Real-Time Architecture For Obstacle Detection, Tracking And Filtering: An Issue For The Autonomous Driving

Dominique Gruyer
IFSTTAR - COSYS - LIVIC
France
dominique.gruyer@ifsttar.fr

Mohamed-Cherif Rahal
VEDECOM Institute
France
mohamed.rahal@vedecom.fr

Guillaume Bresson
VEDECOM Institute
France
guillaume.bresson@vedecom.fr

Valentin Magnier
IFSTTAR - COSYS - LIVIC
France
valentin.magnier@ifsttar.fr

ABSTRACT: *This paper deals with real-time obstacle detection and track- ing using multi-layer LIDAR data. We present two algorithms to cluster raw data coming from LIDAR sensors. The rst algorithm is based on a dynamic clustering approach while the second one relies on the con- nectivity between the laser impacts. Both algorithms take into account the inaccuracy and the uncertainty of the data sources. We propose a tracking approach based on the belief theory to estimate the dynamic state of the detected objects in order to predict their future maneuvers. The objects are then ltered using an intelligent ROI that depends on a dynamic evolution area computed from proprioceptive information of the ego-vehicle. We evaluate and validate the whole chained process on real data-sets.*

Keywords: Computational Geometry, Graph Theory, Hamilton Cycles

Received: 12 October 2016, Revised 18 November 2016, Accepted 9 December 2016

© 2017 DLINE. All Rights Reserved

1. Introduction

In the context of intelligent vehicles, obstacle detection (car, motorcycle, pedes- trian...) is an essential task that must take into

account several constraints like real time, reliability, robustness and accuracy of the detection stage. In addition, it is crucial to remove the existence of false alarms. In a first step of this processing, obstacles must be accurately located in areas of interest. These areas of interest are often represented by the traffic area of the ego-vehicle (detection and identication of the current traffic lane) and/or the maneuvering zone built from the proprioceptive data. The accuracy of this obstacle detection step is important if we then want to use it to assess the dynamics of the obstacles.

Since obstacle detection and tracking is one of the most important task to ensure a fully autonomous driving, many research and industrial works have been carried out in this eld using several kinds of sensors that include vision, laser scanners, radars, infrared cameras, etc. An interesting survey regarding vision (mono and stereo) is proposed in [1]. In [2], the authors present an efficient method for vehicle detection in traffic scenes based on the detection of the shadows underneath the vehicles. Concerning radar, the authors of [3] present a new and efficient tracking algorithm for a vehicle collision warning and collision avoidance system. The target-to-measurement data association is accomplished using a logic decision algorithm based on order statistics. In [4], a single-sensor tracking and multi-sensor fusion (infrared and radar data) algorithm is described. For the tracking part, an IMM ltering is applied where two Kalman filters are running in parallel and interact according to their likelihood.

The LIDAR is also commonly used in obstacle detection and tracking. For example, in [5] a description of a laser-based integrated object and road track- ing is proposed. The tracking process is done using a Kalman Filter. If the lane markings are well estimated an assignment of the vehicles to the available lanes is done. In [6], a method for object detection and tracking is presented. The detection step is based on object models (shapes). In addition to the detection, the authors present a classification system that uses the characteristics of the objects and their velocities to distinguish different classes. The tracking step is realized by a Kalman filter. In [7], an interesting algorithmic chaining using LIDAR raw data is detailed. The first step is a segmentation which creates objects from the set of laser impacts. This step is top down and uses a distance threshold. After that, a feature extraction is performed in order to fit with rectangular shapes. With this detail, the system is not able to detect pedestrians and motorcycles since they do not have rectangular shapes. A data association is performed to measure the change in object position over time, and determine which new seg- ment corresponds to which existing track. Finally, a Kalman filter is used to estimate the position, velocity, acceleration and yaw rate of each object.

LIDAR sensors are interesting because they can provide the positioning of an object with a centimetric accuracy. In addition, laser scans can be generated with a relatively high frequency for real-time operations and for active applications. Depending on the orientation of the obstacles, it is also possible to estimate the width and/or the length of an object in the scene. This gives the possibil- ity to build a bounding box of the obstacle and, by extension, to estimate its orientation.

However, with a single layer laser scanner, the false alarm problem may occur when the vehicle suers from pitch and roll movements. They are mainly caused by important speed changes (acceleration and braking) and by the geometry of the road surface (slope, bank angle, speed bump...). These problems generate laser impacts that hit the road. In this case, it is necessary to implement mecha- nisms to filter these impacts out in order to avoid considering them as potential obstacles. Contrary to this first problem, detection failure can be observed when the plane of the laser impacts goes over obstacles. Finally, the objects which are not on our traffic lane may be dicult both to detect and to filter without the construction of several areas of interest (predicted evolution area and navigation zone).

Obstacle detection using a single layer can thus be dicult without the help of additional processing. A part of the enumerated problems can now be solved using the new generation of multi-layer laser scanners. In this paper, we propose two efficient methods for the processing and the clustering of the laser impacts extracted from different layers. These approaches are independent from the impact processing order. Then, to use all the information generated by the sensor and all layers, and to ensure a robust-enough detection stage, we propose a merging and filtering strategy. In order to validate or to filter out a global object, 3 criteria will be used. These methods will be presented in section 2.

However, obstacles obtained from laser scanner data are punctual completely independent of the time. It is only a snapshot of a situation unrelated to the past time. Therefore, this single stage of detection only allows us to estimate the position of a cluster of impacts which can be seen as a potential punctual obstacle. but not its dynamics. In addition, the temporal independence of detected obstacles prevents us from filtering some remaining false alarms and bad detections. To better address these issues, it is imperative to add a module of multi-object tracking in order to obtain a temporal link between the detected objects at successive times. The output of this process then allows to generate and manage tracks. These developments are discussed in section 3.

In order to manage the computation time of this embedded application handling a set of laser scanners covering a 360° field of view, we propose a fast and efficient approach in order to generate a set of regions of interest from proprioceptive data (i.e. speed and yaw rate from the CAN bus). This ROI is useful to share the laser impacts, to identify obstacles by area, and to filter specic areas. This functionality will be presented in section 4. Finally, we propose some results, a conclusion and perspectives.

