A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking

Cristiano Premebida, Gonçalo Monteiro, Urbano Nunes and Paulo Peixoto

Abstract— **This paper presents a sensorial-cooperative architecture to detect, track and classify entities in semi-structured outdoor scenarios for intelligent vehicles. In order to accomplish this task, information provided by a Lidar and a monocular camera is used in the here proposed system. The detection and tracking phases are performed in the laser space, and the object classification methods work both in laser space (using a Gaussian Mixture Model classifier) and in vision spaces (AdaBoost classifier). A Bayesian sum decision rule is used in order to combine the results of both classification techniques, and hence a more reliable object classification is achieved. Experiments confirm the effectiveness of the proposed architecture.**

I. INTRODUCTION

INTELLIGENT vehicles may have its perception capabilities improved if multi-sensor systems are combined in such a way that more relevant information is available as the result of a proper combination of individual sensor's measurements. This paper describes a multi-module architecture with the purpose of timely processing the sensory information, provided by a laserscanner and a camera, for a collision avoidance system operating in low-speed vehicles moving in Cybercars scenarios [2] . Examples of typical applications in this context are: pedestrian protection, traffic assistant system, pre-crashing warning, etc.

In the proposed architecture, two different classifiers, working in distinct sensor spaces, are combined by means of a practical Bayesian approach in order to provide a higher level inference and meaningful information to achieve a more reliable object classification. A cooperation strategy was also used to establish the coordinate correspondence between a Lidar and a monocular vision camera in order to facilitate the segmentation process and the object detection.

The architecture of the proposed system is subdivided in four subsystems: lidar-based, vision-based, coordinate transformation and tracking-classification subsystems. The lidar-based system detects the entities (objects) in the laser space, estimates its position, size, and gives the class probability for each detected object. The position of the objects is then converted to the camera coordinate system in order to define a region of interest (ROI) in the image space. The vision-based system receives the information from the coordinate transformation module, extracts a window from each detected object and performs an AdaBoost classification. The tracking and the final classification modules process the information from the others subsystems and output the class of the objects and its dynamic behavior (see Fig. 1).

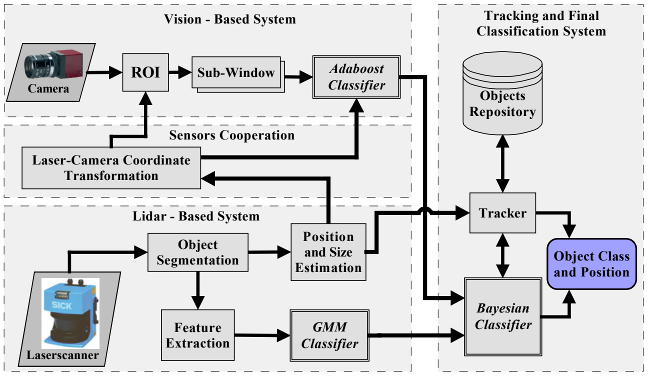

Fig. 1. Multi-module architecture using laser and vision information for object detection, tracking and classification.

Two different classifiers are used to distinguish two object categories: vehicles and pedestrians. A GMM classifier (GMMC) [3] is used to deal with the information provided by the Lidar, and an *AdaBoost* classifier, using *Haar-Like* features, is applied to classify the detected objects in the image frames. In order to combine the classifiers results, a sum decision rule is used based on the *a posteriori* probability provided by each classifier.

II. RELATED WORK

Several works have been carried out using Lidar in multiple target tracking and object classification. In order to classify objects, a variety of approaches have been proposed such as: i) voting schemes [4], ii) methods based on "heuristic" rules, and iii) multi-hypotheses [5]. The first two methods have the disadvantage of not having a self-consistent mathematical framework in order to support its stability and consistency; nevertheless the results presented by the authors show its feasibility. The multi-hypotheses method presented in [5] is based on feature tracking by means of a stochastic Filter, whose main drawback is its high computational cost and the complexity associated to the management of many hypothesis.

