Improving Moving Objects Tracking Using Road Model For Laser Data

Qadeer Baig
University of Grenoble1
Grenoble, France
Qadeer.Baig@imag.fr

Olivier Aycard
University of Grenoble1
Grenoble, France
Olivier.Aycard@imag.fr

***Abstract*—In this paper we have presented a fast algorithm to detect road borders from laser data. Two local search windows, one on right side of the host vehicle and the other on left, are moved right and left respectively from the current position of vehicle in map. A score function is evaluated to know the presence or absence of the road border in current search window. We have used the detected road border information to reduce false alarms in our previous work on DATMO (detection and tracking of moving objects). We also show how these information can be used to infer drivable area and the presence of intersections on the road. Results on data sets obtained from real demonstrator vehicles show that this technique can be successfully applied in real time.**

***Keywords*-SLAM; DATMO; Perception; Occupancy Grid; Road Border Detection; Laser Scanner**

I. Introduction

Perceiving or understanding the environment surrounding a vehicle is an important step in driving assistance systems and for the functioning of autonomous vehicles. It is of utmost importance to be aware of other static or dynamic objects present in the environment along with the host vehicle. Obtention of this knowledge by vehicle is termed as environment perception. Broadly, environment perception includes: getting relative positions of static objects (also called map of the environment), finding position of host vehicle in this map and detecting moving objects and obtaining their properties (position, velocity, direction of movement etc). Intelligent vehicles use on-board sensors, off-board sensors or a combination of them to accomplish surrounding environment perception task.

This environment perception task can be divided into two sub problems: SLAM (Simultaneous Localization And Mapping) and DATMO (Detection And Tracking of Moving Objects). SLAM addresses the problem of an intelligent vehicle navigating an unknown environment. While navigating the environment, the intelligent vehicle seeks to acquire a map thereof, and at the same time it wishes to localize itself using its map [6]. It is regarded as one of the most important problems in the way of building a truly autonomous intelligent vehicle. While SLAM provides the vehicle with a map of the environment, DATMO allows the vehicle to be aware of dynamic entities around, tracking them and enabling the application to predict their future behaviors.

In our previous works [2], [9] we have presented a generic architecture to solve these problems. Our solution using this architecture is based on environment representation using occupancy grids [3] framework. The results of this solution are satisfactory when high resolution laser scanners are used ($<= 0.5°$) as environment perception sensor. However with noisy laser scanners, although solution to SLAM part gives acceptable results but in general we observe many false alarms for detection and tracking of moving objects part. One solution that we have presented [1], especially in an intersection like scenario, is to use multiple sensors (stereo vision along with laser scanner) and perform fusion at object detection level to reduce false alarms and miss detections. But it will be quite attractive if some solution could be presented without requiring an additional sensor.

We have observed that the laser scanner on demonstrator used for current work has following problems due to noise: Data appear vibrating and have reflections from white markings on the road. On the right side near the right edge of the host vehicle especially, the edges of road border appear moving and tend to be curved. It also has longitudinal uncertainty that causes the objects to appear as clutter of points rather than having some uniform outline. From these observations we learned that a successful detection of road borders in laser data can help remove a significant number of false alarms detected on and beyond road borders. This road border detection should be efficient enough to let both SLAM and DATMO run on-line in real time without requiring any extra hardware modifications. In this work we present a simple to implement but efficient road border detection algorithm and use it to reduce false alarms in our DATMO solution.

Many people have already worked on road border detection, especially in vision, lane detection has been a hot field of research. However vision techniques usually based on features extraction are slow, in [4] Kirchner has mentioned pros and cons of usual vision techniques used for this purpose. Moreover using laser scanner he has presented a model based technique for detecting and tracking road borders. Based on clothoid model and Kalman filter it involves many calculations during each step, moreover this work assumes the absence of moving objects in the environment making it infeasible for our work. In [12] Wijesoma presents another