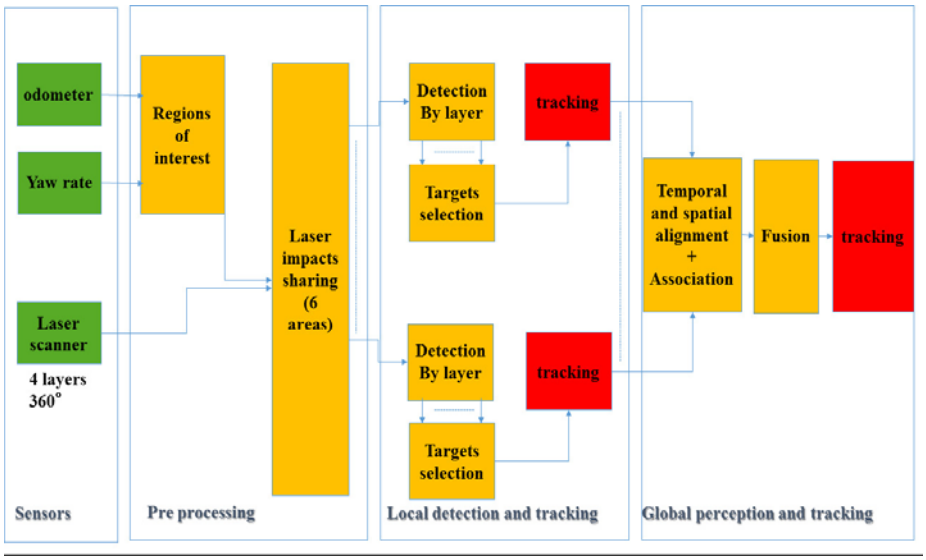

Figure 1. The global architecture of the system

2. Multi-layer detection from raw laser data

2.1 Similarity operator for uncertain data management

Let recall the notion of distance:

$$d : E \times E \rightarrow R^{+}$$

$$(p_i,\ p_j) \rightarrow dij = d\,(p_i,\ p_j)$$

this function *d* must verify the following properties:

1. symmetry: $\forall\ (p_i,\ p_j) \in E^2, d\,(p_i,\ p_j) = d\,(p_j,\ p_i)$
2. separation: $\forall\ (p_i,\ p_j) \in E^2, d\,(p_i,\ p_j) = 0 \Leftrightarrow p_i = p_j$
3. triangular inequality: $\forall\ (p_i,\ p_j,\ p_k) \in E^3, d\,(p_i,\ p_j) \leq d\,(p_i,\ p_j) + d\,(p_j,\ p_k)$

Notice that the third property is optional in case of similarity/dissimilarity func- tions.

There exists, in the literature, a lot of distance and similarity functions that compute the proximity between two measurements, between a measurement and a cluster or between two clusters.We can cite for example: the euclidean distance, Minkowski, Manhattan, Tchebychev , Cannberra, X^2, etc. Unfortunately, the previously cited distance functions do not take into account the accuracy of a measurement and the uncertainty of a class. The distance function must use the modeling of the variances/co-variances of the measurements and the clusters. This distance function must also deal with dierent kind of data and models coming from dierent theories (probabilistic, fuzzy, ensemblist ...etc.).

$$d_{i,j} = \frac{\sqrt{(X_i - X_j)\,(X_i - X_j)^T}}{\sqrt{(g^i - b_i)\,(g^i - b_i)^T} + \sqrt{(g_j - b_j)\,(g_j - b_j)^T}} \qquad (1)$$

The distance used in our algorithm is detailed in the equation (1). This function computes a distance between two clusters C_i and c_j, knowing that a cluster can also be a unique measurement (a singleton).

where X_i and X_j represent respectively the state vectors of the two clusters C_i and C_j, g_i, g_j their centers and b_i, b_j their boundaries.

2.2 Strategies for data clustering

We develop in this section two different algorithms for clustering the data coming from laser scanners. Using the raw data provided by the laser scanners, our aim is to find a robust clustering that provides a partition $P = \{C_1, ...C_k\}$ where each cluster C_j characterizes an object of the scene. The scene can be divided into one or several regions of interest ROI_k for the ego-vehicle. Our partitioning algorithms are non-parametric and do not have an initial or desired number of clusters.

Dynamic clustering The dynamic clustering algorithm used in this work is an iterative process that constructs clusters from a set of points coming from raw laser data. At each step, the algorithm checks if a point, or a cluster, is close to the border of a cluster. From each scan we randomly choose a data point p_j in order to initialize our first cluster and consequently the first object in the scene. For the rest of the points, we search for the closest cluster C_j in the sense of a certain distance or similarity measure d_{ij}. The algorithm 1 shows our dynamic clustering process.

Algorithm 1: The dynamic clustering

input: Raw laser scan: A set of points $\underset{i \in 1..n}{\cup} \{p_i\} \in R^3$;

t_d: the distance threshold for the association

output: A partition in k clusters $P = \{C_1, ...C_k\}$

1 *Random selection of a number* $i \in \{1..n\}$;
2 $P \leftarrow \{\{p_i\}\}$ // First cluster;
3 $E = \{1..n\}/\{i\}$;
4 **foreach** *element* $\in E$ **do**
5 **foreach** $C_j \in P$ **do**
6 *found_closest_cluster = false*;
7 // *find the closest cluster*;
8 **if** $d(element, C_j) < t_d$ **then**
9 *found_closest_cluster = true;*
10 $C = C_j \cup element$;
11 return;
12 **end**
13 **end**
14 **if** *not found_closest_cluster* **then**
15 // *create a new cluster C*;
16 $C = \{element\}$;
17 **end**
18 // *reduce the elements list E*;
19 *E = E/element*;
20 **if** not *found_closest_cluster* **then** ;
21 Update the state vector of the cluster C;
22 $P = P \cup C$;
23 **else** ;
24 $P = P/C_j \cup C$;
25 **end**
26 Return(P)

At each association step of a new laser point p_i to a cluster C_j, several cases are possible:

– p_i is not associated to an existent cluster (line 14 to 16 in 1). In this case, a new cluster C is created. This cluster is represented by a state vector containing the Cartesian coordinates of the laser impact p_i and a noise model given by the specications of the laser scanner. This noise model is used to represent the uncertainty of the resulting cluster.

– In the second case, the laser impact p_i is associated to an existing cluster C_j. In this situation, we recompute the center and variance/co-variance matrix of the resulting cluster.

After the clustering process, for each resulting cluster C_j, a state vector containing its variance/co-variance matrix is added. From this uncertainty, t is possible to compute more accurate information about the width and the height of the clusters as long as at least two faces of an object are visible. Figure 2 shows a result of this clustering algorithm.

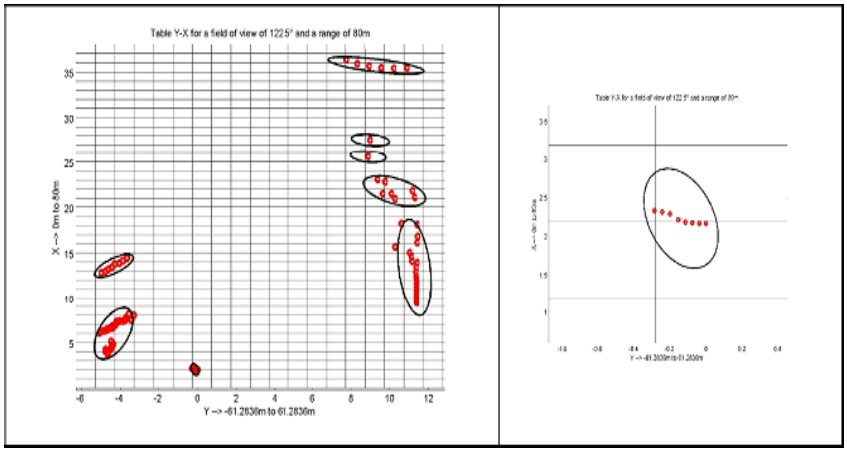

Figure 2. An example of a laser scan clustering for obstacle detection

Once the first step of the clustering is completed, we obtain a partition of k clusters that can be considered as tracks. In order to validate the results, we apply a second pass of the same algorithm using the same distance function in order to aggregate the connected clusters having a high similarity.We show some illustrative results in figure 3.

