10

Multiple Hypothesis Tracking Implementation

Angelos Amditis[1], George Thomaidis[1], Pantelis Maroudis[2],
Panagiotis Lytrivis[1] and Giannis Karaseitanidis[1]
[1]Institute of Communications and Computer Systems,
[2]Institut Supérieur de l' Aéronautique et de l' Espace,
[1]Greece
[2]France

1. Multiple hypothesis tracking

Laserscanner sensors deliver distance measurements from the reflections of objects in the environment. In most automotive applications, the output of the sensor should consist of a list of detected objects. A common architecture for object detection in a laserscanner processing system is the following:

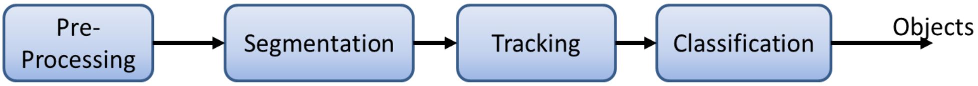


Fig. 1. Laserscanner measurements processing architecture

Tracking algorithm takes as input a list of clustered laserscanner measurements (objects), which for simplicity will be called measurements. The tracking algorithm forms a track whenever there is “enough” evidence that a sequence of measurements represents a real target. Additionally, by using the appropriate filtering techniques the tracking algorithm estimates the kinematic state of the formed track.

A basic step of a tracking algorithm is the measurement to track association. This is a very important procedure since wrong association could mean updating a track with a wrong measurement, initialization of a false track or deletion of a real track if it has erroneously not been associated with a measurement for one or more scans.

When a new set of measurements is outputted by the sensor, each measurement can i) be assigned to existing track, or ii) initiate a new track or iii) be considered as a false alarm. The simplest and most widely spread approach for measurement to track and track to track association, is the global nearest neighbour algorithm (GNN) (Blackman 1999). This approach formulates the most likely track to measurement and new track hypotheses. In the Joint Probabilistic Data Association (JPDA) algorithm (Fortmann T. 1983), multiple track to measurement hypotheses are generated. Hypotheses probabilities are calculated and then assignment hypotheses for each track are merged. With this method, track state is updated using all the measurements that are within the track gate by using a weighted sum of each hypothesis track estimate. These two methods form one track estimate for each track hypothesis. So, GNN and JPDA methods make a "hard" decision on the current scan in cases of conflicting measurement to track association hypotheses.

On the other hand, Multiple Hypotheses Tracking (MHT) methods form alternative association hypotheses in case of observation to track conflict situations. The set of hypotheses is propagated in the next scans with the anticipation that future observations will resolve assignment ambiguities. This method is divided into two approaches, the hypothesis oriented MHT (HOMHT) and the track oriented MHT (TOMHT).

In order to demonstrate the difference between these tracking techniques a practical example will be shown. Assuming there are 2 tracks on scan N and 3 measurements are received in the same scan. The gate formation is illustrated in Fig. 2.

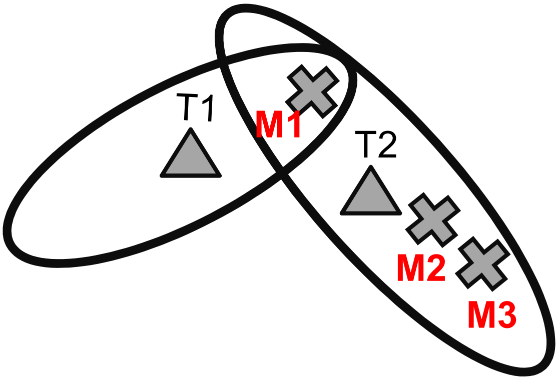


Fig. 2. Gating procedure, tracks are drawn with triangles and measurements are drawn with crosses.

If d_{ij} is the statistical distance between track i and measurement j, and this distance is less than a threshold value (gate size), this pair is candidate for association. Gating is necessary for elimination of unlikely observation to track pairs. Usually a cost function is used for defining the cost of assigning track i to measurement j. The GNN algorithm would use an optimization algorithm for solving this problem, so by using this method T2 would be updated using M2, and T1 would be updated using M1. M3 would initiate a new track. On the other hand, by using JPDA, T2 would be updated by using all measurements and T1 would be updated using M1. The MHT will form different hypotheses by taking into account all the possible sources of a measurement: i) new track, ii) false alarm and iii) existing track.

This chapter aims into giving the reader an overview of the Multiple Hypothesis Tracking approach. This technique is a complicated approach that has a strong mathematical formulation and many variations regarding proposed implementations. This chapter aims to give the reader an overview of the Multiple Hypotheses Tracking philosophy along with some practical examples regarding techniques used in MHT. Reader is encouraged to use this chapter as a starting point of getting familiar with this technique and investigate further with other more focused publications on this topic. Since this chapter aims in making the reader familiar with MHT, complicated mathematical expressions were avoided

2. Hypothesis oriented MHT

Hypothesis oriented MHT presents an exhaustive method of enumerating all possible assignment track to measurement combinations. When a new measurement set is received, observations that fall within a track's validation region set a possible measurement to track assignment.

The idea of propagating multiple assignment hypotheses was first presented in (Singer et. al.,1974), however Reid is considered as the first to present a systematic approach using a hypothesis oriented approach for implementing a multiple hypothesis tracker (Reid 1979). In this approach, when a new set of measurements is received, an existing hypothesis is expanded to a set of new hypotheses by considering all the possible assignments of the tracks contained in the original hypothesis. Each hypothesis contains a set of compatible observation to track assignments, leading to an exhaustive approach of enumerating all the possible assignment combinations

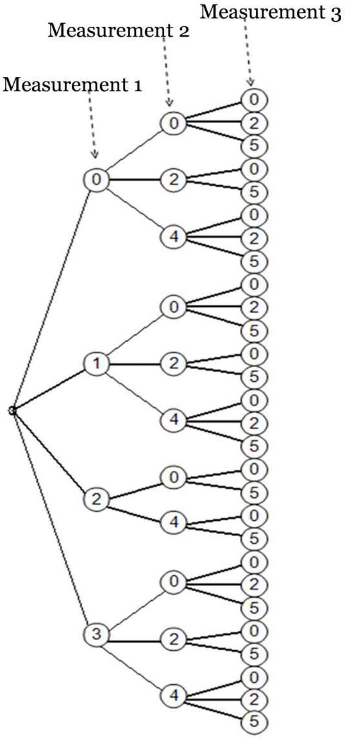
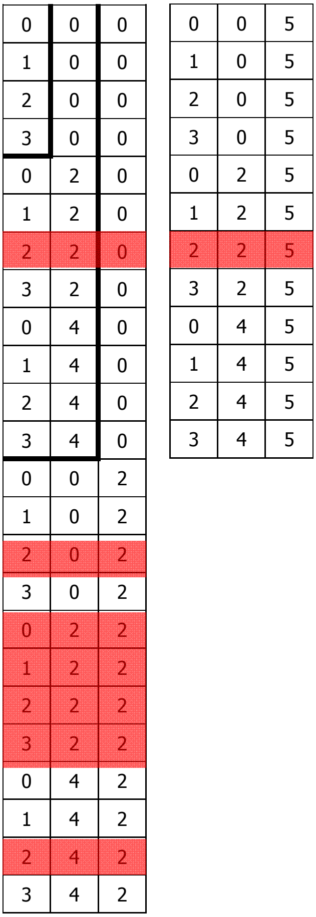


Fig. 3. Tree representation of the formed hypotheses

Taking as example the assignment problem in Fig. 2, the formed hypotheses can be represented in tree (Fig.3) or table representation (Fig. 4). The root of the tree is the original hypothesis – in the current example the original hypothesis is that there are two tracks. The first expansion of the hypothesis tree is done by using all the possible assignments of the first measurement. The numbers in the nodes indicate to which track the measurement is assigned. Regarding the origin of measurement 1, 4 hypotheses can be made: i) false alarm (assignment to the zero track), ii) new track (assignment to track number 4) , iii) and iv) assignment to existing tracks 1 and 2. In the same way the rest of the tree is created. It can be seen that the tree depth is equal to the number of measurements in the current scan. From an implementation perspective, it is more practical to represent the set of hypotheses in a matrix form. After taking into account the first measurement, the hypotheses matrix is a 4x1 matrix. Some lines of the hypothesis matrix are marked in red. These are hypotheses that don't contain compatible assignments (a track is assigned to more than one measurement), thus these lines (branches) of the hypotheses matrix (tree) have to be deleted.



