# A Multi-Target Tracking and GMM-Classifier for Intelligent Vehicles

Cristiano Premebida and Urbano Nunes, *Member*, *IEEE*

***Abstract*— Intelligent vehicles need reliable information about the environment in order to operate with total safety. In this paper we propose a flexible multi-module architecture for a Multi-Target Detection and Tracking System (MTDTS) complemented with a Bayesian object Classification layer based on finite Gaussian Mixture Models (GMM). The GMM parameters are estimated by an Expectation Maximization (EM) algorithm, hence finite-component models were generated based on feature-vectors extracted from object's classes during the training stage. Using the joint mixture Gaussian pdf modelled for each class, a Bayesian approach is used to distinct the object's categories (persons, tree-trunks/posts, and cars) in a semi-structured outdoor environment based on data from a laser range finder (LRF). Experiments using real data scan confirm the robustness of the proposed architecture. This paper investigates a particular problem: detection, tracking and classification of objects in cybercars-like outdoor environments.**

## I. INTRODUCTION

IN the context of cybercars (www.cybercars.org), or in more general ITS and advanced driver assistance system (ADAS) technologies, applications on safe navigation, path following, platooning, obstacle avoidance, collision warning, object/target detection and classification, or a combination of the previous tasks, are typical and interesting problems to be solved. Several works have been done on using laser-scanners in multiple target tracking and classification. In order to classify objects, a voting scheme [4] and a multi-hypotheses approach [5] are examples of proposed methods. Vision based systems are also widely used for object/pedestrian detection [6],[9],[10].

In our case a multi-target detection and tracking system (MTDTS), complemented with a classification module, is proposed to handle the tasks of object detection, tracking and classification in outdoor semi-structured environments (cybercars-like scenarios). A flexible multi-module scheme to deal with this situation is presented, where each module is designed to perform a pre-determined task in a specific manner, but taking into account the whole system (multi-dependency framework). Although this architecture allows the use of other sensors, the results presented in this paper are only from using a laser range finder (LRF), mounted on a vehicle platform.

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).

C. Premebida is supported by the Programme AlBan, the European Union Programme of High Level Scholarships for Latin America, scholarship $n^o$ E04M029876BR.

C. Premebida and U. Nunes are with University of Coimbra, Polo-II, and also with the Institute for System and Robotics, ISR, Coimbra, Portugal. {cpremebida;urbano}@isr.uc.pt

Instead of discussing the details of each module under a general situation, this paper investigates a particular problem: automatic detection-tracking and classification of a set of object's categories of interest (persons, trees/posts and cars) in outdoor environments using data from a 2D LRF. Hence, the major effort of this work has been focused on some modules that we think are more critical and relevant: tracking, data association and classification modules.

In order to make the paper more tractable, a concise overview of our framework and the primary modules description are presented in section 2. Section 3 addresses the tracking and data association, and section 4 is dedicated to object classification. Finally sections 5 and 6 present the experimental results and conclusions, respectively.

## II. MAIN ARCHITECTURE DESCRIPTION

Our experimental platform is a bi-steerable four wheel autonomous vehicle. The control system is composed by three main subsystems [3], which are designated by path-following controller (PFC), vehicle's pose estimator (VPE) and multi-target detection and tracking system (MTDTS). The latter subsystem works in an independent PC connected by a CAN bus throughout the main-system and constitutes the scope of this paper. The essential components of our proposed MTDTS architecture include the modules: data acquisition, segmentation (and pre-filtering), feature extraction, tracking and data association, object classification, a data context-base repository, and a coordinate updating feedback (see Fig. 1).

### A. Primary Modules

Data Acquisition, Coordinate Updating, Pre-filtering/Segmentation, and Feature Extraction, henceforth called Primary Modules are depicted as white-blocks in Fig. 1, and will be described in this section.

**Data acquisition** module is constituted by a 2D LRF, connected through a dedicated CAN bus using a Microcontroller-based RS422-to-CAN bus convertor module. The scan data transfer rate is approximately 36Hz.