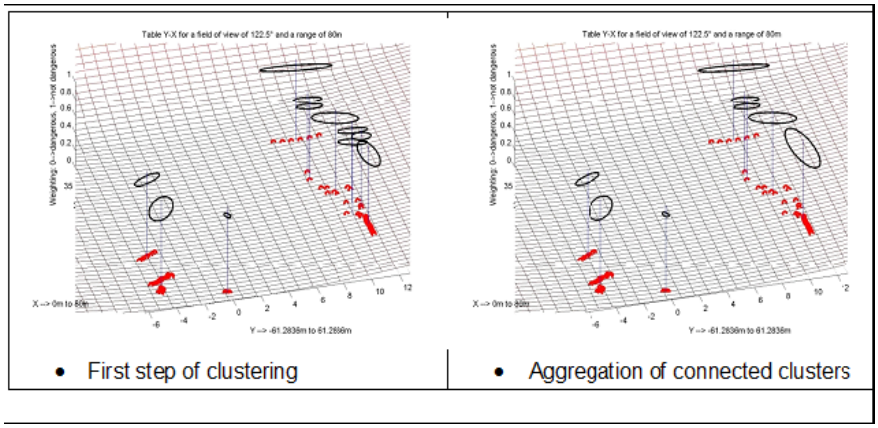

Figure 3. An example of the clustering and aggregation of connected clusters

Connectivity-based clustering approach Our second clustering method is based on laser impact connectivity. We first build a squared matrix which takes into account all the laser impacts, using the same distance measure defined in equation (1). We then

look for mutual laser impact neighbors and merge them in the same cluster. In the case where two impacts are related (determined by a threshold t_d) then the corresponding cell in the matrix (row i, column j) is set to 1. If no relation is found, the cell value is set to 0. If $d_{ij} \leq t_d$ then I_i and I_j belong to the same object, I_i and I_j belong to two different objects otherwise.

This method can easily be parallelized if we are dealing with great amounts of data (e.g. Velodyne). In addition, the resulting matrix is symmetrical and thus only the upper triangle of the matrix needs to be calculated. The values on the diagonal of the matrix are set to 1. With this association step, we have built a global association matrix. From this association matrix, the second stage of our algorithm is to extract a synthetic matrix which contains the objects (the obstacles in the environment) and their association with the laser impacts. The reduction of the matrix is done recursively. by aggregating the impacts having a non-empty intersection. This aggregation step is repeated until all the impacts in the matrix have been treated or the association matrix have not converged to a constant number of lines. During this step, we are looking for paths and connected components in the sense of graph theory (our association matrix in an adjacency matrix). We detail our method in the algorithm 2. Figure 4 shows an application of the connectivity-based algorithm to the detection of pedestrians.

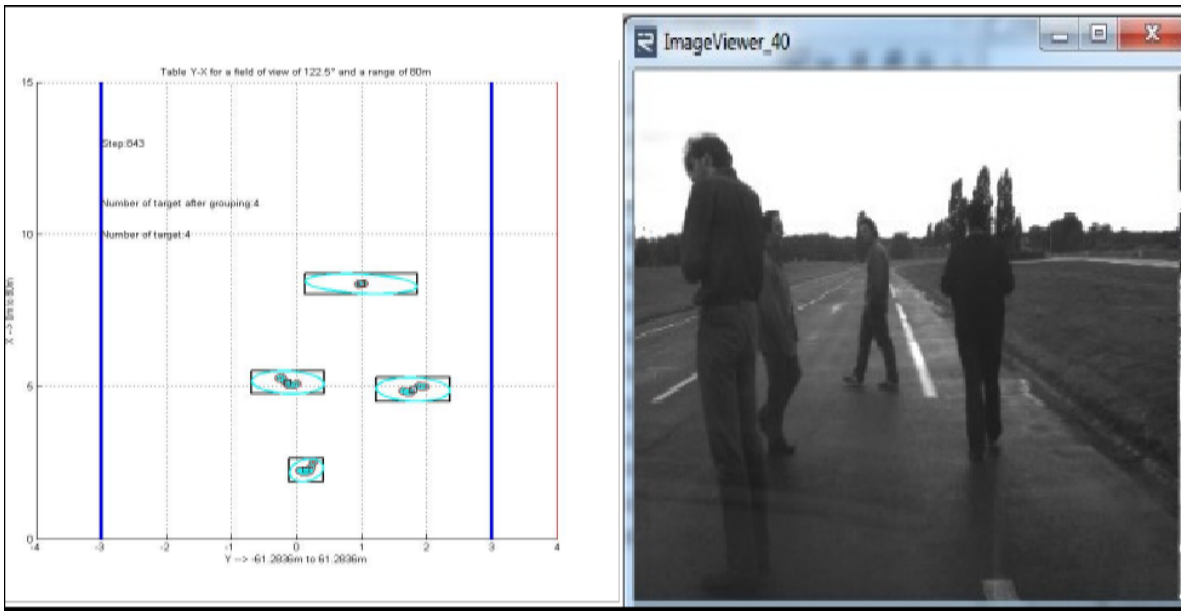

Figure 4. An example of the clustering and aggregation using the connectivity-based method for pedestrian detection

Multi-layer strategy In order to deal with multi-layer LIDARs, we propose a three-step strategy to cluster the raw data. In the first stage, we apply the connectivity-based algorithm on each layer, as described in section 2.2. The output of this stage is a set of objects labeled by layer. The second phase is to apply the same process but on the resulting objects from each layer. We then build a new squared matrix where the indexes of the rows and of the columns are the objects computed in the previous step. In order to make the association between the detected objects on each layer and the others, we re-apply the algorithm 2 where I_i and I_j refer to the objects detected for each layer as shown in the previous algorithm. The result of this approach is a new set of objects where each object is a fusion or an aggregation of closest objects coming from different layers.

In the merging step, we use the following criteria to create new objects and compute their presence confidence:

- The number of impacts per object per layer.
- The number of layers where the object has been detected.
- The total number of impacts in all the layers where the object has been detected.

