

0.10.1 This paper is a postprint of a paper submitted to and accepted for publication in IET Radar, Sonar and Navigation and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at IET Digital Library.

Comparative investigation into Viterbi based and multiple hypothesis based track stitching

L. J. van der Merwe* and J. P. de Villiers*†

*University of Pretoria, Pretoria, South Africa, loodvdmerwe@gmail.com
 pieter.devilliers@up.ac.za †CSIR, Pretoria, South Africa, jdwilliers1@csir.co.za

Abstract—In this study a comparative analysis is performed between a novel Viterbi based and multiple hypothesis based track stitching algorithms. The track fragments in the Viterbi based track stitching algorithm are modelled as nodes in a trellis structure. A sequential Viterbi data association algorithm is then used to solve the trellis and associate track fragments with each other. A Kalman filter is used to determine the possible associations as well as the probabilities of the associations between the track fragments. In the multiple hypothesis track stitching algorithm, the hypothesis based multiple hypothesis tracking (MHT) algorithm is extended to perform track fragment to track fragment associations, rather than associating observations to tracks. Aspects of the developed multiple hypothesis algorithm are compared with implementations of a similar nature. Novel aspects of this research include the modification of the sequential Viterbi algorithm, as well as the extension of the MHT algorithm to solve the track stitching problem. It was found that the sequential Viterbi track stitching algorithm performed somewhat better than the multiple hypothesis track stitching algorithm for similar execution times. The Viterbi based track stitching algorithm is also shown to produce more consistently acceptable results.

Keywords—*track stitching, track fragments, sequential Viterbi, multiple hypothesis.*

I. INTRODUCTION

THE multiple target tracking (MTT) problem has been encountered in a variety of environments. Producing tracks from targets of interest using raw sensor observations has applications in numerous areas including military target tracking, guidance systems, civilian surveillance and threat assessment applications.

The requirement to associate track fragments with each other is inherent to many MTT environments, where occlusions and ambiguous possible track to track associations arise. This problem can be divided into two parts, namely tracking and track stitching [1]. The track of a target can be broken due to a number of reasons while a target is being tracked. These include a sufficient number of missed updates, due to a low sensor detection probability, target occlusions or in cases where the sensor update period might be long. In these cases the target track will usually be dropped by the tracking algorithm and re-initiated at a later time when a sufficient number of updates are received. This results in poor long term track management and in turn affects the performance of higher level algorithms such as threat assessment, target classification, resource assignment and guidance algorithms. The total number of track fragments and therefore, the total

number of possible track fragment associations scales exponentially, thereby increasing the computational complexity of the problem.

The multiple hypothesis tracking (MHT) [2] algorithm can be extended to solve the track stitching problem [3]. This algorithm builds matrix of possible association hypotheses, while deferring the associations to a later stage with the intent of preserving other possible solutions before pruning them. The association matrices can be solved by various algorithms including the auction algorithm [4], the integer programming algorithm [5], or by using an N -scan algorithm along with belief propagation [3]. Track fragments can also be modelled as nodes in a flow networks such as in [6] and [7]. The flow networks are then solved by associating the nodes with neighbouring nodes that are highly correlated. The process of finding track fragments that are highly correlated include a minimum flow algorithm [8] as well as an MHT approach in [7], where target tracks in ambiguous tracking regions are segmented and the resulting segments are then stitched to existing target tracks. Further advancements include estimating the missing data between track fragments using rank minimisation of a Hankel matrix [9], [10], [11] where the missing data is estimated in such a way that it is maximally consistent with the known data. This approach first estimates the model that is maximally consistent with the data, and then estimates the missing data using the estimated model. In [12] a Bayesian network is constructed based on the track fragments and then solved to obtain the most likely target tracks. Pruning is used to limit the number of connections in the Bayesian network, thereby reducing the complexity. An iterative multiple model approach is used in [13] to determine the underlying model, along with a prediction and retrodiction approach with gating to obtain the most likely track fragment associations. Track deletion is used to terminate low quality tracks based on the last update time of the track [14], [15]. Most recently a track fragment partitioning approach based on the time information and a multiple hypothesis filter [16] has been used to stitch track fragments together to aid in guidance systems. Moreover, in [17], the track to track association task is modelled by a track stitching graph extended from a graph based tracking (GBT) algorithm and reduced to a maximum weight bipartite matching problem.

In this paper a Viterbi based track stitching algorithm is presented and compared with the more standard MHT based approach. Certain features representing the track fragments are retained, even in the event that the track is dropped by

the tracking algorithm. The possible track fragment to track fragment associations are then visualised by modelling these track fragments as nodes in a track fragment trellis graph where the arrows between the nodes indicate possible associations [18]. The graph is then solved by using a sequential Viterbi data association algorithm [19], [20], [21] to find the most likely paths through the graph, corresponding to the most probable sequences of track fragments. In this paper, it is assumed that measurements from the sensor does not arrive out of sequence. It is also assumed that the targets are resolved, in that each target can only generate a single point observation at any given time.

The remainder of this paper is presented as follows: In Section 2, methods and algorithms that were developed are presented. In Section 3, the results and findings are presented and discussed. Finally in Section 4, conclusions are provided along with prospective work relating to this research.

II. PROCEDURES AND ALGORITHMS

In this section the procedures and algorithms used to perform MTT, as well as track stitching are discussed. A track maintenance algorithm and Kalman filter, which are used to perform target tracking, are discussed first. Second, a track stitching algorithm is presented, which selectively stores information from the generated track fragments. Nodes in a graphical model represent the stored information. Finally, an approach for solving the graphical model is discussed.