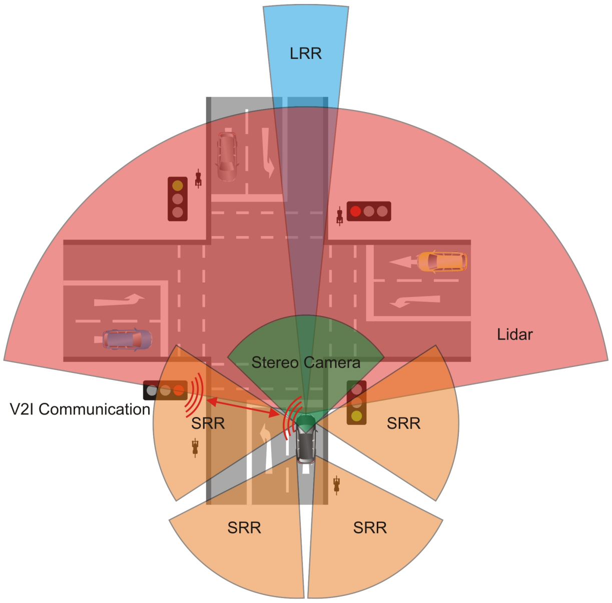

Figure 1. The Volkswagen demonstrator vehicle

method for road border detection based on special laser setup along with strict flat road plan assumptions. In their setup laser is mounted on the vehicle such that its beams hit the road plane at few meters from the vehicle, since road is taken as strictly flat, the laser hit points appear as a line on the ground but this line has sharp edges on the borders due to curbs. The algorithm finds these sharp shifts in the data to detect road borders. Due to the fact that this system requires special orientation of laser scanner and strictly flat road, makes it inappropriate for our purpose. Work by Weiss [11] is closest to what we have done in this paper. However rather than finding road borders they find drivable free space in-front of host vehicle in occupancy grid maps by moving multiple line segments in-front of vehicle in both left and right sides.

The rest of this paper is structured as follows: In next section we present our demonstrator used to get data sets for current work. Section III summarizes our solution of SLAM and DATMO. Our road border detection process is detaild in section IV and its application to false alarm reduction in section V. Results are discussed in section VI and we conclude this work with some perspectives in section VII.

II. Experimental Setup

In this section, we present the demonstrator that has been used to test and validate our work: a Volkswagen demonstrator.

The Volkswagen demonstrator vehicle[1] is equipped with a Lidar with a field of view of 160° with 1° resolution and a maximum range of 150 meters. 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. Our work on this demonstrator is concerned with the processing of Lidar data to solve SLAM and DATMO problems.

III. Grid Based SLAM and DATMO

In this section we summarize our occupancy grid based solution to SLAM and DATMO problems [2], [1]. At any instant of time t the input consists of laser scan z^t and odometry information $u_t = (\nu, \omega)$ (translational and rotational velocities).

A. SLAM

We have used incremental mapping approach based on laser scan matching algorithm to build a local vehicle map. Based on occupancy grid representation the environment is divided into a two dimensional lattice of rectangular cells and we keep track of probabilistic occupancy state of each cell. We build a grid map of $90m \times 108m$ with each cell having dimensions of $0.3m \times 0.3m$. Environment mapping is essentially the estimate of posterior probability of occupancy $P(m \mid x_{1:t}, z_{1:t})$ for each cell of grid m, given observations $z_{1:t} = \{z_1, ..., z_t\}$ from time 1 to time t at corresponding known poses $x_{1:t} = \{x_1, ..., x_t\}$, here $z_t = \{P_i\}$ where $P_i = (x = r_i \cos\theta_i, y = r_i \sin\theta_i)$ is the impact point (or detected target position) by ith laser beam (from its polar coordinates r_i, θ_i) for $\forall i \leq 161$, and $x_t = (x, y, \theta)$ is the vehicle pose. To know these pose values we need to solve the localization problem. We have used a particle filter based on importance sampling for this purpose (as explained in [8]), a total of 300 particles are used. For the given previous pose x_{t-1} and current odometry information u_t we sample different possible positions of the vehicle (i.e. one position of the vehicle corresponds to one particle) from the motion model $P(x_t | u_t, x_{t-1})$. Then we compute the probability of each position (ie, the probability of each particle) using the laser data and a sensor model. Practically, from the pose of each particle, we calculate the fraction of the laser beams terminating in occupied cells of the grid map constructed so far. The pose of the particle getting highest probability is taken as true pose.