Algorithm 2: The connectivity-based algorithm

input : Raw laser impacts: A set of points $E = \bigcup_{i \in 1..n} \{I_i\} \in R^3$

t_d: the distance threshold for the association

output: A partition in k clusters $P = \{C_1, ... C_k\}$

```
1  Initialization step ;
2  P = φ;
3  // build the association matrix Ma;
4  Ma(i, j) = 0, ∀i, j ∈ [1..n] ;
5  foreach Ii ∈ E do
6  |   foreach Ij ∈ E do
7  |   |   if d(Ii, Ij) < td then
8  |   |   |   Ma(i, j) = 1;
9  |   |   end
10 |   end
11 end
12 // aggregation step;
13 convergence = false;
14 while not convergence do
15 |   s = size (Ma, 1) // number of lines of Ma;
16 |   foreach i ∈ [1..s] do
17 |   |   foreach j ∈ [1..s] do
18 |   |   |   if Ma(i, :) ∩ Ma(j, :) ≠ φ then
19 |   |   |   |   // aggregate Ii and Ij ;
20 |   |   |   |   C = Ii ∪ Ij ;
21 |   |   |   |   //remove Ii and Ij ;
22 |   |   |   |   Ma(i, :) = Null;
23 |   |   |   |   Ma(j, :) = Null;
24 |   |   |   |   //Insert C;
25 |   |   |   |   Ma(size(Ma; 1), :) = C;
26 |   |   |   end
27 |   |   end
28 |   end
29 |   convergence = size (Ma, 1) == 1 OR size(Ma, 1) == s;
30 end
31 foreach i ∈ [1..size(Ma, 1)] do
32 |   P = P ∪ Ma(i, :) ;
33 end
34 Return (P)
```

3. Tracking with belief theory

The estimation of the dynamic state of an object is essential in our application in order to predict and anticipate future maneuvers and trajectories so as to avoid collision or hazardous situation. Therefore, once object detection have been achieved, a multi-object tracking algorithm is needed to estimate the dynamic state of the targets and to manage track appearances, disappearances, uncertainties and condences. The position of previously perceived objects (i.e.target) is predicted at the current time using estimators like Kalman filters. These predicted objects are already known objects (i.e. tracks) and will be denoted in what follows by track Y_j . Perceived objects at the current time will be denoted by X_i.

In this tracking approach, the core of the method is to achieve a target-to-track association so as to update the track set. In this context, the proposed multi-object association algorithm is based on the belief theory introduced by Shafer [8]. This theory seems to be the most ecient and relevant approach to answer this topic.

In a general framework, the problem consists in identifying an object designated by a generic variable X among a set of hypotheses Y_i. One of these hypotheses is supposed to be the solution. The current problem consists in associating perceived objects X_i to known objects Y_j. Belief theory allows us to assess the veracity of P_i propositions representing the matching of the dierent objects.

A basic belief mass set, allowing the characterization of a proposition, must be defined. This basic belief mass (mass $m_{\ominus}()$) is defined on a [0, 1] interval. This mass is very close to the one used in probabilistic approaches, except that it is distributed on all the propositions of the referential of definition $2^{\Omega} = \{A/A \subseteq \Omega\} = \{\phi, \{Y_1\}, \{Y_2\}, ..., \{Y_n\}, \{Y_1, Y_2\}, ... , \Omega\}$. This referential is the power set of $\Omega = \{Y_1, Y_2 ... Y_n\}$ which includes all the admissible hypotheses. These hypotheses must also be exclusive $Y_i \cap Y_j = \phi, \forall i \neq j$. The masses thus defined are called basic belief assignment and denoted bba . They verify:

$$sum_{A \subseteq \Omega} m^{\Omega}(A) = 1, A \in 2^{\Omega}, A \neq \phi \qquad (2)$$

The sum of these masses is equal to 1 and the mass corresponding to the impossible case (i.e. Conflict) $m^{\Omega}\{X_i\}(\phi)$ must be equal to 0. In order to succeed in generalizing the Dempster combination rule and thus reducing its combinatorial complexity, the reference frame of definition is limited by the constraint that a perceived object can be connected with one and only one known object. For example, for a detected object, in order to associate it among three known objects, the frame of discernment is: $\Omega = \{Y_1, Y_2, Y_3, Y_*\}$, where Y_i means that X and Y_i are supposed to be the same object. In order to be sure that the frame of discernment is really exhaustive, a last hypothesis noted Y_* is added ([12]). This one can be interpreted as "a target has no association with any of the tracks". In fact, each Y_j represents a local view of the world and Y_* represents the rest of the world. In this context, Y_* means that an object is associated with nothing in the local knowledge set. In our case, the denition of the bba is directly in relation with the data association applications. The mass distribution is a local view around a target X_i and of a track Y_j . The bba on the association between X_i and Y_j will be noted m^{Ω}_j(Xi) () It is defined on the frame of discernment $\Omega = \{Y_1, Y_2, ... , Y_n , Y_*\}$ and more precisely on focal elements $\{Y, \overline{Y}, \Omega\}$ where $\overline{Y}$ means the negation of Y . Each one will respect the following meaning:

$$\begin{cases} m_j^{\Omega}\{X_i\}(Y_j) & X_i \text{ is associated with } Y_j \\ m_j^{\Omega}\{X_i\}(\overline{Y_j}) & X_i \text{ is not associated with } Y_j \\ m_j^{\Omega}\{X_i\}(\Omega) & \text{ignorance on the association between } X_i \text{ and } Y_j \\ m_j^{\Omega}\{X_i\}(Y_*) & \text{reject } X_i \text{ is in relation with nothing} \end{cases}$$

In fact, the complete notation of a belief function is: $m_{S,t}^{\Omega}\{X\}[BC_{S,t}](A)$ with S the information source, t the time of the event, the Ω frame of discernment, X a parameter which takes value in Ω and BC the evidential corpus or knowledge base. This formulation represents the degree of belief allocated by the source S at the time t to the hypothesis that X belongs to A [11]. In order to simplify this notation, we will use the following basic belief function notation m_j^{Ω}\{X\}(A). The t argument is removed because we process the current time without any link with the previous temporal data. The time link is taken into account and processed in the update stage of the Kalman filter. In this mass distribution, X denotes the processed perceived objects and the index j, the known objects (track). If the index is replaced by a set of indices, then the mass is applied to all targets. Moreover, if an iterative combination is used, the mass $m_j^{\Omega}\{X_i\}(Y_*)$ is not part of the initial mass set and appears only after the first combination. It replaces the conjunction of the combined masses $\overline{m_j^{\Omega}\{X_i\}(Y_j)}$. By observing the behavior of the iterative combination with n mass sets, a general behavior can be seen which enables us to express the final mass set according to the initial mass sets. This

allows us to directly compute the final masses without any recurrent stage. The use of a basic belief assignment generator using the strong hypothesis: an object cannot be associated and not associated to another object at the same time allows us to dene new rules. These rules reduce the influence of the conflict (the combination of two identical mass sets will not produce a conflict) and the complexity of the combination [13, 14]. The rules become:

$$m_{1..n}^{\Omega}\{X_i\}(Y_j) = m_j^{\Omega}\{X_i\}(Y_j)\prod_{\substack{a=1\\a\neq j}}^{n}(1 - m_a^{\Omega}\{X_i\}(Y_a)) \quad (3)$$

$$m_{1..n}^{\Omega}\{X_i\}(Y_j,Y_*) = m_j^{\Omega}\{X_i\}(\Omega)\prod_{\substack{a=1\\a\neq j}}^{n} m_a^{\Omega}\{X_i\}(\overline{Y}_a) \quad (4)$$

$$m_{1..n}^{\Omega}\{X_i\}(Y_j, Y_k, Y_*) = m_j^{\Omega}\{X_i\}(\Omega).\, m_k^{\Omega}\{X_i\}(\Omega) \times \prod_{\substack{a=1\\a\neq j\\a\neq k}}^{n} m_a^{\Omega}\{X_i\}(\overline{Y}_a) \quad (5)$$