0	0	0
1	0	0
2	0	0
3	0	0
0	2	0
1	2	0
2	2	0
3	2	0
0	4	0
1	4	0
2	4	0
3	4	0
0	0	2
1	0	2
2	0	2
3	0	2
0	2	2
1	2	2
2	2	2
3	2	2
0	4	2
1	4	2
2	4	2
3	4	2

0	0	5
1	0	5
2	0	5
3	0	5
0	2	5
1	2	5
2	2	5
3	2	5
0	4	5
1	4	5
2	4	5
3	4	5

Fig. 4. Matrix representation of the formed hypotheses

It has to be mentioned that according to the association that has been considered as correct, the state estimation of the a track is not the same in different hypothesis. For example, track 2 estimates in hypotheses [2 4 5], [1 2 5] and [1 0 5] are different since in each case the track is updated with a different measurement.

2.1 Implementation example

In Fig. 5 a basic architecture of the MOMHT is given. This section will give an overview of the basic processing steps of the method.

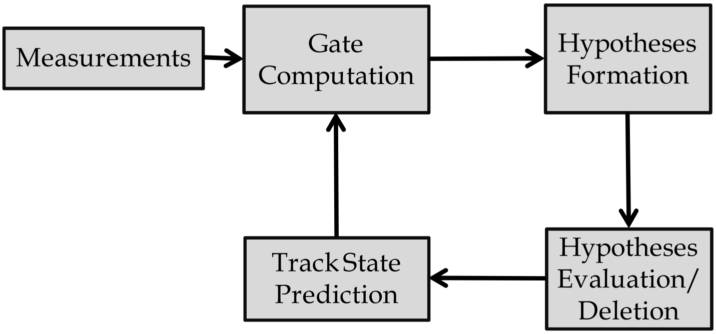


Fig. 5. MOMHT basic architecture

Assuming that the laserscanner processing in scan k provides M possible objects each having a measurement vector z_m, $m = 1...M$. Each hypothesis contains a set of tracks $x_i(k)$, $i = 1...N$, with respective covariance $P_i(k)$. The criterion that a measurement z_m is inside the track gate of size G is based on the innovation vector $z_m - H \cdot x_i$, where H is the measurement matrix:

$$(Z_m - Hx_i)^T B^{-1} (Z_m - Hx_i) \leq G \tag{1}$$

where $B = HP_iH + R$ and R is the measurement noise covariance matrix.

Then the hypothesis formation step is taking place. The gating procedure indicated the compatible track-measurement pairs of the current hypothesis. From an implementation point of view each line of the hypothesis matrix of scan k-1, which represents a hypothesis, is expanded into a new sub-matrix. This recursive procedure is repeated for each line of the hypothesis matrix resulting in the hypothesis matrix of scan k (Fig. 6). The new sub-matrix is expanded by inserting a new column until each measurement is added. When a new measurement is taken into account all the possible origins of this measurement will form the corresponding hypotheses. In any case, each measurement will form two hypotheses, one for assuming the measurement as false alarm and one considering the measurement as new target. Then, according to the gating results, additional hypotheses will be formed in case the measurement is inside one or more track gates. In the first scan, where there are no prior tracks, measurements are considered only as false alarms and new tracks. Attention has to be made that the associations within the hypothesis should be compatible. As a result a track cannot be assigned to two measurements in the same hypothesis.

Gating can be a time consuming process since a measurement set has to be check with the track set of each hypothesis. If the statistical distance calculation involves a matrix inversion, the gate calculation can cost a great part of the total MHT calculation time. During the implementation, the least complex distance measure should be chosen (for example Euclidean), that delivers the required results. In addition, distance measure calculation is affected by the sizes of the used matrixes. For example a four state track vector takes less time for gate computation than a six state vector.

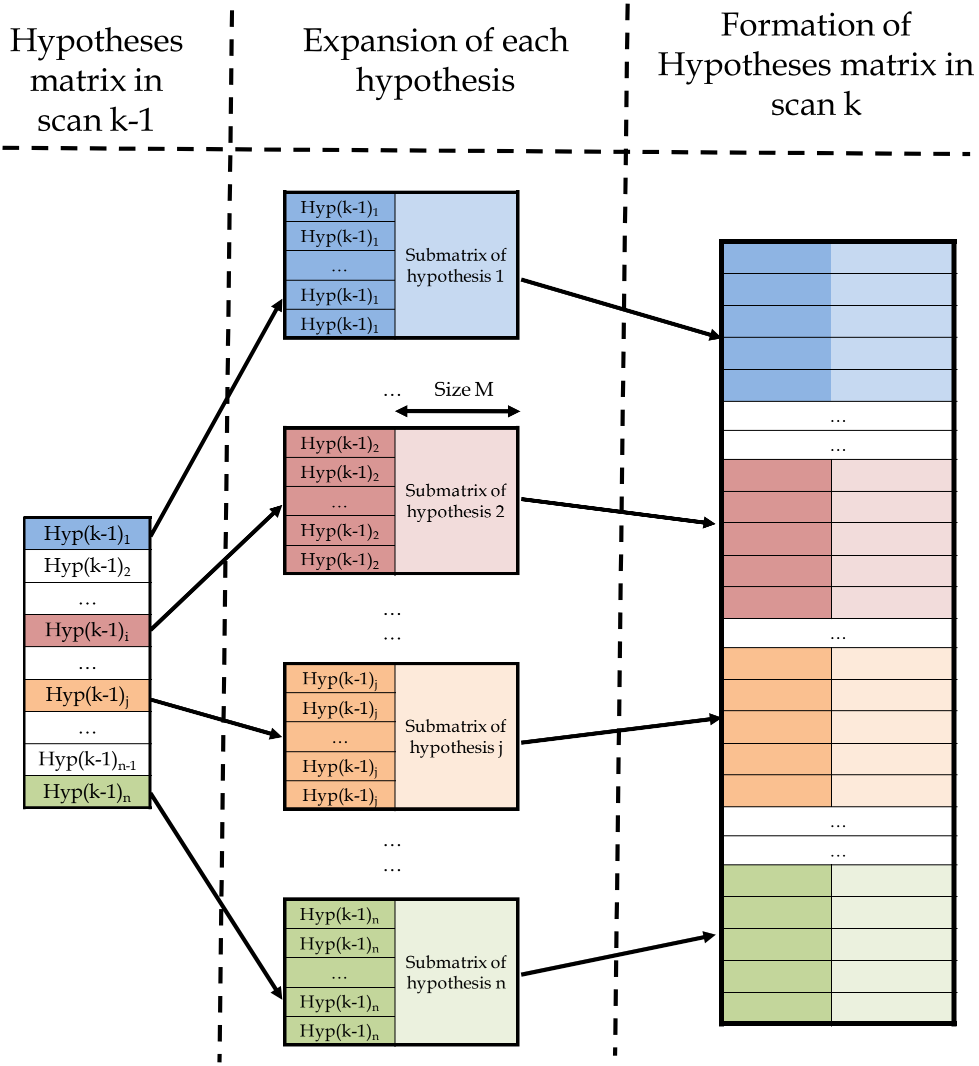


Fig. 6. Expansion of the hypotheses matrix

In parallel to the hypothesis formation, the track states are also updated. If a track has been associated with a measurement in the current scan, the state update is performed using the Kalman filter:

$$\hat{x}(k) = \bar{x}(k) + K[z(k) - H\bar{x}(k)] \tag{2}$$

$$\hat{P}(k) = \bar{P} - \bar{P}H^{T}\left(H\bar{P}H^{T} + R\right)^{-1} H\bar{P} \tag{3}$$

where K is the Kalman gain. As it has already been mentioned, a track with the same ID will have different state vectors in each hypothesis since it has been associated with a different measurement. In addition to the track state update, the hypothesis probability is also calculated and is described in section 4.2.

