



**HAL**  
open science

# Automotive LIDAR objects Detection and Classification Algorithm Using the Belief Theory

Valentin Magnier, Dominique Gruyer, Jérôme Godelle

► **To cite this version:**

Valentin Magnier, Dominique Gruyer, Jérôme Godelle. Automotive LIDAR objects Detection and Classification Algorithm Using the Belief Theory. IV 2017 - IEEE Intelligent Vehicles Symposium, Jun 2017, Los Angeles, United States. 6p, 10.1109/IVS.2017.7995806 . hal-01675243

**HAL Id: hal-01675243**

**<https://hal.science/hal-01675243>**

Submitted on 4 Jan 2018

**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.

# Automotive LIDAR objects Detection and Classification Algorithm Using the Belief Theory

Valentin MAGNIER<sup>1</sup>, Dominique GRUYER<sup>1</sup> and Jerome GODELLE<sup>2</sup>

**Abstract**—In Autonomous driving applications, the LIDAR is becoming one of the key sensors for the perception of the environment. Indeed its work principle which is based on distance ranging using a laser beam scanning the environment allows highly accurate measurements. Among sensors commonly used in autonomous driving applications, which are cameras, RADARs and LIDARs, the LIDAR is the most suited to estimate the shape of objects. However, for the moment, LIDARs dedicated to pure automotive application have only up to four measurement layers (4 laser beams scanning the environment at different height). Hence objects detection algorithm have to rely on very few layers to detect and classify the type of objects perceived on the road scene, that makes them specific.

In this paper we will present an Detection and Tracking of Moving Objects (DATMO) algorithm featuring an object-type classification based on the belief theory. This algorithm is specific to automotive application therefore, the classification of perceived vehicles is between bike, car and truck. At the end of this paper we will present an application of this algorithm in real-world context.

## I. INTRODUCTION

The current automotive trends are mainly focused on the development of automated driving functionality. While for SAE Level 1 and Level 2 of automation, automakers have managed to keep using sensors that are common in the recent automotive industry (RADARs and cameras), the full perception of the environment needed for Level 3 and above requires a more precise estimation of the shape of the objects surrounding the ego-vehicle. For that reason, most of the recent autonomous vehicles projects (Hyundai, Valeo, Audi,...) rely on one or more lidar to achieve an accurate perception of the obstacles. By its conception, a LIDAR sensor provides raw data which take the form of point clouds representing the location of every laser impact on the scene's objects.

Because of cost reductions and also for long range perception, automakers often prefer using 4 layers LIDAR (such as the IBEO Lux or the Valeo Scala) which have a very narrow vertical field of view: 0.8deg between each layer. That allows the perception of obstacles at a high distance with at least one layer (at a distance 100m the vertical distance between each beam is only 1.12m, that allows two

laser beams encountering a car located at 100m of the ego-vehicle). The horizontal precision of the scanning is even better with 0.25deg between each laser beam giving four points to estimate a car's width located at 100m.

The point cloud provided by LIDAR sensors enables to estimate the shape of the objects, moreover, if among these objects some are classified as vehicles their heading angle can be estimated. The heading angle is a very useful information for client applications because, provided a suited evolution model, it allows to estimate the trajectory of these vehicles.

This paper presents a DATMO algorithm based on a 2D point cloud which aims at detecting, tracking, and classifying the vehicles of the road scene. The Belief theory is used for the classification of vehicles through a specific multi-criteria combination operator. This operator combines different features of perceived objects so as to classify them between three categories: bike, car and truck. Objects that are classified as vehicle will be tracked internally with a specific evolution model.

## II. OVERVIEW OF THE ALGORITHM

Our algorithm can be classified among the category of DATMO (detection and tracking of moving objects) algorithms based on LIDAR raw data. Most of the work on LIDAR DATMO applied to automotive context are relatively new since these specific sensor have been considered for mass market only recently (near 2010, with the Valeo Scala). Authors of [5] gives an exhaustive overview of relevant LIDAR DATMO algorithms and present their own approach. They also present the layout of most of LIDAR DATMO algorithms which is usually composed of four steps:

- Point clustering
- Segmentation
- Data-association
- Track update

Their approach copes with the data-association by using points of the measurements that fall inside a validation region determined by the bounding box of the track. The track is updated using the orientation of the measurement's segments.