A. Modelling and target tracking

Before performing track stitching, MTT is performed to generate track fragments. This is accomplished through the following operations:

- 1) First model the bursts of true target detections caused by occlusions,
- 2) then model the true target positions as noisy radar observations, in the presence of false detections,
- 3) then use a Kalman filter, a track management algorithm and a global nearest neighbour (GNN) data association algorithm to generate track fragments.

The resulting track fragments will then be used to perform track stitching. Figure 1 shows a simulation flow of the tracking algorithm, including a target and a false generation processes, where the number of false alarms detections per scan follow a Poisson distribution. AAs can be seen from the figure, the tracking environment is initialised, by generating bursts of true targets. The true target tracks are then modelled as noisy sensor observations in polar coordinates. False targets are then generated, and the sensor observations are inserted into the list of false targets. The false targets result in the creation of false target tracks.

The sequential tracking algorithm then commences. Tracks are first initiated based on the observations of the detections. Next the observations are filtered using Kalman filters. The observations in the following scans are gated, and data association is performed on the observations within the gating area. It is worth noting here that selective information about the

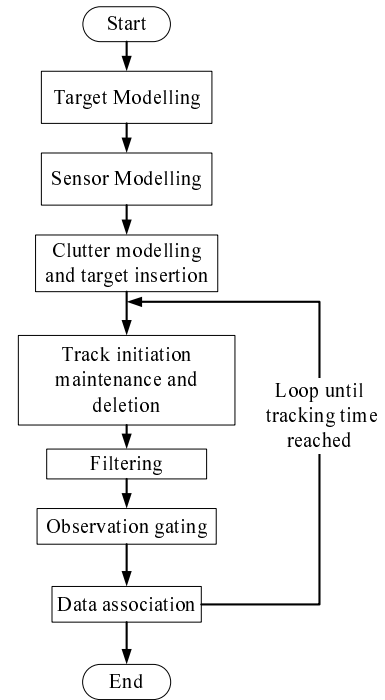


Fig. 1. Simulation flow of the tracking algorithm, including observation and clutter modelling. Certain track fragment features are retained to perform track stitching.

tracks are stored, in order to perform track stitching in real time.

The tracking algorithm used has the following characteristics[14]:

- 1) The true target positions are simulated using a linear Markov model with added white Gaussian noise.
- 2) The noisy sensor observations are modelled in polar coordinates.
- 3) The number of false detections for each scan follows a Poisson distribution.
- 4) The term P_D denotes the probability of detection of the sensor.
- 5) An M out of N algorithm is used to perform track maintenance.
- 6) The single point initiation technique [22] is used to initialise target tracks.
- 7) The ellipsoidal gating region G_0 [14], when no attribute data is contained in the measurement is given by

$$d^2 \leq G_0 = 2 \ln \left(\frac{P_D}{(1 - P_D)(2\pi)^{\gamma/2} \beta_{FA} \sqrt{|\mathbf{S}|}} \right), \quad (1)$$

where the false target rate is given by β_{FA} , the determinant of the filter residual covariance is given by $|\mathbf{S}|$ and γ denotes the dimension of the observation vector.

- 8) A GNN algorithm is used to perform data association using the Auction algorithm.

The true target track fragments are created by modelling the targets in a Markov model as being either in a visible or

occluded state. The transition probabilities between the states are chosen so that the output of the model produces bursts of true track fragments. This can be achieved by ensuring that the transition probabilities between the two states are low when compared with the transition probabilities returning to same nodes. The exact choices for the values of the transition probabilities are arbitrary and should be determined experimentally for a particular application.

B. Track stitching

Selective features of the estimated target tracks are retained, for use in the track stitching algorithm. These features include:

- 1) The filter covariance at the start of the track.
- 2) The track end state.
- 3) The filter covariance at the end of the track.
- 4) The creation time of the track.
- 5) The termination time of the track.

The very first covariance of the track cannot be used as the starting covariance, because of this will be much larger than the following covariances after an update has been received. Therefore, a smaller covariance is used for tracking when compared with the covariance used for track fragment associations. In order to perform track stitching, in this paper, the following assumptions are made:

- 1) A target can generate exactly one point measurement at any given time.
- 2) The sensor measurements does not arrive out of sequence.
- 3) The target will continue on an almost linear course when occluded, although the gating operation used allows for statistical deviations based on the plant noise of the dynamic model.

It can be inferred that an estimated target track can only be the result of exactly one target at time t with the above assumptions in mind. Each track fragment can be associated with only one target, therefore multiple track fragments existing simultaneously are mutually exclusive to any target. This is exploited to reduce the number of possible track fragment to track fragment associations.

A two target example is used to illustrate the operation of the track stitching algorithm. The two targets are denoted as M_1 and M_2 respectively. The targets are partially occluded according to the Markov model discussed earlier. Estimated track fragments $T_1, T_2 \dots T_5$ are created when the tracking algorithm is applied to the resulting radar observations. This scenario is shown in Figure 2. The track fragments in Figure 2 and possible associations, as determined by the gating process can now be modelled as a Markov model as shown in Figure 3. For the purposes of this example, it is assumed that the possible association pairs from the gating process are T_1 and T_3 , T_1 and T_5 , T_2 and T_3 , T_2 and T_4 and finally T_3 and T_4 . In Figure 3 the nodes are referred to as states, where each state represents a track fragment. The number of targets is initially unknown, but can at most be equal to the number of track fragments. A track fragment can either be stitched to a previous track fragment, from the same target or

