A NEW PRUNING/MERGING ALGORITHM FOR MHT MULTITARGET TRACKING

Kevin Buckley Anand Vaddiraju Richard Perry

Department of Electrical and Computer Engineering
Villanova University
Villanova, PA 19085
E-mail: buckley@ece.vill.edu

ABSTRACT

In this paper we develop a new general Viterbi algorithm for multitarget tracking. A standard Multiple Hypothesis Tracking (MHT) formulation, based on Maximum A Posterior (MAP) estimation, is considered for optimally associating measurement data over time to form estimates of the multiple tracks. For estimating K tracks, a trellis diagram of the measurements is used to depict all track set hypotheses as trellis paths. MAP path costs are computed using Kalman filters and *a priori* track set probabilities. A new K-track, List-Viterbi algorithm is then employed to effectively prune the number of evaluated hypotheses to a manageable level. Further hypothesis reduction is implemented through path merging and trellis truncation. The resulting *Viterbi MHT algorithm* is sequential. It is very flexible in that it can handle missed detections, false alarms and number-of-track estimation. It provides an ordered list of best track sets, which can be useful in subsequent data fusion. Compared to MHT algorithms which prune candidate track sets using "gating volumes", the new Viterbi MHT is less prone to loosing tracks.

1. INTRODUCTION

In multitarget tracking [1], we start with time evolving sets of noisy measurements of detected targets and false alarms, and we associate the detections over time to form multiple target tracks. With one class of track estimators, termed *Multiple Hypothesis Trackers (MHT's)*, this is accomplished by considering all hypothesized track sets. A *Maximum A Posterior (MAP)* criterion, which is commonly used to compare hypotheses, can account for missed detections and false alarms. For each hypothesis, a MAP cost is computed using Kalman filter generated innovations and *a priori* track set probability. An effective alternative approach is based on *conditional mean* estimation of the target states. The class of resulting tracking algorithms is referred to as *Bayesian*. Optimum MHT and Bayesian solutions require evaluation of a number of tracks or track sets which is exponentially growing with time.

Numerous approaches to hypothesis *pruning* and *merging* have been developed to reduce the computational burden of MHT and Bayesian trackers. One basic approach to pruning is measurement gating, which has been proposed in both MHT (e.g. [2], [3]) and Bayesian (e.g. [4], [5]) algorithms, and is commonly employed in practice. Another approach to pruning is based on evaluating the likelihoods of the extended tracks (e.g. see [3] for MHT and [5] for Bayesian). Alternatively, for a single track, Singer et al. [4] suggest merging all tracks which share measurements for the past N times. As noted by Bar-Shalom [1], the Probabilistic Data Association Filter (PDAF) [6], an approximate Bayesian tracker, is related to the $N = 0$ case. Joint PDAF (JPDAF) [7], which extends PDAF to multitarget tracking, time recursively extends multiple existing tracks which share measurements in their pruning gates. These practical but suboptimal pruning and merging approaches can generate diverged tracks during target maneuvers or after a few missed detections.

MHT is a discrete paramenter estimation problem, where the measurement-to-track associations are the discrete parameters. Pruning and merging techniques have been proposed for two similar discrete parameter estimation problems: digital communications where the Viterbi algorithm [8] and its extensions (e.g. [9]) are used to prune hypothesized discrete symbol sequences; and abruptly changing systems such as maneuvering target tracks, where a discretized multiple mode model is employed and a variety of approaches have been proposed for pruning or merging hypothesized discrete switching sequences (e.g. Generalized Pseudo Bayesian (GPB) merging [10]; Detection-Estimation Approximation (DEA) pruning/merging [11]; random populations pruning [12]; Interacting Multiple Modes (IMM) merging [13]; and Viterbi algorithm pruning [14]; [15]).