A classification step can be added to determine what is the type of perceived objects (pedestrian, bike, car, truck...). Some works such as [1] and [2] are focused on the pedestrians detection, [4] explores the issue of object classification within a LIDAR DATMO algorithm. They propose a voting strategy where various features of perceived objects are taken into account to determine the type of object. [6] presents another algorithm, which does not use a clustering step, however,

<sup>1</sup> Valentin MAGNIER and Dominique GRUYER are with IFSTTAR (French Institute of Science and Technology for Transport, Development and Networks) LIVIC laboratory, 77 rue des Chantiers, 78000 Versailles, France. valentin.magnier@renault.com, dominique.gruyer@ifsttar.fr <sup>2</sup> Jerome GODELLE is with Renault SAS, Aubevoye Technical Center, 27940 Aubevoye, France. jerome.godelle@renault.com

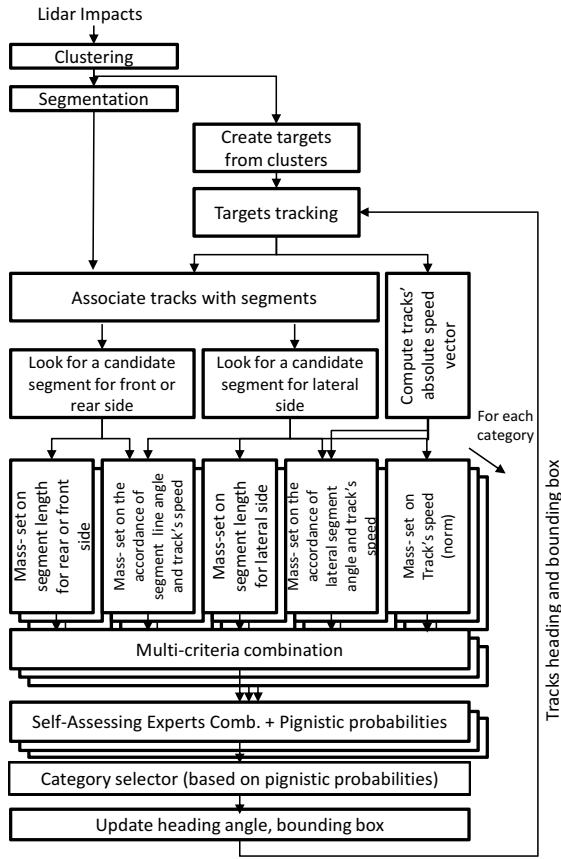


Fig. 1. Data flow overview of our LIDAR DATMO algorithm

they propose a relevant occlusion prediction algorithm. Here again, the object classification is a rule-based scheme. On both of these DATMO algorithms the tracking of objects is achieved thanks to a Kalman filter.

The algorithm proposed in this paper is also composed of five steps. One notice that the objects classification is one of the most difficult issue. To tackle with it, we have chosen to use the Belief theory and using it to merge various criteria in order to select the object type among 'bike', 'car' and 'truck'.

Figure 1 shows the architecture and the data flow. A clustering is applied to raw the data point cloud. Each cluster will be considered as a target. For each target (i.e.: each cluster) a segmentation algorithm is applied. It returns a list of segments associated to each target. A target tracking, deals with all targets so as to find their dynamic state and speed vector. The tracks managed by the tracking also store along the time others information, such as the type of object. A classification algorithm based on the belief theory rely on the segments list linked to each target to determine the type of object, if all necessary conditions are satisfied, the track will be updated with the correct object-type.

#### A. Clustering and segmentation

The clustering stage gathers laser impacts which are supposed to belong to the same objects in separated groups.

