Multiple Pedestrian Tracking using Viterbi Data Association

Asma Azim and Olivier Aycard

***Abstract*— To address perception problems we must be able to track dynamic objects of the environment. An important issue of tracking is the association problem in which we have to associate each new observation with one existing object in the environment. This problem is complex: unfortunately, the number of observations generally does not correspond to the number of objects. Moreover, the number of objects is difficult to estimate since one object might be temporarily occluded or unobserved simply because objects can enter or go out of ranges of vehicle sensors. Moreover, the perception sensors or the object detection process might generate false alarm measurements. In this paper, we propose a new solution to solve the multiple objects tracking problem, using the Viterbi algorithm (VA) [2]. It is an established optimisation technique for discrete Markovian systems that has been extensively used in speech recognition. In this paper, we present an extension of VA to solve multiple objects tracking in clutter environment and show some experimental results on multiple pedestrian tracking and also some quantitative comparisons with MHT algorithms.**

I. INTRODUCTION

To address perception problems we must be able to track dynamic objects of the environment. An important issue of tracking is association problem in which we have to associate each new observation with one existing object in the environment. This problem is complex: unfortunately, the number of observations generally does not correspond to the number of objects. Moreover, the number of objects is difficult to estimate since one object might be temporarily occluded or unobserved simply because objects can enter or go out of ranges of vehicle sensors. Moreover, the perception sensors or the object detection process might generate false alarm measurements.

The data association for multi-target tracking consists in deducing the number of true objects and identifying if each observation corresponds to an already known object being tracked, to a spurious measure or to a new object in the scene that will be tracked. The complexity to solve data association grows exponentially with the number of targets in the scene.

In the literature, data association algorithms are often categorized according to the objective function that they purport to optimize:

- Heuristic approaches typically involve optimizing associations between observations and targets under an explicit objective function.
- Maximum a posteriori (MAP) approaches find the most probable association, given all observations returned so far, then estimate objects with this found association.
- The Bayesian approaches generates optimal filtering predictions by summing over all possible associations, weighted by their probabilities.

Data association algorithms can also be categorized by the way in which they process the measurements:

- Single-scan algorithms estimate the current states of targets based on their previously computed tracks and the current scan of measurements.
- Multi-scan algorithms may revisit past scans when processing each new scan, and can thereby revise previous association decisions in the light of new evidences.

The simplest data association method using a heuristic approach is the Greedy Nearest Neighbor (GNN) [6]. It processes the new observations in some order and associates each with the target whose predicted position is closest, thereby selecting a single association after each scan. The method requires very little computation and is extremely fast. One drawback is its inability of correcting error associations at later steps.

Exact Bayesian data association is even less tractable than the MAP computation. Several pseudo-Bayesian methods have been proposed, of which the best-known is the joint probabilistic data association (JPDA) filter. JPDA is well described in [1] which is a suboptimal single-scan approximation to the optimal Bayesian filter. At each time step, instead of finding a single best association between observations and tracks, JPDA enumerates all possible associations (NP-hard) and computes association probabilities between each observation and each object. JPDA has proved more efficient in a cluttered environments compared with GNN [1] but prone to make erroneous decision since only single scan is considered and the association made in the past is not reversible.

MAP approaches includes the well-known multiple hypothesis tracking (MHT) algorithm [5]. MHT is a multi-scan association algorithm that maintains multiple hypotheses associating past observations with targets and returns a hypothesis with the highest posterior as a solution. The main disadvantage of MHT in its pure form is its computational complexity since the number of hypotheses grows exponentially over time. Various heuristic methods have been developped to control this growth but these methods are applied at the expense of sacrificing the MAP property. However, since the underlying MAP data association problem is NP-hard, so we do not expect to find an efficient, exact algorithm.

