

A generic architecture for dynamic outdoor environment

Olivier Aycard, Trung-Dung Vu, Qadeer Baig, Thierry Fraichard

► **To cite this version:**

Olivier Aycard, Trung-Dung Vu, Qadeer Baig, Thierry Fraichard. A generic architecture for dynamic outdoor environment. ICTAI - IEEE 23rd International Conference on Tools with Artificial Intelligence, Nov 2011, Boca Raton, Florida, United States. pp.585-590, 2011, <10.1109/ICTAI.2011.93>. <inria-00625492>

HAL Id: inria-00625492

<https://hal.inria.fr/inria-00625492>

Submitted on 21 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A generic architecture for dynamic outdoor environment

Olivier Aycard, Trung-Dung Vu, Qadeer Baig, and Thierry Fraichard

Laboratoire d'Informatique de Grenoble - University of Grenoble1, France

firstname.lastname@imag.fr

Abstract—In this paper, we present a generic architecture for perception of an intelligent vehicle in dynamic outdoor environment. This architecture is composed of two levels: a first level dedicated to real-time local simultaneous localization and mapping (SLAM) and a second one is dedicated to detection and tracking of moving objects (DATMO). 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 and on a Volkswagen Demonstrator in the framework of the European Project Intersafe2.

Keywords: Sensor Data Processing, Sensor Data Modeling, Reasoning, Perception

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 [11]. 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 (Fig. 1) 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 [3]. 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

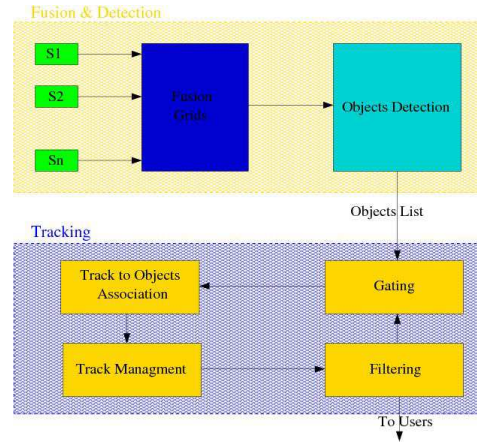


Fig. 1. Architecture of the perception system

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.

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.

The rest of the paper is organized as follows. In the next section, we present the 2 demonstrators used in this work. Description of first level of architecture and Sensor Data Fusion Strategy are described in Section III. Second level is detailed in section V. Experimental results are given in Section IV for the first level and in Section VI and finally in Section VII conclusions and future works are discussed.

II. DEMONSTRATORS

In this section, we present the 2 demonstrators that have been used to test and validate our generic architecture: a Daimler demonstrator and a Volkswagen demonstrator.

A. The Daimler demonstrator



Fig. 2. Left: the Daimler demonstrator vehicle. Right: the Volkswagen demonstrator vehicle.

The Daimler demonstrator vehicle¹ (Fig. 2) is equipped with a camera (only for visualization purpose), two short range radar sensors and a lidar. The radar sensor is with a maximum range of 30m and a field of view of 80°. The radar data are processed by a car supplier and we use the list of detected moving objects as input of our architecture. 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 sensors. The measurement cycle of the sensor system is 40ms.

B. The Volkswagen demonstrator

The Volkswagen demonstrator vehicle² (Fig. 2) is equipped with a Lidar with a field of view of 160° and a maximum range of 150 m. It has 161 laser beams called channels and resolution of 1°. Other sensors installed on this demonstrator include a stereo vision camera, four short range radars (SRR) one at each corner of the vehicle and a long range radar (LRR) in front of the vehicle. Stereo vision processing was done by an other partner [8] of the project.

III. FIRST LEVEL

In this section, we first detail the description of the first level of our architecture: Environment Mapping & Localization, Moving Objects Detection. In the last subsection, we describe the fusion between objects detected by several sensors.

