mathbf{v}_k^i, \mathbf{x}_{k-1}) \mathbf{P}(\mathbf{v}_k^i). \end{aligned}$$

Again, we have assumed that, conditioned on the state, contacts observed at any receiver are independent of those observed at other receivers. By choosing the most likely point in a region to represent the associated discretized point in the state space, quantization noise is reduced while the tree size remains constant.

4 Performance Evaluation

The performance of the stack-based tracker for multistatic sonar has been evaluated on runs A01 and A56 of

the SEABAR'07 experiment. The SEABAR'07 experiment took place in the Mediterranean Sea in October 2007 [8]. The experiment included a single source and three receivers. Both CW and FM pulses were transmitted by the source at one minute intervals. All three receivers were active during part, if not all, of run A01, and receivers 2 and 3 were active during run A56. The target was simulated by a towed echo repeater. In order to provide contact data that better model a true target, the SNR levels of the target contacts were modified to reflect aspect-dependent target strength [9]. The results presented here use contact data modified according to the BASIS target strength model [10].

The source and receiver locations, as well as the true target trajectory, were provided in terms of latitude and longitude. Conversion from time delay to contact range was performed as described in [11]. Source, receiver, and target locations were converted to a 2-D Cartesian coordinate system using the Haversine formula, and the origin of the coordinate system was placed at the source. The source and receiver locations for the SEABAR'07 experiment, along with the target trajectories for runs A01 and A56, are shown in Figure 1.

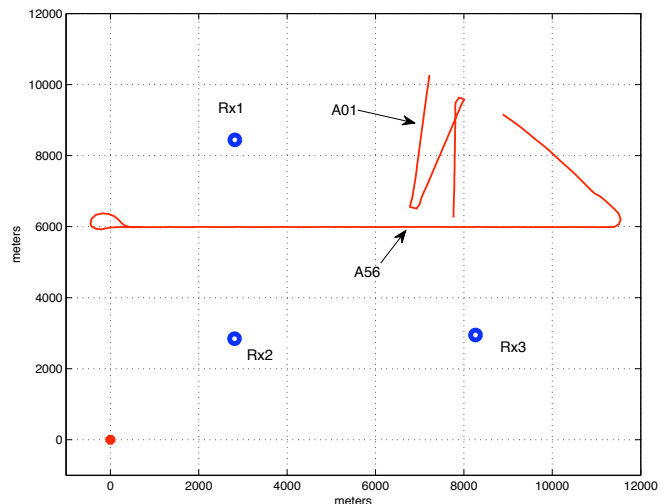


Figure 1: Geometry of the SEABAR'07 multistatic sonar experiment. The source, shown in red, is at (0,0). The three receivers are shown as blue circles, and the target trajectories for runs A01 and A56 are shown in red.

The stack-based tracker employs the Hough transform for track initiation [12]. The surveillance region is divided into 1-km² cells, each of which is evaluated for the presence of a target using peaks in the Hough bins. The common M -of- N approach is employed. The track initiation algorithm is repeated every 10 scans to detect newly arriving targets. The validity of each track

is evaluated at each scan via the stack metric. During tracking, if the metric of an initiated target falls more than 5 dB below the highest metric of the trajectories in track, the target is marked as invalid and is terminated by the algorithm. The value of M/N used for track initiation depends upon the clutter density of the experiment. The values used for runs A01 and A56 are given in Table 1.

The target state vector is given by $\mathbf{x}_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T$, where (x_k, y_k) denotes the position of the target in Cartesian coordinates, and (\dot{x}_k, \dot{y}_k) denotes the target velocity in Cartesian coordinates. Computation of the Bayesian path metric requires that motion and measurement models be provided. Of course, no model is available to describe the motion of the targets in the SEABAR experiment; a simple nearly-constant velocity (NCV) linear motion model [2] is assumed as a default. While extension to interacting multiple models [13] or other more sophisticated models may provide performance improvement, the stack-based search is able to hold track under the NCV model. The assumed motion model is given by