Another interesting technique to solve the MAP problem is to use an optimisation technique. The Viterbi algorithm (VA) [2] is an established optimisation technique for dis-

Asma Azim and Olivier Aycard are with University of Grenoble1 & INRIA Rhône-Alpes, Grenoble, France, (e-mails: Asma.Azim@inrialpes.fr, Olivier.Aycard@inrialpes.fr)

crete Markovian systems that has been extensively used in speech recognition. The VA is essentially a batch algorithm, although in practice it may be used in a fixed-lag processing mode due to merging of paths in the trellis [2]. The application of the Viterbi algorithm to the data association problem in the single target case was covered in [3]. The essential idea is to create a trellis based on the measurements rather than the states. Any path through the trellis corresponds to a sequence of data associations. The Viterbi algorithm is harnessed to determine the shortest or lowest cost path through the trellis. These techniques are not optimal for tracking because the sequence of data associations conditioned on the estimated states is not Markov. Nonetheless convincing results have been achieved over existing single-scan approaches such as JPDA [3]. The so-called Viterbi data association (VDA) approach has been used for multiple objects tracking, although the technique is limited to widely separated targets [4].

In this paper, we present an extension of VDA to solve multiple objects tracking in clutter environment: we propose to solve the problem of multiple pedestrian tracking in clutter environment. The basic idea is to have several instances of VDA running in parallel: one instance for one object. The paper is organized as follow: in next section, we present the experimental platform used in this paper. In section III, we summarize the VDA algorithm in the context of single object tracking. We detail the extension of VDA to multiple objects tracking in section IV. Experimental results are reported in section V. In the last section, we give some conclusions and perspectives.

II. DEMONSTRATOR DESCRIPTION

The demonstrator used to get dataset for this work is the Cycab simulator prepared by INRIA. It provides a graphical interface along with a movable Cycab car. We used an occupancy grid for the internal representation of the external environment. The moving objects are detected using the technique presented by [7]. We construct an occupancy grid map incrementally from laser measurements that can be considered as a background modeling process. And based on the constructed grid map, we are able to identify moving objects when new measurements arrive: If an object is seen in a location previously observed as free space, the object is moving; if free space is observed in a location previously occupied by an object, then that object was moving. The dataset that we have obtained from the Cycab simulator consists of multiple pedestrians. The number of pedestrians varies at different instants.

III. VITERBI DATA ASSOCIATION

The Viterbi algorithm [2] is a recursive algorithm that provides a solution to the discrete linear optimization problem. It is used for finding the most likely sequence of hidden states called the Viterbi path that results in a sequence of observed events, especially in the context of hidden Markov models.

The implementation of the Viterbi algorithm is based on a trellis. A trellis diagram is a type of directed graph (N, A) that consists of a set of nodes N and a set of directed arcs A. The nodes n_i^k are partitioned into ordered sets with the k^{th} set being denoted as $N(k)$, where k represents stages in the trellis $(k = 1, 2, ...T)$. The number of nodes at each stage is denoted by n_k. An important assumption underlying the use of the trellis diagram is that the state can be modeled by a Markov process. Hence, in dealing with the trellis diagram, the set of directed arcs A is a collection of ordered pairs $\{n_i(k-1), n_i(k)\}$ where $k = 2, ..., T$. A path P is a collection of directed arcs that connects an element at stage 1 to an element at stage T. Each directed arc is associated with a metric or a distance label $a_{ij}(k)$. A path metric is defined as the sum of the metrics of all the arcs contained in the path P as

$$d(P) = \sum_{k=2}^{T} a_{ij}(k); \{n_i(k-1), n_i(k)\} \in P \qquad (1)$$

where $d(P)$ is the total metric of the path P.

Starting from the initial stage, the VA successively labels all the nodes in the trellis until the final stage is reached. The optimal state sequence in the trellis is then retrieved by backtracking, starting from the node in the final stage with the smallest metric. In order to use Viterbi Data Association (VDA) technique to resolve the data association problem for single target tracking [3], we assume that each node in the trellis represents an observation. The collection of measurements at k^{th} scan Z_k corresponds to the set of nodes at k^{th} stage of the trellis. Arcs of the trellis are defined as the metric d on the basis of which we can associate the observation to the corresponding track.