The Viterbi algorithm has also been used to prune hypotheses for several approaches to the target tracking in clutter problem. For time-recursive measurement-to-track association, the Viterbi Data Association (VDA) algorithm [16], an extension of an earlier Viterbi based algorithm [17], is a single track algorithm in which measurement gating is incorporated for an over-the-horizon radar application. A K-track Viterbi algorithm has been proposed for MHT [18], but this is only optimum with respect to a simple nonprobabilistic cost function (i.e. the energy required to get from one measurement to the next). A more efficient computational algorithm for the same cost was suggested in [19]. As a hypothesis pruner, the Viterbi algorithm assures the optimum hypothesis is kept only if all hypothesis costs satisfy a "finite state machine" or "Markov" condition [8]. Although this condition is satisfied for the simple nonprobabilistic energy cost used in [18], it is not for the maximum likelihood and MAP costs which are standard for MHT. For MAP based MHT, the single track Viterbi algorithms [16], [17] implement suboptimum sequential pruning, and are prone to loosing tracks for high clutter and missed detection scenarios, especially when extended to the multitrack problem.

In this paper we propose a new general Viterbi algorithm for multitarget tracking. For estimating K tracks, the trellis diagram of the measurements described in [18] is used to depict all track set hypotheses as trellis paths. MAP path costs are computed using Kalman filters and *a priori* track set probabilities. Because the

This research was supported by ONR under Grant N00014-98-1-0892.

hypothesis costs do not satisfy the Markov condition required for Viterbi algorithm pruning optimality, a new K track, list Viterbi algorithm is employed to effectively prune the number of evaluated hypotheses to a manageable level. Further hypothesis reduction is implemented through merging (as in [4]) and trellis truncation [8]. We term the resulting algorithm the *Viterbi MHT algorithm.* It is designed to avoid dropping tracks or track sets.

2. MULTITARGET TRACKING PROBLEM

In the Radar multitarget surveillance problem, the objective is to track multiple target trajectories over time. At each measurement time, noisy measurements are provided from multiple target detections. The measurements for each detection consist of a set of D estimated location and/or velocity parameters. We assume that K, the number of targets is known[1]. Detections can correspond to either targets or false alarms. Missed detections will be accounted for. For MAP estimation, the probability of detection P_d will be assumed known, as will the density function for the number of false alarms per measurement time in the surveillance volume.

2.1. Measurements

We denote as $\mathbf{z}_{m,j}$ the j^{th} the measurement vector at time m. Then, $\mathbf{Z}_m = \{\mathbf{z}_{m,j}; j = 1, 2, ..., J_m\}$ is the set of J_m measurement vectors at time m, and $\mathcal{Z}^n = \{\mathbf{Z}_1, \mathbf{Z}_2, ...\mathbf{Z}_n\}$ denotes the set of all measurement vectors up to time n.

2.2. Hypothesized Tracks

A single hypothesized track is characterized by a measurement vector or a missed detection for each measurement time. At time n let one such track, the l^{th} track, be denoted $\boldsymbol{\theta}_n^l$. The vector of measurements for this track is

$$\mathcal{Z}(\boldsymbol{\theta}_n^l) = \{\mathbf{z}_{1,j_1(l)}, \mathbf{z}_{2,j_2(l)}, \cdots, \mathbf{z}_{n,j_n(l)}\} \tag{1}$$

where the subscript $j_m(l)$ is the measurement index at time m for the l^{th} track. We account for the possibility of a missed detection by letting $j_m(l)$ range from 1 to $J_m + 1$, with $j_m(l) = J_m + 1$ indicating a missed detection. Note that a $\mathbf{z}_{m,J_m+1}$ is not an actual measurement, but indicates a missed detection at time m, and is a notational convenience. There are $L_n' = \prod_{m=1}^{n}(J_m + 1)$ of these hypothesized tracks: $\boldsymbol{\theta}_n^l; l = 1, 2, \cdots, L_n'$. Association of the measurements into a track is the problem of selecting the best $\boldsymbol{\theta}_n^l$, which is equivalent to estimating the discrete parameter $l \in \{1, 2, ..., L_n'\}$.