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \mathbf{G}\mathbf{v}_k, \quad (6)$$

where

$$F = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} \frac{\Delta_t^2}{2} & 0 \\ 0 & \frac{\Delta_t^2}{2} \\ \Delta_t & 0 \\ 0 & \Delta_t \end{bmatrix}. \quad (7)$$

The time between scans, Δ_t is equal to 1 minute. Bearing and range form the observation function, given by

$$\mathcal{H}(\mathbf{x}_k) = \begin{bmatrix} \tan^{-1}(y_k/x_k) \\ \sqrt{x_k^2 + y_k^2} \end{bmatrix}. \quad (8)$$

The covariance matrices of \mathbf{v}_k and \mathbf{w}_k , denoted by Q and R , respectively, are given in Table 1. An SNR threshold of 12 dB is used to determine which contacts are provided to the tracker.

For evaluation of the stack-based multistatic tracker, we first consider run A01 of the SEABAR'07 experiment. This run is particularly challenging, as the target engages in both a hairpin turn and a Crazy Ivan maneuver. The target begins near (7000, 10000) and travels first through the hairpin turn followed by the Crazy Ivan. Figure 2 shows the performance of the stack-based tracker using only FM data from all three receivers. As Figure 2 shows, the stack-based tracker is able to maintain track throughout the target trajectory, including through the hairpin and Crazy Ivan turns. The tracker holds the hairpin turn particularly tightly, showing little to no deviation from the highly nonlinear maneuver. The initiation algorithm generates four false tracks at scan indices 21, 31, 41, and 61. The first and the third false tracks are terminated after

Table 1: Stack-based tracker parameter values used for evaluation on run A01 of the SEABAR experiment.

Contact SNR Threshold	12 dB
\mathbf{P}_D (Prob. of Detection)	0.75
M/N , run A01	3/5
M/N , run A56	2/5
R (covariance of \mathbf{w}_k), FM	$\begin{bmatrix} 0.0019 & 0 \\ 0 & 2500 \end{bmatrix}$
R (covariance of \mathbf{w}_k), CW	$\begin{bmatrix} 0.0019 & 0 \\ 0 & 62500 \end{bmatrix}$
Q (covariance of \mathbf{v}_k)	$\begin{bmatrix} 0.0005 & 0 \\ 0 & 0.0005 \end{bmatrix}$
λ (Forgetting factor)	0.95
Stack Size	256
Discretization Regions	9
Samples Per Region	81

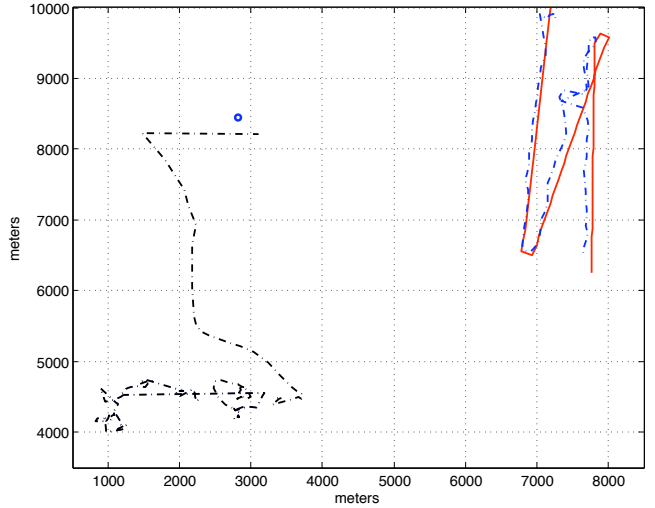


Figure 2: Tracking performance of the stack-based tracker on run A01 using FM contacts from all three receivers. The target trajectory is shown in red, the track estimate is shown in blue, and false tracks are shown in black.

34 and 15 scans, while the second and fourth persist to the end of the experiment.