B. DATMO

Moving objects detection process consists of classifying the laser impact points in the current laser scan as dynamic or static by comparing them with the grid map constructed so far. The principal idea is based on the inconsistencies between observed free space and occupied space in the local map. Laser impacts observed in free space are classified as dynamic whereas the rest are classified as static. Once the classification is done we do a segmentation step to make objects from these observed dynamic points. From

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

these information bounding rectangles of these objects are calculated. Because of sensor noise we get false alarms on this step where a static object is classified as dynamic. We use road border detection technique developed in next section to reduce the number of false alarms detected in this step. Once the road borders have been detected the objects lying on or beyond road borders are taken as false alarms and hence are ignored for tracking. In our previous work [1], we used multi sensor data fusion at this point to achieve this purpose.

For tracking we have used MHT [5] based data association alongwith Interacting Multiple Models (IMM) filtering technique. The details of this technique can be found in our other published work [10].

IV. Road Border Detection

In this section we present details of our road border detection technique which consists of representing the grid map in an intermediate form called hit grid, removing noise, defining a search window, a score function and details about how these search windows are evaluated in an efficient way to find road border.

Since we do not use special setup of laser scanner for detecting road borders, our technique is based on the hypothesis that road curbs are frequently detected by the laser mounted on the host vehicle while driving on the road. If there are parked cars alongside the road we can effectively take them as road border markers or end of drivable road area. Our road border finding process consists of following steps:

1) *Hit grid (HG) representation*; along with the occupancy grid of local map we generate another grid, called hit grid. This grid has same number of cells and spans the same area as that of occupancy grid but each cell of this grid keeps only a count of laser hits detected in area corresponding to this cell. With each new scan reading from laser scanner the count of hits keeps on increasing in cells corresponding to static parts of environment. An advantage of this grid is that, no complicated update formulas need be evaluated for each new observation. The update step only involves incrementing the count of a cell where a laser hit is detected in new observation. If we represent this HG at time instance t as H^t then the value of a cell at the intersection of row i and column j is represented as H^t_{ij}. The update process for this grid consists of finding corresponding cell H^t_{ij} for each laser hit point $z^t_{k=1:161}$ and incrementing its count.

2) *Noise removal*; to filter false hits caused by sensor error and moving objects we perform two operations on each cell of the grid: first, we do the smoothing by adding to the value of each cell the counts of its eight neighbors (a classic smoothing technique in vision), second, we pass this new grid, denoted as S^t and called smoothed hit grid, through a high pass filter. We reset the count value to zero for those cells of S^t which have values less than filter's cut-off value defined as $F_{thr} = Max(S_{ij})/10$. This removes the erroneous values and the cells having count values greater than zero belong to areas having static objects including road borders. Formally these operations are given as:

$$S^t_{ij} = \sum_{i-1}^{i+1}\sum_{j-1}^{j+1} H^t_{kl} \tag{1}$$

$$S_{ij} = \begin{cases} S_{ij} & \text{if } S_{ij} >= F_{thr} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

3) *Search window*; since we suppose that host vehicle is always (excluding when it is on an intersection) oriented parallel to road border, so instead of searching road border everywhere in the grid we search only specific areas in the hit grid. These areas to be searched lie on left and right side of the host vehicle. So we define two search windows of length l and width w (in our experiments $l = 20$ and $w = 1$, in meters, give good results), one on left and the other on right side of the host vehicle. Initial positions of these windows are at 1 meter on right and 1 meter on left of host vehicle. Then in each search iteration the left window is moved 0.9 meter left (maximum 25 iterations) and right window is moved 0.9 meter right (maximum 15 iterations) and a score function is evaluated for the hit counts present in the cells lying under search window. Finally a threshold value is used to decide the presence or absence of road border. In each direction the first window that shows the presence of a road border stops the search in that direction. Figure 2 shows host vehicle along with some search windows in the smoothed hit grid.