Using the notations defined above, we can summarize the Viterbi algorithm for single object tracking as follows:

Step 1: Initialization Step

Assign a value of zero to the label of each node in first stage:

$$d_i(1) = 0, 0 \leq i \leq n_1 \qquad (2)$$

$$\psi_i(1) = 0, 0 \leq i \leq n_1 \qquad (3)$$

Step 2: Recursive Step

Repeat the following procedure for each stage k, where $k = 2, .., T$

- For each node $i = 0, .., n_{k-1}$(at stage $k-1$), calculate the predicted position using Kalman filter:

$$\widehat{x}_i(k/k-1) = \phi\widehat{x}_i(k-1/k-1) \qquad (4)$$

$$P_i(k/k-1) = \phi P_i(k-1/k-1)\phi^T + Q(k-1) \qquad (5)$$

$$S_i(k/k-1) = HP_i(k-1/k-1)H^T + R(k-1) \qquad (6)$$

- For each node $j = 0, .., n_k$ (at stage k), calculate the distance metric $a_{ij}(k)$ of the arc joining nodes $n_i(k-1)$ and $n_j(k)$.
- Assign node $n_j(k)$ with the smallest label as follows:

$$i^* = arg\{\min_{0 \leq i \leq n_{k-1}} \{d_i(k-1) + a_{ij}(k)\}\} \qquad (7)$$

$$Score_j(k) = d_{i^*}(k-1) + a_{ij}(k) \tag{8}$$

$$\psi_j(k) = i^* \tag{9}$$

- Update the target state at each node in stage k:

$$K_j(k) = P_{i^*}(k/k-1)H^T[HP_{i^*}(k/k-1)H^T + R(k)]^{-1} \tag{10}$$

$$P_{i^*}(k/k) = [I - K_j(k)H]P_{i^*}(k/k-1) \tag{11}$$

$$\widehat{x}_j(k/k) = \widehat{x}_{i^*}(k/k-1) + K_j(k)[z_j(k) - H\widehat{x}_{i^*}(k/k-1)] \tag{12}$$

Step 3: Final selection:
Determine the node with the minimum score in the final stage.

$$i^* = arg\{\min_{0 \leq i \leq n_T}\{d_i(k)\}\} \tag{13}$$

$$\widehat{x}(T) = \widehat{x}(T/T) \tag{14}$$

Step 4: Backtracking step
Recover the measurement sequence that terminates with the minimum node score in the final stage. For each stage k,

$$i^*(k-1) = \psi_{i^*(k)}, k = T, T-1, ..., 2 \tag{15}$$

The notation used in the above algorithm are as follows:

- $d_i(k)$ is the metric of node $n_i(k)$
- $\psi_j(k)$ is the predecessor function of the node $n_j(k)$
- i, j are indices of elements in $N(k)$
- $d^*(T)$ is the metric of the shortest path in the trellis diagram.

Each of the nodes in our trellis contains the information about the observations. For the first scan, we obtain the first set of observations $z(1)$. We create a node for each of these observations and assign a value of zero to its accumulated distance metric. For all the preceding scans (k), we use Kalman filter to calculate the predicted position for each of the observations in the previous state $(k1)$ and calculate the Euclidean distance between this predicted position and the new observations obtained in the current scan. This is the distance metric for each of the arc in the trellis. We find the previous observation node $n_i(k1)$ for which this distance metric is minimum and add this metric to the accumulated distance label of the current node $n(k)$. If the current scan is the final scan $(k = T)$, we find the minimum of the distance labels in this stage. The final state of the target is the observation associated with the minimal node. After obtaining this final state, we start to backtrack through the trellis to recover the measurement sequence that ends at this state. This sequence is the Viterbi path or the Viterbi track for the observed pedestrian.

IV. EXTENSION FOR MULTIPLE PEDESTRIANS TRACKING

In order to implement Viterbi algorithm for multiple target tracking, we create a separate instance of Viterbi data association for each of the objects. Thus there is a separate Viterbi trellis and separate track for every object. In each scan, when we receive a new set of observations, we first perform the gating to find the likely observations to be associated with the existing tracks. We have used Mahalanobis distance between the predicted positions of the track and the newly received observation as a measure for gating. We have used kalman filter to calcuated the predicted positions of the tracks at each instance as described in the previous section. After applying gating, the observations that fall outside the prescribed gate can possibly be the potential candidates for new tracks or the false alarms. For all the observations that fall inside the gate, we perform the data association of each observation with all the existing tracks. The observation that gets associated to any of the tracks is marked ”associated” and the remaining observations that are not associated to any of the existing tracks are marked ”unassociated”. These ”unassociated” observations might correspond to a new pedestrian or a false alarm. In the next step, we initialize new instances of vda for each of these ”unassociated” observations. As the number of observations may not always correspond directly to the number of existing tracks in a dynamic environment, we have implemented the mechanism for the creation, suppression and maintenance of the tracks. We have also dealt with the classical problem of split and merge.