Figure 3 shows the performance of the stack-based tracker on run A01 using only CW data from all three receivers. Only bearing and range information are used by the tracker. Augmenting the stack-based tracker to exploit Doppler information is a topic of ongoing work. Like the FM case, the tracker follows the tar-

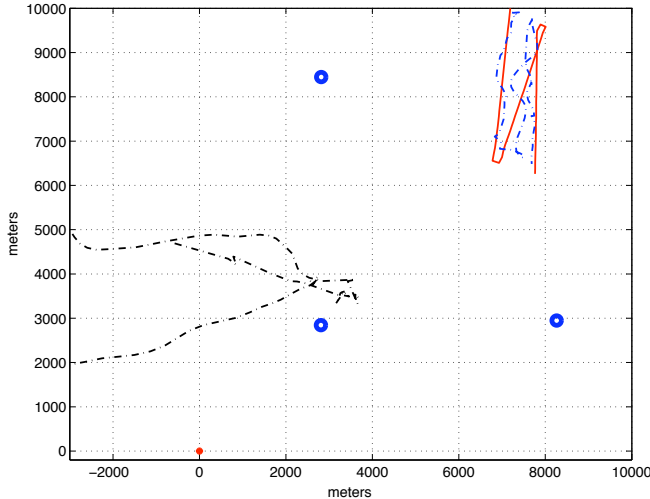


Figure 3: Tracking performance of the stack-based tracker on run A01 using CW contacts from all three receivers. The target trajectory is shown in red, the track estimate is shown in blue, and false tracks are shown in black.

get trajectory throughout its maneuvers. The tracker does not follow the trajectory as closely in this case as it did when FM data was used. We hypothesize that the incorporation of Doppler information will improve the precision of the track estimate. In this experiment, three false tracks have been generated: one at the scan one and two at scan 21. The first false track is killed after 20 scans; one of the remaining false tracks lasts for 59 scans, and the other persists to the end of the experiment. Only parts of the false track are shown in Figure 3.

The performance of the stack-based tracker on run A01 when both FM and CW data are provided is shown in Figure 4. For simplicity, only CW data is used in the Hough transform for track initiation, allowing bin size and thresholds to remain the same as those used when only CW or FM data alone is used for tracking. The tracker maintains track throughout the trajectory, and performance is slightly better than FM alone. Since we have used only CW data for track initiation, the same false tracks appear here and in Figure 3. Using both FM and CW information, all false tracks are killed within 9 scans. When both data types are used, the multistatic likelihood is computed by treating each receiver as two data streams, one for each pulse type. Again, assumptions of independence across these contacts do not necessarily hold in practice, and methods for addressing correlations among contacts need to be further explored.

To evaluate the performance of the stack-based tracker when only a single stream of contact data is available, we consider the scenario in which, for run

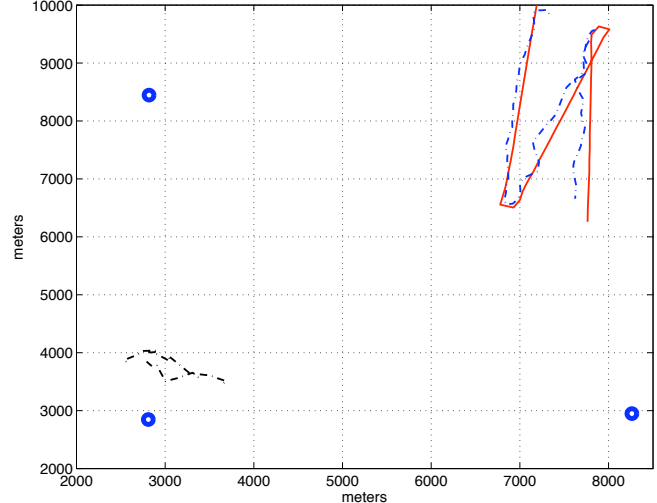


Figure 4: Tracking performance of the stack-based tracker on run A01 using both FM and CW contacts from all three receivers. The target trajectory is shown in red, the track estimate is shown in blue, and false tracks are shown in black. Only CW contacts are used for track initiation.

A01, only CW data from Receiver 2 is provided. The results are shown, along with Receiver 2 CW contacts, in Figure 5. Even when only one receiver is used, the stack-based tracker succeeds in following the target through both maneuvers, though there is some lag in completing the hairpin turn. Note that Receiver 2 does not observe contacts in the region of the false track seen in Figures 3 and 4, and hence no false tracks are initiated when only Receiver 2 data is used.