4) *Score function*; we have defined a very simple score function that can be evaluated very fast. According to this function the score of a search window W consists of the sum of hits of all cells lying under this window:

$$Score(W) = \sum S^t_{ij} | S^t_{ij} \in W \tag{3}$$

5) *Detailed algorithm*; Before going into our model details we will define following mathematical notations (as defined in [7]): The pose (position with orientation) of an object j (real or hypothetical) with respect to a reference frame i is represented as $P_{ij} = [x_{ij}, y_{ij}, \theta_{ij}]^T$. We also define following compounding pose relations for pose transformations: if P_{ij} is the pose of object j w.r.t object i and $P_{jk} = [x_{jk}, y_{jk}, \theta_{jk}]^T$ is the pose of object k w.r.t j then the pose of k w.r.t i denoted as $P_{ik} = [x_{ik}, y_{ik}, \theta_{ik}]^T$ is given as:

$$P_{ik} \equiv \oplus(P_{ij}, P_{jk}) = \begin{bmatrix} x_{jk}\cos(\theta_{ij}) - y_{jk}\sin(\theta_{ij}) + x_{ij} \\ x_{jk}\sin(\theta_{ij}) + y_{jk}\cos(\theta_{ij}) + y_{ij} \\ \theta_{ij} + \theta_{jk} \end{bmatrix}$$

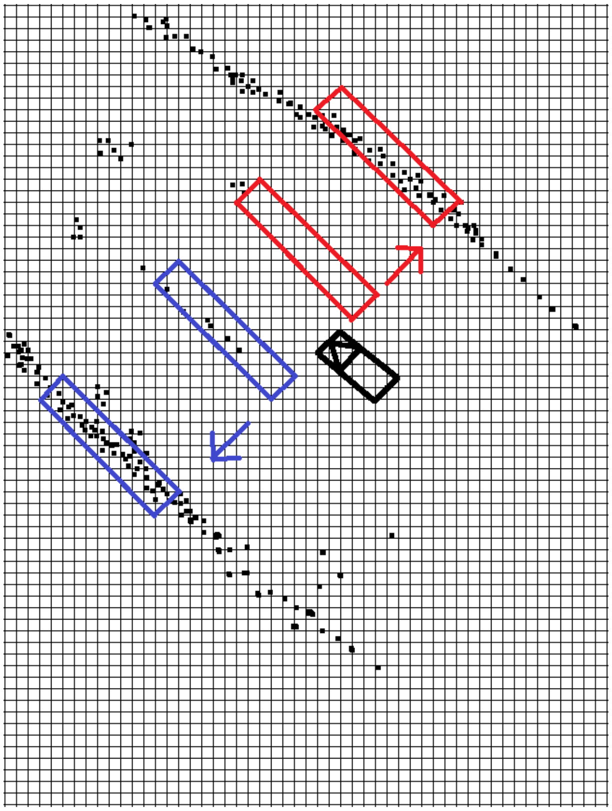

Figure 2. Position of host vehicle and two left (blue) and two right (red) search windows in smoothed hit grid. The two search windows for each left and right side include the initial window and the one with highest score. Windows with highest score lie on road border. Cells having black dots have hit count values non zero. Arrows show the direction of movement of their respective windows.