Vision systems are widely used for object detection [6] and constitute a feasible option to be implemented separately or along with Lidar [7]. Most of the object detection and tracking systems apply a simple segmentation procedure like

This work was supported in part by Portuguese Science and Technology Foundation (FCT), under Grant POSC/EEA-SRI/58279/2004, and by CyberC3 project (European Asia IT&C Programme).

The authors are with Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal. {cpremebida; ggmonteiro; urbano; peixoto}@isr.uc.pt

background subtraction or temporal difference to detect objects [8]. But these approaches have a weakness related to the background changes due to the camera motion. Many authors have dedicated their research to the pedestrians/obstacles detection [9],[10],[11],[12]. Papageorgiou [13] introduced a trainable object detection architecture based on a novel idea of the wavelet template that defines the shape of an object in terms of a subset of the wavelet coefficients of the image. Motivated in part by this last architecture, Viola et. al. [14] presented a machine learning approach for object detection, which is suitable for real-time operation while achieving high detection rates, using *Haar-like* features and an AdaBoost classifier. The vision-based object detection system described in this paper follows the Viola's approach.

III. LIDAR-BASED SYSTEM

In this section, the lidar-based system components: detection, tracking and classification are described (see Fig. 2).

A. Detection and Tracking Objects in Laser Space

To detect the entities, the surrounding is segmented using range information provided by the Laserscanner. Among several possible segmentation methods to be used on 2D laser range images [15], a linear KF-based method has been used to perform the segmentation stage. Within the extracted segments, tracking and data association techniques are performed in the laser referential system, where the objects under tracking are considered to evolve in time according to a stochastic dynamic model driven by process noises. The state and measurement noises are considered zero-mean, mutually independent and white Gaussian sequences with known covariances matrices. Object tracking using a multi-independent KF strategy is performed in the Cartesian space.

The segments/clusters witch define an object are basically a cloud of range-points. The centroid of each cluster is calculated and used as the characteristic-point i.e., the kinematic behavior of the object is described in respect to its centroid. Intrinsically connected with tracking, data association is performed considering two situations:

1) Segment to segment: the process of associating detected segments with other segments (non-classified objects) in the current scan;
2) Segment to object-tracker: the maintenance process, i.e. the association of observed segments with existing objects.

The first situation occurs when one, or more, current segments are "probably" related with a segment under tracking. Therefore, a decision has to be made in order to merge or not the valid segments. To deal with this situation, a combination of rectangular gate and feature-matching technique are used [16]. The second data association problem, i.e., observation-to-tracker association, is solved in a specific manner which accounts for the result of object classification. Hence, when a segment is finally classified, the data association technique is adjusted in accordance with the size and the dynamic behavior of the class into which the object has been classified.

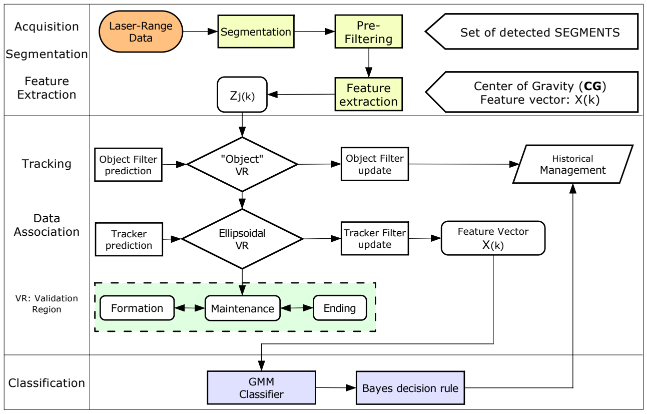

Fig. 2. Sequencial diagram illustrating the laser-based detection and tracking module.