$$m_{1..n}^{\Omega}\{X_i\}(Y_j, Y_k, ..., Y_*) = m_j^{\Omega}\{X_i\}(\Omega).\, m_k^{\Omega}\{X_i\}(\Omega)... \times m_l^{\Omega}\{X_i\}\prod_{\substack{a=1\\a\neq j\\a\neq k\\\cdots\\a\neq l}}^{n} m_a^{\Omega}\{X_i\}(\overline{Y}_a) \quad (6)$$

$$m_{1..n}^{\Omega}\{X_i\}(\overline{Y_j}) = m_j^{\Omega}\{X_i\}(\overline{Y}_j)\prod_{\substack{a=1\\a\neq j}}^{n}(1 - m_a^{\Omega}\{X_i\}(\Omega)) \quad (7)$$

$$m_{1..n}^{\Omega}\{X_i\}(\Omega) = \prod_{a=1}^{n}(1 - m_a^{\Omega}\{X_i\}(\Omega)) \quad (8)$$

$$m_{1..n}^{\Omega}\{X_i\}(\phi) = 1 - \prod_{a=1}^{n}(1 - m_a^{\Omega}\{X_i\}(Y_a)) - \sum_{a=1}^{n} m_a^{\Omega}\{X_i\}(Y_a)\prod_{\substack{b=1\\b\neq a}}^{n}(1 - m_b^{\Omega}\{X_i\}(Y_b)) \quad (9)$$

$mX_i(Y_*)$ is the result of the combination of all "non association" belief masses for X_i. Indeed, new target(s) apparition or loss of track(s) because of field of view limitation or objects occultation, leads us to consider the Y_* hypothesis which models these phenomena.

In fact, a specialized bba can be defined given a local view of X with Y association. In order to obtain a global view, it is necessary to combine the specialized bbas. The combination is possible when bbas are defined on the same frame of discernment and for the same parameter X. In a first step, a combination of $m_j^{\Omega}\{X_i\}()$ with $j \in [1..n]$ is done using formula (3) to (8). The result of the combination gives a mass $m_{1..n}^{\Omega}\{X_i\}()$ defined on 2^{Ω}. We can repeat these operations for each X_i and to obtain a set of p bbas: $m_{1..n}^{\Omega}\{X_1\}()$, $m_{1..n}^{\Omega}\{X_2\}()$, .. $m_{1..n}^{\Omega}\{X_p\}()$, where p is the number of targets and Ω the frame including the n tracks corresponding to the n hypotheses for target-to-track association. In order to get a decision, a pignistic transformation is applied for each $m_{1..n}^{\Omega}\{X_i\}()$ with $i \in [1..p]$. The pignistic probabilities $BetP^{\omega i}X_i(Y_j)$ of each Y_j hypothesis are summarized in a matrix corresponding to the target point of view. However, this first matrix gives the pignistic probabilities for each target without taking into consideration the other targets. Each column is independent from the others. A dual approach is proposed in order to consider the possible association of a track with the targets in order to obtain the track point of view.

The dual approach uses the same bba but combined for each track Y. From the track point of view, the frame of discernment becomes $\Theta = \{X_1, ..., X_m, X_*\}$. The X_* hypothesis models the capability to manage either track disappearance or occultation. For one track Y_j, the bbas are then:

$$\begin{cases} m_i^{\Theta}\{Y_j\}(X_i) = m_j^{\Omega}\{X_i\}(Y_j)\ X_i \text{ is associated with } Y_j \\ m_i^{\Theta}\{Y_j\}(\overline{X}_i) = m_j^{\Omega}\{X_i\}(\overline{Y}_j)\ X_i\, Y_j \text{are not associated} \\ m_i^{\Theta}\{Y_j\}(\Theta) = m_j^{\Omega}\{X_i\}(\Omega) \text{ ignorance on the association between } X_i \text{and } Y_j \end{cases}$$

The same combination (equations (3) to (8)) is applied and gives $m_{1...p}^{\ominus}\{Y_j\}$ (). These operations can be repeated for each Y_j to obtain a set of n bbas:

$m_{1...p}^{\ominus 1}\{Y_1\}$ (), $m_{1...p}^{\ominus 2}\{Y_2\}$ ()... $m_{1...p}^{\ominus n}\{Y_n\}$ () where n is the number of tracks and $\ominus_j$ is the frame based on the association hypothesis for the Y_j parameter. The index j in $\ominus_j$ is now useful in order to distinguish the frames based on association for one specific track Y_j for j $\in$ [1..n].

A second matrix involving the pignistic probabilities $BetP^{\ominus j}\{Y_i\}(X_j)$ about the tracks is obtained. The last stage of this algorithm consists in finding the best decision from the previously computed associations using both of the pignistic probability matrices ($BetP^{\Omega i}\{X_i\}(Y_j)$ and $BetP^{\ominus j}\{Y_j\}(X_i)$). The decision stage is accomplished using the maximum pignistic probability rule. This rule is applied on each column of both pignistic probability matrices. With the first matrix, this rule answers the question which track Y_j is associated with target X_i?

$$X_i = d(Y_j) = \max_i |BetP^{\Omega i}\{X_i\}(Y_j)| \quad (10)$$

With the second matrix, this rule answers the question which target Xi is associated to the track Y_j?

$$Y_j = d(X_i) = \max_j |BetP^{\ominus i}\{Y_j\}(X_i)| \quad (11)$$

Unfortunately, a problem appears when the decision obtained from a pignistic matrix is ambiguous (this ambiguity quanties the duality and the uncertainty of a relation) or when the decisions between the two pignistic matrices are in conict (this conict represents antagonism between two relations resulting each one from a dierent belief matrix). Both problems are solved by using an assignment algorithm known under the name of the Hungarian algorithm [9, 10]. This algorithm has the advantage of ensuring that the decision taken is not good but the best. By the best, we mean that if a known object has been perceived by a defective or poor sensor, then the sensor is unlikely to know what this object corresponds to, and it therefore becomes difficult to ensure that the association is correct. Among all the available possibilities, we must certify that the decision is the best of all possible decisions.

Once the multi-object association has been performed, the Kalman filter associated to each target is updated using the new position of the target. The new dynamic state of each target is estimated, i.e. both linear and angular speeds.

4. Tools and filters for real time autonomous driving

4.1 Dynamic evolution areas

The raw data provided by the laser scanner represents a one-time view of the surrounding environment. However, in this environment, we are generally interested in what happens on the road surface area and especially what evolves above the road surface. These two points mean that we need to filter impacts coming from laser frames to remove unneeded impacts from the obstacle detection phase.