假设修剪

The next step of the algorithm is the reduction of the number of hypotheses. Since this method uses all the possible assignment possibilities, many of the hypotheses will have a very low probability and will be deleted after they have been created. There is a variety of hypotheses reduction techniques which will be discussed in Ch. 5.

回溯所有假设，决定输出哪些假设为航迹

An important aspect of the algorithm is to output a file of confirmed tracks. As already mentioned the essence of MHT is to resolve assignment ambiguities in the future. Usually the number of scans that hypotheses are propagated is fixed, called also as window size. This means that we go a number of scans back in the hypothesis tree and make a decision about the origin of the measurement. Assuming we have a window size equal to 3, on scan N we go back to the columns of the hypothesis matrix corresponding to scan N-2. Each column represents the possible origins of the measurement. If the entries in a particular column are the same, this means that the measurement has a unique origin, so the track that is associated with this measurement has a probability equal to one and can be considered as confirmed. If the measurement does not have a unique origin the authors use the following procedure. In the case mentioned there are two options: a) no track is confirmed from this particular measurement and we wait for future scans in order to confirm a target or b) if a track is more likely to be associated with the measurement then this track can be considered as confirmed. An illustrative example can be seen in Fig. 7. If measurement 2 has a unique origin, track 1 is confirmed. On the other hand if there are two assignment options with probabilities P_1 and P_2, depending on the implementation neither track 1 or track 2 are confirmed or if $P_2 > P_1$ and $P_2 > P_{threshold}$ track 2 is confirmed. Usually, it is useful to keep a list which indicates the hypotheses that contain each track. In this way, the track probability and state vector can be retrieved from each hypothesis.

After track confirmation the question is which is the track state and covariance that will be given to the output. The simplest method is to take the most probable hypothesis and use the state vector from this particular hypothesis. Another option is to combine the estimates, by using for example JPDA or a simple weighted sum using track probabilities, in order to combine the estimates from all the hypotheses that contain the track. In addition to the track confirmation, the implementation should incorporate a track deletion scheme. As mentioned a track's probability is the sum of the probabilities that contain this track. So if a track has a low probability it should be deleted from all the hypotheses. An additional heuristic measure is to delete a track if it has no associations for a continuous number of scans.

As it can be seen the track management scheme is based on the calculation of a probability of each hypothesis. The correct tuning of the parameters in the probabilistic expressions and the right choice in deletion and confirmation thresholds is crucial. Otherwise, wrong parameters can result in premature or delayed track confirmation and deletion. In addition, the propagation of a large number of hypotheses and tracks as a result of bad parameterization can compromise the real time efficiency of the algorithm.

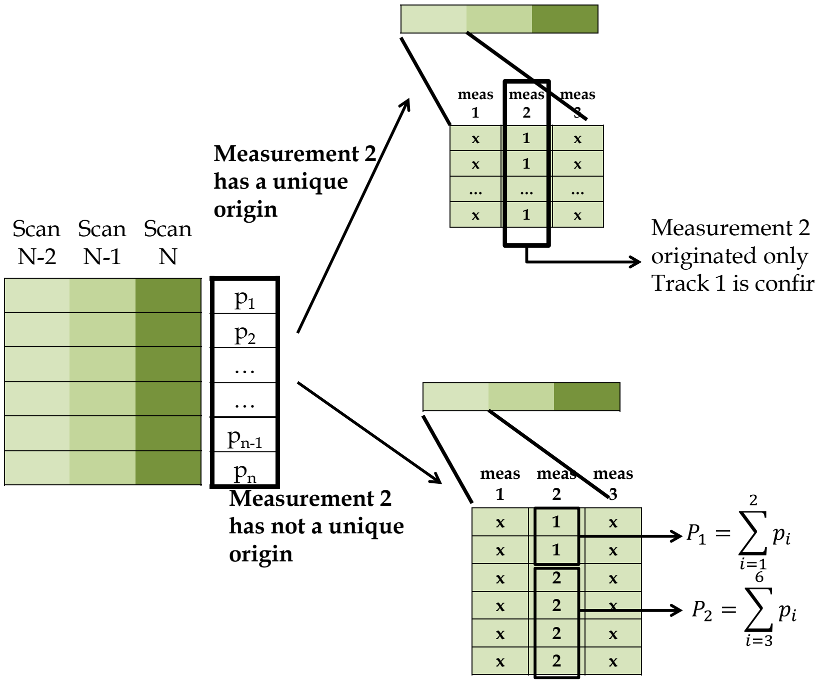

Scan N-2
Scan N-1
Scan N
p_1
p_2
…
…
p_{n-1}
p_n
Measurement 2 has a unique origin
meas 1
meas 2
meas 3
x 1 x
x 1 x
… … …
x 1 x
Measurement 2 from scan is originated only from track 1
Track 1 is confirmed
Measurement 2 has not a unique origin
meas 1
meas 2
meas 3
x 1 x
x 1 x
x 2 x
x 2 x
x 2 x
x 2 x
$P_1 = \sum_{i=1}^{2} p_i$
$P_2 = \sum_{i=3}^{6} p_i$

Fig. 7. Selection of the best measurement- track association

3. Track oriented MHT

Track oriented MHT approach was introduced in (Y.-B. Shalom 1990) and in contrary to the hypothesis oriented approach, it does not maintain hypothesis from scan to scan. When a new measurement set is received, the existing track set is formed into an expanded new track set using all the possible assignments. This method does not maintain a hypothesis set between scans, but it maintains a set of tracks which are not compatible. This set of incompatible tracks is used to form the hypotheses.

In order to point out the differences with HOMHT, the hypothesis formation method will be explained with the same example given in section 2. The original track set contains two tracks. These tracks are represented by the two leftmost trees in Fig. 8. The nodes are numbered according to the measurement that the track has been associated in the current scan. Let's assume that Track 1 is associated with measurement 2 and Track 2 is associated with measurement 1 at the previous scan. Recalling the gating results (Fig. 2), there are 4 possible assignments for Track 1. Assignment to the dummy measurement is indicated with 0 and declares that the track is not assigned to any measurement in the current scan. A matrix representation of the formed tracks is depicted in Fig. 9. The difference with the hypothesis oriented approach is that each track is represented by a tree, and the alternative association hypotheses are represented by the different nodes in the target tree. Each node within the tree is incompatible with all the other tracks of the tree, so the set of tracks in the same tree can represent one target at the most.

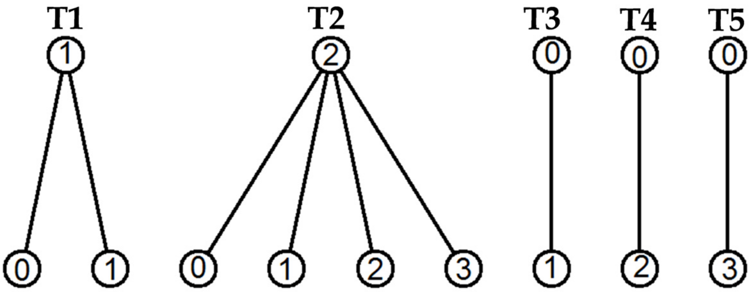


Fig. 8. Tree representation of he formed tracks

0	1
0	2
0	3
1	0
1	1
2	0
2	1
2	2
2	3

Fig. 9. Table representation of the formed tracks

3.1 Implementation example

In this section a general description of the basic TOMHT steps will be described. An overview of sample architecture is depicted in Fig. 10. This is a sample logic and the sequence can be changed or more blocks may be added according to the implementation.