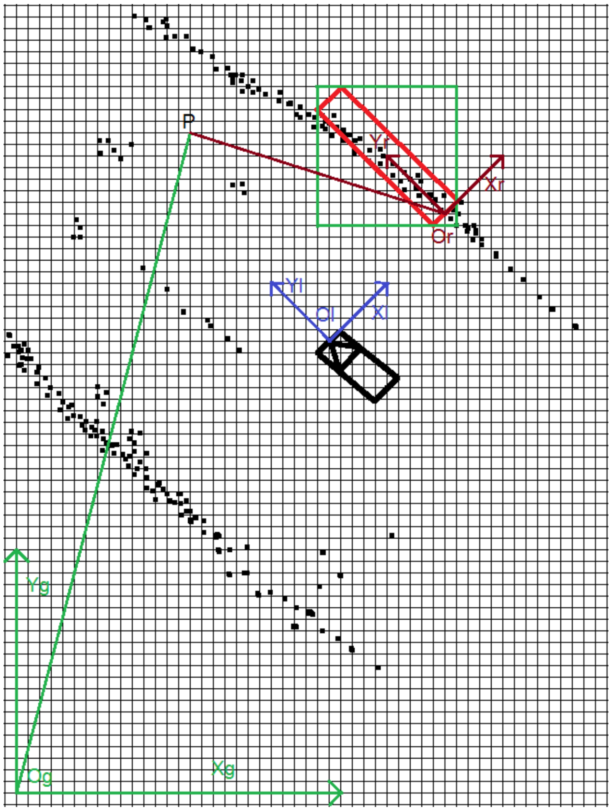

Figure 3. Evaluation of score function for red rectangle. Three reference frames belong to smoothed hit grid O_g, laser frame of reference O_l, and search rectangle frame of reference O_r. Our objectives are to find the coordinates of point P, given in O_g reference frame, w.r.t O_r reference frame and to find row and column of bottom left and top right corners of green rectangle. Cells within green rectangle will be searched only to find the cells lying under red rectangle.

For the pose P_{ij} the reverse pose relationship $P_{ji} = [x_{ji}, y_{ji}, \theta_{ji}]^T$ (pose of i w.r.t j) is defined as:

$$P_{ji} \equiv \ominus(P_{ij}) = \begin{bmatrix} -x_{ij}\cos(\theta_{ij}) - y_{ij}\sin(\theta_{ij}) \\ x_{ij}\sin(\theta_{ij}) - y_{ij}\cos(\theta_{ij}) \\ -\theta_{ij} \end{bmatrix}$$

Considering figure 3, our objective is to find cells lying under red rectangle (our search window) so that score function can be evaluated for them. To proceed we have three frames of reference: grid frame of reference with origin O_g shown in color green, laser frame of reference with origin O_l shown in color blue and search window frame of reference with origin O_r shown in color dark red. The pose of O_l w.r.t O_g is known and is $P_{gl} = [x_{gl}, y_{gl}, \theta_{gl}]^t$. Similarly pose of O_r w.r.t O_l denoted as $P_{lr} = [d, 0, 0]^t$ is also known, because we always take it on X_l at a distance d (that increases with each search iteration by 0.9 meters).

Coordinates of four corners of red rectangle w.r.t O_r are fixed and given as $(-0.5, 0)$, $(0.5, 0)$, $(-0.5, 20)$ and $(0.5, 20)$. So a point $P(x_{lp}, y_{lp})$ lies in red rectangle if $-0.5 <= x_{lp} <= 0.5$ and $0 <= y_{lp} <= 20$. Each cell can be represented by its middle point $P = [x_{gp}, y_{gp}]^t$, the coordinates of this point can be easily calculated w.r.t O_g from its row and column using following equations

$$x_{gp} = col * CellSize + CellSize/2 \tag{4}$$

$$y_{gp} = row * CellSize + CellSize/2 \tag{5}$$

But to calculate the coordinates of same point P w.r.t O_r i.e $P = [x_{rp}, y_{rp}]^t$ we need following operations: Pose of O_r w.r.t O_g:

$$P_{gr} = \oplus(P_{gl}, P_{lr}) \tag{6}$$

Pose of O_g w.r.t O_r:

$$P_{rg} = \ominus(P_{gr}) \tag{7}$$

Pose of P w.r.t O_r:

$$P_{rp} = \oplus(P_{rg}, P_{gp}) \tag{8}$$

Using these relations we can calculate the coordinates of middle point of any gird cell w.r.t O_r and then we can easily check if this point lies in the red rectangle or not.

Now we have the tool to know if a cell lies under red rectangle or not. So to evaluate the score function one brute force method can be to loop on all cells of the grid, evaluate above equations on them and sum the score for cells which are in red rectangle and ignore the others. But it will be quite inefficient to evaluate all the cells when the red rectangle is very small as compared to whole grid. To solve this problem we will restrict our search to cells in green rectangle, but first we need to calculate row and column of bottom left and top right corners of this rectangle. To calculate this, suppose $P_{rp} = [x_{rp}, y_{rp}, 0]^t$ is the pose of a corner of red rectangle w.r.t O_r the pose of same corner w.r.t O_g can be calculated using following relation:

$$P_{gp} = \oplus(P_{gr}, P_{rp}) \tag{9}$$

This way we can calculate coordinates of all four corners of the red rectangle w.r.t O_g. Then minimum of the four x and y values will give bottom left corner of green rectangle w.r.t O_g and maximum of the four x and y values will give top right corner. Finally using reverse transforms of equations 4 and 5 we can calculate rows and columns of cells on these corners.