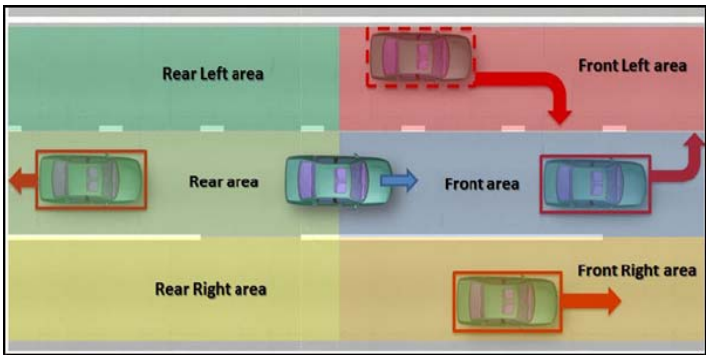

Figure 5. An example of the decomposition of the surrounding space into 6 ROIs

To eliminate the hazardous impacts, a predicted evolution area of the vehicle is used. The generation of this area is based on proprioceptive data coming from embedded sensors (INS or odometer). From these sensors, we use the yaw rate and the velocity information of the ego-vehicle in order to generate its potential future positioning. The range of this Region of Interest is fixed with a time value which allows us to adapt the filtering to specic situations (Warning, hazardous, safe areas).

In our case, we consider 6 areas of interest around the ego-vehicle, corresponding to specic traffic zones and potential maneuvers. These regions are shown in gure 5. In this figure, 3 frontal zones (left, middle, right) and 3 rear zones can be observed. In the decomposition that we have chosen, the right and left lateral areas are shared by both the front and rear areas of interest. The
number of ROIs can easily be expanded.

4.2 Modeling of an area

Our areas of interest are built from vehicle dynamics observers (the yaw rate ω (rad / s), the longitudinal velocity v (m/s)) and a time range S (s). In the case of a stationary ego-vehicle (which would thus give an empty interest area), it is necessary to set a minimum speed (e.g. a few m/s) in order to always have a non-empty area (0 m of distance range). This is important when the vehicle is waiting at a stop sign or at a traffic light. Then, according to the desired shape of the region of interest, we tune the near-half width *nd* and the far-half width *fd* (see figure 6).

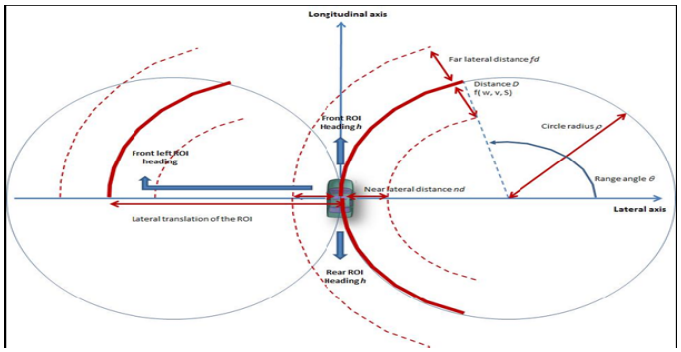

Figure 6. Illustration of the ROI building

In addition to these lateral parameters, the translation and heading (front, rear) parameters allow us to position the region of interest in the vehicle environment. From all these parameters, two variables are calculated: the radius ρ and the maximum angle θ. The radius of the circular arc representing the potential vehicle trajectory is obtained with $\rho = v/\omega$. The distance covered on this circular arc in 1 second is given by the equation: $D = \rho \times \omega \times S$. The angle corresponding to the distance on the arc in n seconds is obtained from: $\theta = D/\rho$. By replacing D and ρ by their respective definitions, we obtain: $\theta = \rho \times \omega^2 \times S/v$. If ω is negative then $\theta = \theta + \pi$. If the heading of the region of interest is oriented towards the rear then $\theta = -\theta + 2 \times \pi$.

$$\begin{cases} \theta_{inf} = 0 \ \theta_{sup} = \theta & \text{if } \theta \in [0, \pi/2] \\ \theta_{inf} = \theta \ \theta_{sup} = \pi & \text{if } \theta \in [\pi/2, \pi] \\ \theta_{inf} = \pi \ \theta_{sup} = \theta & \text{if } \theta \in [\pi, 3\pi/2] \\ \theta_{inf} = \theta \ \theta_{sup} = 2\pi & \text{if } \theta \in [3\pi/2, 2\pi] \end{cases}$$

This information, which identifies the processed circle quarters, allows us to calculate the coefficients of the polynomial modeling the circular arc. Then, depending on the heading (front and rear) of the area of interest, we build this area with a second-order polynomial, which gives us:

- In a front area case: $a = \dfrac{f_d - n_d}{\theta_{sup} - \theta_{inf}}$, $b = f_d - a \times \theta_{sup}$.
- In a rear area case: $a = \dfrac{n_d - f_d}{\theta_{sup} - \theta_{inf}}$, $b = f_d - a \times \theta_{sup}$.

Finally, the ROI provided to the laser-impact filtering module is built from the following set of parameters: $ROI = [\theta, \rho, [a, b], h, [\theta_{inf}, \theta_{sup}], T]$, with θ the angle, ρ the radius, a and b the polynomial coefficients, h the heading of the ROI, *inf* and *sup* the limits of the used quarter circle, T the lateral translation of the ROI. These parameters can now be used to filter laser data knowing by applying the two following equations: $\rho_{impact} = \sqrt{(}(x + \rho)^2 + y^2)$ and $\alpha = \arctan 2\,(y, (x + \rho))$.

An impact is validated as present in an area of interest if it validates the 2 following propositions:

1. Validation test for the longitudinal distance:

- If $\alpha \in]\pi/2, 3\pi/2[$ then $\rho_{impact} = -\rho_{impact}$
- The impact is validated if $\alpha \in [\theta_{inf}, \theta_{sup}]$ or $\alpha + 2\pi \in [\theta_{inf}, \theta_{sup}]$

2. Validation test for the lateral distance:

- First, we calculate the distance between the radius of the ROI and the radius of the laser impact:

$d^2 = (\rho_{impact} \cos \alpha - (\rho + T) \cos \alpha)^2 + (\rho_{impact} \sin\alpha - (\rho + T) \sin \alpha)^2$

- Then, we calculate the lateral range depending on the coefficients of the polynomial and the angle given by the impact: $l = a \times \alpha + b$
- The impact is validated if $d \leq l$

5. Experiments and discussion

We validate our approach on a real experiment presenting urban, peri-urban and highway environments in an open-road context. Our vehicle is equipped with 5 4-layer IBEO LUX LIDARS located so as to cover 360° around the car. This sensors are running at 25 Hz, that is they deliver a new set of data each 40 miliseconds.

5.1 Evaluation of the detection, tracking and filtering

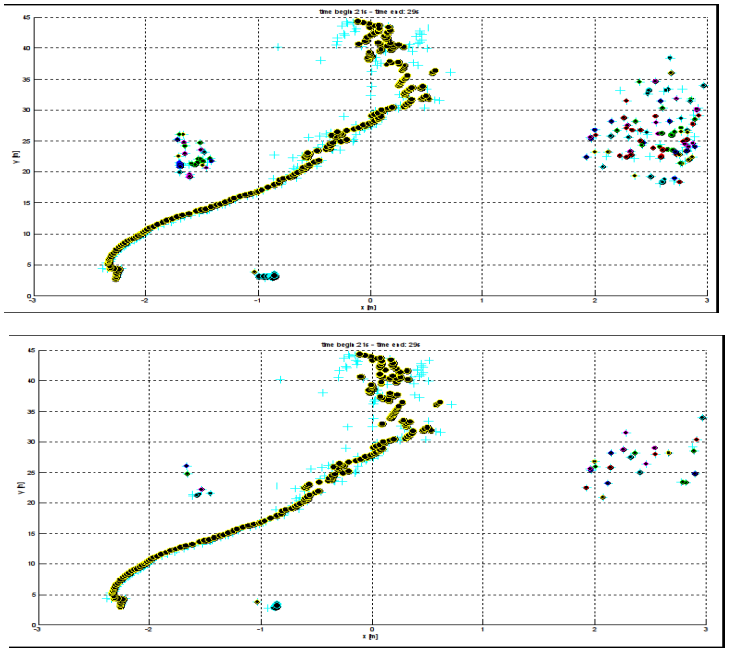

Figure 7. (Top) Target detection and tracks using one layer. (Bottom) Target detection and tracks using two layers

We propose, here, to study the eect of the number of layers on the validation of a target.