For a given hypothesized track $\boldsymbol{\theta}_n^l$, measurement noise is assumed additive, Gaussian and temporally white. The trajectory and measurements are assumed to evolve in time according to the state/measurement equations[2]

$$\mathbf{x}_{m+1}^l = \mathbf{\Phi}_m \mathbf{x}_m^l + \mathbf{w}_m^l \tag{2}$$

$$\mathbf{z}_{m,j_m(l)} = \mathbf{H}_m \mathbf{x}_m^l + \mathbf{u}_m^l \tag{3}$$

where at time m and for hypothesized track $\boldsymbol{\theta}_n^l$, $\mathbf{x}_m^l$ is the state vector, and $\mathbf{\Phi}_m$ and $\mathbf{H}_m$ are the state transition and output matrices respectively. The $\mathbf{w}_m^l$ and $\mathbf{u}_m^l$ vectors are zero mean, mutually independent, white and Gaussian with known covariance matrices $\mathbf{Q}_m^l$ and $\mathbf{R}_m^l$ respectively.

For each hypothesis, the measurement innovations sequence $\{\mathbf{v}_{1,j_1(l)}, \mathbf{v}_{2,j_2(l)}, \cdots, \mathbf{v}_{n,j_n(l)}\}$ and corresponding covariance matrices $\{\mathbf{S}_{1,j_1(l)}, \mathbf{S}_{2,j_2(l)}, \cdots, \mathbf{S}_{n,j_n(l)}\}$, generated by a hypothesis-specific Kalman filter computation, are to be used in optimum track estimation. Note that a $\mathbf{v}_{m,j_m(l)}$ corresponding to $\mathbf{z}_{m,J_m+1}$ is not an actual innovation, but is a result of a missed detection at time m, and again is a notational convenience. In such a case, the Kalman state is just the predicted state (i.e. no measurement is available to update the state).

2.3. Hypothesized Track Sets

Now consider a set of K tracks. Assume tracks can not share detections, and again assume missed detections are possible[3]. It can be shown that the number of hypothesized track sets is[4]

$$I_n' = \left(\sum_{j=0}^{K} \begin{pmatrix} J_1 \\ j \end{pmatrix} \right) \cdot \left(\prod_{m=2}^{n} \left(K! \sum_{j=0}^{K} \frac{\begin{pmatrix} J_m \\ j \end{pmatrix}}{(K-j)!} \right) \right) \quad . \tag{4}$$

Let the i^{th} hypothesized K-track set be represented as

$$\boldsymbol{\tau}_n^i = \{\boldsymbol{\theta}_n^{l_1(i)}, \boldsymbol{\theta}_n^{l_2(i)}, \cdots, \boldsymbol{\theta}_n^{l_K(i)}\} \quad , \tag{5}$$

where the superscript $l_k(i)$ denotes the k^{th} track of the i^{th} track set. $\mathcal{Z}(\boldsymbol{\tau}_n^i)$ will be used to denote the measurement data associated with the i^{th} K-track set. Multiple Hypothesis Tracking (MHT) algorithms aim to determine at time n the best from the I_n' hypothesized track sets.

3. MAP BASED TRACK ESTIMATION

Using a MAP formulation which incorporates prior track set probabilities, at time n we have the following optimization problem for determining the measurement-to-track associations. For $\boldsymbol{\tau}_n^i; i = 1, 2, \cdots I_n'$

$$\max_{\boldsymbol{\tau}_n^i} \quad p\left(\boldsymbol{\tau}_n^i | \mathcal{Z}^n\right) = \frac{p\left(\mathcal{Z}^n | \boldsymbol{\tau}_n^i\right) \cdot p\left(\boldsymbol{\tau}_n^i\right)}{p\left(\mathcal{Z}^n\right)} \quad . \tag{6}$$

$p\left(\mathcal{Z}^n\right)$ is a normalizing factor which serves to make the possible values of $p\left(\boldsymbol{\tau}_n^i | \mathcal{Z}^n\right)$ sum to unity, and can be ignored since it is independent of $\boldsymbol{\tau}_n^i$. $p\left(\boldsymbol{\tau}_n^i\right)$ is the *a priori* probability of the set of tracks, and is derived from knowledge of the sensors, targets, noise, clutter, jammers and preprocessors, as manifested for example as the probability of missed detections and the false alarm rate.