The performance of the stack-based tracker has also been evaluated on run A56 of the SEABAR'07 dataset. The target trajectory for run A56 begins to the left near (600, 6000) with a loop and then travels straight for approximately 12 km before making a final turn. Figure 6 shows the performance of the stack-based tracker on run A56 when CW contacts are used for track initiation and both CW and FM contacts are used for tracking. Receivers 2 and 3, which were active for part or all of the run, are shown along with the target trajectory, the track estimate, and two false tracks. The stack-based tracker maintains track throughout the run, including the loop at the beginning and the turn near the end. Initiation with CW contacts results in four false tracks. Two are initiated at the first scan and last for 6 and 14 scans, respectively. Another false track is initiated at scan 51 and persists through 39 scans. At time index 91, a final false track is initiated and lasts 10 scans.

Finally, the performance of the stack-based tracker on run A56 when only CW data is provided for initiation and only FM data is provided for tracking is shown in Figure 7. The tracking performance using

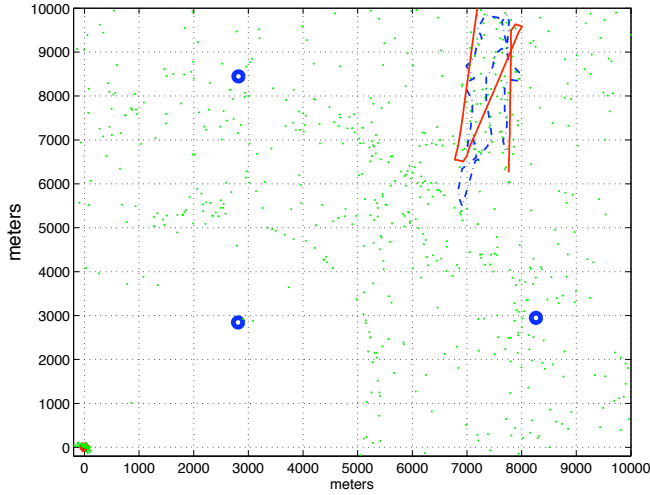


Figure 5: Tracking performance of the stack-based tracker on run A01 using only CW contacts observed at Receiver 2. The target trajectory is shown in red, and the track estimate is shown in blue. Contacts are shown in green.

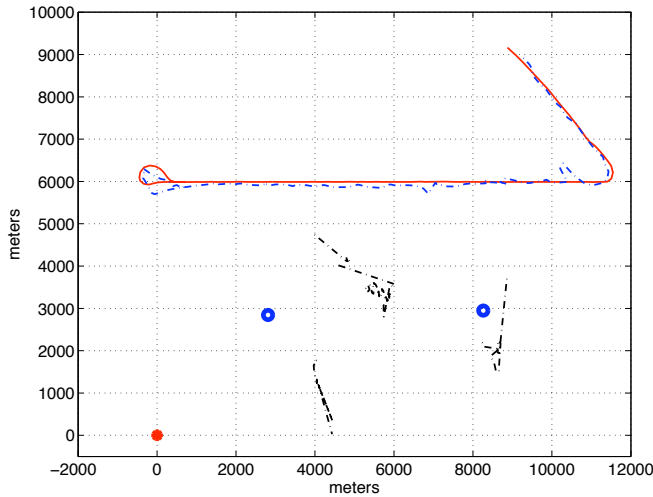


Figure 6: Tracking performance of the stack-based tracker on run A56. The target trajectory is shown in red, the track estimate is shown in blue, and the false tracks are shown in black. Both CW and FM contacts are used for tracking. Only CW contacts are used for track initiation.

only FM data is very similar to that using both FM and CW data. In both cases, the target track is maintained throughout the run. Similar to Figure 6, four false tracks are identified; they are terminated in 8, 20, 12, and 23 scans, respectively. The observation that tracking based on FM data alone can perform as well as tracking based on both FM and CW data is a useful

one, as considering only a subset of the total available contacts allows a slight reduction in algorithm complexity. We hypothesize, however, that incorporation of Doppler information in the stack-based tracker may significantly increase the performance benefit of CW data. Future work will address this issue.