B. Feature Extraction

Within the segments under tracking, probable objects of interest, a feature vector is extracted in order to perform the classification stage. This feature vector, used in the GMMC, has the following components:

f1) Segment centroid: or gravitational center;

f2) Normalized Cartesian dimension: this feature corresponds to the root mean square of the width (ΔX) and the length (ΔY) dimensions:

$$f2=\sqrt{\Delta X^2+\Delta Y^2} \tag{1}$$

f3) Internal std: denotes the standard deviation of the range-points with respect to the Segment centroid:

$$f3=\sqrt{\frac{1}{n-1}\sum_{j}\|r_n-\bar{x}\|} \tag{2}$$

f4) Radius: it is the radius of a circle extracted to the segment points. The circle fitting method used is based on [GUIV];

f5) Mean average deviation from median: it is given by: where $\tilde{x}$ is the median, defined as:

$$\tilde{x}=\begin{cases} X_{(k+1)/2} & \text{if K is odd} \\ \frac{1}{2}(X_{k/2}+X_{(k/2)+1}) & \text{if K is even} \end{cases} \tag{3}$$

Initially the GMMC was implemented with three feature components and the estimated velocity [IROS2006], however the actual GMMC version was augmented with features $f4$ and $f5$. The utilization of the later two feature components were inspired in [Arras].

C. Lidar-Based Classifier

This section describes the GMMC, which was implemented based on data from the Lidar. Each object category is modeled by a finite-GMM distribution whose parameters were estimated during a supervised training. A Maximum A Posteriori (MAP) decision rule is used to calculate the *a posteriori* probability of each object to belong to the classes of interest.

The object classes are modeled by a weighted combination of Gaussian probability density functions (PDF) which are referred to in this context as Gaussian components of the mixture model describing a class (object category). In a GMM model, the probability distribution of a multi-dimensional random vector $\mathbf{x}$ is a mixture of M Gaussian probability density function (GPDF) (4), defined as follows:

$$p(\mathbf{x}|\Theta) = \sum_{m=1}^{M} \alpha_m p(\mathbf{x}|\theta_m) \tag{4}$$

where $\theta_1, \ldots, \theta_M$ are the parameters of the Gaussian distributions and $\alpha = [\alpha_1, \ldots, \alpha_M]$ is the weighted vector, such that $\sum_{m=1}^{M} \alpha_m = 1$. The complete set of parameters that specify the mixture model is $\Theta = (\alpha; \theta_1, \ldots, \theta_M)$, with each parameter $\theta_m = (\mu_m, \Sigma_m)$ consisting of a mean vector μ and a covariance matrix Σ. Considering a d-dimensional feature-vector X, the likelihood of each class (q_i) are described as linear combinations of Gaussian mixture probability density functions:

$$p(X|q_i, \Theta^i) = \sum_{m=1}^{M} \alpha_m^i p(X|\theta_m^i) \tag{5}$$

where each GPDF component is given by

$$p(X|\theta_m^i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_m^i|}} \exp[-\frac{1}{2}(X-\mu_m^i)^T (\Sigma_m^i)^{-1}(X-\mu_m^i)] \tag{6}$$

The GMM parameters[1] for each object class were estimated using the expectation-maximization (EM) algorithm, i.e., for a set of N labeled feature-vectors ($\mathscr{X}^N = X_1, X_2, \ldots, X_N$) the EM algorithm [17][18] calculates M Gaussian parameters-vector that maximizes the joint likelihood (7) among the GPDF-components:

$$p(\mathscr{X}^N|q_i, \Theta^i) = \prod_{j=1}^{N} p(X_j|q_i, \Theta^i) \tag{7}$$

To select which of the categories (q_i), modelled by the GMM parameters Θ^i, fits the current observation feature-vector $X(k)$, i.e at current time interval k, a practical maximum *a posteriori* decision rule based on the likelihood (5) and on the prior probability $P(q_i|\Theta^i)$ is used. Considering the attributes equiprobable, the posterior probability $P(q_i|X, \Theta^i)$ for all categories is calculated as:

$$P(q_i|X, \Theta^i) = \frac{p(X|q_i, \Theta^i) P(q_i|\Theta^i)}{P(X)} \tag{8}$$

where $P(X)$ is the marginal likelihood (normalized factor). By knowing the initial prior probability for each class, our classification method outputs the maximum a posteriori probability for each observed vector. To decide which is the most "likely" class q_i, the MAP decision rule is used as follows:

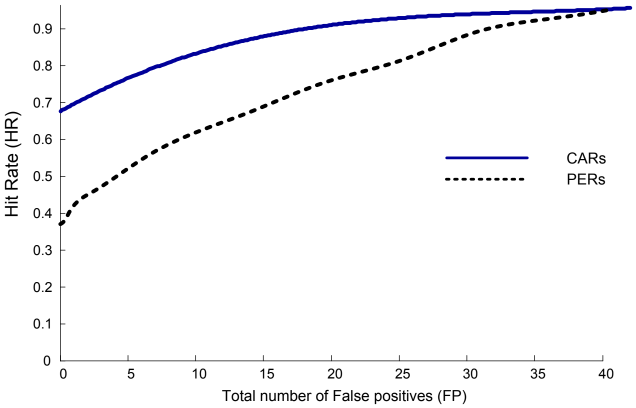

Fig. 3. ROC curve of the GMM classifier based on Lidar data to detect pedestrians and Vehicles.

$$Object \in q_i \quad if \quad P(q_i|X(k), \Theta^i) = \max(P(q_u|X(k), \Theta^u)) \tag{9}$$

where $u = 1, 2, \ldots, N_{\mathscr{C}}$, and $N_{\mathscr{C}}$ is the total number of classes. The key idea of our algorithm is that the current *a priori* probability, after initialization, is set with the value of the past *a posteriori* probability, as shown in the Algorithm 1.

Algorithm 1 GMMC using a MAP decision rule

```
Input: $X(k)$: observed feature vector;
Output: MAP for each detected object (segment).
k: time frame;
i=1,2,...,Nc: class index;
$\Theta^i$: trained model for each class;
$P(\Theta^i_{k=0})$: initial prior probability;
while k do
    $N_S$: number of current detected segments;
    $j = 1, 2, ..., N_S$: segment index ($S_j$);
    $X_j(k)$: feature-vector for each segment;
    for $j = 1 : N_S$ do
        $P(\Theta^i_k)_j = P(\Theta^i|X_j(k-1))_j$;
        for $i = 1 : Nc$ do
            $P(\Theta^i|X_j(k))_j = \frac{P(X_j(k)|\Theta^i)_j P(\Theta^i_k)_j}{P(X)}$
        end for
        $MAP_j = \max_i P(\Theta^i|X_j(k))_j$
    end for
end while
```

IV. VISION-BASED SYSTEM

This system uses a set of *Haar-Like* features (see Fig. 4) to extract the information from the given image. The detection of the objects is performed using these features as an input to an AdaBoost classifier. The AdaBoost classifier is then trained to perform the detection of the objects on the environment. The main goal of this learning algorithm is to find a small set of *Haar-Like* features that best classifies the object class, rejecting most of the background objects, and to construct a robust classifier function. To support this

[1] To avoid singularities and facilitate the EM convergence, the number of Gaussian components was restrict to 1, 3 and 5.

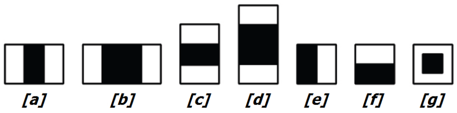

Fig. 4. Subset of the Haar-like features prototypes used in the object detection. *a*, *b*, *c* and *d* are the line features, *e* and *f* the edge features and *g* is the center-surround feature.

purpose, a weak learning algorithm is designed to select the single feature which best separates the positive and negative examples. For each feature, the weak learner determines the optimal threshold classification function, so that the minimum number of examples are misclassified. A weak classifier (10) $h_j(x)$ consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the inequality sign. The value 1 represents the detection of the object class and 0 represents a non-object. Each of these classifiers *per si* is not able to detect the object category. Rather, it reacts to some simple feature in the image that may be related to the object. The final classifier is constructed with the weighted sum of the weak classifiers and is represented by the Equation 11.

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_i(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$H(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The detection of the objects is done by sliding a search window through each sub-image and checking whether an image region at a certain location is classified as the object class. Initially, the detection window is of the same size of the classifier (15×37 for pedestrians and 30×30 for cars), then the window's size is increased by β until the size of the window is equal to the sub-image size ($\beta = 1.05$). This process is repeated for each of the object class trained. A classifier is constructed for each of the objects that we want to classify (pedestrians and cars).