A. Environment Mapping & Localization

To map the environment and localize in the environment, we propose an incremental mapping approach based on occupancy grid to map the environment and a fast and robust method to localize the vehicle in its environment.

1) Environment mapping using Occupancy Grid Map:

Using 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. 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 $P(m|x_{1:t}, z_{1:t})$ for each cell of grid m , given observations $z_{1:t} = \{z_1, \dots, z_t\}$ from time 1 to time t at corresponding known poses $x_{1:t} = \{x_1, \dots, x_t\}$ from time 1 to time t .

Here we apply Bayesian Update scheme [10] that provides an elegant recursive formula to update the posterior under log-odds form:

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

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

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

The desired probability of occupancy, $P(m|x_{1:t}, z_{1:t})$, can be easily recovered from the log-odds representation. Since the updating algorithm is recursive, it allows for incremental map updating when new sensor data arrives.

2) Localization of the vehicle in the 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 [1] is a popular way to correct vehicle location. Unfortunately, sparse data and dynamic entities in outdoor environment cause problems of correspondence finding in ICP-style methods which affect the accuracy of matching results.

To solve this problem, we used a particle filter. We predict different possible positions of the vehicle (ie, one position of the vehicle corresponds to one particle) using the motion model and compute the likelihood of each position (ie, the probability of each particle) using the laser data and a sensor model. 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 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 .

For the motion model, we adopt the probabilistic velocity motion model similar to that of [10]. The vehicle motion u_t is comprised of two components, the translational velocity v_t and the yaw rate ω_t .

¹This demonstrator has been used in the framework of the European project PReVENT-ProFusion: www.prevent-ip.org/profusion

²This demonstrator has been used in the framework of the European project Intersafe2: www.intersafe-2.eu

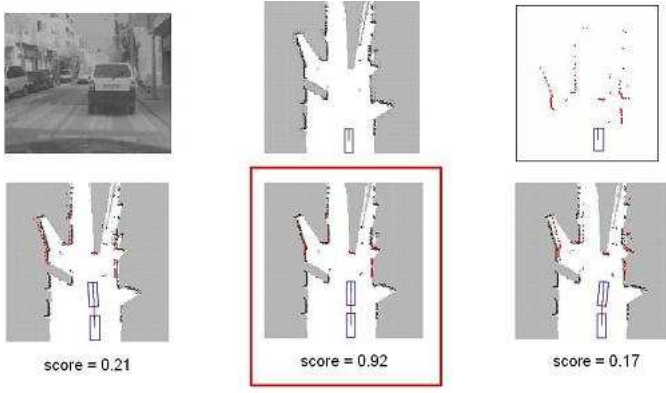


Fig. 3. An example of localization on the Daimler demonstrator

For the measurement model $P(z_t | x_t, M_{t-1})$, mixture beam-based model is widely used in the literature [6]. However, the model comes at the expense of high computation so 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 that its projected end-point falls into. 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 estimated at time t (follows (1)), we can write the measurement model under the sum following:

$$P(z_t | x_t, M^{t-1}) \propto \sum_{k=1}^K \{P(M_{hit_t^k}^{t-1}) | M_{hit_t^k}^{t-1} \text{ is occupied}\} \quad (3)$$

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.

The resulting pose \hat{x}_t of equation 2 is then used to generate a new map M_t according to (1):

$$M_t = M_{t-1} \cup \{\hat{x}_t, z_t\} \quad (4)$$

An example of the localization algorithm is shown in Figure 3. The color of each grid map cell indicates its probability of being occupied: gray=unknown, white=free, black=occupied. 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. In this example, we have 3 hypothesis (ie, 3 particles) and we compute the probability of these 3 particles using laser data and a sensor model. The particle in the bottom middle has the highest score, so it corresponds to the real position of the vehicle. At each point in time, only one local map is maintained 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 initialized.