The window giving highest score greater than a threshold value is taken as belonging to road border and the line connecting the middle points of this search rectangle (red) along the length is taken as road border.

V. Applying Road Borders to Detect False Positives

After we have detected the road borders, the process of removing false positives from the detected moving objects in the current laser scan consists of following steps:

1) See if the detected object lies to the left or to the right of the host vehicle. This is important to check it against appropriate road border.

2) Transform the rectangle (only four corner points) of the detected moving object to the reference frame of the search window of the detected road border on the left or right side depending on the inferred position of the object in step 1. To do this, suppose the pose of one corner point P of the object rectangle w.r.t O_g is $P_{gp} = [x_{gp}, y_{gp}, 0]^t$. Then the pose of this point w.r.t O_r denoted as $P_{rp} = [x_{rp}, y_{rp}, 0]^t$ is given by equation 8. Then this point is a false positive if $x_{rp} >= -0.5$ for right road border or $x_{rp} <= 0.5$ for left road border. We repeat this process for all four points of object rectangle, this object is taken as valid moving object if none of its corner points is a false positive.

3) If object is a false positive then remove it from the list of detected dynamic objects before passing to tracking module.

VI. Results

Figure 4 shows the results of smoothing process of a scene. In the smoothed hit window road borders are easily visible. Two moving cars in front of host vehicle have no traces in the smoothed grid.

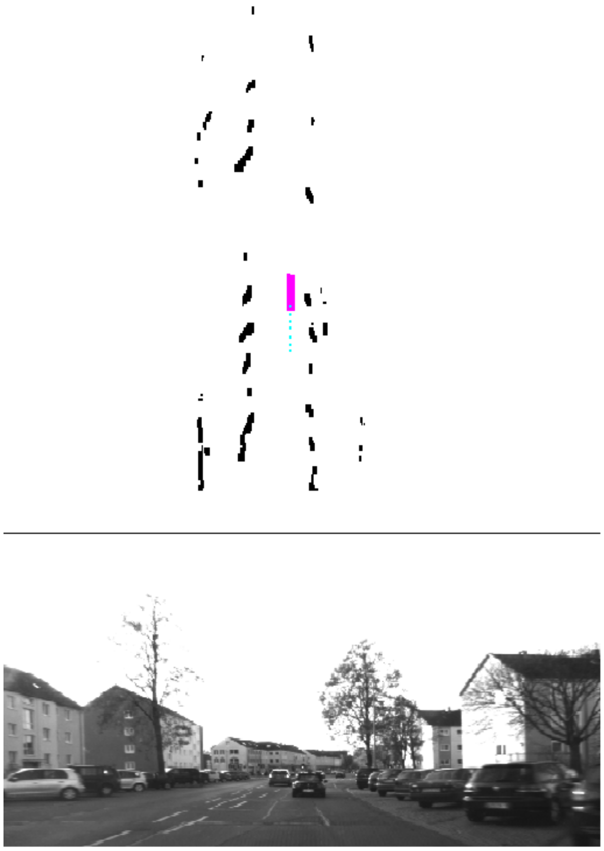

Figure 4. Smoothed hit grid (top) with host vehicle (pink rectangle) for an actual road scene (bottom), smoothing process removed the laser hits detected on moving cars.

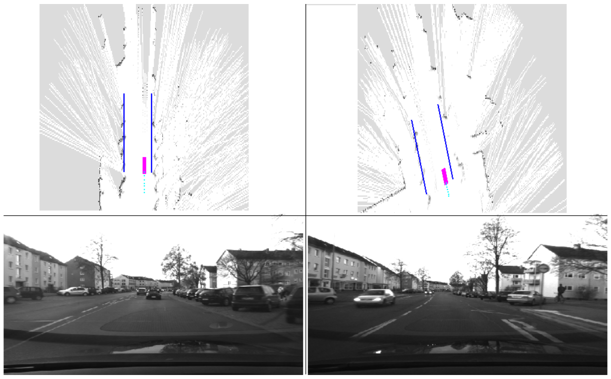

Figure 5. Detected road borders shown in map view for two scenarios.