$p\left(\mathcal{Z}^n | \boldsymbol{\tau}_n^i\right)$ is the joint probability density function of the measurements $\mathcal{Z}^n$ conditioned on the track set $\boldsymbol{\tau}_n^i$. For a given $\boldsymbol{\tau}_n^i$, we can partition $\mathcal{Z}^n$ into the data associated with the tracks, $\mathcal{Z}(\boldsymbol{\tau}_n^i)$, and data associated with false alarms, $\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^i)$. We then have

$$p\left(\mathcal{Z}^n | \boldsymbol{\tau}_n^i\right) = p\left(\mathcal{Z}(\boldsymbol{\tau}_n^i) | \boldsymbol{\tau}_n^i\right) \cdot p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^i) | \boldsymbol{\tau}_n^i\right) \quad , \tag{7}$$

[1] Estimation of K, using a Viterbi algorithm approach, is discussed in [20].

[2] To simplify the discussion we assume linear state/measurement equations.

[3] These assumptions allow for representation of unresolved targets as a combination of detections and missed detections.

[4] $\begin{pmatrix} A \\ B \end{pmatrix}$ represents the number of combinations of A things taken B at a time.

where, under the assumption that false alarms are uniformly distributed over the surveillance volume Y,

$$p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^i)|\boldsymbol{\tau}_n^i\right) = \left(\frac{1}{Y}\right)^{\sum_{m=1}^{n} F_m^i} , \tag{8}$$

where F_m^i is the number of false alarms at time m given hypothesized track set $\boldsymbol{\tau}_n^i$. In terms of the Kalman filter generated measurement innovations, for the track measurements,

$$p\left(\mathcal{Z}(\boldsymbol{\tau}_n^i)|\boldsymbol{\tau}_n^i\right) = \prod_{p=1}^{K} p(\mathcal{Z}(\boldsymbol{\theta}_n^{l_p(i)})) = \prod_{p=1}^{K}\prod_{m=2}^{n} p(\mathbf{v}_{m,j_m(l_p(i))}) \tag{9}$$

$$= d_n^i \exp\left\{-\frac{1}{2}\sum_{p=1}^{K}\sum_{m=2}^{n} \mathbf{v}_{m,j_m(l_p(i))}^T \mathbf{S}_{m,j_m(l_p(i))}^{-1} \mathbf{v}_{m,j_m(l_p(i))}\right\} \tag{10}$$

where for each track p the product (or sum) over m does not include any missed detections since as noted earlier measurements and innovations for missed detections do not exist. Keeping this in mind,

$$d_n^i = \prod_{p=1}^{K}\prod_{m=2}^{n} \frac{1}{(2\pi)^{D/2} \cdot \sqrt{\det(\mathbf{S}_{m,j_m(l_p(i))})}} . \tag{11}$$

The MAP cost for estimating the best K tracks is described by (6), along with (8), (10), (11) and a specific form of $p\left(\boldsymbol{\tau}_n^i\right)$.