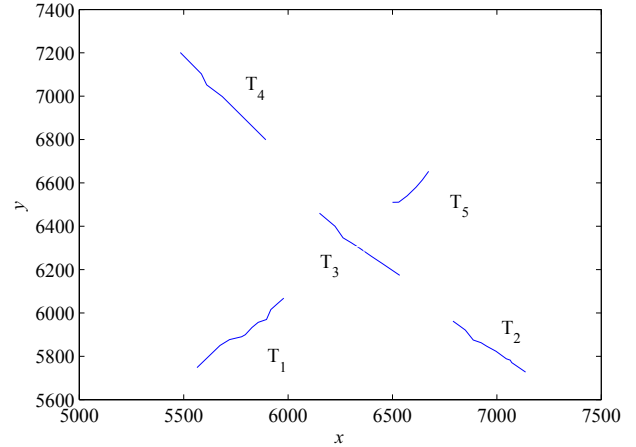


Fig. 2. The track fragments created by tracking the partially occluded targets.

a track fragment can be the result of a new target (false or real). If unassociated, each of the track fragments can be a target on its own. A transition to a different node represents that the target that generated the former track fragment is in the latter track fragment. Transitions returning to the same state are reserved for the no association hypothesis. As can be seen from Figure 3 ambiguity exists where track fragment T_3 can be associated with either of the two targets. Because of the assumption that observations and therefore the tracks are in the correct chronological order, these diagrams do not allow traversing from a later state to an earlier one.

The diagram in Figure 3 can be reconfigured into a trellis diagram as can be seen in Figure 4. For the sake of clarity, the invalid associations were removed. The invalid associations are indicated by a zero likelihood in implementation. In Figure 4, the columns of the graph represent the discrete times when the track fragments were inserted into the trellis diagram. A new column is only inserted in the graph when a new track fragment is confirmed. The trellis only has to be solved when a new track fragment is inserted. The grey nodes and arrows are inserted to preserve the rectangular structure of the graph. These association probabilities of the track fragment to itself (represented by horizontal grey arrows) are set equal to one, $p(T_n|T_n) = 1$.

1) *Solving the track graph trellis:* To solve the track graph trellis, a sequential Viterbi algorithm is used, and is briefly reviewed here.

The sequential Viterbi algorithm: The Viterbi [19] algorithm is a related algorithm of the sum-product belief propagation (message passing) algorithm and is often known as the min-sum, or max-product algorithm. The Viterbi algorithm solves the problem of maximising the probability of a sequence [20]. The goal is to find the sequence in the set of states, x , that maximises a global function, g , i.e. the sequence of most probable states. The most probable sequence in x can be

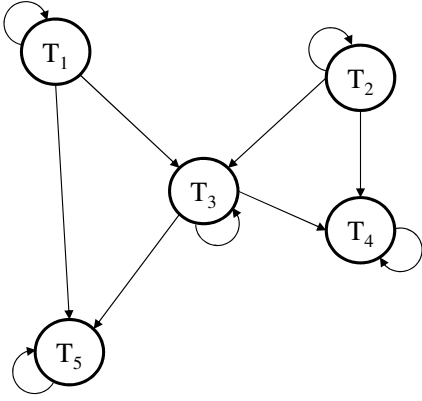


Fig. 3. The state diagram indicating possible transitions. The number of track fragments T_1 to T_5 , represented by a state transition diagram, where the state transitions show how paths generated by an initially unknown number of targets enter track fragments (states) in sequences $[T_1, T_2, T_5]$, $[T_1, T_3, T_4]$, $[T_2, T_3, T_4]$, $[T_2, T_3, T_5]$, $[T_1, T_5]$, $[T_1]$, $[T_2]$, $[T_3]$, $[T_4]$ and $[T_5]$

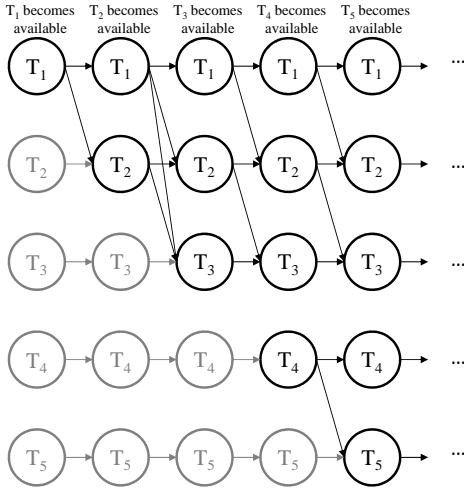


Fig. 4. The track graph trellis represented by the state diagram in Figure 3. The grey nodes are inserted to preserve the square structure of the graph.

defined by the term

$$\arg \max_x g(x), \quad (2)$$

where $g(x)$ is the inverse of the Viterbi cost. The Viterbi algorithm can be summarised as follows. For a Hidden Markov Model with state space S , initial probabilities π_i of being in state i and transition probabilities $a_{i,j}$ of transitioning from state i to state j , outputs y_1, \dots, y_T are observed. The most likely state sequence, x_1, \dots, x_T , that produces the observations is given by the recurrence relations in [20]

$$p_{1,k} = p(y_1|k)\pi_k, \quad (3)$$

$$p_{n,k} = p(y_n|k) \max_{x \in S} (a_{x,k} p_{n-1,x}). \quad (4)$$

The quantity $p_{n,k}$ is the probability of the most probable state sequence responsible for the first n observations, with the final

state k . The Viterbi path can be retrieved by storing pointers of which state x was used in equation 4. Let $\text{Pt}(k, n)$ be the function that returns the value of x used to compute $p_{n,k}$. Then,

$$x_T = \arg \max_{x \in S} (v_{T,x}),$$

$$x_{n-1} = \text{Pt}(x_n, n).$$

The algorithm can now backtrack to find the most probable state sequence (Viterbi path) using the pointer function, Pt . The probabilities can of course be implemented as costs (inverse probabilities), in this case the Viterbi algorithm aims to minimise the Viterbi cost, where the Viterbi cost is represented by the inverse of $g(x)$