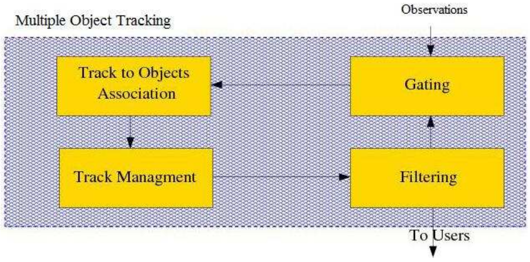

Fig. 1. Architecture of the Multiple Object Tracking

A. *Maintenance of an object*

We illustrate the Viterbi data association technique for multiple objects tracking using the following example. In the start, we received two observations for moving objects and initialized two instances of VDA, VDA$_1$ and VDA$_2$ corresponding to each of these observations. In the next scan, we received 2 observations again. We made the associations of both the tracks VDA$_1$ and VDA$_2$ with each of these observations. We first associated VDA$_1$ to each of the observations and computed corresponding distance metrics for each association to find the best association at this time step. Then we associated VDA$_2$ to each observation and found the best association for that as well. This process is shown in Fig. 2 where the Viterbi association is represented by the solid arrows and the others by dashed arrows.

B. *Creation of an object*

After iterating the new observations for all the existing tracks, we look for the observations which are not used yet. These unused observations are potential candidates for new tracks. We create a new track for each of these unused observations but those tracks are not confirmed yet. If those

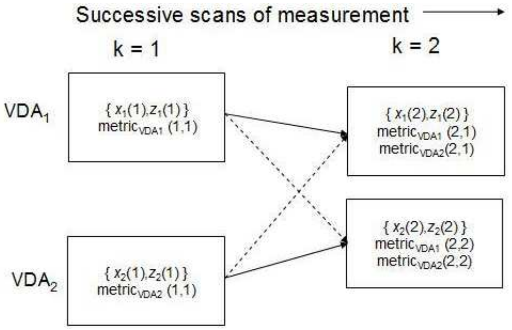

Fig. 2. Track maintenance

newly created tracks get associated to the observations for 5 consecutive scans then those are marked as confirmed objects and are displayed on the occupancy grid. In the other case, those observations are assumed to be false alarms and are suppressed subsequently. Using the same example, in the next few time steps, we assume that we received 2 observations each and associated them with the current tracks similarly. After a few scans we received 3 observations instead of two at the k^{th} stage. We associated each of the tracks VDA_1 and VDA_2 with these observations. Two of those got associated to the existing tracks while the third observation remained unused. We created a third instance of VDA, track VDA_3 for this observation but this newly created track is tentative and not confirmed yet. Fig. 3 illustrates this process. We have omitted the initial stages and shown the previous and current stage only to make the figure look more clear. For the preceding scans, we associate all the tracks, including VDA_3, with each of the observations received. If there are observations which are associated to VDA_3 for 3 consecutive scans, as shown in Fig. 3, then we confirm its status that it is the track for a pedestrian. Otherwise, if we do not receive observations for this track in next 3 scans, we consider it to be a false alarm and suppress this track by destroying the VDA instance and making it available for any new object.

