

Mapping of environment, Detection and Tracking of Moving Objects using Occupancy Grids

Trung-Dung Vu, Julien Burlet and Olivier Aycard
Laboratoire d'Informatique de Grenoble, France
firstname.lastname@inrialpes.fr

Abstract—In this paper, we present a real-time algorithm for local simultaneous localization and mapping (SLAM) with detection and tracking of moving objects (DATMO) in dynamic outdoor environments from a moving vehicle equipped with a laser scanner and two radars. To correct vehicle location from odometry we introduce a new fast implementation of incremental scan matching method that can work reliably in dynamic outdoor environments. After a good vehicle location is estimated, the surrounding map is updated incrementally and moving objects are detected without a priori knowledge of the targets. Detected moving objects are finally tracked by a Multiple Hypothesis Tracker (MHT) coupled with an adaptive IMM (Interacting Multiple Models) Filter. The experimental results on datasets collected from different scenarios such as: urban streets, country roads and highways demonstrate the efficiency of the proposed algorithm on a Daimler Mercedes demonstrator in the framework of the European Project PReVENT-ProFusion2.

I. INTRODUCTION

Perceiving or understanding the environment surrounding of a vehicle is a very important step in driving assistant systems or autonomous vehicles. The task involves both simultaneous localization and mapping (SLAM) and detection and tracking of moving objects (DATMO). While SLAM provides the vehicle with a map of static parts of the environment as well as its location in the map, DATMO allows the vehicle being aware of dynamic entities around, tracking them and predicting their future behaviors. It is believed that if we are able to accomplish both SLAM and DATMO in real time, we can detect every critical situations to warn the driver in advance and this will certainly improve driving safety and can prevent traffic accidents.

Recently, there have been considerable research efforts focusing on these problems [13][9][18][19]. However, for highly dynamic outdoor environments like crowded urban streets, there still remains many open questions. These include, how to represent the vehicle environment, how to obtain a precise location of the vehicle in presence of dynamic entities, and how to differentiate moving objects and stationary objects as well as how to track moving objects over time.

In this context, we design and develop a generic architecture to solve SLAM and DATMO in dynamic outdoor environments. This architecture (Fig. 2) is divided into two main parts: the first part where the vehicle environment is mapped, fusion between different sensors is performed and moving objects are detected; and the second part where previously detected moving objects are verified and tracked.



Fig. 1. The Daimler Mercedes demonstrator car.

This architecture is currently used in the framework of the European project PReVENT-ProFusion¹. The goal of this project is to design and develop generic architectures to perform perception tasks. In this context, our architecture has been integrated and tested on two demonstrators: a Daimler-Mercedes demonstrator and a Volvo truck demonstrator [7]. In this paper, we give an overview of our architecture, focusing on the description of the first level of the architecture and show some results on the Daimler-Mercedes demonstrator moving at high speed.

The rest of the paper is organized as follows. In the next section, we present the Daimler Mercedes demonstrator. A brief overview of our architecture is given in section III. Description of first level of architecture is detailed in Section IV. Second level is summarized in section V. Experimental results are given in Section VI and finally in Section VII conclusions and future works are discussed.

II. THE DAIMLER MERCEDES DEMONSTRATOR

The Daimler Mercedes demonstrator car is equipped with a camera, two short range radar sensors and a laser scanner (Fig. 1). The radar sensor is with a maximum range of 30m and a field of view of 80°. The maximum range of laser sensor is 80m with a field of view of 160° and a horizontal resolution of 1°. In addition, vehicle odometry information such as velocity and yaw rate are provided by the vehicle

¹www.prevent-ip.org/profusion

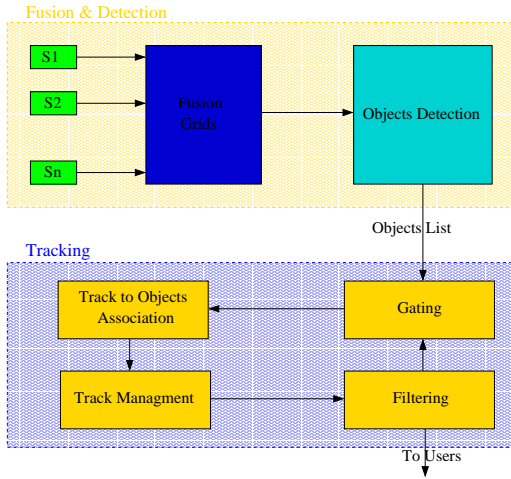


Fig. 2. Architecture of the perception system

sensors. The measurement cycle of the sensor system is 40ms. Images from camera are for visualization purpose.