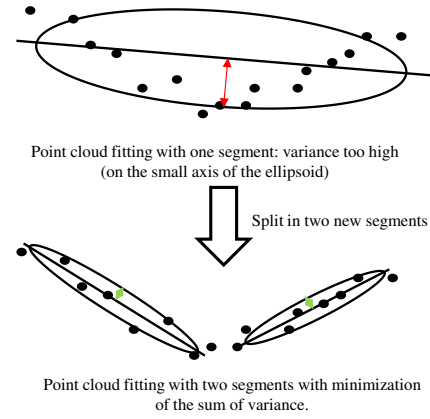


Fig. 2. Visual representation of the segment splitting algorithm

Some information will be calculated for each cluster like the bounding-box's size (on the ego-vehicle's axis coordinates) and its center's position as well as the average distance between the points which used to compute a trust on the target. The clustering algorithm is simply based on the gathering of points which are close to their neighbors. A threshold on the distance between points (calculated with the Mahalanobis distance) is used to build the clusters.

The segmentation algorithm is recursive and based on segment splitting, hence we named it Recursive Best Segment Split (RBSS). This algorithm tries to fit a segment to a group of points. If the points variance (more precisely the small axis of the variance's ellipsoid) is below a determined threshold, the segment is kept, otherwise, it is split in two parts. The splitting points  $i$  is determined by minimizing the sum of the variances of the two new segments' points. Then the two new segments will be analyzed with the same process.

In the following algorithm,  $p_{begin,segment_k}$  corresponds to the first point of  $segment_k$  and  $p_{end,segment_k}$  to its last point.

#### Algorithm 1 Recursive Best Segment Split algorithm

```

while  $p_{end,segment_k} \neq p_{end,cluster}$  do
  if  $var(p_{begin,segment_k} \dots p_{end,segment_k}) > threshold$ 
  then
    find  $i$  such as:
    
$$argmin \left( \begin{array}{l} var(p_{begin,segment_k} \dots p_{i,segment_k}) \\ + var(p_{i+1,segment_k} \dots p_{end,segment_k}) \end{array} \right)$$

     $segment_k \leftarrow p_{begin,segment_k} \dots p_{i,segment_k}$ 
     $segment_{k+1} \leftarrow p_{i+1,segment_k} \dots p_{end,segment_k}$ 
     $listof\ segments \leftarrow listof\ segments + segment_{k+1}$ 
  else
     $k \leftarrow k + 1$ 
  end if
end while

```

This algorithm has been built following observations made in [7]. Authors of this paper, after having reviewed some of the most used segmentation algorithm (Split-and-Merge, Line Regression, Incremental, RANSAC), conclude that the

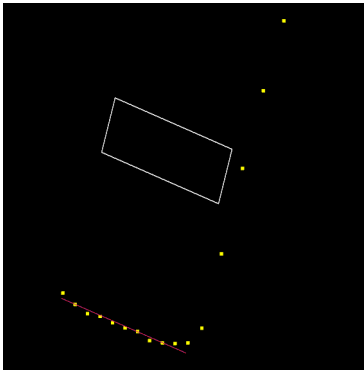


Fig. 3. RBSS segment fitting (red line) on LIDAR impacts (yellow dots) on the rear of a vehicle

Split-and-Merge and Incremental algorithms have good performance in speed and accuracy. However the Incremental algorithm has the drawback to have a different behavior according to the direction of the point cloud parsing. Our algorithm is therefore closer to the Split-and-Merge algorithm. However, the simple Split-and-merge algorithm, has the drawback to be sensible to clutter because the splitting point is determined by the maximum normal distance from the point to the segment. Our algorithm however deals with that by computing the minimal sum of variances of the two parts resulting of the the original segment split. Therefore, our algorithm is less sensible to noise. Finally, in the RBSS algorithm, the segment orientation is determined by the angle of the long axis of the segment's variance ellipsoid and not joining the begin and end points of a segment again to avoid the effect of noise.