As in section 2.1, the first step of the tracking algorithm is the gate computation. The gates are calculated using for example equation(1). All the possible assignments result in the formation of new tracks or updates of existing ones. The target tree expansion begins with the assumption that tracks are updated with the dummy measurement, meaning that the track is not updated with any measurement (indicated as node "0" in the target tree). Additionally all measurements spawn a new track. Then all the possible measurements that can be associated with a track are used to update the track state. This is indicated as different branches that are formed in the target tree. Taking as example track T2 in Fig. 8, the track may be assigned to the dummy measurement and also to measurements 1, 2 and 3. In parallel to the track formation step, track probabilities are also calculated (sect. 4.2).

The next step is the reduction in the number of tracks. Each track is assigned with a probability. Tracks that have a probability lower than a defined threshold will be deleted. Additionally similar tracks (sec. 5.2) can also be merged, thus the overall number of tracks can be further reduced

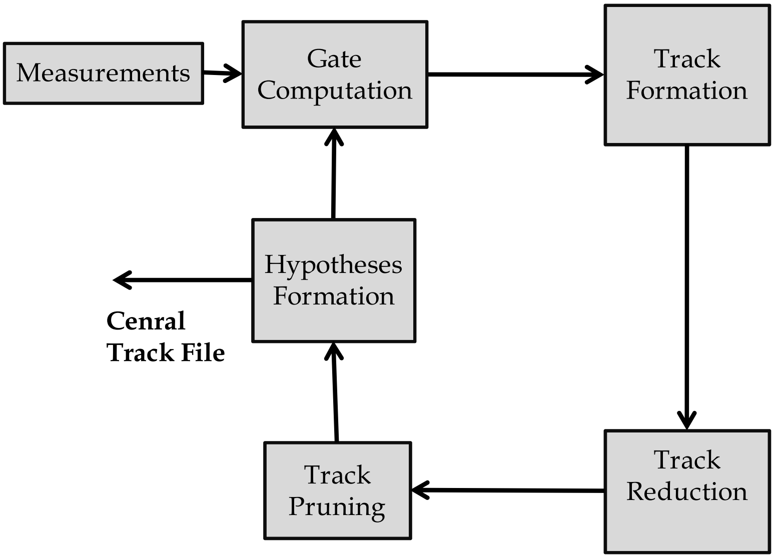


Fig. 10. Architecture of TOMHT algorithm

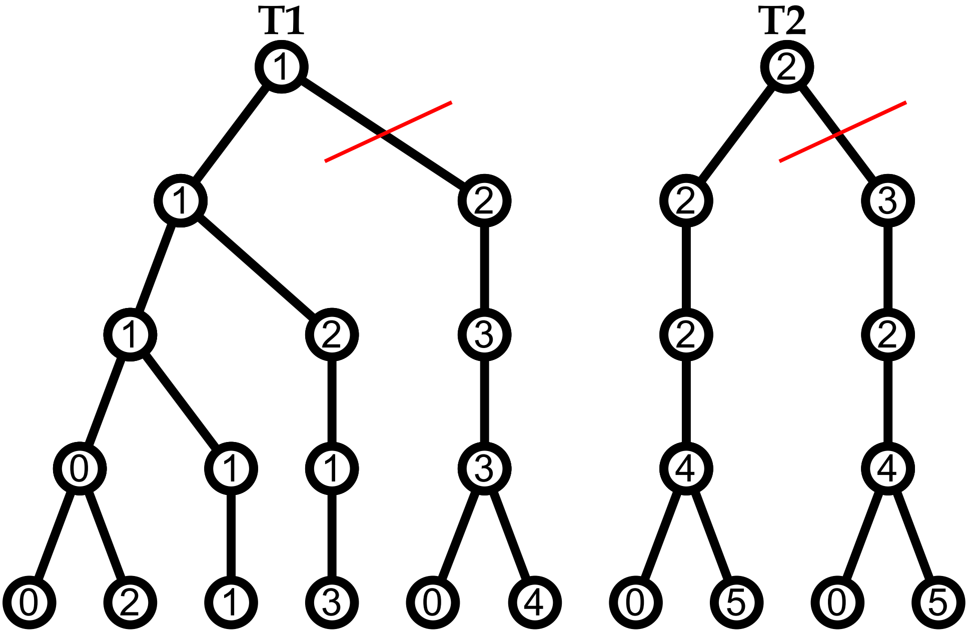
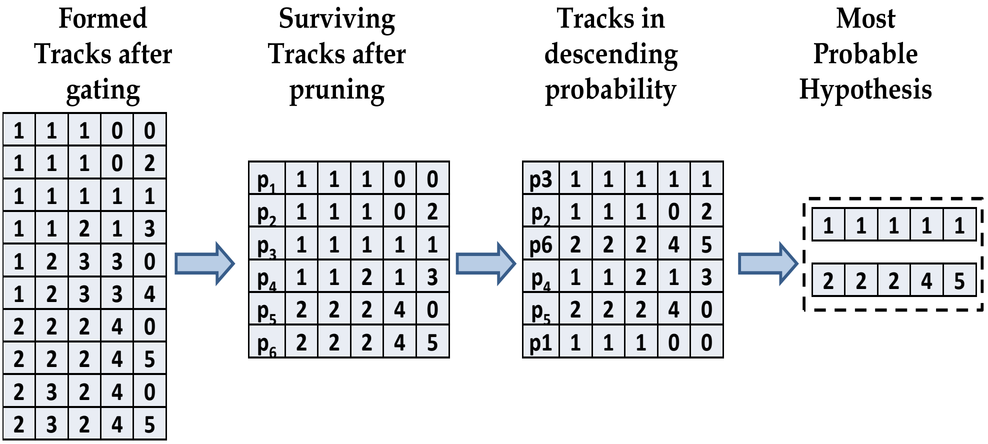


Fig. 11. Target tree pruning after solving the multi-dimensional assignment problem

After the track reduction step, the target trees have to be pruned. Similarly to HOMHT we have to go back a number of scans in the past and decide which track-measurement assignment is correct. For the rest of the section the following example will be used. Let's assume we have a 5 scan window and the track hypotheses trees are formed as illustrated in Fig. 11. In this particular association problem we have to decide if T1 is associated with measurement 1 or 2 in scan N-3 and whether T2 is associated with measurement 2 or 3. Assuming that the correct track pairs are T1-M1 and T2-M3, all tracks that do not stem from these associations will be deleted. In order to perform the tree pruning, the multi-dimensional assignment problem has to be solved. A possible solution that also has been used by the authors is the use of Lagrangian relaxation algorithm (Fisher 1981). The output of the algorithm will be a set of S-tuples that represent a set of compatible tracks. In our example the two 5-tuples will be [1 1 1 1 1] and [2 2 2 4 5]

The final step of the presented TOMHT architecture is the formation of the most probable hypothesis. The number of trees indicates the maximum number of possible tracks. The total number of tracks is equal to the number of nodes in the target tree that correspond to the last scan or equal to the number of rows in the hypotheses matrix. As mentioned before, this set of tracks is incompatible. In order to form the most probable hypothesis that contains a set of compatible tracks, the tracks are arranged into descending probability order. The most probable track is added to the hypothesis. Then, the next most probable track is added. This track should be compatible with the rest of the track in the hypothesis. This procedure is continued until there are no more compatible tracks that can be added to the hypothesis. The steps of composing the most probable hypothesis in our example are depicted in Fig. 12. Apart from the best hypothesis, the m-best hypotheses may also be formed (Popoli et. al.,2001;Fortunato et. al., 2007). Tracks that are not contained in the m-best hypotheses can also be deleted.