III. GENERAL ARCHITECTURE

We design and develop a generic architecture (Fig. 2) to solve SLAM and DATMO in dynamic outdoor environments.

In the first part of the architecture, to model the environment surrounding the vehicle, we use the Occupancy Grid framework developed by Elfes [6]. Compared with feature-based approaches, grid maps can represent any environment and are specially suitable for noisy sensors in outdoor environments where features are hard to define and extract.

In general, in order to perform mapping or modelling the environment from a moving vehicle, a precise vehicle localization is essential. To correct vehicle locations from odometry, we introduce a new fast laser-based incremental localization method that can work reliably in dynamic environments. When good vehicle locations are estimated, by integrating laser measurements we are able to build a consistent grid map surrounding of the vehicle. Finally by comparing new measurements with the previously constructed local vehicle map, dynamic objects then can be detected.

Finally, sensor data coming from different sensors are fused.

In the second part, detected moving objects in the vehicle environment are tracked. Since some objects may be occluded or some are false alarms, multi objects tracking helps to identify occluded objects, recognize false alarms and reduce mis-detections.

IV. FIRST LEVEL

In this section, we detail the first level of the architecture: Environment Mapping & Localization, Moving Objects Detection and Sensor Data Fusion.

A. Environment Mapping & Localization

Since our radar sensors provide pre-filtered data at object level as lists of target points, so to perform mapping, only laser data is used. To our safety vehicle navigation purpose,

a good global map is not necessary, so that the problem of revisiting or loop closing in SLAM is not considered in this work. For this reason, we propose an incremental mapping approach based on a fast laser scan matching algorithm in order to build a consistent local vehicle map. The map is updated incrementally when new data measurements arrive along with good estimates of vehicle locations obtained from the scan matching algorithm. The advantages of our incremental approach are that the computation can be carried out very quickly and the whole process is able to run online.

1) *Notation:* Before describing our approach in detail, we introduce some notations used in the paper. We denote the discrete time index by the variable t , the laser observation from vehicle at time t by the variable $z_t = \{z_t^1, \dots, z_t^K\}$ including K individual measurements corresponding to K laser beams, the vector describing an odometry measurement from time $t-1$ to time t by the variable u_t , the state vector describing the true location of the vehicle at time t by the variable x_t .

2) *Occupancy Grid Map:* In the occupancy grid representation, the vehicle environment is divided into a two-dimensional lattice M of rectangular cells and each cell is associated with a measure taking a real value in $[0, 1]$ indicating the probability that the cell is occupied by an obstacle or not. A high value of occupancy grid indicates the cell is occupied and a low value means the cell is free. Suppose that occupancy states of individual grid cells are independent, the objective of a mapping algorithm is to estimate the posterior probability of occupancy $P(M_i | x_{1:t}, z_{1:t})$ for each cell of grid M_i , given observations $z_{1:t} = \{z_1, \dots, z_t\}$ at corresponding known poses $x_{1:t} = \{x_1, \dots, x_t\}$.

In the literature, many methods are used for occupancy grid mapping, such as Bayesian [6], Dempster-Shafer [12] and Fuzzy Logic [11]. Here we apply Bayesian Update scheme [17] that provides an elegant recursive formula to update the posterior under log-odds form:

$$\log O(M_i | x_{1:t}, z_{1:t}) = \log O(M_i | x_{1:t-1}, z_{1:t-1}) + \log O(M_i | z_t, x_t) - \log O(M_i) \quad (1)$$

where $O(a|b) = \text{odds}(a|b) = P(a|b) / (1 - P(a|b))$

In (1), $P(M_i)$ is the prior occupancy probability of the map which is set to 0.5 representing an unknown state, this makes this component disappear. The remaining probability $P(M_i | x_t, z_t)$, is called the *inverse sensor model*. It specifies the probability that a grid cell M_i is occupied based on a single sensor measurement z_t at location x_t . In our implementation, it is decided by the measurement of the nearest beam to the center mass of the cell.