The track graph can now be solved by maximising the association probability for the associations at each node, in each column of the track graph. This is akin to finding the most probable paths (lowest cost) through the track graph, where the paths are all mutually disjoint [21], i.e. the paths do not share any common nodes. Assuming that the track graph has columns, L , and rows, M . The path probability is given in equation 6, while the path cost is given in equation 5 for a path terminating at column l .

$$D_m = D_{l-1,m^*} + \min(\delta_{m,l-1}^G), \quad (5)$$

$$p(m) \propto \frac{1}{D_m}, \quad (6)$$

where D_{l-1,m^*} is the path cost in the previous column, and was found to terminate in row m^* , and D_m is the new path cost, terminating in row m . The quantity $\min(\delta_{m,l-1}^G)$ is the minimum transition cost (i.e. the cost associated with moving from a node in one column of the trellis to a node in the subsequent column) from column $l-1$ to the node in row m , column l . The quantity $\delta_{m,l-1}^G$ represents the cost of moving from a node in the current column to a node in the next column. The quantity $p(m)$ is the probability associated with the path cost D_m . Again the probabilities can be normalized by ensuring the probabilities of the paths likelihoods sum to unity.

Any of these two equations can be used to determine the most likely paths. The solution to the track graph is therefore similar to a multiple Viterbi algorithm, where the paths are mutually disjoint.

The no association (horizontal transitions) are assigned the constant value of the new track probability, $p(m_{NT})$ or the corresponding new track cost. This was determined through Monte Carlo simulations. Messages of the current path likelihood are propagated from a node in the current column, to nodes in next column in the graph. These path likelihoods are updated using equation 6. Once the end of the graph is reached, the algorithm backtracks through the most likely nodes to obtain the set of most likely paths. These paths then represent the most likely track-to-track associations. For the example problem presented previously, two solution paths are found, representing the two target tracks consisting of a number of track fragments. These solution paths are shown in Figure 5. The number of solutions paths through the trellis, is not known *a priori*. The algorithm keeps on solving the trellis for solution

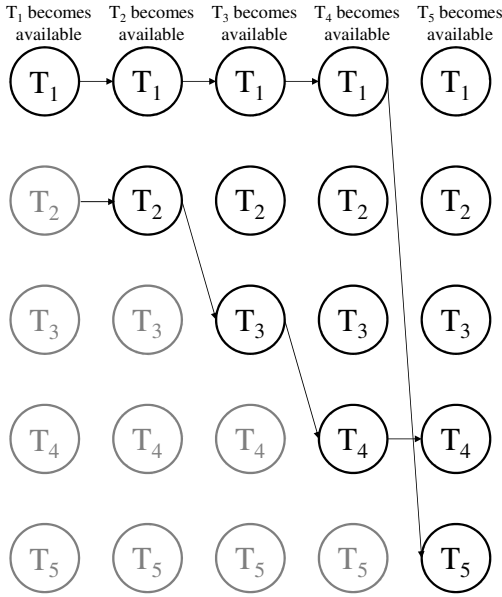


Fig. 5. The two most probable solution paths through the trellis. Track fragments T_1 and T_5 are associated with target M_1 , while fragments T_2 , T_3 and T_4 are associated with target M_2 .

paths through the graph up to the point where each of the track fragments (nodes) have been used in a solution path.

2) *Track fragment to track fragment associations*: An ellipsoidal gating algorithm is used to reduce the number of track fragment to track fragment associations. This is performed by predicting the current position of a target from each earlier track fragment using a Kalman propagator (i.e. a Kalman filter coasts up to the current point in time for each previous track fragment). As the Kalman filter coasts, the filter covariance increases to account for the uncertainty in the target position. The increase in the filter covariance in turn increases the size of the ellipsoidal gate. Each track fragment is now evaluated in turn. If the estimated starting position of the current track fragment falls within the gate around the predicted position of a previous track fragment, an association may then be made between the two track fragments with a certain association probability. The track fragment to track fragment association probability is defined in the next subsection. An ellipsoidal gating region is used as in the target tracking algorithm, given by

$$G_i = 2 \ln \left(\frac{P_{D_T}}{(1 - P_{D_T})(2\pi)^{\gamma/2} \beta_{FA} \sqrt{|S|}} \right), \quad (7)$$

where P_{D_T} denotes the probability of a detecting a target resulting from the M/N track initiation process and is generally lower than the probability of detecting an observation P_D , resulting in a larger gate. This probability is generally slightly lower than that of P_D resulting in a larger gate. In equation 7, γ refers to the dimension of the estimated state vector of the track and the determinant of the innovation covariance is given by $|S|$. The false target rate is given by β_{FA} . A binary association matrix can now be constructed by using

the associations and the time restrictions imposed as described earlier. This association matrix is square with the dimensions equal to the number of track fragments.