B. Moving Objects Detection

After a consistent local grid map of the vehicle is constructed, moving objects can be detected when new laser measurements arrive by comparing with the previously constructed grid map. The principal idea is based on the inconsistencies between observed free space and occupied space in the local 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 it can be static or dynamic and we set the unknown status for the object in this case.

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. We finally compute the centroid of each detected object.

The output of detection process consists of a list of detected objects $D_{objects}^t = \{O_D\}$ where $O_DF = (r_D, \theta_D)$ where r_D is the range of the detected object and θ_D is the angle of the detected object in polar coordinate system.

C. Fusion between several sensors

In this work, we use multiple sensors on the vehicle to observe the surrounding environment, since a single sensor can provide only a limited view of the environment. But using multiple sensors inherently requires to perform fusion between them at some appropriate level so as to get optimal results. The most common way to perform fusion is after tracking [4]. In our case, sensors are embedded on a moving vehicle at high speed and due to field of view and range of sensors, so moving objects quickly go out of sight before they are confirmed as tracks. So we perform fusion before tracking (ie, at detection level).

In this subsection, we give details of object detection level fusion between two or more sensors. As input to the fusion process we have two or more lists of objects. We believe that an object detection level fusion between these two lists can complement each other thus giving more complete information about the states of objects in the environment. If we have more than 2 lists of detected objects, we proceed them 2 by 2. This fusion process consists of following two steps:

1) *Object Association*: In this step, we determine which detected objects of the first list are to be associated to which detected objects of the second list using nearest neighbor technique. A detected object of the first list O_1^i is associated to a detected object O_2^j of the second list if $dist(r_1^i, r_2^j) < thr_r$ where r is the maximum depth between r_1^i and r_2^j and θ_1^i is close to θ_2^j .

The positional uncertainty of a detected object increases with depth, so we have defined a distance threshold function $thr(r)$ based on the depth as:

$$thr(r) = 0.10 \times r$$

Here, to take into account uncertainty in depth of detected objects, we consider that the distance in depth between two detected objects should be less than the 10% of the depth of the further detected object.

2) *Position information fusion*: This step works on the pair of objects associated with each other in the previous step and fuses their position information (range and θ). We model the position uncertainty using 2D Gaussian distribution for both objects. Suppose $P_1 = [r_1, \theta_1]^T$ is the centroid position of an object detected by the first sensor and R_1 the associated covariance and suppose that $P_2 = [r_2, \theta_2]^T$ is the centroid position of the same object detected by the second sensor and R_2 the associated covariance. Using Bayesian fusion, the probability of fused position $P = [r_F, \theta_F]^T$ and R the associated covariance are given as:

$$P = \frac{P_1/R_1 + P_2/R_2}{1/R_1 + 1/R_2} \quad \text{and} \quad 1/R = 1/R_1 + 1/R_2$$

3) *Fusion Output*: The result of this fusion process is a new list of fused objects. This list also has all objects detected by the first (resp. second) sensor which could not be associated with objects detected by the second (resp. first) sensor. We keep unassociated objects because they may correspond to dynamic objects which may not have been detected by all the sensors due to occlusion or noise on the sensor. More formally, the output of fusion process consists of fused list of objects $F'_{objects} = \{O_F\}$ where $O_F = (r_F, \theta_F, SensorCount)$. For each object, we have position (centroid) information and a count for number of sensors detecting this object.

IV. EXPERIMENTAL RESULTS ON FIRST LEVEL

A. SLAM + Moving Objects Detection

The results of local SLAM and detection of moving objects on the Daimler demonstrator are shown in Figure 4. In our implementation, the width and height of local grid map are set to 80 m and 200 m respectively, and the grid resolution is set to 20 cm. Every time the vehicle arrives at 40 m from the grid border, a new grid map is created. The local SLAM and DATMO is run for every new laser scan.

The images in the first row represent online maps and objects moving in the vicinity of the vehicle are all detected and tracked. The current vehicle location is represented by blue box along with its trajectories after corrected from the odometry. The red points are current laser measurements that are identified as belonging to dynamic objects. The green boxes indicate detected moving objects with corresponding tracks shown in dark-yellow. The second row are images for visual references to corresponding situations.