It is easy to see that the desired probability of occupancy, $P(M_i | x_{1:t}, z_{1:t})$, can be recovered from the log-odds representation. Moreover, since the updating algorithm is recursive, it allows for incremental map updating when new sensor data arrives.

3) *Localization in Occupancy Grid Map*: In order to build a consistent map of the environment, a good vehicle localization is required. Because of the inherent error, using only odometry often results in an unsatisfying map. When features can not be defined and extracted, direct scan matching techniques like ICP [3] can help to correct the odometry error. The problem is that sparse data in outdoor environments and dynamic entities make correspondence finding difficult. One important disadvantage of the direct scan matching methods is that they do not consider the dynamics of the vehicle. Indeed we have implemented several ICP variants [15] and found out that scan matching results are unsatisfactory and often lead to unexpected trajectories of vehicle. This is because matching only two consecutive scans may be very hard, ambiguous or weakly constrained, especially in outdoor environment and when the vehicle moves at high speeds.

An alternative approach that can overcome these limitations consists in setting up the matching problem as a maximum likelihood problem. In this approach, given an underlying vehicle dynamics constraint, the current scan's position is corrected by comparing with the local grid map constructed from all observations in the past instead of only with one previous scan. By this way, we can reduce the ambiguity and weak constraint especially in outdoor environment and when the vehicle moves at high speeds. Mathematically, we calculate a sequence of poses $\hat{x}_1, \hat{x}_2, \dots$ and sequentially updated maps M_1, M_2, \dots by maximizing the marginal likelihood of the t -th pose and map relative to the $(t-1)$ -th pose and map:

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \{P(z_t | x_t, M_{t-1}) \cdot P(x_t | \hat{x}_{t-1}, u_t)\} \quad (2)$$

In the equation (2), the term $P(z_t | x_t, M_{t-1})$ is the measurement model which is the probability of the most recent measurement z_t given the pose x_t and the map M_{t-1} constructed so far from observations $z_{1:t-1}$ at corresponding poses $\hat{x}_{1:t-1}$ that were already estimated in the past. The term $P(x_t | \hat{x}_{t-1}, u_t)$ represents the motion model which is the probability that the vehicle is at location x_t given that the vehicle was previously at position \hat{x}_{t-1} and executed an action u_t . The resulting pose \hat{x}_t is then used to generate a new map M_t according to (1):

$$M_t = M_{t-1} \cup \{\hat{x}_t, z_t\} \quad (3)$$

Now the question is how to solve the equation (2), but let us first describe the motion model and the measurement model used.

For the motion model, we adopt the probabilistic velocity motion model similar to that of [17]. The vehicle motion u_t is comprised of two components, the translational velocity v_t and the yaw rate ω_t . Fig. 3 depicts the probability of being at location x_t given previous location x_{t-1} and control u_t . This distribution is obtained from the kinematic equations, assuming that vehicle motion is noisy along its rotational and translational components.

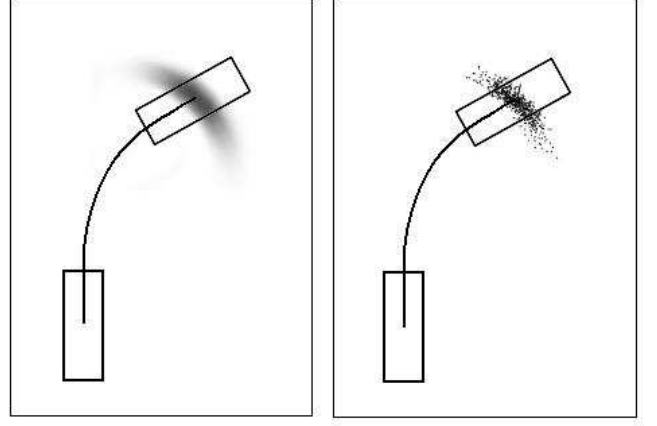


Fig. 3. The probabilistic velocity motion model $P(x_t | x_{t-1}, u_t)$ of the vehicle (left) and its sampling version (right).