Formed Tracks after gating

1	1	1	0	0
1	1	1	0	2
1	1	1	1	1
1	1	2	1	3
1	2	3	3	0
1	2	3	3	4
2	2	2	4	0
2	2	2	4	5
2	3	2	4	0
2	3	2	4	5

Surviving Tracks after pruning

p_1	1	1	1	0	0
p_2	1	1	1	0	2
p_3	1	1	1	1	1
p_4	1	1	2	1	3
p_5	2	2	2	4	0
p_6	2	2	2	4	5

Tracks in descending probability

p3	1	1	1	1	1
p_2	1	1	1	0	2
p6	2	2	2	4	5
p_4	1	1	2	1	3
p_5	2	2	2	4	0
p1	1	1	1	0	0

Most Probable Hypothesis

1	1	1	1	1
2	2	2	4	5

Fig. 12. Hypothesis formation in TOMHT

4.1 Window size

An important parameter of the MHT algorithm is the number of scans that the track history is kept, called also the window size. The larger the window size, the larger the number of hypotheses that are created. If the window size is N and $\mathrm{n_i}$ is the number of measurements in scan i, the tree depth in HOMHT will be $\sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{N} \cdot \mathrm{n_i}$. TOMHT consists of multiple trees, each one having depth equal to N. Window size is a design parameter and should be chosen so that it is large enough to resolve assignment ambiguities but also not result in an unmanageable number of hypotheses.

4.2 Hypotheses probability

As mentioned, for each hypothesis a probability has to be calculated. In Reid's original work the probability P_i^k of hypothesis $\Omega_\mathrm{i}^\mathrm{k}$ formed from hypothesis $\Omega_\mathrm{i}^{\mathrm{k}-1}$ is calculated as follows:

$$\begin{aligned} P_i^k = \frac{1}{c} P_D^{N_{DT}} \left(1 - P_D\right)^{\left(N_{NGT} - N_{DT}\right)} \beta_{FT}^{N_{FT}} \beta_{NT}^{N_{NT}} \\ \times \left[\prod_{m=1}^{N_{DT}} N\left(Z_m - H\bar{x}, B\right) \right] P_g^{k-1} \end{aligned} \tag{4}$$

Where:

P_D = Probability of detection

β_{FT} = Density of false targets

β_{NT} = Density of previously unknown targets that have been detected

N_{DT} = Number of measurements associated with prior targets

N_{FT} = Number of measurements associated with false targets

N_{NT} = Number of measurements associated with new targets

$Z_m - H\bar{x}$ and B are the innovation vector and innovation covariance matrix.

Log Likelihood Ratio (LLR) is a more convenient value to be calculated. Many formulas exist for the calculation of this value. The basic advantage is that if $L(k-1)$ is the LLR of track at scan $\mathrm{k}-1$ the LRR $L(k)$ at scan k will be (Blackman 2004):

$$L\ (k) = L\ (k-1) + \Delta L(k) \tag{5}$$

where $\Delta L(k)$ can be calculated using the original work in (Sittler 1964) or expressions like (Bar-Shalom et. al., 2007). Another advantage of LLR is that is a dimensionless quantity and it can also be easily converted to the probability P_T of a true target

$$P_T = e^{LLR} / \left[1 + e^{LLR}\right] \tag{6}$$

5. Hypothesis and track management

Multiple Hypothesis Tracking can result in an exponential increase of hypotheses, making this approach impossible without the use of hypotheses reduction techniques. In this section a number of techniques regarding hypotheses and track management will be presented, including some practical application examples.

5.1 Hypotheses clustering

Clustering is a well explored topic in computer science (Bouguettaya & Le-Viet, 1998; Steinbach et. al., 2000) whereas it has also been applied to target tracking (Czink et. al., 2006). Reid had indicated the need of a clustering scheme in order to improve the efficiency of his HOMHT algorithm implementation (Reid 1979). As a result, clustering was integrated into the HOMHT algorithm architecture

Clustering is not a direct way of reducing hypotheses number in HOMHT. However it presents a way of dividing a large tracking problem into smaller tracking sub-problems. Each cluster consists of a separate hypothesis tree, so the number of hypotheses that are managed in each cluster is smaller than it would be in the case that the tracking problem was handled in one hypothesis tree. For example, it is not necessary to gate a measurement with all the tracks, but only to those tracks that belong to a particular cluster. In addition, with the use of multi core processors, it is possible to process each cluster separately.

A cluster management algorithm for MHT mainly consists of three sub-routines: cluster initialization, cluster merging and cluster splitting. Cluster management should not allow clusters to grow uncontrollably. In this case, the additional processing overhead of cluster management algorithm will not be covered but the gains from the reduction in MHT processing time.

The first step is to distribute the measurements into existing clusters. For every measurement i the distance $\mathrm{d_{ij}}$ from each cluster j is calculated. The calculation of the distance can be the defined as the statistical distance between the measurement and the cluster centroid for example. If p_i and x_i are the probability and the estimate of track i in the cluster, the cluster centroid can be defined as follows:

$$x_c = \frac{\sum_i p_i x_i}{\sum_i p_i} \tag{7}$$

Alternatively, $\mathrm{d_{ij}}$ can be considered as the closest distance between the measurement and the cluster's tracks. If $\mathrm{d_{ij}}$ is smaller than a threshold value then the measurement is assigned to the particular cluster. If the measurement is assigned to more than one cluster, these clusters are merged, a process that will be managed by the cluster merging routine. Finally, if the measurement is not assigned to any cluster, a new cluster is initialized.

Cluster merging is the second routine of the clustering algorithm. If a measurement is associated with two clusters, these two clusters have to be merged. The merging method should combine the hypotheses that are contained in the two clusters. This involves the combination of the tracks and measurements that are contained in these two separate clusters. Additionally, a combined hypothesis matrix and tree has to be formed while the combined hypotheses probabilities have to be calculated. Assuming we have the case of Fig. 13 where clusters A and B have to be merged. Cluster A has a two row hypothesis matrix whereas cluster B has a three row matrix. Numbers above each column represent the scan of the corresponding measurement. The final matrix combines the rows and columns of the initial matrixes, whereas the probability of the merged hypothesis will be the product of the probabilities of the merged hypotheses.

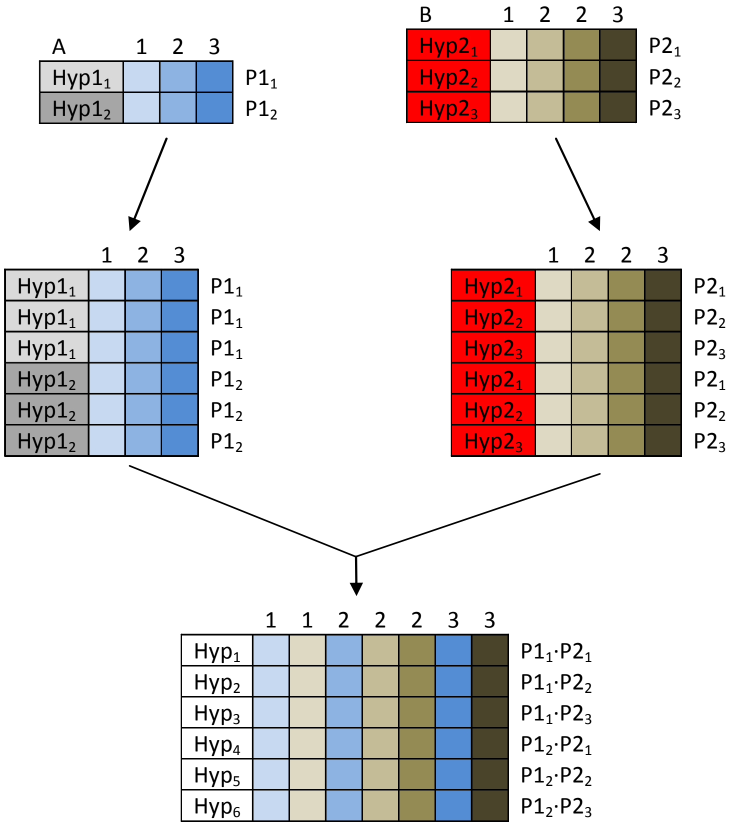


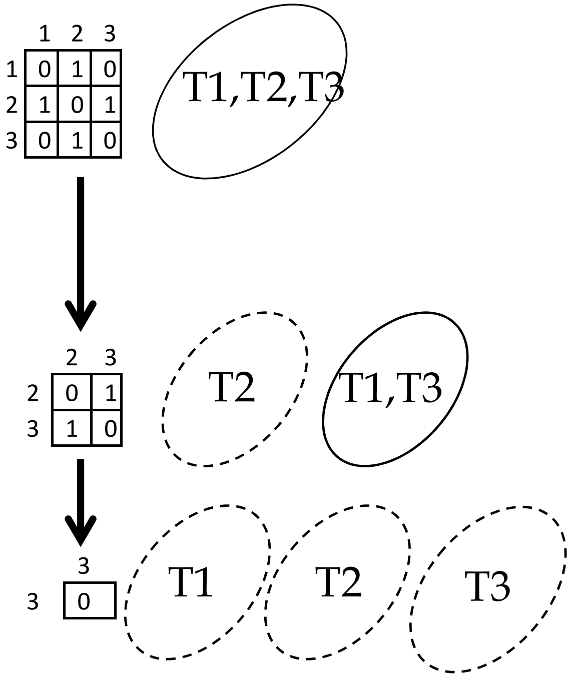
Fig. 13. Merging of two hypotheses matrixes