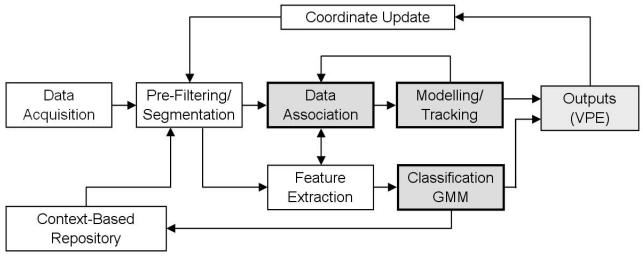


Fig. 1. MTDTS system architecture overview

**The Coordinate updating** is necessary to take into consideration the incremental vehicle displacement along the trajectory; such values are estimated by the VPE subsystem in a local reference frame. The innovations ($\upsilon_{DA}$), in the Data Association stage are, therefore calculated as

$$\upsilon_{DA} = \begin{cases} \upsilon_x = z_x - (\hat{z}_x - \Delta x_R) \\ \upsilon_y = z_y - (\hat{z}_y - \Delta y_R) \end{cases} \quad (1)$$

where $(\Delta x_R, \Delta y_R)$ are the vehicle displacement, $(z_x, z_y)$ and $(\hat{z}_x, \hat{z}_y)$ are the measurement and estimated values (positions) respectively.

**Pre-filtering** aims to perform the following tasks:
• filtering the incoming data by means of a Kalman Filter (KF) segmentation approach;
• discard segments with less than two points;
• transform the incoming scan data from Laser frame to the vehicle reference frame.

**Segmentation** aims to detect break-points (discontinuities that indicate groups of range points probably belonging to the same object), i.e. its purpose is to identify Segments ("clusters"). Among several possible segmentation methods to be used on 2D laser range images, a linear KF-based method [2] has been selected in our work.

Considering a full scan as an ordered sequence of $N_{\mathscr{S}}$ measurement points in the form $\mathscr{S} = \{(r_l, \theta_l) | l = 1, ..., N_{\mathscr{S}}\}$, where $(r_l, \theta_l)$ denotes the polar coordinates of the $l^{th}$ *scan point*, a group of scan points that constitute a segment $S_j$ can be expressed as

$$S_j = \{(r_n, \theta_n)\}, \quad n \in [l_{begin}, l_{end}], \quad j = 1, \ldots, N_S \quad (2)$$

where $N_S$ is the number of segments detected in the current scan, $l_{begin}$ and $l_{end}$ are the first and the last scan points that define each segment. A segment can also be defined in Cartesian coordinates ($x_n = r_n \cos\theta_n, \quad y_n = r_n \sin\theta_n$) if convenient. Applying, as input, a complete sequence of $N_{\mathscr{S}}$ scanned points, the segmentation algorithm [2] outputs for each *j*-th detected segment an array of values starting from the first beam-point ($l_{begin}$) until the last beam-point ($l_{end}$).

A simple criteria was adopted to discard or merge spurious points, i.e. disperse range points, which can be summarized by the following steps:

```
If   v_k ≤ γ_k   &   v_{k+1} ≤ γ_{k+1}
Then: merge range points;
Else: a breakpoint is detected;
```

where $v_k = \upsilon_k^T S_k^{-1} \upsilon_k$ is the gate (or validation region) which depends on the innovation $\upsilon_k$ and its associated covariance matrix. The gate thresholds $\gamma_k$ and $\gamma_{k+1}$ are assigned from a $\chi_1^2$ test table, corresponding to a determined gate probability.

**Feature extraction** stage aims to extract relevant information from the incoming scan points that constitute each segment, which is crucial for the reliability and robustness of the whole system. This information is used in the classification and can be useful for object geometrical representation. Once one or more objects are detected by the LRF, the segments of points (2) associated with each object are processed in order to extract two feature vectors designed by **f** and **g**. The former is calculated at every time interval *k* and constitutes a "crucial" feature-vector. This vector has four components:
$f_1$: object centroid (mass center);
$f_2$: normalized Cartesian dimension;
$f_3$: internal *std* (calculated from the centroid);
$f_4$: object speed (estimated from the tracking stage);
The last three components are used directly as inputs of Classification module, while $f_1$ is the characteristic-point used in the tracking module.

The second feature vector (**g**) is extracted only and after a segment $S_j$ is classified as one of the object's categories. This feature vector constitutes the Cartesian parameters of a geometrical-primitive, which can be a circle (center and radius) or a line-segment (line coefficients; line start and line endpoint) depending of the object's category. For representation/visualization purposes, the objects classified as trees-trunks, posts or persons are represented by circles, while car-like vehicles are represented by line-segments.

