

Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments

Trung-Dung Vu, Olivier Aycard and Nils Appenrodt

Abstract—In this paper, we present a real-time algorithm for online simultaneous localization and mapping (SLAM) with detection and tracking of moving objects (DATMO) in dynamic outdoor environments from a moving vehicle equipped with laser sensor and odometry. 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 using Global Nearest Neighborhood (GNN) method. The experimental results on datasets collected from different scenarios such as: urban streets, country roads and highways demonstrate the efficiency of the proposed algorithm.

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.

In this context, the paper proposes to solve SLAM and DATMO in dynamic outdoor environments with a vehicle moving at high speed using laser sensor and odometry. Recently, there are considerable research efforts focusing on these problems [12], [7], [16], [17]. 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.

To model the environment surrounding the vehicle, we use the Occupancy Grid framework developed by Elfes [5]. Compared with feature-based approaches [8], grid maps can represent any environment and are specially suitable for noisy sensors in outdoor environments where features are hard to define and extract. Grid-based approaches also



Fig. 1. The DaimlerChrysler demonstrator car.

provide an interesting mechanism to integrate different kinds of sensors in the same framework taking the inherent uncertainty of each sensor reading into account.

In general, in order to perform SLAM and DATMO from a moving vehicle, a precise vehicle localization is essential. Direct scan matching techniques like ICP [2] and its variants [13] are popular ways to correct vehicle location from odometry. The most evident flaw of these ICP-style scan matching methods is that the measurement uncertainty is not taken into account. Especially, 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 overcome these problems, we introduce a new fast incremental scan matching method based on a local grid map and it does not need to find corresponding features and can work reliably in dynamic environments. When good vehicle locations are estimated, by integrating sensor measurements we are able to build a consistent grid map surrounding of the vehicle. And by comparing new measurements with the constructed local vehicle map, dynamic objects can be detected and tracked.

The proposed algorithm for solving SLAM and DATMO is tested on data collected from the DaimlerChrysler demonstrator car equipped with a camera, two short range radar sensors and a laser scanner (Fig. 1). In addition, vehicle odometry information such as velocity and yaw rate are provided by the vehicle sensors. In this paper we use only data from laser scanner and odometry. Images from camera are for visualization purpose. Experimental results show that our algorithm can perform both SLAM and DATMO in real

T.D. Vu and O. Aycard are with LIG & INRIA Rhône-Alpes, Grenoble, France, (e-mails: Trung-Dung.Vu@inrialpes.fr, Olivier.Aycard@imag.fr)

N. Appenrodt is with DaimlerChrysler AG, Research and Technology, Department REI/AU, Ulm, Germany, (e-mail: Nils.Appenrodt@daimlerchrysler.com)

time for different types of dynamic outdoor environments.

The rest of the paper is organized as follows. In the next section, we describe our approach to mapping and localization in dynamic outdoor environments. Algorithm for detecting and tracking moving objects is presented in Section III. Experimental results are given in Section IV and finally in Section V conclusions and future works are discussed.

II. LOCALIZATION AND MAPPING

For safety navigation applications, basically 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 scan matching algorithm which involves in building 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.

A. Notation

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 .

B. 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 [5], Dempster-Shafer [11] and Fuzzy Logic [10]. Here we apply Bayesian Update scheme [15] 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

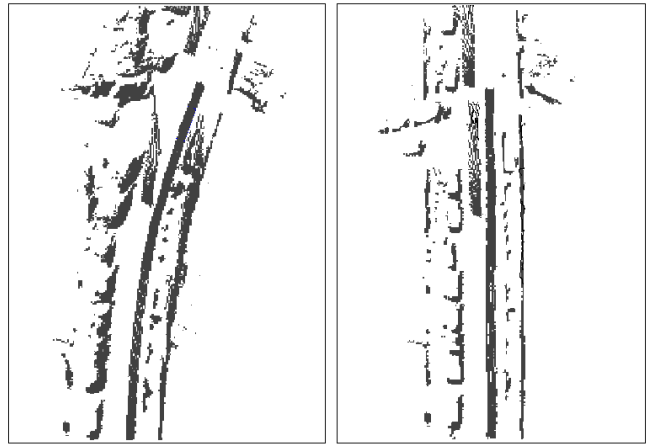


Fig. 2. Hit maps build directly from raw laser data collected from a vehicle moving along a straight street: with vehicle localization using odometry (left); and using results of scan matching (right). Note that the scan matching results are not affected by moving objects in the street. See Fig. 6 for the resulting occupancy grid map.

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

Fig. 3b shows an example of an occupancy grid map, incrementally constructed from laser measurements during the vehicle's movement. The color of each grid map cell indicates the probability that corresponding space being occupied: gray=unknown, white=free, black=occupied.