For the measurement model $P(z_t | x_t, M_{t-1})$, mixture beam-based model is widely used in the literature [8][9]. However, the model come at the expense of high computation since it requires ray casting operation for each beam. This can be a limitation for real time application if we want to estimate a large amount of measurements at the same time. To avoid ray casting, we propose an alternative model that only considers end-points of the beams. Because it is likely that a beam hits an obstacle at its end-point, we focus only on occupied cells in the grid map. A voting scheme is used to compute the probability of a scan measurement z_t given the vehicle pose x_t and the map M_{t-1} constructed so far. First, from the vehicle location x_t , individual measurement z_t^k is projected into the coordinate space of the map. Call hit_t^k the grid cell corresponding to the projected end-point of each beam z_t^k . If this cell is occupied, a sum proportional to the occupancy value of the cell will be voted. Then the final voted score represents the likelihood of the measurement. Let $P(M_t^i)$ denote the posterior probability of occupancy of the grid cell M_t^i estimated at time t (following (1)), we can write the measurement model under the sum following:

$$P(z_t | x_t, M_{t-1}) \propto \sum_{k=1}^K \{P(M_{t-1}^{hit_t^k}) \text{ so that } M_{t-1}^{hit_t^k} \text{ is occupied}\} \quad (4)$$

The proposed method is just an approximation to the measurement model because it does not take into account visibility constraints, but experimental evidences show that it works well in practice. Furthermore, with a complexity of $O(K)$, the computation can be done rapidly.

It remains to describe how we maximize (2) to find the correct pose \hat{x}_t . Hill climbing strategy in [16][9] can be used but may suffer from a local maximum. Exploiting the fact that the measurement model can be computed very quickly, we perform an extensive search over vehicle pose space. A sampling version of the motion model (Fig. 3 right) is used to generate all possible poses x_t given the previous pose

x_{t-1} and the control u_t . The resulting pose will be the pose at which the measurement probability achieves a maximum value. Because of the inherent discretization of the grid, the sampling approach turns out to work very well. In practice, with a grid map resolution of 20 cm, it is enough to generate about four or five hundreds of pose samples to obtain a good estimate of the vehicle pose with the measurement likelihood that is nearly unimproved even with more samples. The total computational time needed for such a single scan matching is about 10 ms on a low-end PC. An example of scan matching result is shown in Fig. 4. The most likely vehicle pose is obtained when the laser scan is aligned with the occupied parts of the map and at the same time the vehicle dynamics constraint is satisfied.

Besides the computational effectiveness, one attraction of our algorithm is that it is not affected by dynamic entities in the environment. Since we only consider occupied cells, spurious regions in the occupancy grid map with low occupancy probability that might belong to dynamic objects do not contribute to the sum (4). Since a large part of measurements belong to static objects, the voting scheme ensures that measurement likelihood reach a maximum only when the laser scan is aligned with the static parts of the environment. To some meaning, measurements from dynamic entities can be considered as outliers of the alignment process. This property is very useful for moving object detection process that will be described in the next section.

4) *Local mapping*: Because we do not need to build a global map nor deal with loop closing problem, only one online map is maintained at each point in time representing the local environment surrounding of the vehicle. The size of the local map is chosen so that it should not contain loops and the resolution is maintained at a reasonable level. Every time the vehicle arrives near the map boundary, a new grid map is reinitialized. The pose of the new map is computed according to the vehicle global pose and cells inside the intersection area are copied from the old map.

B. Moving Objects Detection

After a consistent local map of the vehicle is constructed from SLAM, moving objects can be detected when new measurements arrive. The principal idea is based on the inconsistencies between observed free space and occupied space in the local grid map. If an object is detected on a location previously seen as free space, then it is a moving object. If an object is observed on a location previously occupied then it probably is static. If an object appears in a previously not observed location, then we can say nothing about that object.

Another important clue which can help to decide a object is dynamic or not is evidence about moving objects detected in the past. For example, if there are many moving objects passing through an area then any object that appears in that area should be recognized as a potential moving object. For this reason, apart from the local static map M as constructed

by SLAM described in the previous section, a local dynamic grid map D is created to store information about previously detected moving objects. The pose, size and resolution of the dynamic map is the same as those of the static map. Each dynamic grid cell store a value indicating the number of observations that a moving object has been observed at that cell.