To obtain the a posteriori probability of the AdaBoost classifier it was used the formulation called *Logistic Correction* [19]. Initially, in AdaBoost, each of the examples in a training set (x_i, y_i) have the same weight. At each step i a weak learner h_i is trained on the weighted training set. The error of h_i determines the model weight α_i and the future weight of each training example. Assuming $y_i \in \{-1, 1\}$ and $h_i \in \{-1, 1\}$, the output of the boosted model is given by (12), where T is the number of weak classifiers.

$$F(x) = \sum_{i=1}^{T} \alpha_i h_i(x) \tag{12}$$

It is shown that AdaBoost builds an additive logistic regression model for minimizing $E(\exp(-yF(x)))$. They show that $E(\exp(-yF(x)))$ is minimized by:

$$F(x) = \frac{1}{2} \log \frac{P(y=1|x)}{P(y=-1|x)} \tag{13}$$

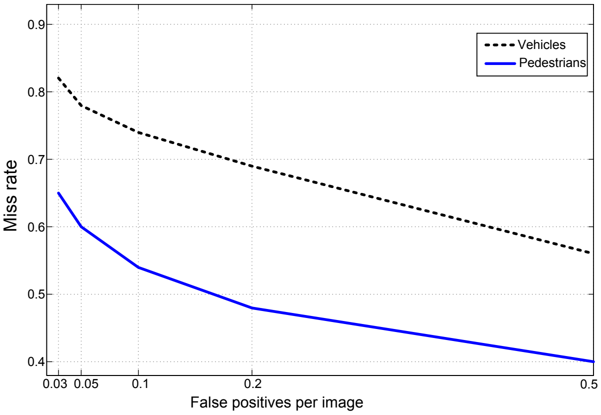

Fig. 5. DET curve for the Adaboost pedestrian and vehicles detection system.

This suggests applying a logistic correction in order to get back the posterior probability:

$$P(q_i|X) \triangleq P(y=1|x) = \frac{1}{1+\exp(-2F(x))} \tag{14}$$

Some preliminary experiments were made with the AdaBoost classifier in order to test its robustness and accuracy. Two training data sets have been used: for pedestrians, the INRIA dataset [20], and, for cars, the CALTECH dataset [21]. The INRIA dataset is composed by 2416 pedestrian images and 12180 background images, for the training phase, and 997 pedestrian images in a variety of realistic scenes, for the test phase. The CALTECH dataset is composed by 3698 car images and 13690 background images for the training phase, and 337 car images, in the test phase. All the tests were made in the same conditions of [22]. The results of these are presented in the DET curves shown in Fig. 5. The AdaBoost classifier runs in real-time ($\approx 13 fps$).

V. COORDINATE TRANSFORMATION SYSTEM

The task of multi-sensor cooperation leads to establishing a correspondence between the measurements gathered by distinct sensors. In this case, it is necessary to find a correspondence between the camera reference and the Lidar reference system. The coordinate transformation subsystem (see Fig. 6) is used to calculate this correspondence. Thereby the result of the object's position and size estimation, in the Lidar space, is used to construct a ROI in the image frame by means of a set of coordinate transformations. The ROI is formed in the image plane as the result of a correspondence between the objects under tracking (in the laser reference) and the objects in the image frame, in order to facilitate the process of detection in the vision-based system and to decrease the computational time of the AdaBoost classifier.

The advantages of using a Laserscanner in cooperation with the vision system are: simpler segmentation process and data processing; the laser measurements are not very sensitive under weather changes and consequently the whole system become more robust. Moreover the laser sensor has a good accuracy in the distance/depth measurements,

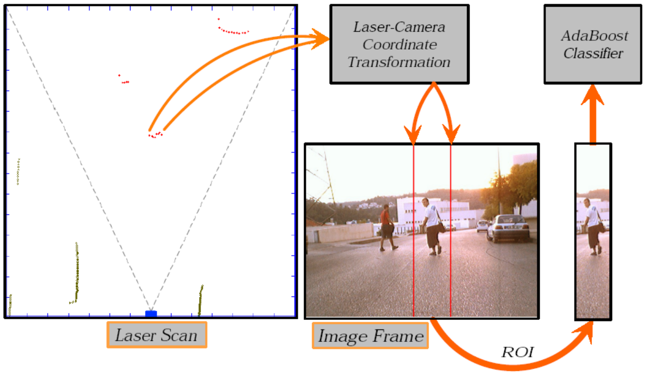

Fig. 6. Sensory cooperation between the Laserscanner and the monocular camera.

while vision systems are cheap and have a very detailed information of the surrounding.