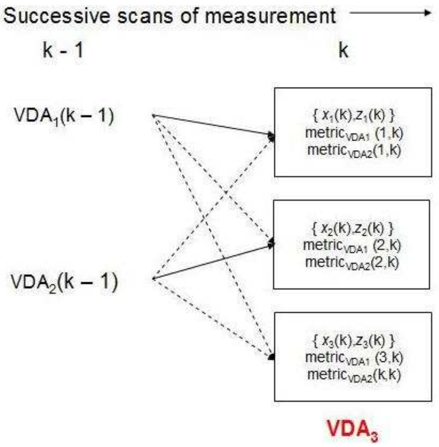

Fig. 3. Track creation

C. Deletion of an object

Similar to the track creation mechanism, a track deletion or suppression mechanism is also implemented in our technique. If at any stage, we find no observation associated to a track, we mark it as not observed. We predict the position of the object from its previous observations. This predicted position is considered to be the current position of that object. If the object is not observed for 3 consecutive scans, we assume that it has moved out of the scanning area and its track is deleted. Continuing with the same example, we assume that after a few more scans, we did not receive any observation associated to track VDA_2. We estimated the position of VDA_2 using Kalman filter and considered this estimated position to be actual position of this track and marked it as not observed. The reason for doing so is that the pedestrian whose track it is may be temporarily occluded by some object or some other pedestrian and may become visible again after a short while. But if we do not receive the observations associated to VDA_2 for 3 consecutive scans, it shows this pedestrian is no more in the viewing area therefore we delete the track VDA_2 as shown in Fig. 4.

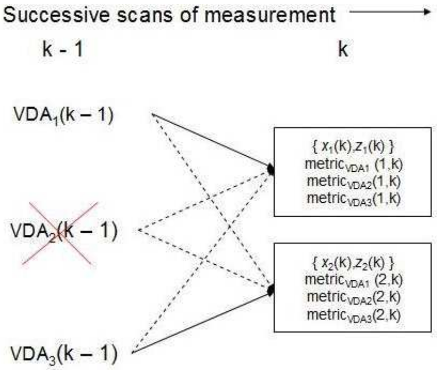

Fig. 4. Track deletion

D. Merge problem

We assume that initially we have two independent tracks for a couple of pedestrians. Then, at some later stage, they come so close to each other that we receive only one observation for both of them. In this case, keeping the previous record, and using the filtering technique for predicted positions, we associate the single observation with both the tracks thus to get a merged track for two pedestrians. Fig. 5 is an illustration of this problem. In scan $k-1$, we had tracks for two pedestrians, VDA_1 and VDA_2. In scan k, we received only one observation $z_1(k)$. We used Viterbi algorithm to make the associations and found that both VDA_1 and VDA_2 got associated to $z_1(k)$. Then we found the predicted positions for both the pedestrians. In the next scan, we again received a single observation and associated it with both tracks similarly. Thus we obtained a merged track for the two pedestrians.

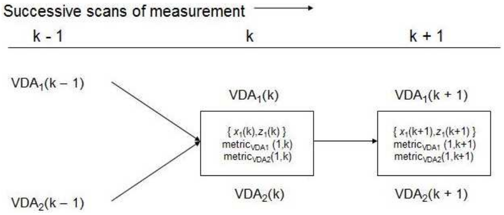

Fig. 5. Track merge

E. Split problem

Splitting of tracks is the case opposite to merging. In the situations when there are two pedestrians walking together such that one is hiding the other, we receive only one observation for both. As a consequence, only one track is created for both. Later, when they split their paths, we start receiving separate observations for them. In this case, we split their track in to two and the previous observations are associated to history of both tracks. Fig. 6 is an illustration of the split problem. At scan $k-1$, we had a single observation for the two pedestrians with merged tracks VDA$_1$ and VDA$_2$. At scan k, we received a couple of observations $z_1(k)$ and $z_2(k)$ and found the best associations between the existing tracks and those new observations. We found that VDA$_1$ got associated to the $z_1(k)$ and VDA$_2$ to $z_2(k)$. Thus we split the merged tracks for two pedestrians as illustrated in Fig. 6.

A relatively complex situation of track splitting arises in the case when both the pedestrians were moving in the same direction and with same speed from the very start. In this case, we receive a single observation for both the pedestrians from the first scan thus we create only one instace of VDA corresponding to that observation. It carries on to be a single instance as long as we keep on receiving a single observation. But, at the instance when at least one of the pedestrians (or both) deviates from his path, we get two separate observations corresponding to them. Then we calculate the best association for the existing track and the other observation is used to create a new instance of VDA. Thus, in this case, we have the track for the second pedestrian only from the instance we received a separate observation for him. We used the same example to illustrate this with the help of Fig. 7.