From these remarks, our moving object detection process is carried out in two steps as follows. The first step is to detect measurements that might belong to dynamic objects. Here for simplicity, we will temporarily omit the time index. Given a new laser scan z , the corrected vehicle location and the local static map M computed by SLAM and the dynamic map D containing information about previously detected moving objects, state of a single measurement z^k is classified into one of three types following:

$$state(z^k) = \begin{cases} static & : M_{hit^k} = occupied \\ dynamic & : M_{hit^k} = free \text{ or } D_{hit^k} > \alpha \\ undecided & : M_{hit^k} = unknown \end{cases}$$

where hit^k is the coordinate of the grid cell corresponding to the end-point of the beam z^k and α is a predefined threshold.

The second step is after dynamic measurements are determined, moving objects are then identified by clustering end-points of these beams into separate groups, each group represents a single object. Two points are considered as belonging to the same object if the distance between them is less than 0.3 m.

Fig. 5 illustrates the described steps in detecting moving objects. The leftmost image depicts the situation where the vehicle is moving along a street seeing a car moving ahead and a motorbike moving in the opposite direction. The middle image shows the local static map and the vehicle location computed by SLAM and the current laser scan is drawn in red. Measurements which fall into free region in the static map are detected as dynamic and are displayed in the rightmost image. After the clustering step, two moving objects in green boxes are identified and correctly corresponds to the car and the motorbike.

Note that our map updating procedure makes use of results from moving object detection step. Measurements detected as dynamic are not used to update the map in SLAM. For unknown measurements, a priori we will suppose that they are static until latter evidences come. This will help to eliminate spurious objects and result in a better map.

C. Fusion with radars

After moving objects are identified from laser data, we confirm the object detection results by fusing with radar data and provide the detected objects with their velocities. For each moving object detected from laser data as described in the previous section, a rectangular bounding box is calculated and the radar measurements which lie within the box region are then assigned to corresponding object. The velocity of the detected moving object is estimated as the average of these corresponding radar measurements.