A. Calibration

The Lidar and the camera are mounted in a common base, where their axes are parallel and "ideally" aligned. The calibration procedure is necessary to obtain a mapping expression to transform points in the laser reference system $\{L\}$ to the camera reference system $\{C\}$ and then to the image plane. Considering that the laser and the camera are parallel and aligned with one another, and using a flat target ("checkerboard") positioned at different distances to the laser-camera common base, the transformation between $\{L\}$ and $\{C\}$ was obtained under a quadratic error minimization criteria. Various images and laser measurements taken at different positions of the target were used to estimate this coordinate transformation. The camera's intrinsic and extrinsic parameters were calculated using a Matlab Toolbox [23].

Once the range points, that define the segments detected by the Laserscanner, are transformed to the camera reference frame and hence are projected in the image plane, a ROI is constructed in the image plane corresponding to each object that rely in camera's FOV. With the Ladar information it is only possible to obtain the horizontal limit of the object in the image. If it is assumed that the vehicle/robot moves on a "flat" surface, it is geometrically easy to obtain the bottom limit of the ROI. The top limit of the ROI can be estimated using the distance to the object and the maximum height for all categories.

VI. COMBINING CLASSIFIERS

The data from the camera and the Lidar is used by different classifiers whose results are fused. Inspired in [24] it was used a sum rule to combine the classifiers outputs discussed previously. Based on the *a posteriori* probability calculated by each classifier, the sum decision rule is used to ultimate classify an object.

Let us consider the number of classifiers as NC, and the feature vector used by the ith classifier denoted by $\mathscr{X}_i$. Let us assume that each class q_j is represented by a class-conditional probability density function $p(\mathscr{X}_i|q_j)$ and its a priori probability of detection $P(q_j)$. Given the PDF and the a priori probability, the following classical decision rule can be stated:

$$\begin{aligned} &assign \;\; Object \longrightarrow q_j \;\; if \\ &P(q_j|\mathscr{X}_1,\ldots,\mathscr{X}_{NC}) = \max_k P(q_k|\mathscr{X}_1,\ldots,\mathscr{X}_{NC}) \end{aligned} \tag{15}$$

Assuming that the features vectors are conditionally statistically independent, and that the posterior probability of each classifier do not deviate dramatically from the prior probability, after some mathematical formulations [24], a "practical" combinational Bayesian decision rule is stated as:

$$\begin{aligned} &assign \;\; Object \longrightarrow q_j \;\; if \\ &(1-NC)P(q_j) + \sum_{i=1}^{NC} P(q_j|\mathscr{X}_i) = \\ &\max\nolimits_{k=1}^{N_{\mathscr{C}}}[(1-NC)P(q_k) + \textstyle\sum_{i=1}^{NC} P(q_k|\mathscr{X}_i)] \end{aligned} \tag{16}$$

The "sum" decision rule depends on the prior probability of occurrence of each class q_j and the posterior probabilities yielded by the respective classifiers.

VII. EXPERIMENTAL RESULTS

The experimental were conducted in some outdoor environments, constrained to low speed moving object (under 20 Km/h) and up to 20 m of distance from the vehicle (Fig. 7 shows one of the vehicles used in the experiments). The sensory devices were mounted on the front of the vehicle approximately 65cm above the ground. Our setup is composed of a LMS200 connected to the PC through a RS422-USB conversor and a firewire camera.

The system was tested using data sequences (image frames and laser points) of pedestrians and cars in different size, pose and lighting conditions. The Hit Rate (HR) and the number of False Positives (FP) of each classifier are shown in the Tables I and II. Fig. 8 shows some detected objects, in the image space, during the experimental trial.