3) *Track fragment to track fragment association probability*: The track fragment to track fragment association probabilities are defined in this subsection. The viability of assigning one track to another is described by these probabilities. The likelihoods of assigning an earlier track T_i to a later track T_j is illustrated by Equations 8 to Equation 11.

$$P_t^{ij} = P_t^i + P_t^j \quad (8)$$

$$\hat{\Delta}_t^{ij} = \hat{x}_t^i - \hat{x}_t^j \quad (9)$$

$$\delta_t^{ij} = \hat{\Delta}_t^{ij\top} [P_t^{ij}]^{-1} \hat{\Delta}_t^{ij} \quad (10)$$

$$p(T_i|T_j) \propto \frac{1}{\delta_t^{ij}} \quad (11)$$

In Equations 8 to 11 it is assumed that the estimation errors $\hat{x}^i - x^i$ and $\hat{x}^j - x^j$ are independent of each other [23] therefore no cross-covariance term is subtracted. This assumption is not necessarily true and is a simplification which gives at least the worst case estimate of the covariance. The result is that a larger combined covariance is obtained in Equation 8 because of the lack of the cross-covariance term, which leads to a larger ellipsoidal gate for all associations. P_t^{ij} refers to the combined covariance of the two track fragments in Equations 8 to 11. The difference between the state estimations of the two tracks is given by $\hat{\Delta}_t^{ij}$. Finally, δ_t^{ij} refers to the association cost.

All the association costs δ^{ij} entering a node in Figure 4 are normalised, resulting in the approximate association probabilities $p(T_i|T_j)$ (note the proportionality in Equation 11. This is achieved by ensuring that these probabilities all sum to unity. The term P_t^i in Equations 8 to 11 refers to the covariance at the end of an earlier track fragment, while the covariance at the start of a later track fragment is referred to as P_t^j . The combined covariance at time t is referred to as P_t^{ij} . The estimated starting position of the later track fragment is denoted as \hat{x}_t^j , while the estimated ending position of an earlier track fragment is denoted as \hat{x}_t^i . The association scores are inverted in 11 and normalized to determine the association probabilities.

The track fragments are stitched together with the missing position estimations linearly interpolated. The linear interpolation performed is not ideal, but is used here, since track interpolation is not the focus of this study. A better approach is to estimate the missing observations and then filter over these observations as in [9] [10].

The manner in which track splitting and merging is handled warrants some discussion. Unlike the approach in [8], where an explicit track merging and splitting algorithm is used, this approach purges unassociated tracks and handles track splits by instantiating new target tracks. In particular, the unassociated track fragments need to be purged from memory while the associated track fragments need to be permanently fused. This is required to reduce the computational complexity of the track fragment to track fragment association, thereby ensuring real time data association. A sliding window approach is used to

perform this. Let the size of the window be K . The number of elapsed time steps since the track has ended is recorded. If the amount of time steps elapsed exceeds the size K of the window, where no association was made to the particular track fragment, the track is purged from memory [14], [15]. If an association was made to a particular track fragment, the number of time steps in which the association is confirmed is recorded. If this exceeds $K/2$, the two tracks are fused together subject to the following rules:

- 1) The new track fragment has the starting parameters of the earlier track.
- 2) The new track fragment has the ending parameters of the later track.
- 3) The missing estimated positions between the two track fragments are linearly interpolated. Ideally, interpolation should be performed by utilising underlying motion model.
- 4) The two individual track fragments are removed from memory and replaced with the new track fragment.
- 5) All previous assumptions regarding track independence and gating techniques and time constraints as mentioned earlier applies to the new track fragment.

III. MULTIPLE HYPOTHESIS TRACK STITCHING

Traditional MHT's typically cannot track targets over extended periods of time because of the large number of hypotheses that frequently gets generated [2] [7]. In a typical MTT environment however, the number of tracks and track fragments that get generated are significantly fewer than the number of observations. Track fragments instead of observations are used as the basic building blocks of track hypotheses. While the same approach is used in [7] to create an extended duration MHT track stitching algorithm, the approach in this paper simply extends the hypothesis based MHT algorithm to use track fragments instead of observations. The approach in [7] categorises track fragments into ambiguous and pure fragments and assumes that no false alarm detections are present while the approach in this paper does not.

The same track to track gating technique is used as discussed in the previous section to generate a binary validation matrix for each hypothesis Θ_{k-1}^i , where i denotes the index of the hypothesis, and $k-1$ denotes the previous time step.

From the generated validation matrix for the current hypothesis an assignment matrix is generated as was described in the previous section. The likelihoods l_{ij} in the validation matrix now denotes the track fragment to track association likelihood and is given by

$$l_{ij} = \frac{P_{D_T}^j p_{k|k-1}^j(\hat{\mathbf{x}}_{0_k}^i)}{(1 - P_{D_T}^j P_G^j)},$$

where P_{D_T} refers to the probability of detection of a track. The quantity $p_{k|k-1}^j(\hat{\mathbf{x}}_{0_k}^i)$ is the predicted Gaussian measurement density of the j th target at the first estimated position of track fragment i and the term P_G^j denotes the gate probability of the j th target. The false alarm rate, usually used in multiple hypothesis tracking in the assignment matrix

is replaced by β_{FT} to denote the false track rate.

The best solution (highest probability) of the N_h solutions produced by Murty's algorithm for each of the hypotheses is now selected and these hypotheses are updated (stitched) using the corresponding track fragments. The best solution in each of the hypotheses now form the new set of hypotheses Θ_k^i .

For the hypotheses where a new track is the most probable, the track fragment is added to the list of tracks and other track fragments can now be stitched to it. The tracks and track fragments that did not have a valid update in a fixed number of consecutive scans N_P (pruning interval) are pruned and are not considered in subsequent hypotheses. The value of N_P is dependent on the application and should be chosen to be larger than the maximum expected occluded time on the target.