V. EXPERIMENTAL RESULTS

We have verified our technique by using different datasets generated by the Cycab simulator using laser sensor. The datasets contained multiple pedestrians having arbitrary motion in different directions. We use the background subtraction method to identify the stationary objects in the environment and show them using an occupancy grid. Then we detect the moving pedestrians and Viterbi data association is used to track the motion of these pedestrians. In our occupancy grid, the grey cells show the fact that the space is empty and white cells mean that scanner detected objects there and so this space is occupied. The rest of the cells having black color mean that we do not know whether they are occupied or not, either they are out of the view of the Cycab or are behind the occupied cells. The bold lines show the tracks for the pedestrian and the rectangles represent their final position. We have used different colors to represent the track for each pedestrian. The green dot shows the position of the Cycab itself. We have taken the datasets consisting of measurements for 200 scans.

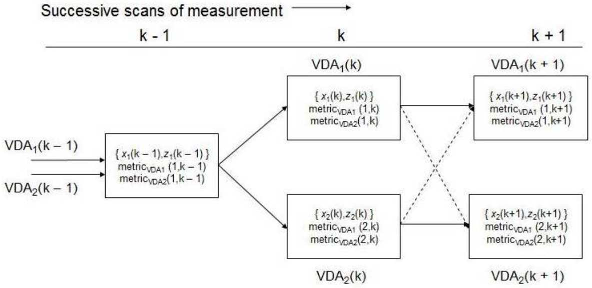

Fig. 6. Track split

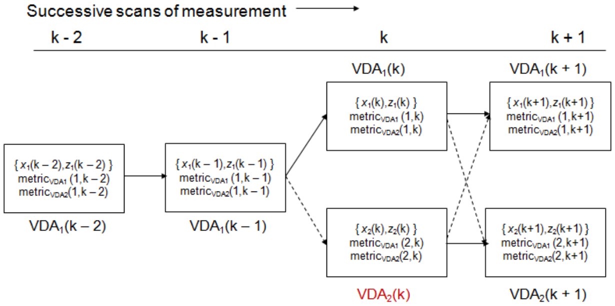

Fig. 7. Track split and creation of a new track

We have tested our technique for single pedestrian tracking first. For the tracking of multiple pedestrians, we started with the relatively simple scenario of two or three pedestrians moving such that they did not cross each other. All of them were moving in roughly the same direction. After testing our technique for pedestrians having mutually exclusive tracks, we tested it for the ones who have crossing tracks. An example of the results we obtained for multiple pedestrian tracking is given in Fig. 8. The leftmost image depicts the situation where we had six pedestrians moving in different

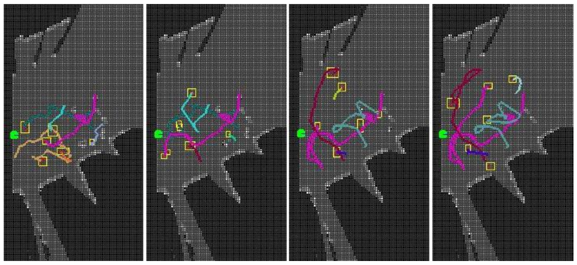

Fig. 8. An example of multiple pedestrian tracking using Viterbi data association. See text for more details.

directions. There tracks are represented with the lines in different colors. The next image represents the situation after a few more scans. At that instance, there are only five pedestrians in the viewing area. The three pedestrians whose tracks were represented by brown, orange and skyblue lines had moved out of the view of Cycab, thus there tracks were deleted. While a couple of new pedestrians had moved in the view, for whom the new tracks were created which are represented by lines of light green and maroon colors. The two images of the right are representing the tracks of pedestrians after next few scans respectively.

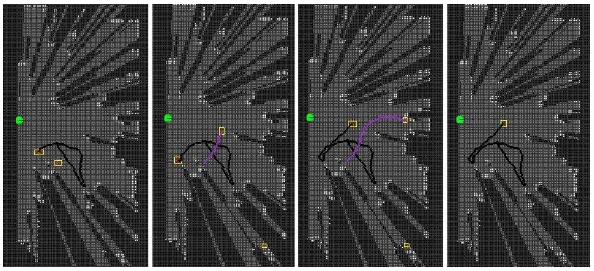

Fig. 9. An example of multiple pedestrian tracking using Viterbi data association. See text for more details.