Experimental results show a good robustness of the RBSS algorithm (as picture 3 shows). It is well suited to find car's side, rear or front based on point cloud even if is the car has a curved shape.

### B. Targets tracking

The targets tracking ensure the following of perceived objects along the time. Objects returned at each time stamp by the tracking algorithm are called "tracks". We remind briefly the three steps of a tracking algorithm:

- 1) Prediction: positions of tracks are calculated at the current time-stamp.
- 2) Data-association: each target is associated with a corresponding track. This step is one of the main issue of tracking algorithm.
- 3) Correction: The tracks' position are updated with the targets' one.

For our implementation, we chose a simple Nearest Neighbor tracking algorithm which associates targets with tracks only by selecting the target which is the closest to a track.

Nevertheless, we add another stage between the Data-Association and the Correction, which we called Object Classifier. The purpose of this stage is to use the information of the segmentation to modify if necessary the way the tracks will be updated according to the result of the object

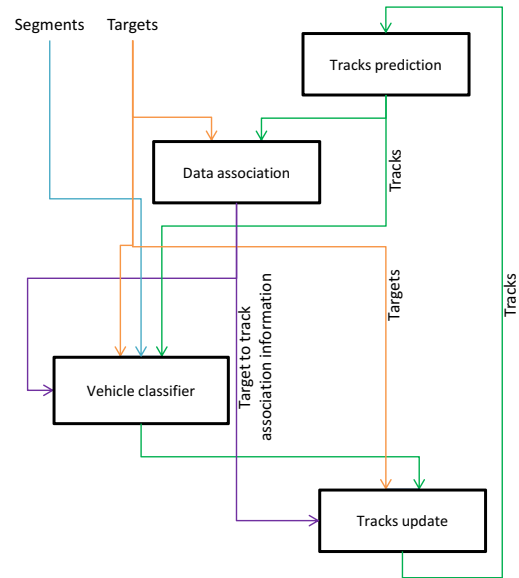


Fig. 4. Tracking and classifier architecture

classification algorithm. This stage is located just after the data-association stage because it requires information from the targets, the corresponding list of segments and from tracks (speed vector, current classification state,...).

## III. OBJECT CLASSIFICATION USING THE BELIEF THEORY

### A. The Belief Theory

The Belief Theory, also known as Dempster-Shafer Theory, aims at assessing the truth of hypothesis (which are basic assumption that can be qualified with a degree of confidence) by combining sources of information. Hypotheses are often denoted "H". The degree of confidence of an hypothesis is called a "mass" and is denoted  $m(H)$  such as  $m(H) \in [0, 1]$ . If the mass is equal to 1, the hypothesis is supposed to be true. The fact that a mass on a hypothesis is null does not mean that the hypothesis is false but that we have no information regarding its truth. A source of information quantifies the confidence of an hypothesis according to the piece of evidence qualifying this hypothesis. The group of all informative hypotheses on which a source of information is able to compute a mass is called "Basic Belief Assignment". The sum of masses on the hypotheses of the BBA is equal to 1. The group of all the individual hypothesis (called "singletons") is named "frame of discernment" is written  $\Omega$ .

$$\Omega = \{H_1, \dots, H_N\}$$

As explained before, one of the main purpose of the Belief Theory is to combine multiple sources of information. To do so, the Dempster conjunctive combination rule is often used (see equation 1) It sums the masses on hypotheses whose conjunction results in the hypothesis on which we want to assess the truth.

$$m_{1,2}(H_z) = \sum_{H_x \cap H_y} m_1(H_x) \cdot m_2(H_y) \quad (1)$$

Notice that the subscripts behind the letter  $m$  refer to the sources of information involved in the determination of the mass.