4. TRELLIS DIAGRAM FORMULATION

To derive an equivalent cost from which a time recursive trellis structure representation can be derived, take the negative natural log of (6), ignoring $p\left(\mathcal{Z}^n\right)$. For the i^{th} track set, the following equivalent MAP optimization problem is obtained:

$$\min_{\boldsymbol{\tau}_n^i} \quad \Lambda^n\left(\boldsymbol{\tau}_n^i\right) = \sum_{m=1}^{n} \Lambda_m(\boldsymbol{\tau}_n^i) \tag{12}$$

where the time m incremental cost is

$$\Lambda_m(\boldsymbol{\tau}_n^i) = \lambda_m(\boldsymbol{\tau}_n^i) + \sum_{p=1}^{K} \lambda_m(\boldsymbol{\theta}_n^{l_p(i)}) \tag{13}$$

where $\lambda_m(\boldsymbol{\tau}_n^i) = \ln\{p\left(\boldsymbol{\tau}_n^i\right)\}$ and the p^{th} track incremental cost at time m is

$$\begin{aligned} \lambda_m(\boldsymbol{\theta}_n^{l_p(i)}) &= \frac{1}{2}\ln(\det(\mathbf{S}_{m,j_m(l_p(i))})) + \frac{D}{2}\ln(2\pi) \qquad (14) \\ &+ \frac{1}{2}\mathbf{v}_{m,j_m(l_p(i))}^T \mathbf{S}_{m,j_m(l_p(i))}^{-1} \mathbf{v}_{m,j_m(l_p(i))} . \end{aligned}$$

For K track estimation, under the assumptions that tracks can not share detections, and that missed detections (as many as K) are possible, we choose a trellis, depicted in Figure 1, in which each state represents a set of K measurements and/or missed measurements. (This is the trellis used in [18].) For stage m with J_m measurements, there are

$$M_m = \sum_{j=0}^{K} \binom{J_m}{j} \tag{15}$$

states. Each branch from stage $m-1$ to m has a set of $K!/(K - T_m^i)!$ permutations associated with it, specifying the possible orders in which the K measurements and/or missed measurements represented by the state at stage m are used[5]. Now for each hypothesized K-track set, the incremental cost assigned to a branch is the sum of the costs of K hypothesized tracks. So, for hypothesized K-track set $\boldsymbol{\tau}_n^i$, the branch cost from stage $m-1$ to m is $\Lambda_m(\boldsymbol{\tau}_n^i)$. The MAP K-track estimation problem is now one of finding the minimum-cost path through this trellis.

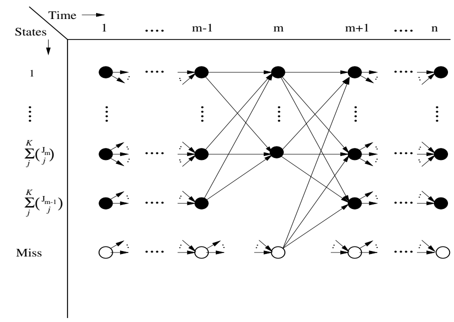

Figure 1: Trellis Diagram for general K.

5. THE MHT VITERBI ALGORITHM

The multitarget tracking algorithm we propose incorporates the list Viterbi algorithm, along with the K-track trellis formulation of the multitarget tracking problem [18], and MAP cost computation based on Kalman filter innovations and prior track set probabilities. The algorithm determines a list of L feasible K-tracks sets as a list of L paths through the measurement set trellis.

Algorithm

1) *Setup:* For each time $m = 1, 2, \cdots n$, allocate arrays of size M_m-by-L for the predecessor state indices, predecessor state L-best indices, permutation number of the current state measurement indices to track associations, and current total minimum negative log costs. Allocate Kalman filters for each track in each L-best sets of tracks for each state.

2) *Initialization:* For $m = 1,\ l = 1$, the current costs are set using (17) for the *a priori* probabilities, since no Kalman filter innovations are available initially. The Kalman filters are initialized using the available measurements for time $m = 1$, and using predicted values for time $m = 2$.

3) *Iteration:* For each time $m = 2, 3 \cdots n$, and for each state $j = 1, 2 \cdots M_m$

   3.1) Add incremental costs (13) from the L-best of each previous state to the total previous costs, and temporarily store these hypothesized new costs.