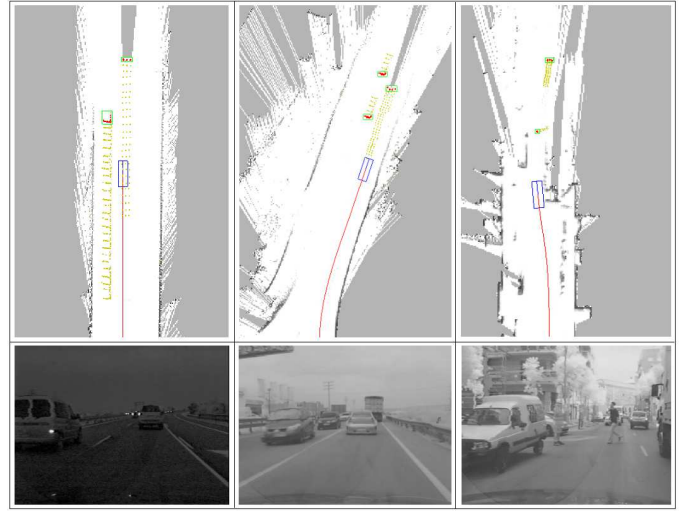


Fig. 4. Experimental results show that our algorithm can successfully perform both SLAM and DATMO in real time for different environments.

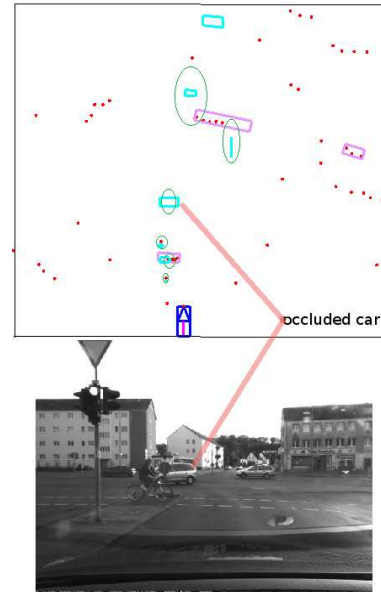


Fig. 5. Laser and Stereo vision objects fusion, see the text for details.

B. Fusion

Figure 5 shows an interesting fusion scenario between stereo-vision and lidar on the Volkswagen demonstrator, objects shown in cyan color are the objects detected by stereo vision whereas the objects shown by light violet rectangles are the laser detected objects, red dots are raw laser impact points. An oncoming car that was being detected by laser until last scan is now occluded by a cyclist whereas stereo system was able to detect this car. So this mis detection by laser will be filled in by stereo during fusion hence giving a smooth track. The increased position uncertainty with depth for stereo vision objects can also be seen in the figure (green ellipses).

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]. Regarding tracking techniques, Bayesian filters [2] 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 describe the four different parts of our architecture (figure 1) to solve the different parts of multi-objects tracking: gating, association, track management and filtering.

More details about these different parts are outlined next.

A. Gating

In this part, taking as input predictions from previous computed tracks and newly detected objects, a gating is performed. It consists in, according to an arbitrary distance function, determine the detected objects which can be associated with tracks. Also during this stage, clustering is performed in order to reduce the number of association hypothesis. It consists in making clusters of tracks which share at least one detected object. In the next stage, association can be performed independently for each cluster decomposing a large problem in smaller problems which induce generation of less hypothesis.

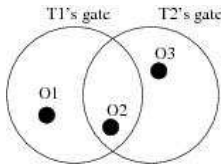


Fig. 6. Example of association problem

If we take as an example the situation depict by the Fig. 6, in this stage one set is computed as T_1 and T_2 share object O_2 . Also according to gates, objects O_1 and O_2 can be assigned to T_1 and objects O_2 and O_3 to T_3 .