The focus here is not to present a track stitching version of MHT, since MHT is the de facto standard method for solving the target tracking problem [7]. As such the equations for MHT are not provided here, but can be obtained in classical texts such as [14]. MHT is used here as a baseline for comparison

IV. RESULTS AND FINDINGS

In this section the track stitching results are presented. Two cases are presented to illustrate performance of the track stitching algorithm. A comparison is also made to a multiple hypothesis track stitching approach. In both implementations the MATLAB code used to simulate the algorithms was optimised and vectorised as far as possible to ensure compatible sets of results.

A. Track fragment generation

Two cases are presented in this section to illustrate the performance of the sequential Viterbi track stitching algorithm. In both cases the following transition probabilities are used when generating the true target tracks.

$$\begin{aligned} p(\text{Visible}|\text{Visible}) &= 0.8, p(\text{Visible}|\text{Occluded}) &= 0.2, \\ p(\text{Occluded}|\text{Visible}) &= 0.4, p(\text{Occluded}|\text{Occluded}) &= 0.6 \end{aligned}$$

Here, V denotes the target in the *Visible* state, while O denotes the target in the *Occluded* state as described by the Markov model in Section 2. In the two cases presented here, two true target tracks are generated with track fragmentation induced.

B. Track fragment epochs

It is assumed that the observations arrive in sequence, therefore, it can be inferred that the track fragments are created in a chronological order. Of particular interest is the time intervals during which the track fragments existed. Figure 6 shows the surviving times for track fragments for an example scenario. As can be seen overlapping occurs between the survival times of certain tracks, therefore these tracks could not have been generated by the same target. This fact is exploited to reduce the number of valid associations.

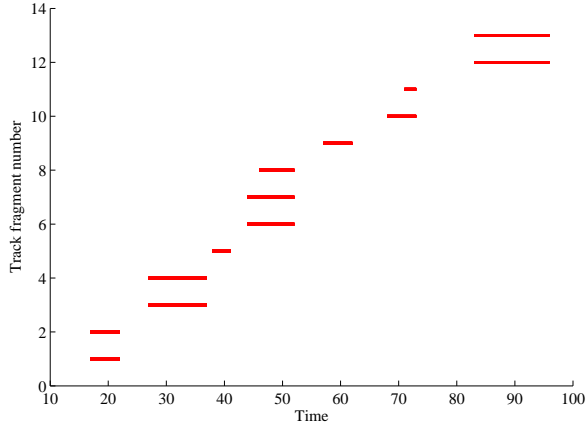


Fig. 6. The time intervals during which the track fragments existed.

C. Track stitching

The true tracks of the two targets for this case is shown in Figure 7 with the estimated track fragments superimposed. The track fragments were stitched in real time using the sequential Viterbi track stitching algorithm. The result of the track stitching algorithm is shown in Figure 8. As can be seen from Figure 8, the sequential Viterbi algorithm successfully stitched the track fragments together to create two estimated target tracks.

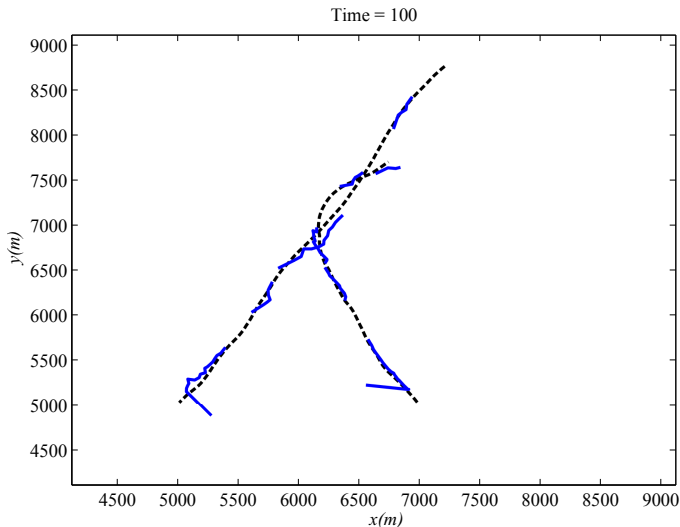


Fig. 7. Two true target tracks with the estimated track fragments superimposed. The units are in meters (m).

Figures 9 and 10 show a case where three targets cross and where the tracks are stitched successfully.

It was found that the Viterbi track stitching algorithm performed poorly in cases whenever there is a lack of track fragments where the targets cross. In these case the MH track stitching algorithm could stitch the track fragments together correctly when a large enough number of hypotheses

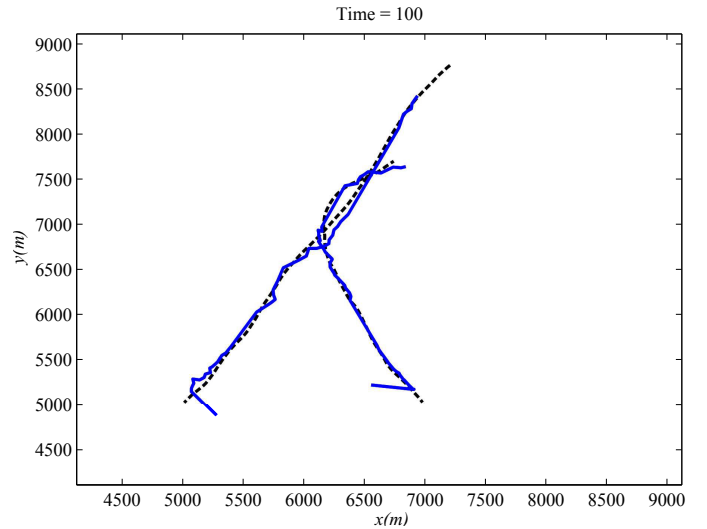


Fig. 8. Two true target tracks with the stitched tracks superimposed. The units are in meters (m).