   3.2) Select the L-best of the hypothesized new costs and update the associated Kalman filters.

4) *Results:* Output the final L-best sets of tracks.

[5] Because each state represents all permutations of the corresponding K measurements/missed-events, each path through the trellis represents a number of hypothesized track sets.

Although the algorithm is described above in block form, it is time-sequential, and at any time during the iterations can produce current estimates of the best tracks.

5.1. Additional Hypothesis Merging and Truncation

The list-Viterbi implementation prunes the number of hypotheses out of stage m to $L \cdot M_m$. The value of L required to achieve a certain level of performance can be reduced by merging similar paths into a state prior to pruning the number of paths to L. This way, the list of L will contain diverse hypotheses, which is the objective (i.e. we want to keep some less-likely, significantly different hypotheses on the chance that they will become part of more likely hypotheses later in time).

In order to merge similar paths, a criteria must be established to determine whether paths are similar enough to be merged. The criteria used in our simulation code follows. Tracks are merged if:

1. they end on the same trellis state and state permutation, or if some of the K tracks end in a sequence of missed target events and the last actual measurements used for these tracks (at some previous trellis stage) are the same while the rest of the K tracks have common measurements back to the this previous stage; and
2. the trellis states used for the track sets coincide for at least two stage indices, and differ for no more than one stage index.

At each trellis state, the similar paths are first identified. For each set of similar paths, the paths are probabilistically averaged (as in [4]), i.e. the associated Kalman filter states are averaged, weighted by their normalized probabilities. The resulting averaged Kalman states then replace the states for the most likely path in the set, and the other similar paths are pruned.

6. NUMERICAL EXAMPLES

Letting $J_m = T_m^i + F_m^i$, where T_m^i is the number of measurement vectors at time m used in the K track set $\boldsymbol{\tau}_n^i$ and F_m^i is the number of assumed false alarms, we have that

$$p\left(\boldsymbol{\tau}_n^i\right) = \prod_{m=1}^{n} P_d^{T_m^i} \left(1 - P_d\right)^{K - T_m^i} p(F_m^i) \ , \qquad (16)$$

where $p(F_m^i)$ is the Poisson distributed probability of F_m^i false alarms. Then, referring to (13), the *a priori* incremental cost is

$$\begin{aligned} \lambda_m(\boldsymbol{\tau}_n^i) \ &= \ F_m^i \ln(Y) - T_m^i \ln(P_d) \qquad (17) \\ &- \ (K - T_m^i)\ln(1 - P_d) - \ln(p(F_m^i)) \ . \end{aligned}$$

First, two parabolic target tracks consisting of $n = 21$ $\mathcal{X}$ and $\mathcal{Y}$ position values were generated. Gaussian measurement noise with a variance of 0.01 was added to the target tracks. "False detect" events were generated using a Poisson distribution with a false alarm rate of 2, and the false detections themselves were generated using a uniform distribution over the range [0,4]. Probability of detection was assumed to be 0.7, and for the purpose of illustrating that our algorithm can "regain" a target even after a series of missed detections, the lower track was forced to have missed detections from $\mathcal{X} = 1.5$ to 2.5. Other simulation parameters were K=2 and L=16. Figure 2 shows the $\mathcal{X}$ and $\mathcal{Y}$ position values for

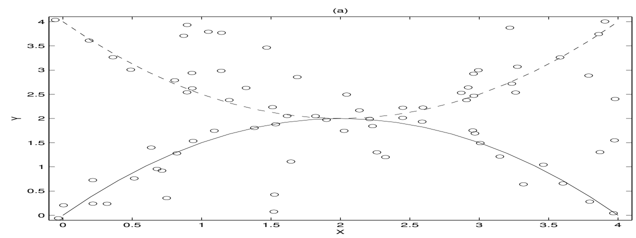

Figure 2: True Tracks and Measurements

the true tracks, and the noisy measurements used as input to the algorithm.