### B. The multi-criteria operator applied to classification

The Dempster conjunctive combination rule can be generalized to  $K$  sources of information providing information on a common hypothesis :

$$m_{1..K}(H) = \prod_{k=1..K} (1 - m_k(\bar{H})) - \prod_{k=1..K} (m_k(\Omega)) \quad (2)$$

where  $m_k(\Omega)$  is the mass on the “unknown” hypothesis. The “unknown” hypothesis is the default hypothesis, that does not provide any information. It gathers the remaining mass of a BBA in order to get its sum equal to 1. The mass on the unknown hypothesis resulting from the combination of experts can also be calculated from the combination of different sources of information.

$$m_{1..K}(\Omega) = \prod_{k=1..K} (m_k(\Omega)) \quad (3)$$

Finally a new mass can be introduced:  $m_{1..K}(\emptyset)$ . It corresponds to the conflict between the sources of information. Notice that, to be able to generate conflict, the sources of information must be able to provide a mass on the negation of a hypothesis  $m(\bar{H})$ .

$$m_{1..K}(\emptyset) = 1 - \prod_{k=1..K} (1 - m_k(\bar{H})) - \prod_{k=1..K} (1 - m_k(H)) + \prod_{k=1..K} (m_k(\Omega)) \quad (4)$$

We use this multi-sources combination rules as a multi-criteria combination operator. Indeed, while comparing two entities, each criteria of comparison can be considered as an individual source of information assessing the similarity between these two entities. Regarding to our application that is LIDAR objects classification, various features characterizing the perceived objects will be compared to standard values specific to different types of object. For example, the width of the perceived object will be compared to typical width values specific to bikes, cars and trucks. Actually, we do not use only one value as the “typical” but rather a range of values.

For each criterion and for each vehicle type (bike, car, truck) a range of acceptable values is specified. A range is specified with four values “minimum”, “low”, “high” and “maximum”. Beyond “minimum” and “maximum”, the comparison is rejected without doubt, between “low” and “high” the comparison is accepted without doubt, finally in the ranges “minimum” to “low” and “high” to “maximum” the comparison is accepted with a doubt. These ranges

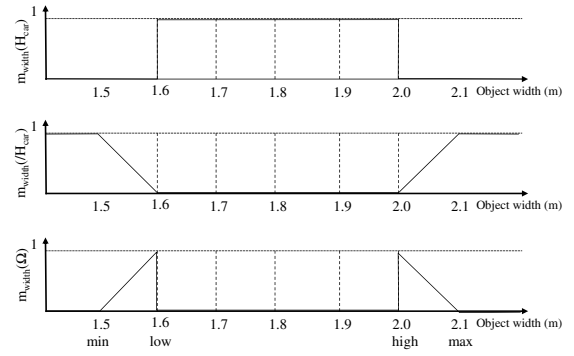


Fig. 5. Mass generative function related to the width of an object for type “car”

are used for mass-generative functions. Mass-generative functions generate a mass-set (triplet of masses) which represents the accordance of the an object’s feature with the values specific to a type of vehicle. Figure 5 shows an example of a mass-generative function dedicated to assess the object’s width for the vehicle type “car”. This mass-generative function produces a triplet of masses on  $\{m_{width}(H_{car}), m_{width}(\bar{H}_{car}), m_{width}(\Omega)\}$ . This triplet is combined with triplets corresponding to the others criteria thanks to the multi-criteria combination operator. The multi-criteria combination results in a mass-set composed of four masses:

$$\left\{ \begin{array}{l} m_{multi-crit.}(\emptyset), m_{multi-crit.}(H_{car}), \\ m_{multi-crit.}(\bar{H}_{car}), m_{multi-crit.}(\Omega) \end{array} \right\}$$

Once one multi-criteria combination mass-set has been calculated for each vehicle type, we need to combine them together in order to get a combined mass-set for each vehicle type. This combined mass-set takes into account the interactions between vehicle’s types and their consistency. For instance, after the multi-criteria combination, if there is mass indicating that an object is classified as a truck and an other mass as a bike at the same time, there is an inconsistency and, more precisely, a conflict. The Self-Assessing Experts Combination enables the detection of conflict [8], moreover, it normalizes the sum of masses on vehicle-type hypothesis (see equations 5, 6 and 7).