Cluster splitting is the process where one track of a cluster is removed from this cluster and initiates a new cluster. For this purpose we have to set a similarity criterion for tracks within the cluster. A simple method is to use the Euclidian or Mahalanobis (8) distance in order to calculate the distance d_{ij} between the tracks.

$$d_{ij}^2 = \left(\hat{x}_i - \hat{x}_j\right)^T \left[\left(P_i^T + P_j^T\right)^{\mathrm{T}}\right]^{-1} \left(\hat{x}_i - \hat{x}_j\right) \tag{8}$$

where $\hat{x}_i$, P_i and $\hat{x}_j$, P_j are the state estimate and covariance of tracks i and j respectively.

If this distance is greater than a define value, these tracks are considered to be candidate for splitting. Let's assume we have 3 tracks, we form a 3x3 matrix which contains binary values. A value of "1" at entry (i,j) indicates that these two tracks are "far away" so they can be assigned to different clusters. A value of "0" indicates the opposite. This example is illustrated in Fig. 14. All the columns are examined sequentially, if the sum of the column is greater than zero, then this track is removed, along with the corresponding row and matrix, and a new cluster is initiated. In addition, all the hypotheses that contained the particular track are removed from the hypotheses matrix.



$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{array}$$

$$\begin{array}{c|cc} & 2 & 3 \\ \hline 2 & 0 & 1 \\ 3 & 1 & 0 \end{array}$$

$$\begin{array}{c|c} & 3 \\ \hline 3 & 0 \end{array}$$

Fig. 14. Cluster splitting procedure

A final routine that a clustering algorithm should include is the cluster deletion. Clusters that do not contain any tracks should be deleted. Similar clustering strategy applied to tracks and not hypotheses can be used also in TOMHT.

5.2 Hypotheses reduction

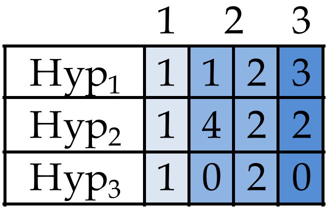
In this section some methods for hypotheses reduction will be presented. As it has been described hypotheses are reduced when a decision for an assignment in the past is made.

In addition, hypotheses deletion is an efficient measure for keeping algorithm calculation time under control. Hypotheses reduction techniques for HOMHT and TOMHT will be described separately.

5.2.1 Hypotheses reduction in HOMHT

The simplest way of reducing the number of hypotheses is to delete those hypotheses that have a probability below a certain threshold. Since HOHMT is an exhaustive method that enumerates all the possible assignments, a large part of those assignments will be highly unlikely, so they can be deleted. However, this presents the danger of deleting some useful hypotheses also. For example when a track has just been created, the hypotheses that contain it will have a low probability. Some of these hypotheses have to survive deletion and wait for the next scans in order to calculate whether the probability will be increased or not. A useful practice according to authors is that each column in the hypotheses matrix that corresponds to the last scan should contain not only "0" but at least one track number. As a result, we will allow alternative hypotheses for this measurement to be propagated and when we decide about the origin of the measurement the correct assignment will be made. Of course this can result in an unacceptable high number of hypotheses. Authors suggest that a maximum number of hypotheses and tracks should be set in the algorithm. If the hypotheses or track number exceeds these thresholds, additional hypotheses deletion should take place. This will ensure that the system will always be able to run real time, even if tracking performance is sacrificed. Apart from hypotheses, the number of tracks that are contained in the hypotheses can be reduced. A track probability is the sum of hypotheses probabilities that contain it. Low probability tracks are deleted from the hypotheses that contain them.

Apart from hypotheses deletion, similar hypotheses can be merged in order to reduce the total number of hypotheses. Two hypotheses are merged if they satisfy one or more similarity criteria. In general, two hypotheses can be merged if they contain at least the same number of tracks and these tracks are "close enough". Assuming we have 3 hypotheses as indicated in Fig. 15. Hyp$_1$ and Hyp$_2$ contain 2 tracks whereas Hyp$_3$ contains 2 tracks.



	1	2		3
Hyp$_1$	1	1	2	3
Hyp$_2$	1	4	2	2
Hyp$_3$	1	0	2	0

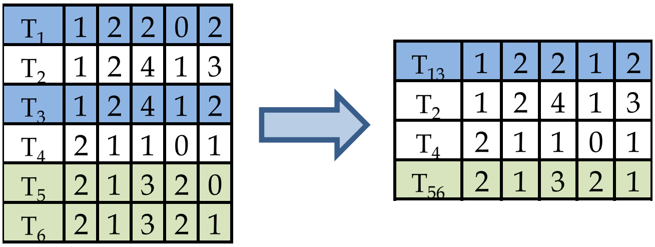
Fig. 15. Hypotheses merging example

The first two hypotheses are candidate for merging. The distance measure $d_{ij}, i = j = 1...3$, between the tracks of the two hypotheses is calculated. If all the tracks of the first hypotheses have a distance from a track of the second hypotheses below a defined threshold, these two hypotheses can be merged. The probability of the merged hypotheses will be the sum of the individual hypotheses probabilities. The estimates of the merged tracks can be calculated using JPDA which results in a weighted sum of the individual estimates.

5.2.2 Track reduction in TOMHT

As in MOMHT, a first step is to delete low probability tracks. In TOMHT, low probability tracks can also be deleted before and after the hypothesis formation step. Deletion thresholds should be adapted carefully since new tracks have a low probability; it may happen that hypotheses which contain new tracks may be deleted. A useful practice is to have each measurement of the current scan associated with at least one track. In this way, if the measurement represents a real target, the track will have an increased probability after the next scan, otherwise it will be deleted. On the other hand, tracks with a long history have relatively high probabilities, so a large number of "old" tracks may describe the same target

Similar tracks can also be merged. Again, a similarity criterion has to be set. For example, two tracks can be merged if they share N measurements in the past and are closer than a defined threshold are merged. If one track is associated with a measurement and another track is associated with the dummy measurement, these two tracks are considered as compatible in the particular scan. In Fig. 16, for example tracks 1 and 3 are considered as compatible, so they can be merged. The same applies to tracks 5 and 6. On the other hand, tracks 2 and 3 are not compatible since they are associated with a different measurement in the last scan.

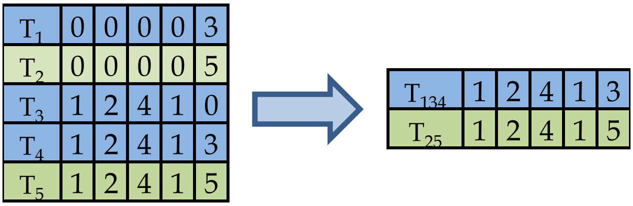


T_1	1	2	2	0	2
T_2	1	2	4	1	3
T_3	1	2	4	1	2
T_4	2	1	1	0	1
T_5	2	1	3	2	0
T_6	2	1	3	2	1

T_{13}	1	2	2	1	2
T_2	1	2	4	1	3
T_4	2	1	1	0	1
T_{56}	2	1	3	2	1

Fig. 16. Track merging example

A possible danger in track merging is the situation depicted in Fig. 17. If the similarity criteria are satisfied, the five tracks will be merged into two tracks. These two tracks are not compatible, so in the track extraction process only one track is outputted. However the correct hypothesis may be that there are two real targets represented by measurements 3 and 5 of the last scan. This example is used in order to show that the similarity criteria have to be set carefully or that the necessary measures have to be taken so that enough tracks are maintained.