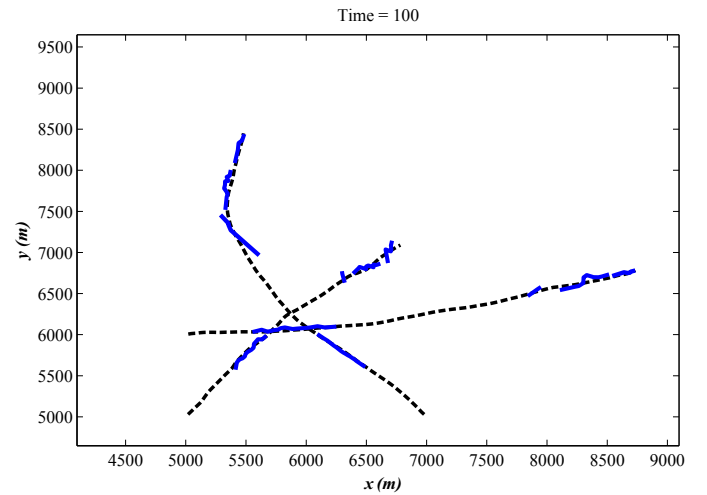


Fig. 9. Three true target tracks with the estimated track fragments superimposed. The units are in meters (m).

are considered. A comparison between these two algorithms follow in the next section.

V. MH TRACK STITCHING VS. SEQUENTIAL VITERBI TRACK STITCHING

A Monte Carlo analysis of 250 simulation runs was performed on the sequential Viterbi track stitching algorithm and the multiple hypothesis track stitching algorithm. Three discrete levels of ambiguity were chosen to compare the two algorithms. Table I shows the average correct track stitching probabilities for three different levels of ambiguity. The levels of ambiguity relates to the ambiguity in the simulated scenarios and increases as the ambiguity increases. In the table the number of time steps before pruning a track fragment is denoted by $N_{P_{MHT}}$ and $N_{P_{Vi}}$ for both algorithms respectively.

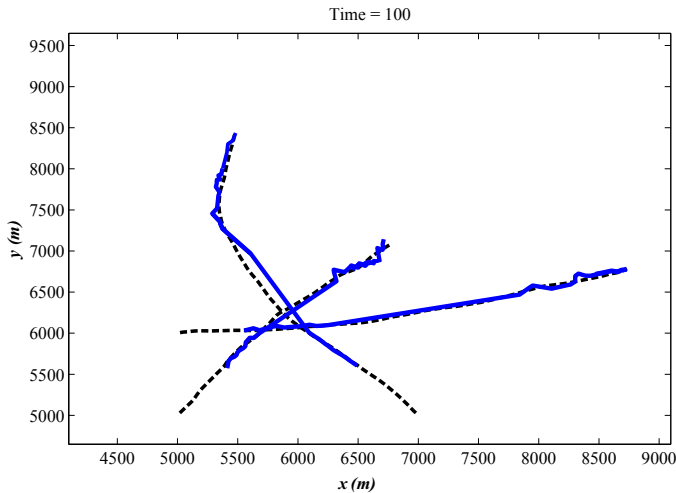


Fig. 10. Three true target tracks with the stitched tracks superimposed. The units are in meters (m).

The number of times steps before fusing two associated track fragments in the sequential Viterbi track stitching algorithm is denoted as N_F , while the number of hypothesis considered in the multiple hypothesis track stitching algorithm is denoted by N_h .

These variables influence the execution times as well as the performance of the two algorithms where chosen in such a way as to obtain similar execution times between the two algorithms. This allows for a fair comparison between the performances of the two algorithms.

$$\begin{aligned} N_{P_{Vi}} &= 15 \text{ time steps} \\ N_F &= 10 \text{ time steps} \\ N_{P_{MHT}} &= 5 \text{ time steps} \\ N_h &= 5 \text{ hypotheses} \end{aligned}$$

TABLE I. EVALUATION OF THE MULTIPLE HYPOTHESIS TRACK STITCHING ALGORITHM AND THE SEQUENTIAL VITERBI TRACK STITCHING ALGORITHM

Level of ambiguity	Viterbi track stitching		MH track stitching	
	Average execution time over 250 time steps (seconds)	Correct track stitching probability	Average execution time over 250 time steps (seconds)	Correct track stitching probability
Level 1	14.22 ± 4.14	0.79 ± 0.21	14.38 ± 4.84	0.67 ± 0.29
Level 2	16.13 ± 4.08	0.71 ± 0.24	15.22 ± 4.61	0.62 ± 0.31
Level 3	16.88 ± 4.22	0.67 ± 0.31	16.32 ± 4.54	0.58 ± 0.34

Figure 11 shows a comparison of the two algorithms for the first level of ambiguity in Table I. Figure 11 also shows the mean and $1-\sigma$ covariance Gaussian ellipse for the two cases. From the means it can be seen that the sequential Viterbi track stitching algorithm performs somewhat better than the multiple hypothesis track stitching approach for similar execution times.