Circle and line-segment extraction is itself not a complicated problem and can be solved using several known methods, such as a classical least-square algorithm or a Kalman filter-based method [8]. The advantage of using geometrical features data rather than the scanned range-points is that the former is clearer, easier to interpret and smaller than the latter. In the end, for each object a structure variable is computed as follows

```
S_feature[j]   {
   int   class; //1: tree/post; 2:car; 3:persons
   int   ftype; //1: circle; 2:line
   float   f; //first feature vector
   float   g; //geometrical feature vector   }
```

## III. TRACKING AND DATA ASSOCIATION

In this section, object modeling, object tracking and data association are described. It is to worth note that the measurement vector ($z_k$) used in these modules is the "characteristic point" defined by the first component of the feature vector **f**, i.e. the observation vector is the Cartesian coordinates of the centroid (position) of each detected object.

### A. Model

The object's classes of interest are sub-divided in three distinct groups. Hence three kinds of models [7] are considered:
• A static model: related to tree trunks and posts;
• A white acceleration model: car-like vehicles;
• A Wiener process acceleration model: related to pedestrian.

### B. Tracking

Tracking is done using a multiple independent linear Kalman Filter (KF) approach. This process is based on an individual KF for each target-segment that starts using a

white acceleration model. Once a segment is classified the corresponding filter is used with the suitable object's model as described in the previous section. For all trackers three mandatory steps are performed: track initialization; track maintenance; track ending (time of life).

A tracker is initialized (created) if a segment $S_j$ remains in the sensor field of view at least more than two consecutive scans, i.e. to avoid "false alarms". Tracker maintenance is performed during the stages of prediction/updating and data association. Finally, a tracker is destroyed if the object disappears, due to occlusion or missing association, during more than three predicted-cycles. After a segment is finally classified the current model is then changed to the corresponding object model and the filter (tracker) is updated with the current values of the state estimation covariance matrix, $P(k|k)$, and the current predicted state vector.

### C. Data Association

Closely connected with tracking, data association is the module discussed in this section. The following two cases of data association are considered:

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 existed objects.

Solving the first problem is necessary when one, or more, current segments are "probably" related with a segment being tracked. This is not trivial and demands much effort specially when more than one observed segment actually constitute just one tracked non-classified object. In general this situation occurs due to partial occlusion or noise measurements. In order to deal with "non-classified" segment-to-segment association, we propose the following procedure:

1) A rectangular-gate is constructed around the estimated centroid position of each detected segment:

$$|z-\hat{z}| = |\tilde{z}| \leq K_G \sigma_r \tag{3}$$

2) If do not exist "conflicts" in the rectangular validation region, then each segment is treated as an individual object;

3) Else, if any conflict occur, a more restricted ellipsoidal-gate is used, in order to be more confident about that segment association:

$$\upsilon^T \cdot S^{-1} \cdot \upsilon \leq \gamma \tag{4}$$

4) If the conflict persists, then the "validated" segments are merged and treated as a single object.

The residual standard deviation ($\sigma_r$) in (3) depends on the values of the observation and prediction variances. The gate thresholds $K_G$ (3) and $\gamma$ (4) are obtained from tables of the Chi-square distribution (Gaussian case) and also depend on the measurement dimension [7].

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, rather than using a general approach. Therefore, when a segment is labelled to a determined class the data association technique has to be adjusted to that class, thereby three situations are considered:

- For Class1 (tree trunks or posts): a circular validation region (*VR*), centered at the estimated position, is used;
- For Class2 (persons): centered ellipsoidal *VR* is applied;
- For Class3 (car-like vehicles): a rectangular *VR* is employed.

A realistic and common case to be solved during the data association stage occurs, for example, when the LRF detects the legs of a walking pedestrian (see Fig. 2). It is obvious that the measured points that correspond to the legs are actually two segments close each other (segments identified by square and circle markers form the actual observation segment for the same object). This kind of situation is also a common occurrence in other classes of objects.