C. Scan Matching against Occupancy Grid Map

In order to build a consistent map of the environment, good vehicle localization is required. Because of the inherent error, using only odometry often results in unsatisfied maps (see Fig. 2 left). When features can not be defined and extracted, direct scan matching techniques like ICP [9] can help to correct the odometry error. The problem is that sparse data in outdoor environments and dynamic entities make correspondence finding difficult. One more important disadvantage of the direct scan matching methods is that they do not consider dynamics of the vehicle. Indeed we have implemented several ICP variants [13] 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 dynamic 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 [14], [7]. 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. Mathematically, we calculate a

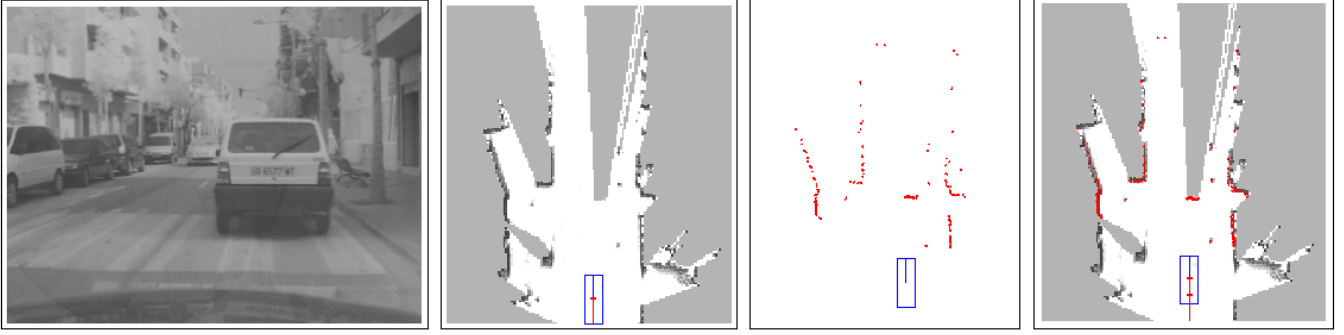


Fig. 3. An example of scan matching. From left to right: reference image; local map created so far M^{t-1} and previous vehicle pose x_{t-1} ; laser measurement at time t ; and matching result is obtained by trading off the consistency of the measurement with the map and the previous vehicle pose.

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 . 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 [15]. The vehicle motion u_t is comprised of two components, the translational velocity v_t and the yaw rate ω_t . Fig. 4 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.

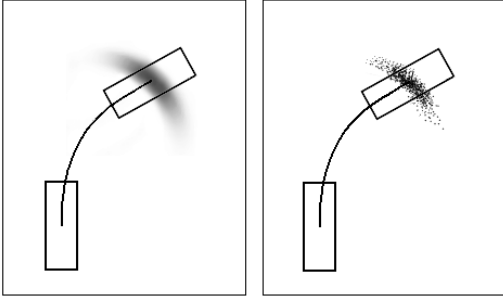


Fig. 4. The probabilistic velocity motion model $P(x_t | x_{t-1}, u_t)$ of the vehicle (left) and its sampling version (right).

For the measurement model $P(z_t | x_t, M^{t-1})$, mixture beam-based model is widely used in the literature [6], [7]. 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 cells corresponding to the projected end-points. 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 $P(M_i^t)$ denote the posterior probability of occupancy of the grid cell M_i estimated at time t (following (1)), we can write the measurement model under the sum following:

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

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 [14], [7] 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. 4 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

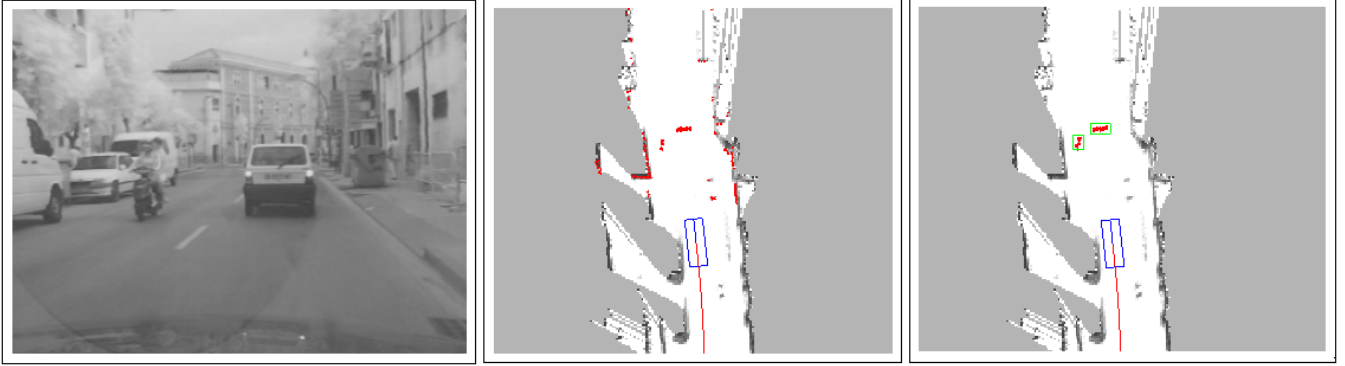


Fig. 5. Moving object detection example. See text for more details.