Results of road border detection for two scenarios are shown in figure 5. The detected road borders are shown in map view instead of smooth hit grid.

Figure 6 shows the detected road borders in moving objects tracking view. Red dots are the current laser hits and

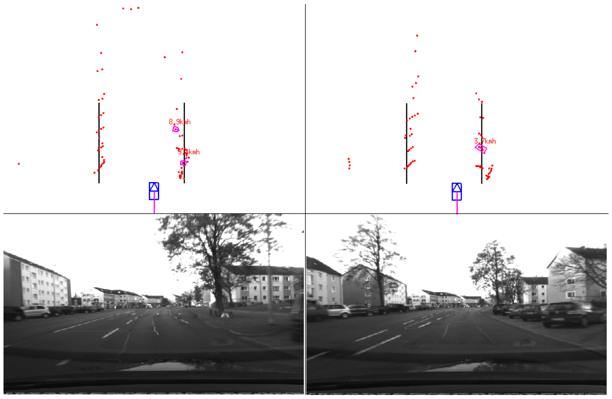

Figure 6. False positives (static objects detected as moving due to sensor noise) detected on road borders, with this knowledge of road borders we can successfully identify them as false positives.

Table I
Statistics of false positives (FP) with and without road border information for three data sets.

Data Set	FP without road border	FP with road border
1	70	45
2	103	80
3	150	95

small pink rectangles with speed captions are the detected moving objects by our DATMO. In both of these examples they are false alarms due to noisy laser data. It can be seen that these false positives lie on the detected road borders, which allows us to infer them as false positives and filter them out. In these figures they are shown before this filtration just to show them as false positives.

Table I shows empirically observed quantitative results for detected false positives using road border information.

Another important result is the detection of intersections on road. Our simple hypothesis in this regard is that, if no road borders are detected at some instance then host vehicle is on an intersection. Some states of intersection detection are shown in table II. Some result videos can be seen here [2].

Table II
Statistics of intersection detection against ground truth for three data sets.

Data Set	Total intersection	True positives	False positives
1	2	2	1
2	1	1	1
3	4	4	4

VII. Conclusion and Perspective

In this paper we have presented our work on road border detection in laser data. The presented technique, based on search window and a score function, is very fast and runs in real time along with our SLAM and DATMO solution. After detecting road borders we have used them to remove false moving objects detected on and beyond road borders. This have improved the tracking of moving objects.

In future we plan to improve this technique to detect legal moving objects on or beyond road borders, like pedestrians walking on the pavement.

References

[1] Q. Baig, O. Aycard, TD. Vu, and T. Fraichard. Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario. In *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011.

[2] Q. Baig, TD. Vu, and O. Aycard. Online localization and mapping with moving objects detection in dynamic outdoor environments. In *IEEE Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, August 2009.

[3] A. Elfes. *Occupancy grids: a probabilistic framework for robot percpetion and navigation*. PhD thesis, Carnegie Mellon University, 1989.

[4] A. Kirchner and Th. Heinrich. Model based detection of road boundaries with a laser scanner. 1998.

[5] D. B. Reid. A multiple hypothesis filter for tracking multiple targets in a cluttered environment. Technical Report D-560254, Lockheed Missiles and Space Company Report, 1977.

[6] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.

[7] R. Smith and P. Cheeseman. On the representation of spatial uncertainty. *International Journal of Robotic Research*, 1987.

[8] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

[9] 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.

[10] TD. Vu, J. Burlet, and O. Aycard. Grid-based localization and local mapping with moving objects detection and tracking. *International Journal on Information Fusion, Elsevier*, 2009. To appear.

[11] T. Weiss, B. Schiele, and K. Dietmayer. Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids. In *Intelligent Vehicles Symposium (IV)*, Istanbul, Turkey, June 2007.

[12] W.S. Wijesoma, K.R.S. Kodagoda, A.P. Balasuriya, and E.K. Teoh. Road edge and lane boundary detection using laser and vision. In *International Conference on Intelligent Robots and Systems*, Hawaii, USA, November 2001.

[2] https://sites.google.com/site/qadeerbaig/road-border-stuff