$$m_{1..N}(H_i) = m_i(H_i) \prod_{\substack{j=1..N \\ j \neq i}} (m_j(\Omega) + m_j(\bar{H}_j)) \quad (5)$$

$$m_{1..N}(\Omega) = \prod_{i=1..N} (m_i(\Omega) + m_i(\bar{H}_i)) - \prod_{i=1..N} m_i(\bar{H}_i) \quad (6)$$

$$m_{1..N}(\emptyset) = 1 - \sum_{i=1..N} m_{1..N}(H_i) - m_{1..N}(\Omega) + \sum_{i=1..N} m_{1..K}(\emptyset) \quad (7)$$

This normalization is used just after in order to calculate pignistic probabilities. Pignistic probabilities, were introduced by SMET [9] in order to be able to take decision from information originally modeled using the Belief theory. The conversion formula is:

$$BetP(W) = \sum_{\substack{W \subseteq \Omega \\ w \in W}} \frac{1}{|W|} \frac{m(W)}{1 - m(\emptyset)} \quad (8)$$

For our vehicle classification application, equation 8 becomes (example for object type “car”):

$$BetP(H_{car}) = \frac{m(H_{car})}{1 - m(\emptyset)} + \frac{1}{3} \frac{m(\Omega)}{1 - m(\emptyset)}$$

Finally the vehicle’s type is chosen according to the corresponding maximum pignistic probability. Figure 6 presents the complete layout of the vehicle classifier.

### C. Vehicle type assignment

A vehicle-type can be assigned to a track if we have enough clues (criteria) to determine the type. The criteria used in our algorithm are object’s width, object’s length, consistency of the speed vector’s direction with the rear/front side, consistency of the speed vector’s direction with the lateral side and absolute value of its speed.

The results of the RBSS segmentation algorithm provides a list of segments for each track. Among them, we try to find candidate segments for rear and lateral sides for each vehicle type based on comparison of their length and the typical values for specified vehicle type. If one candidate is found for either rear or lateral side, the directions of these candidate segments is compared with the speed vector’s direction of the track. A lateral side segment should be parallel with the speed vector whereas a rear side segment should be perpendicular. A maximum deviation of  $25^\circ$  is tolerated.

Finally the absolute speed of the track is also taken as another criterion.

If an information on a criterion is not available (for example, if only the rear side of a vehicle is perceived, we have no information on the lateral side’s length), the mass-set corresponding to this criterion is set to full “unknown”:  $m_{length}(\Omega) = 1$ , therefore it will have no influence on the computation of the final pignistic probability on the type.

The vehicle type can be assigned when the maximum pignistic probability on a type exceed a determined threshold. The same vehicle-type of a track has to be validated several times before being actually assigned to the track.

When a track has vehicle-type assigned, its motion model is switched to a bicycle motion model in order to improve the tracking precision especially during poor target detections.

Notice that the heading angle of the track is calculated from the angle of its rear, lateral or both segments.

If a vehicle type is already assigned to a track, the tracking can cope with poor target detection (targets with no usable segments), during the data-association stage, by estimating which side of the vehicle has actually been observed, and then process to the track-update stage by taking into account