B. Association

In this part, taking as input clusters of tracks and detected objects validated by the gating stage, association hypothesis are evaluated. By considering likelihood of objects with tracks, new track apparition probability and non-detection probability, an association matrix is formed.

Let be $L(o_i, t_j)$ the function giving the likelihood of object i with track j , P_{NT} the new track apparition probability and P_{ND} the non detection probability. Taking as an example the situation in the Figure 6, the association matrix is written:

$$\begin{pmatrix} L(o_1, t_1) & -\infty & P_{NT} \\ L(o_2, t_1) & L(o_2, t_2) & P_{NT} \\ -\infty & L(o_3, t_2) & P_{NT} \\ P_{ND} & P_{ND} & -\infty \end{pmatrix}$$

Thus a possible association hypothesis corresponds to a valid assignation in the matrix of detected objects with tracks *i.e* one unique element in each row and each column is chosen to compose the assignation. In order to reduce the number of hypothesis, only the m-best association hypothesis are considered. The m-best assignment in the association matrix are computed using the Murty method [7] which computes the m-Best assignations in the matrix and by this way we obtain the m-Best Hypothesis.

C. Track management

In this third stage, using the m-Best Hypothesis resulting of the association stage, the set of tracks, is maintained *i.e* tracks are confirmed, deleted or created.

New tracks are created if a new track creation hypothesis appears in the m-best hypothesis. A new created track is confirmed if it is updated by detected objects after a fixed number of algorithm steps (three in our implementation). Thus spurious measurements which can be detected as objects in the first step of our method are never confirmed.

If a non-detection hypothesis appears and so to deal with non-detection cases (which can appear for instance when an object is occulted by an other one, tracks without associated detected objects are updated according to their last associated objects and next filtering stage becomes a simple prediction. But if a track is not updated by a detected object for a given number of steps, it is deleted.

D. Filtering

In this filtering stage, according to previously computed predictions, estimations are performed for each association of all hypothesis and new predictions are computed for the gating stage. Regarding filtering techniques, there exists several kinds of filters, the most classical is the well known Kalman filter. But in all kinds of filters, the motion model is the main part of the prediction step. Since in dynamic outdoor scenario there may be different types of objects (vehicles, motor bikes, pedestrians etc) moving in different directions using different motion modes, a single motion model based filtering technique is not sufficient.

To deal with these motion uncertainties, Interacting Multiple Models (IMM) [9] have been successfully applied in several applications [2]. The IMM approach overcomes the difficulty due to motion uncertainty by using more than one motion model. The principle is to assume a set of motion models as possible candidates of the true displacement model of the object at one time. To do so, a bank of elemental filters is ran at each time, each corresponding to a specific motion model, and the final state estimation is obtained by merging the results of all elemental filters according to the distribution probability over the set of motion models (in the next part we note μ this probability). By this way different motion models are taken into account during filtering process.

We have seen that four motion models (constant velocity, constant acceleration left turn and right turn) are sufficient to successfully track objects in dynamic outdoor environment. We use four Kalman filters to handle these motion models.

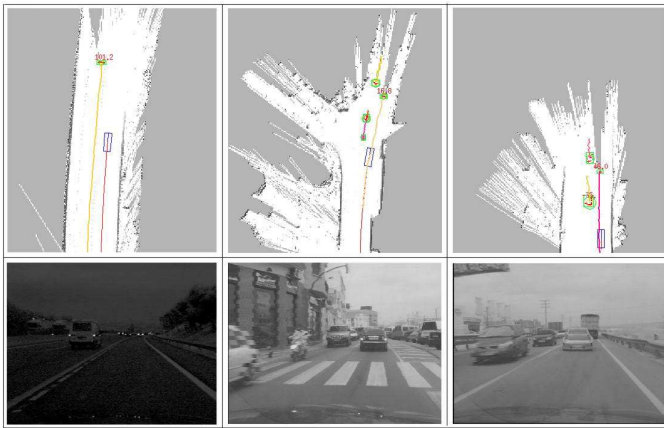


Fig. 7. Experimental results show that our algorithm can successfully perform both SLAM and DATMO in real time for different environments

Finally the most probable trajectories are computed by taking the most probable branch and we select one unique hypothesis for one track tree.