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 conventional PC. An example of scan matching result is shown in Fig. 3. 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 (see Fig. 2 right). Since we only consider occupied cells, spurious regions in the occupancy grid map that might belong to dynamic objects do not contribute to the sum (4). The voting scheme ensures that measurement likelihood reach a maximum only when the measurement is aligned with the static parts of the environment. To some meaning, measurements from dynamic entities can be considered as outliers. This property is very useful for moving object detection process that will be described in the next section.

D. 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 initialized. 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.

III. OBJECT DETECTION AND TRACKING

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 location.

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

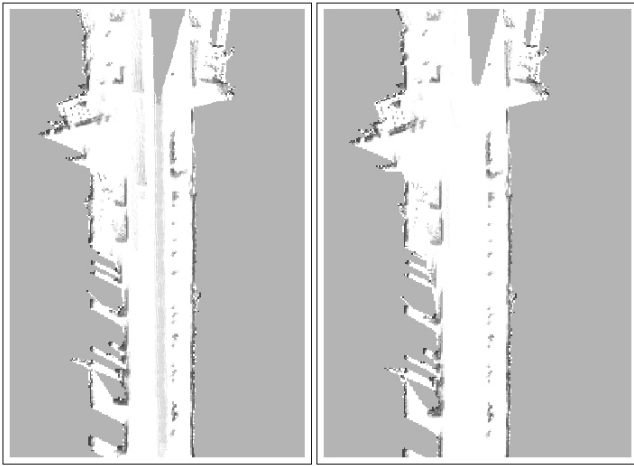


Fig. 6. Occupancy grid maps built with and without filtering out detected moving objects.

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. Fig. 6 shows two occupancy grid maps constructed from the same laser data in Fig. 2 with and without filtering out dynamic measurements. We can see that the left one built without the filtering step results in many fuzzy regions.

Once we are able to detect moving objects we need to track them in order to estimate their states and predict their behaviors in the future. Tracking multiple moving objects is a classical problem. In the general case this problem is very hard, however it has been shown experimentally that simple methods are good enough to cope with urban scenarios [16]. In our current work, a simple object tracking scheme as described in [1] using Global Nearest Neighborhood (GNN) and Kalman filter is employed to track detected objects. A replacement using MHT [4] and Adaptive IMM [3] is ongoing.

IV. EXPERIMENTAL RESULTS

Our proposed approach for SLAM and DATMO is tested on datasets collected with the DaimlerChrysler demonstrator car using only laser sensor and odometry. The maximum measurement range of laser sensor is 70 m with a horizontal field of view of 160° and a resolution of 1° . The vehicle was driven through different kinds of scenarios such as city streets, country roads and highways with a maximum speed of 120 kph and the data is collected every 40 ms.

In our implementation, the width and height of local grid map are set to 160 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 results of local SLAM and DATMO are shown in Fig. 7. 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.

In Fig. 7, the leftmost column depicts a highway scenario where the demonstrator car is moving at a very high speed of about 120 kph while two cars moving in the same direction in front of it are detected and tracked. In the middle is the situation where the demonstrator is moving at 80 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. In the third situation, the demonstrator moving quite slowly at 25 kph in a crowded city street. Our system detects and tracks both the pedestrian moving in front of the vehicle and the car moving far ahead. Temporary stationary objects like another standing pedestrian and several other cars parked nearby are considered as static objects. 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 with data cycle in real time.

V. 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 with moving object detection and tracking from a vehicle moving 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 build, moving objects are detected and tracked reliably.

Future works include integrating information from radar and exploring how radar can help to improve the moving object detection and tracking process. We also intend to incorporate object models and road models that give a more meaningful representation of detected objects with specific shapes and sizes instead of only sets of contour points as in our current work.