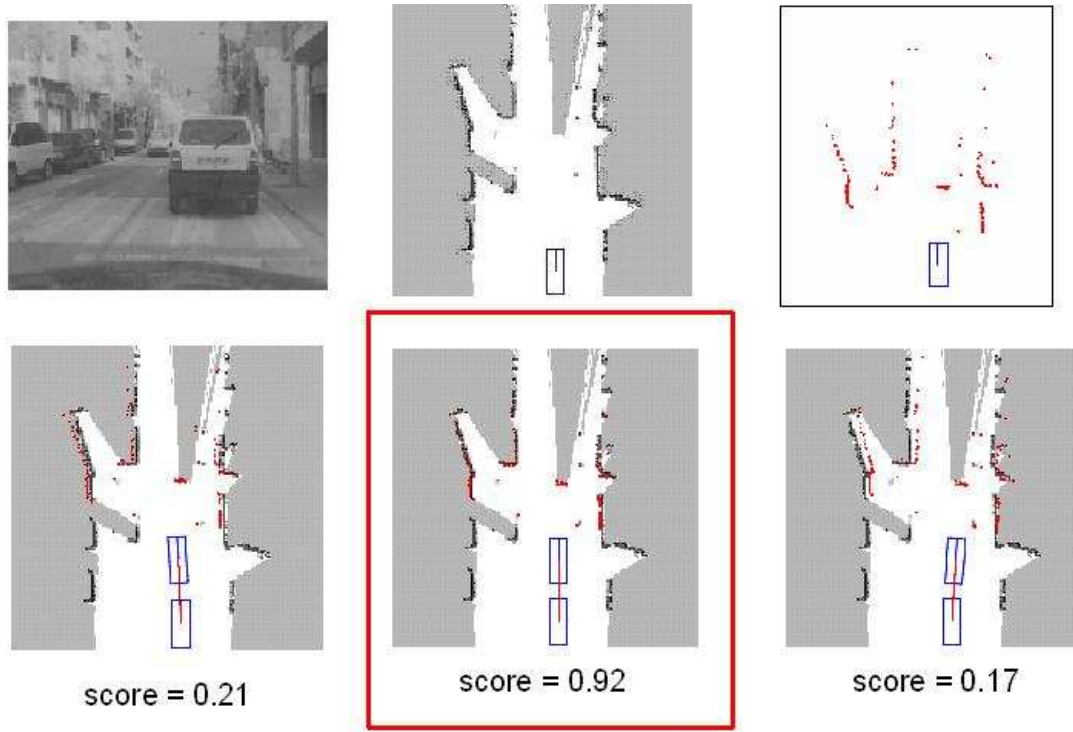


Fig. 4. An example of localization

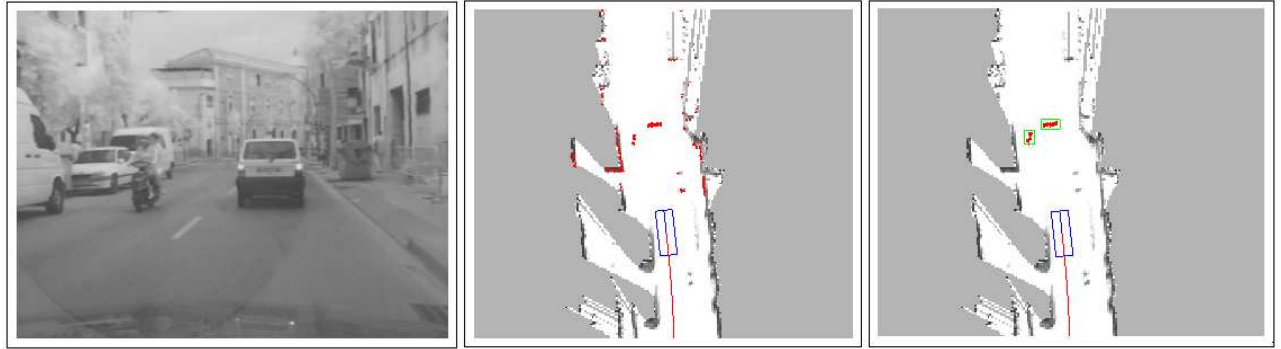


Fig. 5. Moving object detection example. See text for more details.

Figure 6 shows an example of how the fusion process takes place. Moving objects detected by the Laserscanner are displayed in red with green bounding boxes. The targets detected by two radar sensors are represented as small circles in different colors along with corresponding velocities. We can see in the radar field of view, that two objects detected by the Laserscanner are also seen by two radars so that they are confirmed and their velocities are estimated. Radar measurements that do not correspond to any dynamic object or fall into another region of the grid are not considered.

V. SECOND LEVEL

In general, the multi objects tracking problem is complex: it includes the definition of tracking methods, but also association methods and maintenance of the list of objects currently present in the environment [2][17]. Regarding tracking techniques, Bayesian filters [1] are generally used. These filters

require the definition of a specific motion model of tracked objects to predict their positions in the environment. Using this prediction and some observations, in a second stage, an estimation of the position of each object present in the environment is computed.

In this section, we briefly summarize the four different parts of the second level of the architecture (figure 2) to solve the different parts of multi-objects tracking (More details could be found in [5]):

- The first one is the gating. In this part, taking as input predictions from previous computed tracks, we compute the set of new detected objects which can be associated to each track.
- In a second part, using the result of the gating, we perform objects to tracks association and generate association hypothesis, each track corresponding to a



Fig. 6. Moving object detected from laser data is confirmed by radar data.

previously known moving object. Output is composed of the computed set of association hypothesis. This task is solved using the Multiple Hypothesis Tracking (MHT) algorithm [2].

- In the third part called tracks management, tracks are confirmed, deleted or created according to the association results and a pruned set of association hypothesis is output.
- In the last part corresponding to the filtering step, estimates are computed for 'surviving' tracks and predictions are performed to be used the next step of the algorithm. However, defining a suitable motion model is a real difficulty. To deal with this problem, Interacting Multiple Models [10][14] have been successfully applied in several applications. In [4], we have developed a fast method to adapt on-line IMM according to trajectories of detected objects and so we obtain a suitable and robust tracker.