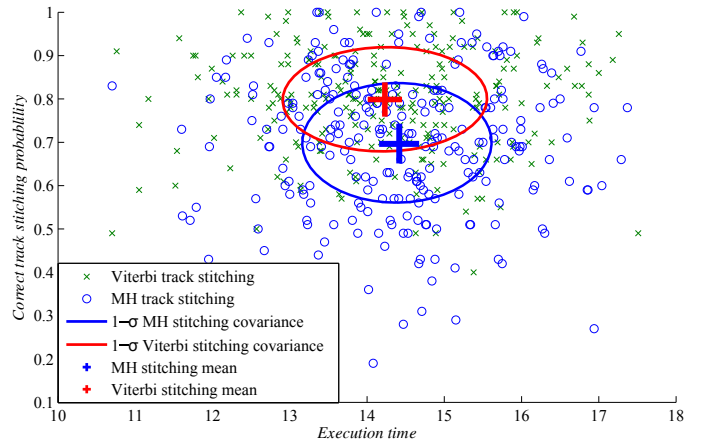


Fig. 11. A comparison between MH track stitching and sequential Viterbi track stitching.

VI. CONCLUSION

In this paper it was shown that track stitching can be performed by using graphical models and message passing. The tracks were modelled as nodes in a lattice track graph structure. A method to solve the track graph was also provided. The hypothesis based multiple hypothesis tracking algorithm was extended to perform track to track associations instead of observation to track associations. It was shown that the sequential Viterbi algorithm performs somewhat better than the MHT track stitching algorithm for similar execution times, and, the Viterbi algorithm produces more consistent results.

It should be noted that if the number of hypotheses, N_h , were to be increased the execution time of the MHT track stitching algorithm would also increase due to the combinatorial explosion. This increase would however cause the MHT track stitching algorithm to consider more solutions and therefore perform better than the Viterbi based algorithm.

Future work includes extending the research to account for multi-sensor systems, it also includes evaluating the algorithm with real world data, and higher target densities as well as modelling the track fragments using a more general graphical model. These models can then be solved by using more advanced algorithms [4], [14], [15], [19]. Additional future work also includes extending this work to a three dimensional space and comparing the track stitching algorithms considered in this study with other track stitching algorithms in the literature. In this study a simple linear interpolation of the missing positions was performed. Future work includes considering more advanced smoothing algorithms to interpolate in between the track fragments that satisfy the motion model [9], [10].

REFERENCES

- [1] R. Ivey, J. Horn, and R. Merket, "Long-duration fused feature learning aided tracking," *Multisensor, Multisource Information Fusion: Architecture, Algorithms and Applications*, Apr. 2012.
- [2] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843 – 854, Dec. 1979.

- [3] Z. Chen, L. Chen, M. Cetin, and A. Willsky, "An efficient message passing algorithm for multi-target tracking," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, Jul. 2009, pp. 826–833.
- [4] D. Bertsekas, "A distributed asynchronous relaxation algorithm for the assignment problem," in *Decision and Control, 1985 24th IEEE Conference on*, vol. 24, 1985, pp. 1703–1704.
- [5] C. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *Automatic Control, IEEE Transactions on*, vol. 22, no. 3, pp. 302–312, Jun. 1977.
- [6] G. Castanon and L. Finn, "Multi-target tracklet stitching through network flows," in *Aerospace Conference, 2011 IEEE*, Mar. 2011, pp. 1–7.
- [7] C.-Y. Chong, G. Castanon, N. Coopriker, S. Mori, R. Ravichandran, and R. Macior, "Efficient multiple hypothesis tracking by track segment graph," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, Jul. 2009, pp. 2177–2184.
- [8] Z. Wu, T. H. Kunz, and M. Betke, "Efficient track linking methods for track graphs using network-flow and set-cover techniques," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1185–1192.
- [9] T. Ding, M. Sznaiier, and O. Camps, "Fast track matching and event detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. vol.2, Jun. 2008, pp. 1–8.
- [10] —, "A rank minimization approach to fast dynamic event detection and track matching in video sequences," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 4122–4127.
- [11] R. Lublinerman, M. Sznaiier, and O. Camps, "Dynamics based robust motion segmentation," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, June 2006, pp. 1176–1184.
- [12] A. J. Abrantes, J. S. Marques, and J. M. Lemos, "Long term tracking using bayesian networks," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, June 2002, pp. 609–612 vol.3.
- [13] S. Zhang and Y. Bar-Shalom, "Track segment association for gmti tracks of evasive move-stop-move maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 1899–1914, July 2011.
- [14] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [15] S. S. Blackman, *Multiple-target tracking with radar applications*, ser. Artech House radar library. Norwood, Mass. Artech House, 1986. [Online]. Available: <http://opac.inria.fr/record=b1105067>
- [16] D. Salmond, "Tracking and guidance with intermittent obscuration and association uncertainty," in *Information Fusion (FUSION), 2013 16th International Conference on*, 2013, pp. 691–698.
- [17] S. Mori and C.-Y. Chong, "Performance analysis of graph-based track stitching," in *Information Fusion (FUSION), 2013 16th International Conference on*, 2013, pp. 196–203.
- [18] L. van der Merwe and J. de Villiers, "Track-stitching using graphical models and message passing," in *Information Fusion (FUSION), 2013 16th International Conference on*. IEEE, 2013, pp. 758–765.
- [19] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
- [20] J. Forney, G.D., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [21] G. Pulford, "Multi-target viterbi data association," in *Information Fusion, 2006 9th International Conference on*, Jul. 2006, pp. 1–8.
- [22] M. Mallick and B. L. Scala, "Comparison of single-point and two-point difference track initiation algorithms using position measurements," *Acta Automatica Sinica*, vol. 34, no. 3, pp. 258–265, 2008.
- [23] Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.