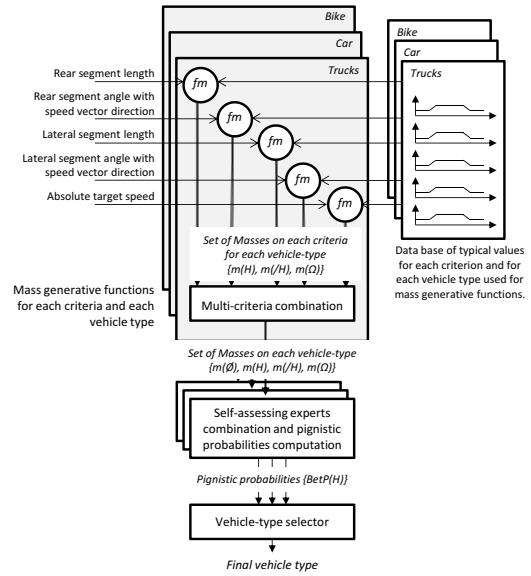


Fig. 6. Vehicle type classifier architecture

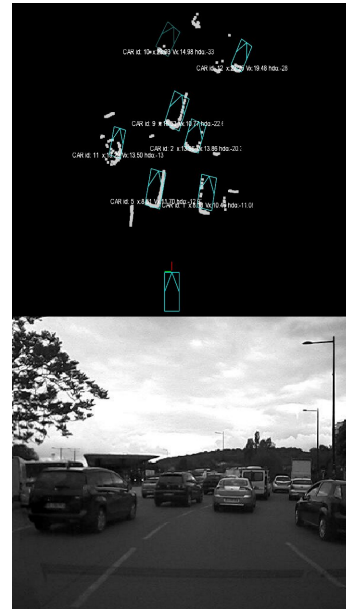


Fig. 7. Driving scenario (rectangles in cyan represent the detected cars)

only the position of the target and not its heading angle. This strategy allows to keep a good tracking despite poor lidar detections.

## IV. EXPERIMENTAL RESULTS

For the tests in real conditions we have used an experimental vehicle (Renault Espace) equipped with an IBEO Lux LIDAR, enabling a  $110^\circ$  horizontal field of view in front of the vehicle. During the scenario, our experimental vehicle is driven on a congested urban road with large crosses (Boulogne-Billancourt near Paris, see Figure 7) with a relatively heavy traffic during approximately 180s.

We have manually classified the vehicles located into the field of view of the lidar sensor. The classification time is

Type	Vehicle	Time Visible	Classif. duration	Classif. OK
car	Citroen C3	00:10.6	00:01.3	yes
car		00:32.1	00:03.3	yes
car	Renault Twingo	00:34.3	00:01.1	yes
minivan	Citroen Berlingo	00:47.3	00:02.0	yes
minivan	Renault Traffic	00:51.7	00:02.2	yes
ped		00:52.0	00:02.8	yes
ped		00:54.0	00:01.2	yes
car	Renault Vel Satis	00:54.8	00:01.9	yes
ped		00:55.1	00:01.8	yes
car	Citroen C3	00:56.3	00:00.5	yes
car	Renault Twingo	00:58.8	00:00.3	yes
car	Peugeot 407	01:01.0	00:00.6	yes
car	VW Golf plus	01:03.7	00:02.3	yes
ped		01:05.5	00:00.8	yes
car	Citroen C3	01:10.7	00:00.2	yes
car	Lancia Ypsilon	01:11.6	00:00.7	yes
minivan	Peugeot Partner	01:14.0	00:01.0	yes
motorbike		01:15.2	00:00.8	yes
car	Renault Espace	01:16.5	00:00.3	yes
car	Peugeot 307	01:16.7	00:00.3	yes
bus		01:20.2	00:00.7	yes
ped		01:18.1	00:00.3	yes
ped		01:20.5	00:01.2	yes
car	Peugeot 1007	01:21.7	00:01.3	yes
car	VW Tiguan	01:26.0	00:01.1	yes
car	Skoda Roomster	01:27.0	00:00.3	yes
car	VW Touran	01:30.2	00:00.2	yes
minivan	Citroen Berlingo	01:31.2	00:00.3	yes
car	VW Polo	01:33.7	00:00.2	yes
car	Toyota Prius	01:35.7	00:00.5	yes
car	Renault Clio	01:37.8	00:00.4	yes
car	Ford Fiesta	01:37.9	00:01.0	yes
car	Renault Laguna	01:40.3	00:01.2	yes
car	Mercedes Benz Vito	01:45.4	00:00.2	yes
car	VW Beetle	01:46.2	00:01.0	yes
car	Ford Fiesta	02:05.9	00:00.9	yes
car	Peugeot 5008	02:04.0	00:01.2	yes
car	Peugeot 508	02:07.9	00:00.4	yes
car	Mercedes Benz Vito	02:07.0	00:01.0	no
car	VW Touran	02:08.4	00:01.9	no
car	VW Polo	01:59.0	00:02.8	yes
minivan	Citroen Berlingo	02:16.4	00:00.7	yes
car	Toyota Yaris	02:18.4	00:00.3	yes
car	Fiat Punto	02:24.2	00:00.5	yes
car	Toyota Yaris	02:40.9	00:01.1	yes