T_1	0	0	0	0	3
T_2	0	0	0	0	5
T_3	1	2	4	1	0
T_4	1	2	4	1	3
T_5	1	2	4	1	5

T_{134}	1	2	4	1	3
T_{25}	1	2	4	1	5

Fig. 17. Track merging resulting in one track

6. Results

In this section some indicative results of MHT implementation will be presented in order to demonstrate the performance gains of the MHT algorithm.

In the first set of tests, five scenarios recorded from an IBEO LUX laserscanner system in highway scenarios were used. The scenarios differed in the amount of traffic that was encountered. Theses series of tests were conducted in order to measure the savings in execution time after applying a clustering algorithm in HOMHT. The sensor setup is illustrated in Fig. 18. Laserscanner network topology. The laserascanner raw data are processed by a fusion module (Fig. 19) whose output consists of a list of detected objects. This list is then given as input to the HOMHT algorithm.

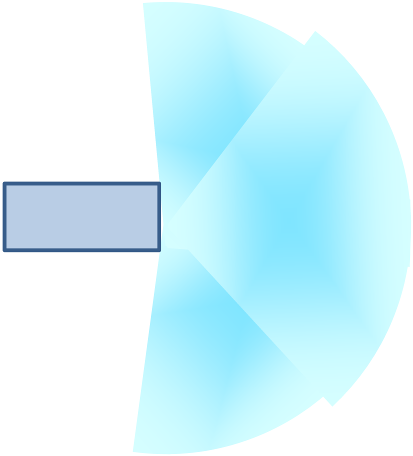


Fig. 18. Laserscannner network topology

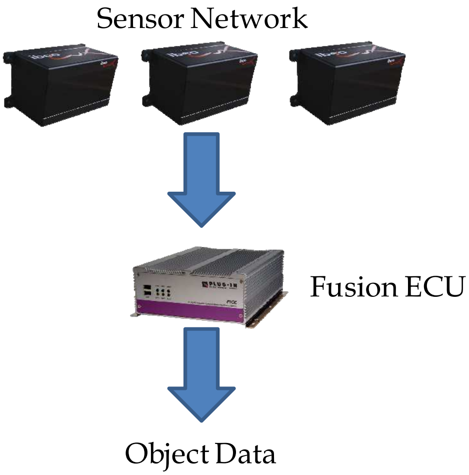


Fig. 19. Laserscannner processing architecture

Two algorithms were implemented in a C++ environment, the first one without clustering and the second one with the clustering scheme presented in section 5.1. The use of a profiling software made possible to extract the calculation times indicated in Table 1

The results indicated that the average execution times can be improved by almost 92%. Of course, this improvement heavily depends on the sensor used and also by the scenario. However, since a multilayer laserscaner network has a great range, it is not unusual to receive up to 40 targets in a highway scenario, or even double this number in urban environments. As it can be seen, the algorithm without clustering is impossible to run real time, since the calculation time exceeds the refresh rate of a usual sensor. It has also to be mentioned, that the tracking performance of the two algorithms does not differ. Clustering does not deteriorate at all the performance gains of the MHT technique.

	No Clustering			Clustering			Difference		
	Total	Mean	Max	Total	Mean	Max	Total	Mean	Max
Scen 1	5437	3	199	1573	1	9	**71.08%**	**71.03%**	**95.46%**
Scen 2	20311	20	447	1618	2	7	**92.04%**	**91.98%**	**98.42%**
Scen 3	51100	51	21127	17447	18	80	**65.86%**	**65.86%**	**99.62%**
Scen 4	81740	47	37345	25633	14	150	**68.64%**	**69.62%**	**99.60%**

Table 1. Execution times in milliseconds for the MHT algorithm with and without clustering

Another interesting test was to measure the difference in calculation times by changing the width of the observation area. For a given scenario, only the measurements that had a lateral distance below a certain limit were given as input to the MHT. In this way, it is possible to measure, roughly, how the calculation times increase as the number of measurements increase. The comparison chart is depicted in Fig. 20. This chart shows that execution time of a Multiple Hypotheses Tracker that incorporates clustering processing steps is increased by far less compared to an MHT without clustering. However, efficiency of the clustering mainly depends on the density of the measurements.

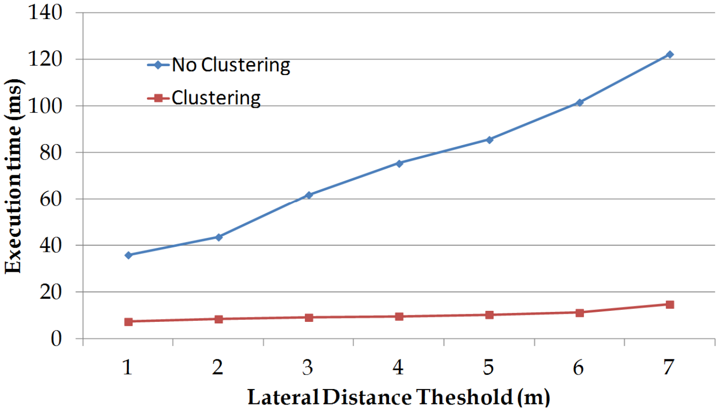


Fig. 20. Execution time vs observation area chart

In order to test the performance of the HOMHT algorithm and identify the benefits over the conventional GNN, a series of simulations of a typical driving scenario were performed. The scenario to be examined is: two parallel moving targets. In simulated measurements (122 in total), a variable random Gaussian noise was added, with standard deviation varying from 0.1 to 2 meters. In addition, random clutter (40% of the total measurements) was added, simulating measurements from non-vehicle objects. The measures that were calculated are i) ID losses, ii) Correct Associations ratio, iii) False Alarms ratio and iv) Estimation error. Tracking results for both methods are shown in Fig. 21 and Fig. 22. These figures were extracted from the scenario with measurement noise of 1.2m. Red circles correspond to true observations while black squares represent measurements originated from random clutter. The estimated position for each target is shown with colourful crosses. Consecutive crosses of the same colour correspond to the same track ID. The plots clearly show that MHT maintains the track throughout the scenario while GNN fails to provide a good tracking result. A detailed analysis of these scenarios and results can be found in (Spinoulas 2010) and (Thomaidis et. al.,2010).

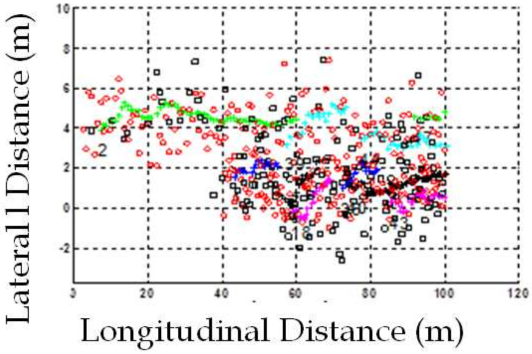


Fig. 21. Tracking output of GNN

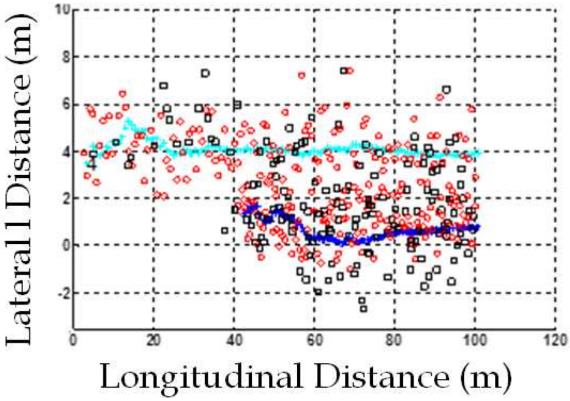


Fig. 22. Tracking output of MHT

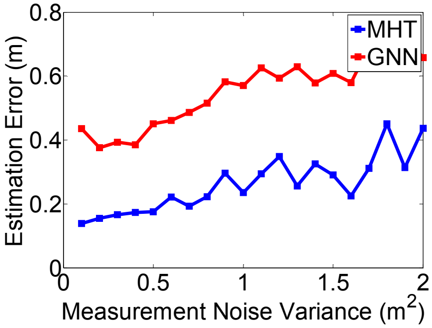


Fig. 23. Estimation error of MHT and GNN

In Fig. 23, the estimation error of both algorithms is plotted. As it can be seen MHT provides more accurate estimation than GNN. This is explained by the fact that MHT has better association results; it associates more often a track with the correct measurement. If a track is updated with the wrong measurement, then the track estimation will degrade or the track may be lost at all.