VI. EXPERIMENTAL RESULTS ON SECOND LEVEL

Fusion and tracking results on the Daimler demonstrator are shown in Figure 7. In this figure, the difference with figure 4 is the estimation of the trajectory of each tracked object: information on velocities is displayed next to detected objects if available.

In Figure 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.

Figure 8 shows an interesting scenario on the Volkswagen demonstrator at intersection, left image shows the tracking results based only on laser data, the car behind the cyclist is occluded in last few frames giving insufficient impact points to be detected as a moving object, the right image shows the tracking with fusion, car was also partially detected by stereo vision and hence in the fused results it was tracked successfully.

VII. CONCLUSIONS AND FUTURE WORKS

We have presented an approach to accomplish online mapping and moving object tracking simultaneously. Exper-

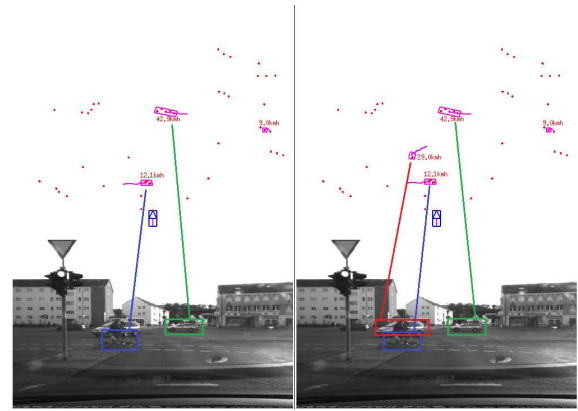


Fig. 8. Fusion and tracking results. Left(laser only): car occluded by cyclist is not being tracked. Right(laser and stereo): with fusion car was tracked successfully.

imental results³ have shown that our system can successfully perform a real time mapping of the local environment, localization of the ego vehicle and moving objects detection and tracking from a vehicle at high speeds in different dynamic outdoor scenarios.

This architecture has also been integrated, tested and validated on 3 industrial demonstrators: a Daimler demonstrator, a Volkswagen demonstrator and a Volvo truck [5] as well.

Future works include integrating information from camera to perform classification of moving objects (pedestrians, bicycles or motorcycles, cars and bus or truck) and predicting future trajectories of ego vehicle and other moving objects to detect potential collisions.

REFERENCES

- [1] P. Besl and N. McKay. A method for registration of 3d shape. *Trans. Pattern Analysis and Machine Intelligence*, 12(2), 1992.
- [2] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 2000.
- [3] A. Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.
- [4] Nikos Floudas, Panagiotis Lytrivis, Harris Avgoustidis, Malte Ahrholdt, George Thomaidis, and Angelos Amditis. Track based multi sensor data fusion for collision mitigation. In *IEEE International Conference on Fusion*, 2008.
- [5] Ruben Garcia, Olivier Aycard, Trung-Dung Vu, and Malte Ahrholdt. High level sensor data fusion for automotive applications using occupancy grids. In *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2008.
- [6] D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7):579–598, 2003.
- [7] K. G. Murty. An algorithm for ranking all the assignments in order of increasing costs. *Operations Research*, 16:682–687, 1968.
- [8] S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, S. Bota, I. Haller, C. Pantile, M. Drulea, and C. Golban. *On Board 6D Visual Sensors for Intersection Driving Assistance Systems*. Advanced Microsystems for Automotive Applications, Springer, 2010.
- [9] 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.
- [10] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [11] C.-C. Wang. *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.

³More results and videos can be found at <http://emotion.inrialpes.fr/~aycard/Videos>.