TABLE I  
RESULTS OF DETECTION AND CLASSIFICATION

determined from the moment when a vehicle is fully visible. Indeed, to avoid false alarms we need to initialize tracks only if rear or both rear and lateral sides are visible. Nevertheless, once initialized, track can be maintained even if only the lateral side is visible.

During this scenario we got valid detection rate of 96% (three non-detection of nearly stand still vehicles) and a miss-classification rate of 4% (due to partially hidden cars). The average classification time is 1.0 second. Table I gives the results of our algorithm during the 180s experimental scenario.

These performances are promising for automotive applications, the proposed scenario was complex and difficult to tackle due to the high number of hidden vehicles.

## CONCLUSION

In this work, we have presented an innovative LIDAR DATMO algorithm which is able to classify objects of the road scene. If objects are classified as vehicle, their heading angle and vehicle type is returned. The classification process relies on a segmentation algorithm named RBSS which is well suited for that kind of automotive applications because it is able to deal with noisy data and is not based on random algorithm (as the RANSAC for example). The determination of the vehicle type is achieved using a multi-criteria combination operator based on the belief theory, leading to a pignistic probabilities relative to each type of vehicle.

The tracking algorithm could be improved by using a multi-criteria data-association (like in [3]) taking into account for example the size of bounding boxes in order to avoid the association of a track classified as vehicle with inconsistent targets (that could come from a crash barrier for example).

## ACKNOWLEDGMENT

We would like to thank Benoit LUNETTI for its valuable help in set-up of the experiment. We also thank Renault and the IFSTTAR (french institute on transportation and road-network) for their support on this work.

## REFERENCES

- [1] K.Ch. Fuerstenberg, K.C.J. Dietmayer, and V. Willhoelt. Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. *Intelligent Vehicle Symposium, 2002. IEEE*, 1:31–35, 2002.
- [2] S Gidel, P Checchin, C Blanc, T Chateau, L Trassoudaine, Lamea U M R Ubp-cnrs, Université Blaise Pascal, and Landais Aubière. Pedestrian Detection using a Multilayer Laserscanner. In *RFIA*, Amiens, France, 2008.
- [3] Valentin Magnier, Dominique Gruyer, and Jerome Godelle. Implementation of a Multi-criteria Tracking based on the Dempster-Shafer Theory. *Intelligent Vehicles Symposium (IV), 2015 IEEE*, 2015.
- [4] a. Mendes, L.C. Bento, and U. Nunes. Multi-target detection and tracking with a laser scanner. *IEEE Intelligent Vehicles Symposium, 2004*, pages 796–801, 2004.
- [5] Christoph Mertz, Robert Navarro-Serment, Luis MacLachlan, Rybski Paul, Steinfeld Aaron, Arne Suppe, Christopher Urmson, Vandapel Nicolas, and Martial Hebert. Moving Object Detection with Laser Scanners. *Journal of Field Robotics*, 30(1):17–43, 2012.
- [6] F Nashashibi and a Bargeton. Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation. *2008 IEEE Intelligent Vehicles Symposium*, (1):847–852, 2008.
- [7] Viet Nguyen, Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 1768–1773, 2005.
- [8] Cyril Royère, Dominique Gruyer, Véronique Cherfaoui, Heudiasyc Cnrs, and Université De Technologie De Compiègne. Data association with believe theory. *Information Fusion, 2000. FUSION 2000*, pages 1–7, 2000.
- [9] Philippe Smets. Decision Making in the TBM : the Necessity of the Pignistic Transformation. *International Journal of Approximate Reasoning*, pages 1–14, 2004.