TABLE I

COMPARISION BETWEEN MHT AND VDA

Sub-sequences	total frame	total objects	correct detected		false alarms	
			MHT	VDA	MHT	VDA
1	875	94	82	79	15	17
2	693	149	136	133	23	21

Fig. 9 is an illustration of the track creation and deletion using our technique. In this case, there are a couple of pedestrians moving in arbitrary directions but the difference is that the second pedestrian appeared at a later stage. Initially there was only one pedestrian as shown in the leftmost image. After a few scans, the second pedestrian was detected and his track was initiated. Then after a few more scans, the second pedestrian, whose track is shown in purple, moved out of the view of the Cycab. When there was no observation associated to its track for 5 consecutive scans, the track for this pedestrian was deleted and for rest of the scans, there was only one pedestrian being tracked, as shown in the rightmost image.

In order to demonstrate the performance of the Viterbi data association for multiple target tracking, we provide a numerical comparison of this technique with its nearest counterpart, i.e mulitple hypothesis tracking (MHT). Due to the pruning of the association hypothesis, Viterbi data associations has proved to be very efficient and less time consuming as compared to MHT. We choose some typical subsequences from the dataset and perform an evaluation for both MHT and VDA. In each frame of data we labeled all pedestrians in the viewing area. For each subsequence, we count how many pedestrians are detected and tracked. Wrong detections (false positives) are also counted. The results of the evaluation are summarized in Table I.

The results in the table show that, for the first subsequence, consisting of 875 frames and 94 tracks, VDA detected 84% of the tracks correctly in comparison to 87.2% of MHT. While in the second subsequence, having 693 frames and 149 tracks, the percentage of correct detection for VDA was 89.3% and 91.2% for MHT. These results illustrate that the Viterbi data association has a very consistent and high percentage of correct detection for multiple pedestrian tracking in complex scenarios while a very low rate of false detection. The performance of this technique is quite comparable with that of the MHT with an additional advantage of less time consumption due to pruning of association hypothesis. Using this technique, we do not need to keep track of all possible hypothesis of data association and can still achieve a very good performance due to the ability to backtrack and revise the decision at later stages. Though it is not a perfect solution as it becomes more time consuming with an increasing number of objects to be tracked, but if we compare it with MHT, which is the best solution theoretically, VDA is still reasonably less time consuming. Another limitation is the situation in which the tracks of a large number of objects cross each other very often. In such situations, we might have the problem of track swapping and false detection.

VI. CONCLUSION

In this paper, we present an extension of VDA to multiple objects tracking. To show the interest of our method, we present some experimental results on multiple pedestrian tracking. Moreover, quantitative comparisons with MHT is presented and commented.

In other work [8], we propose a ”model-based solution” to the problem of multiple objects tracking. To perform data association, we used MCMC method. We plan to extend the approach presented in this paper to model-based solution and compare computational time, qualitative and quantitative results.

VII. ACKNOWLEDGEMENTS

The work is supported by the European project INTERSAFE-2[1]

REFERENCES

[1] Y. Bar-Shalom and T.E. Fortman. *Tracking and Data Association*. Academic Press, 1988.

[2] G. David Forney. The viterbi algorithm. *Proceedings of The IEEE*, 61(3):268–278, 1973.

[3] A. S. Gad and Farooq M. Viterbi based data association techniques for target tracking. *SPIE*, 5096(37), 2003.

[4] G. W. Pulford. Multi-target viterbi data association. In *Proc. of the IEEE Int. Conf. on Information Fusion*, 2006.

[5] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[6] S.Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. A K Peters, 1999.

[7] TD. Vu, O. Aycard, and N. Appenrodt. Online localization and mapping with moving objects tracking in dynamic outdoor environments. In *IEEE International Conference on Intelligent Vehicles*, 2007.

[8] Trung-Dung Vu and Olivier Aycard. Laser-based detection and tracking moving object using data-driven markov chain monte carlo. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009.

[1] http://www.intersafe-2.eu