In this example, the ego-vehicle is overtaken by another vehicle coming from the left and reaching the ego-lane in front of the ego-vehicle. The roadsides are composed of tall grass creating clutter that can generate false alarms.

In the following figures, the points correspond to the relative position of targets to the ego vehicle. Arrows represent targets and circles, tracks. The color of the circle outer ring is the identifier of a track. The four graphs depict the eect of using respectively 1, 2, 3 and 4 layers to validate targets.

As we can observe, the increase in the number of layers used to validate targets leads to drastic improvements: noisy tracks generated by the grass on the road sides and tracks generated by the road surface are removed. When three layers are used to validate targets, the noise is almost totally filtered.

Nevertheless, a drawback of using a multi-layer target validation is that, due to the vertical angle between the layers, long distance detection can be affected. Indeed, when a vehicle is suciently far away from the ego-vehicle (more than 50m), very often, there only is one layer reaching it, the others being too high or too low compared to the vehicle height. We can observe that the overtaking vehicle is being detected up to 45m using one layer instead of 37m using the four layers.

From these observations, validating targets with an adaptive function so as to be more efficient would be an interesting solution. This function would adapt the values of the 3 criteria (number of layers, numbers of impacts by layer, impacts for selected layers) according to the detection distance in order to get the optimal compromise between clutter filtering and detection range.

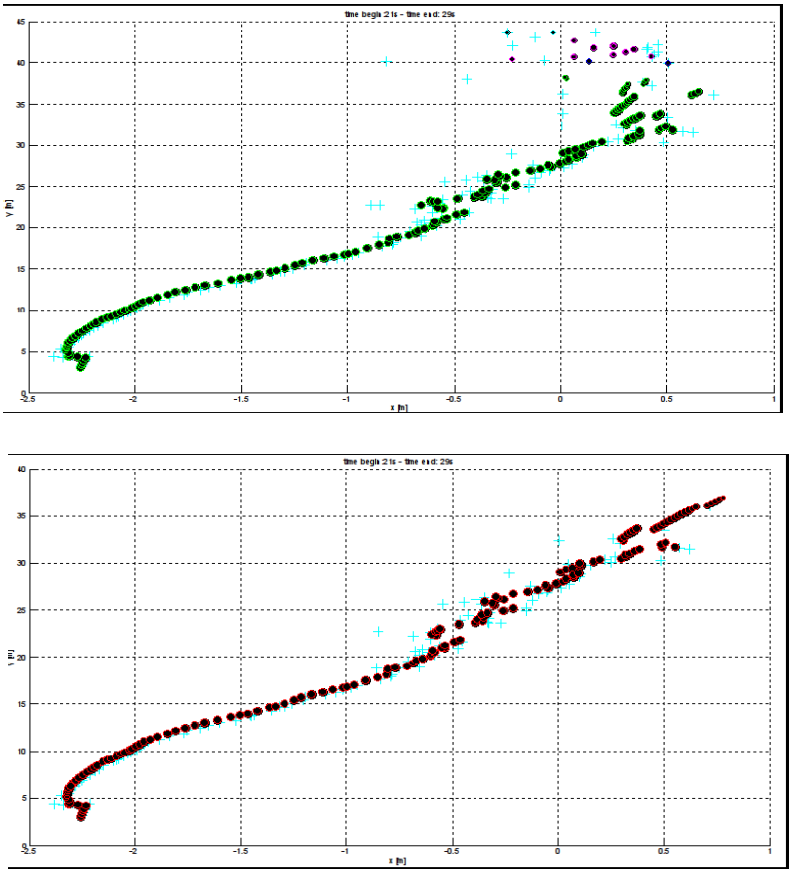

Figure 8. (Top) Target detection and tracks using three layers. (Bottom) Targes detection and tracks using four layers

5.2 Evaluation of the execution time

We examine here the execution time of the most important blocks of our architecture that are : The laser impact sharing which affects each laser impact to a ROI, the detection and the tracking step.

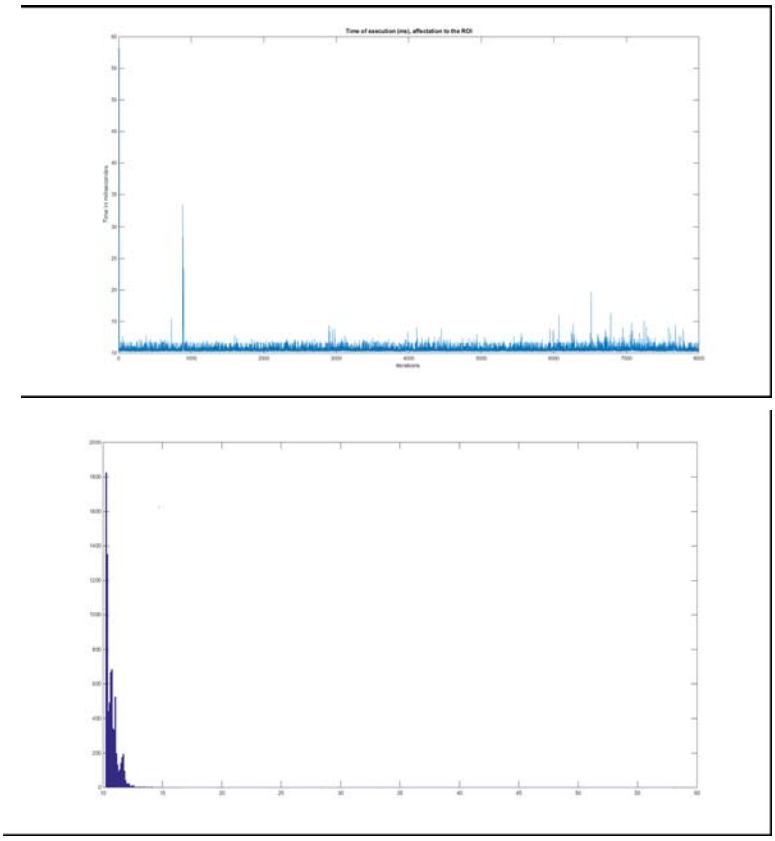

Figure 9. (Top)The execution time and (bottom) the associated histogram for generation of ROI and the affectation step obtained form a sequence of 10 minutes in real traffic situations

Figure 9 shows the execution time for generation of ROI and the affectation step obtained form a sequence of 10 minutes in real trac situations, the average execution time of these two steps is around 10 milliseconds.