VI. EXPERIMENTAL RESULTS

The detection and tracking results are shown in Fig. 7. The images in the first row represent online maps and objects moving in the vicinity of the vehicle are detected and tracked. The current vehicle location is represented by blue box along with its trajectories after correction from the odometry. The red points are current laser measurements that are identified as belonging to dynamic objects. Green boxes indicate detected and tracked moving objects with corresponding tracks displayed in different colors. Information on velocities is displayed next to detected objects if available. The second row are images for visual references to corresponding situations.

In Fig. 7, the leftmost column depicts a scenario where the demonstrator car is moving at a very high speed of about 100 kph while a car moving in the same direction in front of it is detected and tracked. On the rightmost is a situation

where the demonstrator car is moving at 50 kph on a country road. A car moving ahead and two other cars in the opposite direction are all recognized. Note that the two cars on the left lane are only observed during a very short period of time but both are detected and tracked successfully. The third situation in the middle, the demonstrator is moving quite slowly at about 20 kph in a crowded city street. Our system is able to detect and track both the other vehicles and the motorbike surrounding. In all three cases, precise trajectories of the demonstrator are achieved and local maps around the vehicle are constructed consistently. In our implementation, the computational time required to perform both SLAM and DATMO for each scan is about 20 – 30 ms on a 1.86GHz, 1Gb RAM laptop running Linux. This confirms that our algorithm is absolutely able to run synchronously data cycle in real time. More results and videos can be found at <http://emotion.inrialpes.fr/~tdvu/videos/>.

VII. CONCLUSIONS AND FUTURE WORKS

We have presented an approach to accomplish online mapping and moving object tracking simultaneously. Experimental results have shown that our system can successfully perform a real time mapping and moving object tracking from a vehicle at high speeds in different dynamic outdoor scenarios. This is done based on a fast scan matching algorithm that allows estimating precise vehicle locations and building a consistent map surrounding of the vehicle. After a consistent local vehicle map is built, moving objects are detected and are tracked using an adaptive Interacting Multiple Models filter coupled with an Multiple Hypothesis tracker.

VIII. ACKNOWLEDGMENTS

The work is supported by the European project PREVENT-ProFusion, and partially by the Délégation Générale pour L'Armement (DGA).



Fig. 7. Experimental results show that our algorithm can successfully perform both SLAM and DATMO in real time for different environments

REFERENCES

- [1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filter for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 2002.
- [2] Y. Bar-Shalom and T.E. Fortman. *Tracking and Data Association*. Academic Press, 1988.
- [3] P. Besl and N. McKay. A method for registration of 3d shape. *Trans. Pattern Analysis and Machine Intelligence*, 12(2), 1992.
- [4] J. Burlet, O. Aycard, A. Spalanzani, and C. Laugier. Adaptive interactive multiple models applied on pedestrian tracking in car parks. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing (CN), 2006.
- [5] J. Burlet, T.D. Vu, and O. Aycard. Grid-based localization and online mapping with moving object detection and tracking. Technical Report 167687, INRIA, 2007.
- [6] A. Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.
- [7] N. Floudas, A. Polychronopoulos, O. Aycard, J. Burlet, and M. Ahrholdt. High level sensor data fusion approaches for object recognition in road environment. In *IEEE International Conference on Intelligent Vehicles*, 2007. [pdf].
- [8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.
- [9] D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7):579–598, 2003.
- [10] E. Mazar, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(1):103–123, 1998.
- [11] G. Oriolo, G. Ulivi, and M. Vendittelli. Fuzzy maps: a new tool for mobile robot perception and planning. *J. of Robotic Systems*, 14(3):179–197, 1997.
- [12] D. Pagac, E.M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14, 1998.
- [13] E. Prassler, J. Scholz, and P. Fiorini. Navigating a robotic wheelchair in a railway station during rush hour. *Int. Journal on Robotics Research*, 18(7):760–772, 1999.
- [14] X. Rong Li and Vesselin P.Jilkov. A survey of maneuvering target tracking-part v: Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 2003.
- [15] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. 2001.
- [16] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *ICRA*, 2000.
- [17] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [18] C.-C. Wang. *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.
- [19] Denis F. Wolf and Gaurav S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19, 2005.