Fig. 24 presents the percentage of correct associations and the ID changes of the two algorithms. Correct associations indicate the percentage of scans that a track was associated with the correct measurement. ID changes present the number of times that the ID of the track was changed; indicating the number of times that a track was lost and re-initialized. As it can been seen, MHT has a higher percentage of correct associations over GNN. In addition the two algorithms have a degrading performance as noise levels increase. The second plot of Fig. 24. indicates that MHT has less track losses than GNN method.

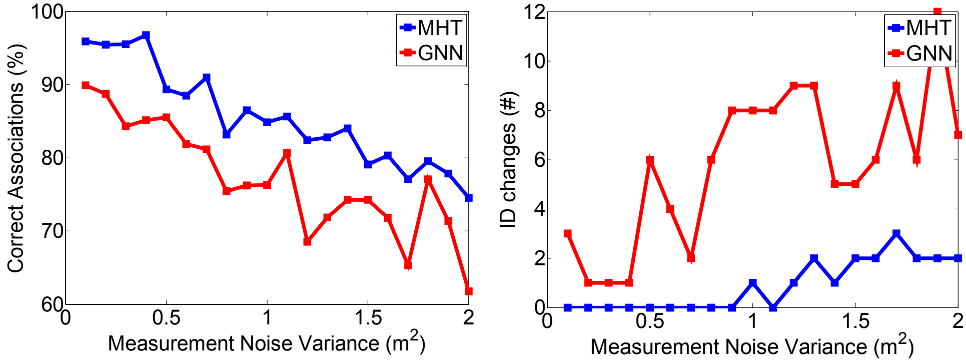


Fig. 24. Comparative results of MHT and GNN

7. Conclusions

This chapter presented practical aspects in the implementation of a Multiple Hypothesis Tracker. The two basic algorithmic approaches where presented, but of course alternative algorithms such as the Bayesian MHT (Koch 1996) have been proposed.

The MHT tracking algorithm yields better results than other methods which propagate only one association hypotheses. Undoubtedly this performance advantage comes at a cost. This method is much more complicate in the implementation than other tracking methods since it involves many processing steps. In addition each step can be implemented in a variety of ways. The propagation of many hypotheses as it has been mentioned also implies increased needs in terms of computational time. However, with the advent of modern fast CPUs and the use of optimization methods like the ones presented in this chapter, the implementation of a real time MHT is feasible.

8. Acknowledgment

The authors would like to thank Leonidas Spinoulas for his contribution in the work presented in this chapter.

9. References

Blackman, S., Multiple Hypothesis Tracking for multiple target tracking, *IEEE Trans. Aerospace and Electronic Systems,* vol. 19, no.1, pp. 5-18, 2003

Bouguettaya, A. Le-Viet, Q. Data Clustering Analysis in a Multidimensional Space, *International Journal on Information Sciences*, Vol 112, Iss. 1-4, pages 267-295. 1998.

Bar-Shalom, Y. (Ed.)(1990), *Multitarget-Multisensor Tracking: Advanced Applications*, ArtechHouse, 096483122, Nonvood, MA.

Bar-Shalom, Y. Blackman, S., Fitzgerald, R. J. Dimensionless Score Function for Multiple Hypotheses Tracking, *IEEE Trans. Aerospace and Electronic Systems* vol. 49, no.1, pp. 392-400, 2007

Blackman, S. Popoli, R.(1999), *Design and analysis of modern tracking systems*, Arthech House,1580530060, Nonvood, MA.

Czink, N. Del Galdo, G. Mecklenbräuker, C. A novel automatic cluster tracking algorithm, *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Helsinki, Finland. 11.09.2006 - 14.09.2006.

Fisher, M. L. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, vol. 27, no.1 pp. 1-18, 1981

Fortmann, T. Bar-Shalom. Y. Scheffe. M., Sonar tracking of multiple targets using jointprobabilistic data association, *IEEE Journal of Oceanic Engineering*, Vol. 8, Iss. 3,pp.173 - 184, July 1983

Fortunato, E. Kreamer, W. Mori, S. Chee-Yee, Chong. Castanon, G. Generalized Murty's Algorithm With Application to Multiple Hypothesis Tracking, *Proceedings of 10th International Conference on Information Fusion, 2007,* 978-0-662-45804-3, Quebec, 9-12 July 2007

Kock, W. Retrodiction for Bayesian multiple-hypothesis/multiple-target tracking in densely cluttered environment, *O.E. Drummond (Ed.) Proc of SPIE Signal and Data Processing of Small Targets*, pp.429-440, vol. 2759, May 1996

Popoli, R. Pattipati, K. Bar-Shalom, Y. m-Best S-D Assignment Algorithm with Application to Multitarget Tracking, *IEEE Trans. Aerospace and Electronic Systems.* vol. 37, no.1, pp. 22-39, 2001

Reid D. , An algorithm for tracking multiple targets, *IEEE Trans. on Automatic Control*, vol. 24, issue 6, pp. 423-432, Dec.1979.

Singer, R. Sea, R. Housewright K., Derivation and evaluation of improved tracking filter for use in dense multitarget environments , *IEEE Trans. Information Theory*, vol. 20, pp. 423–432, Jul. 1974.

Sittler, Robert W. An Optimal Data Association Problem in Surveillance Theory, *IEEE Trans. on Military Electronics*, vol. 8 iss.2, pp. 125 – 139, April 1964

Spinoulas, L. Object tracking using Multiple Hypothesis Tracking for road environment perception. *Diploma Thesis,* National Technical University of Athens, Feb. 2010.

Steinbach, M. Kapyris, G. Kumar, V. A comparison of Document Clustering Techniques. *Proc. of KDD-2000 Workshop TextMining*. Aug. 2000

Thomaidis, G. Spinoulas, L. Lytrivis, P. Ahrholdt, M. Grubb, G. Amditis, A. Multiple hypothesis tracking for automated vehicle perception, Proc. IEEE Intelligent Vehicles Symposium (IV), 2010, 978-1-4244-7866-8, San Diego, 21-24 June 2010

Laser Scanner Technology

Edited by Dr. J. Apolinar Munoz Rodriguez

ISBN 978-953-51-0280-9
Hard cover, 258 pages
Publisher InTech
Published online 28, March, 2012
Published in print edition March, 2012

Laser scanning technology plays an important role in the science and engineering arena. The aim of the scanning is usually to create a digital version of the object surface. Multiple scanning is sometimes performed via multiple cameras to obtain all slides of the scene under study. Usually, optical tests are used to elucidate the power of laser scanning technology in the modern industry and in the research laboratories. This book describes the recent contributions reported by laser scanning technology in different areas around the world. The main topics of laser scanning described in this volume include full body scanning, traffic management, 3D survey process, bridge monitoring, tracking of scanning, human sensing, three-dimensional modelling, glacier monitoring and digitizing heritage monuments.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821