The figure 10 summarizes the time execution and the associated histogram for the steps of detection and tracking. We can observe some peaks on the Top graph coincides with crowded situations like intersections and trac lights with a lot of obstacles surround the vehicle, the average execution time for this step is 95 miliseconds. We must mention that this step can be computed in parallel such that we can divide the execution time per the number of the desired ROI.

6. Conclusion

We have proposed a new perception architecture involving adaptive obstacle detection and tracking with dynamic Region of Interest (ROI) generation. In order to improve the detection stage, we have proposed an algorithmic strategy to deal with multi-layer laser scanners. The approach is based on two different algorithms to cluster the raw data. The first one is based on a dynamic clustering approach while the second one is base on the connectivity between the laser impacts. Both take into account the inaccuracy and the uncertainty of the data sources. In order to manage and estimate the speed of obstacles, a tracking method has been added. This innovative method, based on the belief theory, allows us to take into account the data imperfections and to identify the ambiguous and conflicting situations. Moreover, this tracking method manages efficiently

track appearance,disappearance, propagation and re-association. Finally, a new method is presented to divide the road surface into six Regions of Interest using the proprioceptive information coming from the CAN bus.

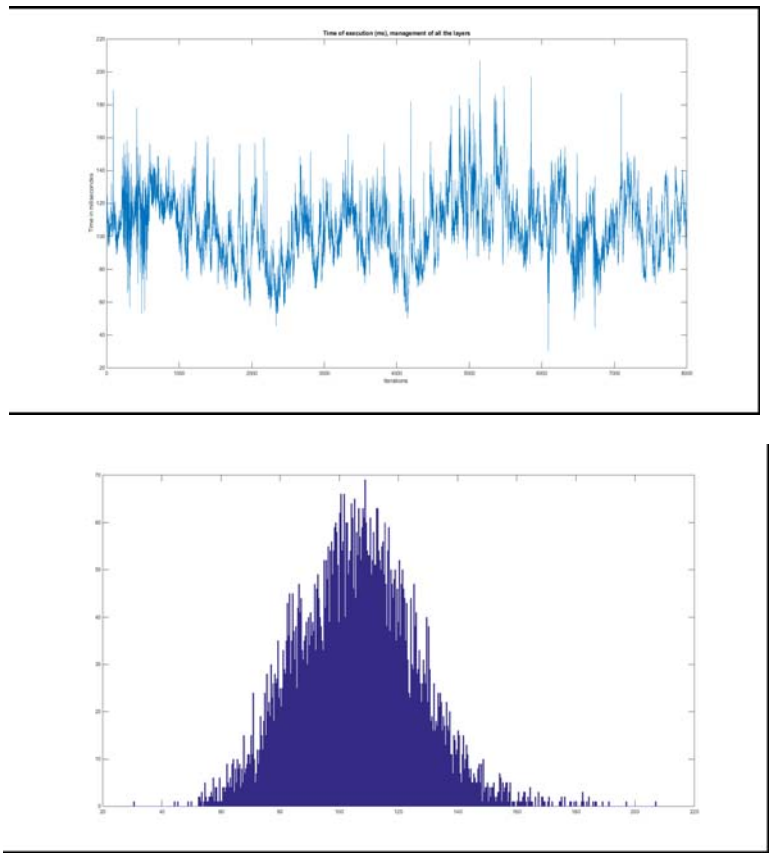

Figure 10. (Top) The execution time and (Bottom) the associated histogram for detection step obtained form a sequence of 10 minutes in real traffic situations

Our approach was tested and validated on a real experiment. The obtained results are very encouraging. The ROI generation provides an efficient way to reduce the computational time and oers a high level view of what is happening in the environment.

In the future, we will extend our approach to build a local dynamic perception map taking into account the other elements of the road scene. We plan to introduce some cooperation mechanisms between these elements to ensure a robust and reliable perception task for the planning stage. In this case, the use of the road surface sharing will be very useful to help the decision module to take the right decision at the right moment with the relevant information.

References

[1] Sun, Z., Bebis, G., Miller, R. (2006). On-road vehicle detection: A review, *IEEE Trans. Pattern Anal. Mach. Intell.*, 28 (5) 694711, May.

[2] Tzomakas, C., von Seelen, W. (1998). Vehicle detection in trac scenes using shadows, Ruhr Univ. Bochum: Institut fuer Neuroinformatik, Bochum, Germany, IRINI 98-06.

[3] Lee, M.-S. Kim, Y.-H. (2003). An ecient multitarget tracking algorithm for car applications, *IEEE Trans. Ind. Electron.*, 50 (2) 397399, Apr.

[4] Mobus, R., Kolbe, U.(2004). Multi-target multi-object tracking, sensor fusion of radar and infrared, *In:* Proc. IEEE Intell. Vehicles Symp. 2004, p. 732737.

[5] Kirchner and Ameling, C. (2000). Integrated obstacle and road tracking using a laser scanner, *In*: Proc. IEEE Intell. Vehicles Symp., Dearborn, MI, 2000, p. 675681.

[6] Streller, D., Furstenberg, K., Dietmayer, K. Vehicle and object models for robust tracking in trac scenes using laser range images, *In*: Proc. IEEE Int. Conf. Intell. Transp. Syst. , 2002, p. 118123.

[7] MacLachlan, R., Mertz, C. (2006). Tracking of moving objects from a moving vehicle using a scanning laser rangender, *In*: *Proc. IEEE Intell. Transp. Syst.* , Sep. vol. 2006, p.301306.

[8] Shafer, G. A (1976). Mathematical theory of evidence, Princeton University Press.

[9] Kuhn, H. W. (1955). The Hungarian method for assignment problem, Nav. Res. Quart., 2.

[10] Ahuja, R. K., Magnanti T. L., Orlin J. B. (1993). Network Flows, theory, algorithms, and applications, Editions Prentice-Hall.

[11] Denoeux, T., Smets P. (2006). Classication using Belief Functions: the Relationship between the Case-based and Model-based Approaches, *IEEE Transactions on Systems, Man and Cybernetics* B , 36 (6) 1395-1406

[12] Gruyer, D., Demmel, S., Magnier, V., Belaroussi, R. (2016). Multi-Hypotheses Tracking using the DempsterShafer Theory, application to ambiguous road context, *In*: Information Fusion, Elsevier, V. 29, May 2016, 40-56.

[13] Labayrade, R., Royere, C., Gruyer, D., Aubert, D. (2005). Cooperative Fusion for Multi-Obstacles Detection with the Use of Stereovision and Laser Scanner, in the special issue Robotics Technologies for Intelligent Vehicles in Autonomous Robots (Springer science), on, 19 (2) 117-140.

[14] Magnier, Valentin., Gruyer, Dominique., Godelle, Jerome (2005). Implementation of a multi-criteria tracking based on the Dempster-Shafer Theory, *In*: IEEE Intelligent Vehicles Symposium 2015 (IV 2015), 28 June-1 July, Seoul, Korea.