Figure 3 shows the best paths, the total cost of which is 117.35. Notice that the one track starts out following the lower target, but switches to the upper target around the region where the true tracks intersect. Figure 4 shows the fourth best paths, which have a total cost of 118.41, just slightly higher than the cost of the best paths, and which correctly follow both the targets. Notice that during missed-detect events, the lower track follows a path tangential to the true track, but "regains" the target again when detections are recorded.

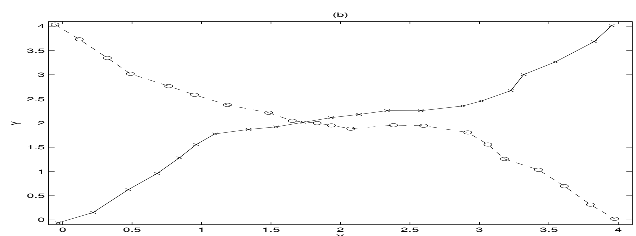

Figure 3: Best set of two tracks, cost=117.35

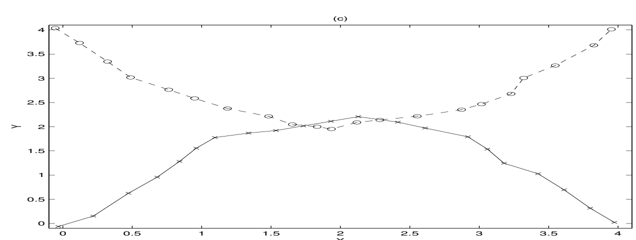

Figure 4: 4^{th} best set of two tracks, cost=118.41.

For the second simulation, Figure 5(a) shows an example scenario with a single target traveling in the positive X direction vs. time. In this figure, the solid line is the true target track. "x" marks the positions of false detections, which were generated randomly and uniformly over the observation space with a Poisson distribution average of two false detections per unit time. The "x"'s enclosed in circles represent measurements associated with the target, generated using a noise variance of 0.01. Four sequential missed target events occur in the range $X = 1.4 \ \ldots \ 2.0$, and three additional missed target events occur in the range $X = 2.8 \ \ldots \ 4.0$. The probability of detection is 0.6.

Figure 5(b) shows the tracks produced by running our algorithm for various values of L and path merging strategy. The track la-

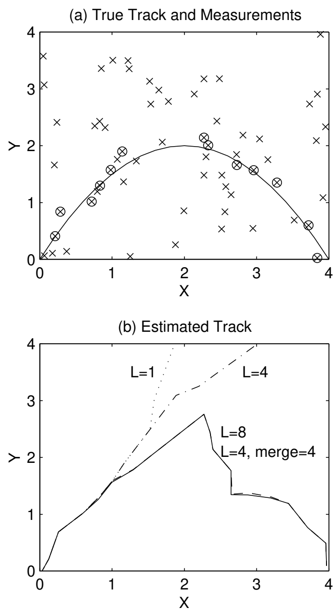

Figure 5: Single Track with Missed Target Events: (a) true track and measurements with false detections; (b) estimated track vs. values of L and merging strategy.

beled $L = 1$ is from the standard Viterbi algorithm, and the tracks labeled $L = 4$ and $L = 8$ are from our List Viterbi algorithm with no path merging. For $L = 1$ and $L = 4$, the estimated tracks diverge from the true track as soon as the first missed target events occur; they follow the linear predictions from the Kalman filter and/or false detections. For $L = 8$, the estimated track starts to diverge from the true track for $X \geq 1$, then reacquires the true track for $X \geq 2$.

The track labeled $L = 4$, merge=4, is from our List Viterbi algorithm using a track merging period of 4. This track is indistinguishable from the $L = 8$ track, and reacquires the true track with a lower computational and memory cost.