In the vision based system, the classification performance of cars is better than that of the pedestrians. This happens because pedestrians are non-rigid bodies. In other words, the shape and size of a pedestrian varies greatly, and therefore the model of pedestrians is much more complex than that of rigid objects. For the laser based system, the classification results for car like category was, in this context, less accurate than for pedestrian class. This is more evident when the detected cars and the ego-vehicle are moving. A remarkable disadvantage on using LMS200 in outdoor scenario, far from 15 m, is its poor capacity to detect cars is some circumstances.

TABLE I

CLASSIFICATION RATES FOR PEDESTRIANS.

	Hit Rate (%)	False Positives (%)
AdaBoost classifier	69.5	33.2
GMM classifier	67.4	21.1
Combining classifier	82.9	6.0

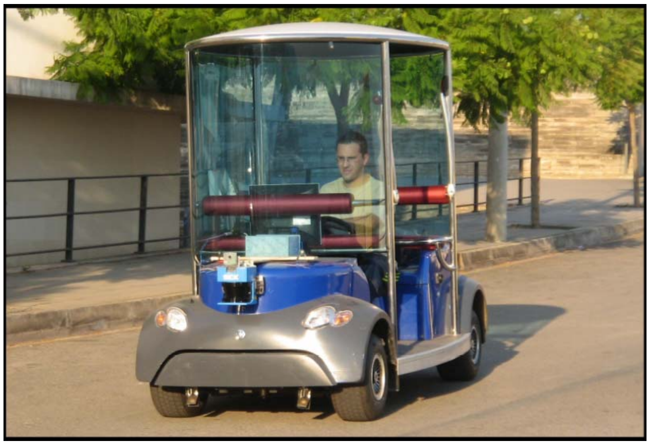

Fig. 7. Vehicle used in the experiments conducted in an outdoor scenario. The Lidar and the camera are mounted on the front of the vehicle.

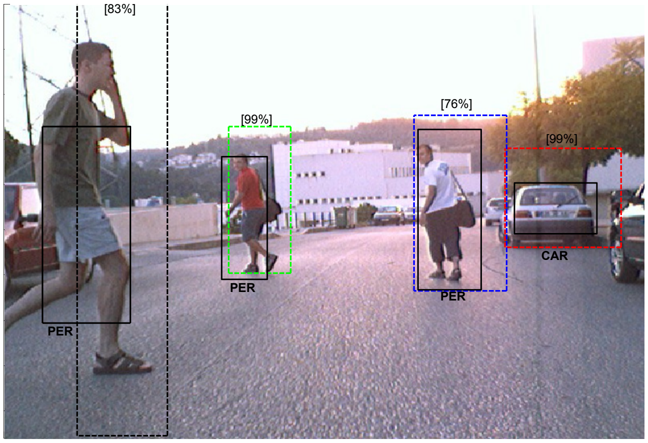

Fig. 8. Output of the vision-based system. The vertical lines represent the ROI defined by the laser.

VIII. CONCLUSIONS AND FUTURE WORK

A multi-modal system for detecting, tracking and classifying objects in outdoor environment was presented. The objects of interest were restricted to pedestrians and cars. A cooperative technique was implemented in order to “fuse” the information from a lidar and a monocular vision camera. Some results were obtained for each classifier separately and also for a classifier combination scheme, employed to improve the performance of the system. This compound classification technique can be summarized as a practical Bayesian framework that combines, in a probabilistic sense, the a posteriori probability of each classifier by means of a sum rule. Preliminary experiments, conducted on some real urban street scenes, demonstrate that the system is able to detect and classify pedestrians and cars in various positions, shapes, sizes and colors with a good accuracy.

In the future, we intend to integrate more cameras in the system to increase the field of view and consequently the perception of the surrounding environment of the autonomous vehicle. Moreover, new methodologies/classifiers will be studied. It is also planned to investigate new combinations of classifiers to improve the global system performance.

TABLE II

CLASSIFICATION RATES FOR VEHICLES.

	Hit Rate (%)	False Positives (%)
AdaBoost classifier	66.7	20.4
GMM classifier	78.8	16.5
Combining classifier	84.0	8.6

REFERENCES

[1] Kai Arras, O. Mozos and W. Burgard, “Using Boosted Features for the Detection of People in 2D Range Data”, *IEEE International Conference on Robotics and Automation* - ICRA, pp. 3402-3407, Rome, Italy, 2007.