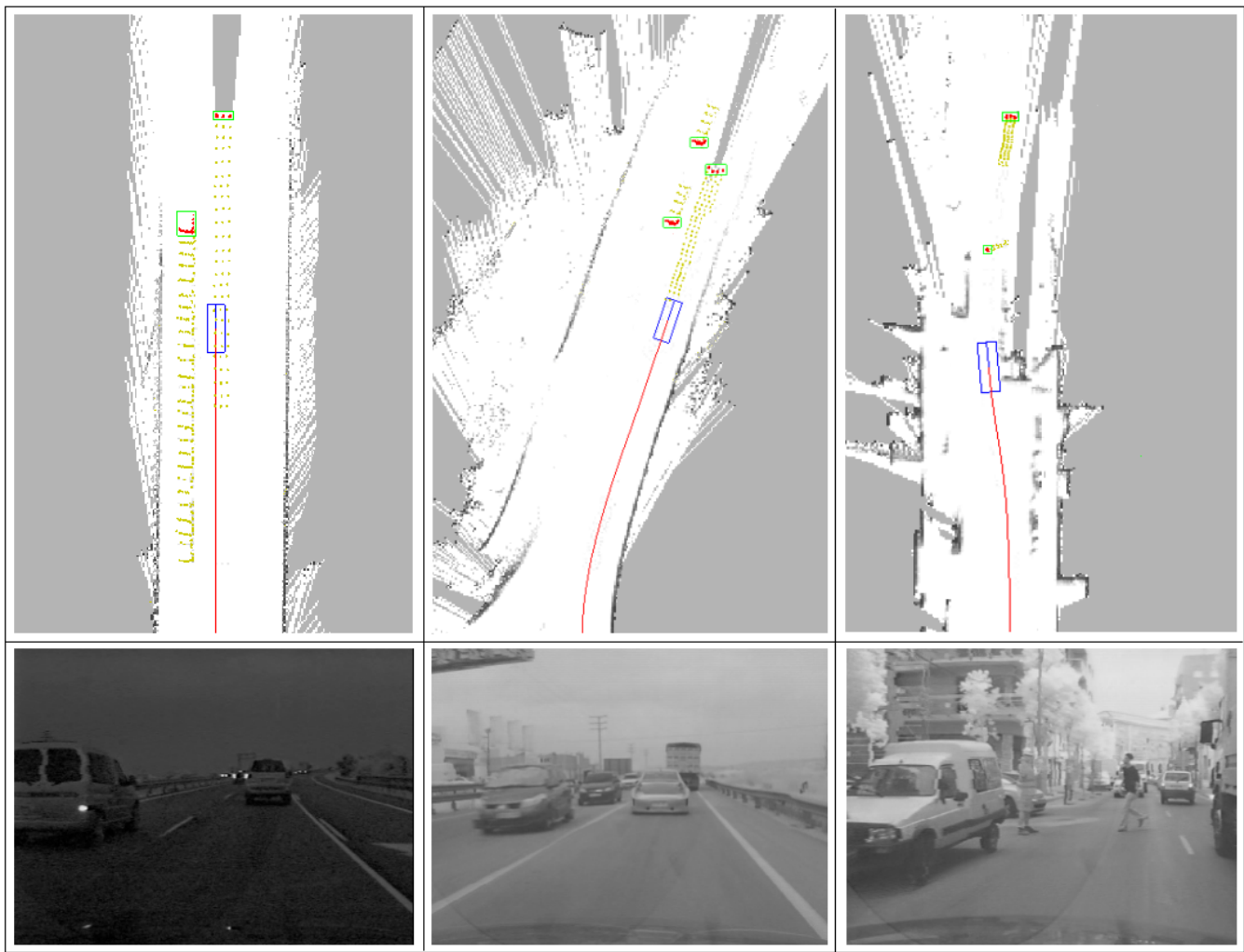


Fig. 7. Experimental results show that our algorithm can successfully perform both SLAM and DATMO in real time for different environments.

VI. ACKNOWLEDGMENTS

The work is supported by the European project PREVENT-ProFusion¹

REFERENCES

- [1] O. Aycard, A. Spalanzani, J. Burlet, C. Fulgenzi, T.D. Vu, D. Raulo, and M. Yguel. Pedestrian tracking using offboard cameras. 2006.
- [2] P. Besl and N. McKay. A method for registration of 3d shape. *Trans. Pattern Analysis and Machine Intelligence*, 12(2), 1992.
- [3] J. Burlet, O. Aycard, A. Spalanzani, and C. Laugier. Adaptive interactive multiple models applied on pedestrian tracking in car parks. 2006.
- [4] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation and evaluation of reid's multiple hypothesis tracking algorithm for visual tracking. *Pattern Recognition, 1994. Vol. 1*, 1, 1996.
- [5] A. Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.
- [6] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.
- [7] D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7):579–598, 2003.
- [8] J. Leonard and H. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. *IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, 3, 1991.
- [9] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 1997.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. *Third Intl. Conf. on 3D Digital Imaging and Modeling*, 2001.
- [14] 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.
- [15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [16] C.-C. Wang. *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.
- [17] Denis F. Wolf and Gaurav S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19, 2005.

¹<http://www.prevent-ip.org/profusion>