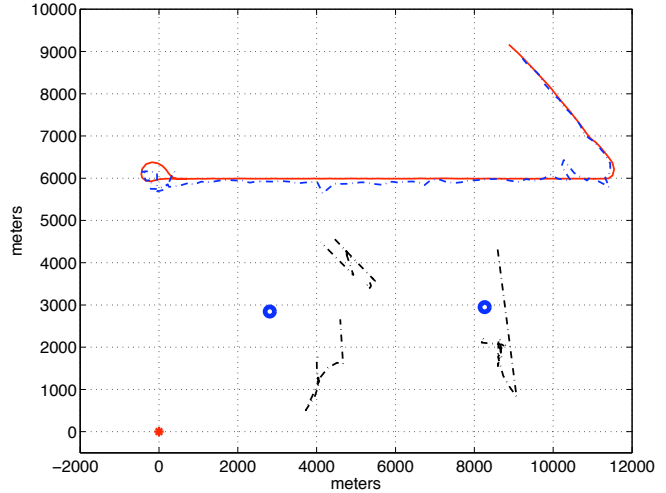


Figure 7: Tracking performance of the stack-based tracker on run A56. The target trajectory is shown in red, the track estimate is shown in blue, and the false tracks are shown in black. Only FM contacts are used for tracking, and only CW contacts are used for track initiation.

5 Conclusions

A tree-search based approach to target tracking in clutter has been extended to address multistatic sonar tracking, and its performance has been evaluated on data obtained from the SEABAR'07 experiment. Performance results show that the stack-based tracker is able to follow the target trajectory closely and to maintain track through highly nonlinear maneuvers. In fact, the stack-based tracker is able to maintain track throughout run A01 even when data from only a single receiver is available. The proposed track management approach, which includes track initiation via the Hough transform and track termination via metric comparison, has been shown to be effective, as well, producing at most four false tracks for runs A01 and A56.

Based on these results, the stack-based tracking algorithm shows promise for multistatic tracking and warrants further development and evaluation. In addition to further evaluation on real and simulated datasets, avenues for future work include incorporation of Doppler information in the tree-search path metric, development of more accurate methods for fusion of data from multiple source-receiver pairs, improvement of the track

termination algorithm, and exploration of distributed algorithms for stack-based tracking.

References

- [1] J. K. Nelson and H. Roufarshbaf, "A tree search approach to target tracking in clutter," in *Proceedings of the 12th International Conference on Information Fusion*, July 2009, pp. 834–841.
- [2] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley and Sons, New York, 2001.
- [3] M. S. Schlosser and K. Kroschel, "Limits in tracking with extended Kalman filters," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1351–1359, 2004.
- [4] L. Stone, T. Corwin, and C. Barlow, *Bayesian Multiple Target Tracking*, Artech House, Norwood, MA, 2002.
- [5] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, New York, NY, 1999.
- [6] G. D. Forney, "Convolutional codes III: Sequential decoding," *Information and Control*, vol. 25, no. 3, pp. 267–97, July 1974.
- [7] H. Roufarshbaf and J. K. Nelson, "Target tracking via a sampling stack-based approach," in *Forty-Third Asilomar Conference on Signals, Systems and Computers*, November 2009, pp. 1327–1331.
- [8] F. Ehlers, "SEABAR'07 Cruise Report," 2008, NATO Undersea Research Center, La Spezia, Italy.
- [9] D. Grimmett, "Specular-cued multistatic sonar tracking on the SEABAR'07 dataset," in *Proceedings of the 12th International Conference on Information Fusion (FUSION)*, July 2009, pp. 1576–1583.
- [10] R. F. Gragg, "The BASIS-3D acoustic target strength model," Tech. Rep. NRL/FR/7140-07-101052, Naval Research Laboratory, Washington, D.C., 2002.
- [11] S. Coraluppi, "Multistatic sonar localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 964–974, October 2006.
- [12] B. D. Carlson, E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough transform. I. System concept," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 102–108, January 1994.
- [13] H. A. P. Blom and Y. Bar-Shalom, "Multisensor tracking of a maneuvering target in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, pp. 176–189, March 1989.