[2] Cybercars. Cybernetic technologies for the cars in the city. [online], www.cybercars.org, 2001.

[3] C. Premebida, and U. Nunes, “A multi-target tracking and GMM-classifier for intelligent vehicles”, *in 9th International IEEE Conference on Intelligent Transportation Systems*, Toronto, Canada, 2006.

[4] A. Mendes, L. C. Bento and U. Nunes, “Multi-target detection and tracking with laser range finder”, *in IEEE Intelligent Vehicles Symposium*, Parma, Italy, 2004.

[5] D. Streller and K. Dietmayer, “Object tracking and classification using a multiple hypothesis approach”, *IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004.

[6] U. Franke and S. Heinrich, “Fast obstacle detection for urban traffic situations”, *IEEE Trans. Intell. Transport. Syst.*, vol. 3, pp. 173-181, Sept. 2002.

[7] J. Neira, J.D. Tard, J. Horn and G. Schmidt, “Fusing range and intensity images for mobile robot localization”, *in IEEE Trans. Robotics and Automation*, Vol. 15, No. 1, pp 76- 84, 1999.

[8] D. Toyh and T. Aach. “Detection and recognition of moving objects using statistical motion detection and Fourier descriptors.” *in ICAP 03*. 2003.

[9] M. A. Sotelo, I. Parra, D. Fernandez and E. Naranjo. “Pedestrian Detection using SVM and Multi-feature Combination.” *in Proc. of the IEEE ITSC 2006*. Toronto. Canada. 2006.

[10] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, “Shape-based pedestrian detection” *in Proc. IEEE Intelligent Vehicles Symposium.* Dearborn, Michigan, USA, October, 2000.

[11] D. M. Gavrila and V. Philomin, “Real-time object detection for smart vehicles,” *in Proc. of the Seventh IEEE International Conference on Computer Vision*. ISBN: 0-7695-0164-8, 1999.

[12] R. Labayrade, C. Royere, D. Gruyer, and Aubert, “Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner,” *in Proc. International Conference on Advanced Robotics.* pp. 1538-1543, 2003.

[13] M. Oren, C. Papageorgiou and P. Sinha. “Pedestrian detection using wavelet templates.” *in Proc. IEEE CVPR 1997*. 1997.

[14] P.Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features.” *in IEEE CVPR 2001*, 2001.

[15] C. Premebida, and U. Nunes, ”Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications”, *in Proc. 5th National Festival of Robotics, Scientific Meeting (ROBOTICA)*, Coimbra, Portugal, 2005.

[16] Y. Bar-Shalom and X.R. Li, *Multitarget-Multisensor Tracking: Principles & Techniques*,YBS Publishing; 1995.

[17] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, John Wiley and Sons, New York, NY; 1997.

[18] P. Paalanen, J. K. Kamarainen, J. Ilonen, and H. Klviinen, “Feature representation and discrimination based on Gaussian mixture model probability densities - practices and algorithms”, *Research Report-95*, Lappeenranta University of Technology, Dep. of Information, 2005. (www.it.lut.fi/project/gmmbayes/).

[19] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 38(2), 2000.

[20] Dalal, N. and Triggs, B., ”Histograms of Oriented Gradients for Human Detection”, *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, (2005), 886–893.

[21] L. Fei-Fei, R. Fergus and P. Perona, ”One-Shot Learning of Object Categories”. *IEEE Trans. Pattern Recognition and Machine Intelligence*. In press, (2004).

[22] L. Oliveira, P. Peixoto and U. Nunes, ”A Hierarchical Fuzzy Integration of Local and Global Feature-based Classifiers to Recognize Objects in Autonomous Vehicles”,*In: IEEE ICRA 2007 Workshop on ”Planning, perception and navegation for intelligent vehicles”*, Rome, 2007.

[23] Camera Calibration Toolbox for Matlab. [online], www.vision.caltech.edu/bouguetj/calibdoc/, 2006.

[24] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On combining classifiers”, *in IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 20, no. 3, pp. 226–239, 1998.