7. CONCLUSION

In this paper we describe a time-recursive K-track L-list Viterbi algorithm for MAP measurement-to-track association. This algorithm can handle missed detections, false alarms and track initialization. Compared to direct computation of the solution of the MAP problem, this algorithm manages computational cost by time recursively eliminating from further consideration hypothesized track sets which have too great a MAP cost.

8. REFERENCES

[1] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Trans. on Automatic Control,* Vol. 23, No. 4, pp. 618-626, Aug. 1978.

[2] P. Smith and G. Buechler, "A branching algorithm for discriminating and tracking multiple objects," *IEEE Trans. on Automatic Control,* Vol. 20, No. 1, pp. 101-104, Feb. 1975.

[3] C. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *IEEE Trans. on Automatic Control,* Vol. 22, No. 3, pp. 302-312, June 1977.

[4] R. A. Singer, R. G. Sea and K. M. Housewright, "Derivation and evaluation of improved tracking filters for use in dense multitarget environments," *IEEE Trans. on Info. Theory,* Vol. 20, No. 4, pp. 423-432, July. 1974.

[5] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, Vol. AC-24, No. 6, pp. 843-854, Dec. 1979.

[6] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica,* Vol. 11, pp. 451-460, Sept. 1975.

[7] T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Jour. of Oceanic Eng.,* Vol. 8, No. 3, pp. 173-183, July 1983.

[8] G. D. Forney, "The Viterbi algorithm," *Proc. of the IEEE,* Vol. 61, pp. 268-278, No. 3, Mar. 1973.

[9] N. Seshadri and C. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. on Communications,* Vol. 42, No. 2/3/4, pp. 313-323, Feb./March/April 1994.

[10] C. Chang, M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Trans. on Aerospace & Elect. Systems,* Vol. 14, No. 3, pp. 418-425, May 1978.

[11] V. J. Mathews and J. K. Tugnait, "Detection and estimation with fixed lag for abruptly changing systems", *IEEE Trans. on Aerospace & Elect. Systems,* Vol. 19, No. 5, pp. 730-739, Sept. 1983.

[12] Hajime Akashi and Hiromitsu Kumamoto. Detection and Estimation for Abruptly Changing Systems. *Automatica*, Vol. 13:429–434, 1982.

[13] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple Model Algorithm for systems with Markovian switching coefficients", *IEEE Trans. on Automatic Control,* Vol. 33, No. 8, pp. 780-783, Aug. 1988.

[14] J. K. Tugnait, "Detection and estimation for abruptly changing systems", *Automatica,* Vol. 18, pp. 607-615, 1982.

[15] A. Averbuch, S. Itzikowitz and T. Kapon, "Radar target tracking - Viterbi versus IMM," *IEEE Trans. on Aerospace & Elect. Systems,* Vol. 27, No. 3, pp. 550-562, May 1991.

[16] B. LaScala and G. W. Pulford, "Viterbi data association tracking for over-the-horizon radar," *Proc. Int. Radar Sym.,* Vol. 3, pp. 155-164, Sept. 1998.

[17] T. Quach and M. Farooq, "Maximum likelihood track formation with the Viterbi Algorithm," *Proc. 33-rd IEEE Conf. on Decision and Control,* pp. 271-276, Dec. 1994.

[18] J. K. Wolf, A. M. Viterbi and G. S. Dixon, "Finding the best set of K paths through a trellis with application to multitarget tracking," *IEEE Trans. on Aerospace & Elect. Sys.,* Vol. 25, No. 2, pp. 287-296, Mar. 1989.

[19] D. Castanon, "Efficient algorithms for finding the K best paths through a trellis," *IEEE Trans. on Aerospace & Elect. Sys.,* Vol. 26, No. 2, pp. 405-410, Mar. 1990.

[20] R. Perry, A. Vaddiraju and K. Buckley, "Trellis structure approach to multitarget tracking," *Proc. Sixth Annual Workshop on ASAP*, Mar. 1999.