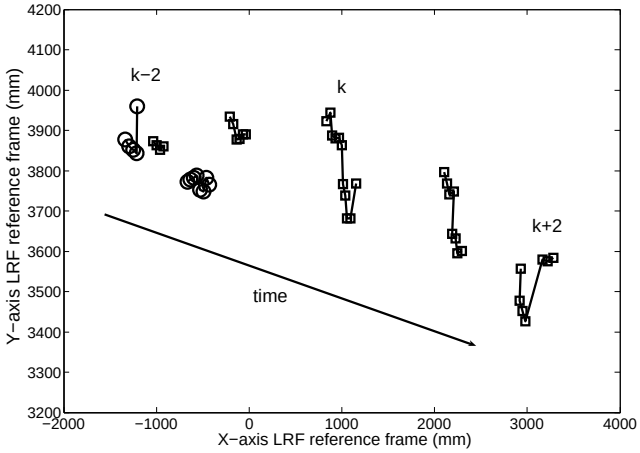


Fig. 2. Splitting and merging scan data for the same object at five instants of time.

## IV. CLASSIFICATION

A classifier algorithm do not usually work suitably with raw data, then a vector of structured data (feature array) is used as aforementioned.

The detected objects are classified into one of the three predefined classes using finite-components GMMs (with weights $\alpha_m$) estimated using samples collected in a real outdoor scenario. The categories were considered with equi-probable features, and a supervised learning training was used to determine the GMMs model parameters.

For object classification a Bayesian classifier, that outputs in each interval of time which class/category fits more likely the current observed segment $S_j$, has been used.

### A. GMM

A Gaussian mixture model is a weighted combination of Gaussian probability density functions (pdf) which are referred 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 $x$ is a mixture of $M$ Gaussian probability density function $p(x|\theta_m)$ defined as follows

$$p(x|\Theta) = \sum_{m=1}^{M} \alpha_m p(x|\theta_m) \quad (5)$$

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 $\mu$ and a covariance matrix $\Sigma$. Considering a $d$-dimensional feature-vector $\Omega$, the mixture Gaussian pdf for each *i-th* class, which is modelled by $\Theta^i$, is calculated as

$$p(\Omega|\Theta^i) = \sum_{m=1}^{M} \alpha_m^i p(\Omega|\theta_m^i) \quad (6)$$

where each pdf component is given by

$$p(\Omega|\theta_m^i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_m^i|}} \exp[-\frac{1}{2}(\Omega - \mu_m^i)^T (\Sigma_m^i)^{-1} (\Omega - \mu_m^i)] \quad (7)$$

The expectation-maximization (EM) algorithm is a general method of finding the locally maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values. In this work, the GMM parameters for each object's class model, $\Theta^i$, were estimated using a EM algorithm [1], i.e. for each set of $N$ labelled feature-vectors ($\Omega^N = \Omega_1, \Omega_2, \ldots, \Omega_N$) the EM algorithm calculates $M$ Gaussian parameters-vector that maximizes the joint likelihood among the pdf-components:

$$p(\Omega^N|\Theta^i) = \prod_{j=1}^{N} p(\Omega_j|\Theta^i) \quad (8)$$

### *B. Bayesian Classifier*

Once we have modelled each class with a finite Gaussian mixture pdf, to select (classify) which category ($q_i$), modelled by the parameters $\Theta^i$, fits the current observation feature-vector $\Omega_k$, i.e observation at current time interval $k$, a Bayesian decision framework based on the log-likelihood and on the log-prior probability is used.

Computing the log-likelihood has great advantages. First of all, the computational requirements are favorable because it turns the product (8) into a summation. Moreover for Gaussian distribution, it avoids the computation of the exponentials in the pdf expression (7), which becomes:

$$\log(p(\Omega|\theta_m^i)) = \begin{cases} -0.5[d\log(2\pi) + \log(|\Sigma_m^i|) + \\ (\Omega - \mu_m^i)^T (\Sigma_m^i)^{-1} (\Omega - \mu_m^i)] \end{cases} \quad (9)$$

Furthermore, since $log(.)$ is a monotonically growing function, the log-likelihoods have the same relations of order as the likelihoods, thus allowing us to use the former directly to classify the objects.

Considering the features equi-probable, the logarithm of the posterior probability $\log(P(\Theta^i|\Omega,))$ for all categories is proportional to the sum of the log-likelihood with the prior probability's logarithm:

$$\log(P(\Theta^i|\Omega)) \approx \log(p(\Omega|\Theta^i)) + \log(P(\Theta^i)) \quad (10)$$

Rather than trying to estimate the posterior probability, it is much more practical and convenient to resort to Bayes'law, which makes use of likelihoods and the prior probability. The former is calculated for each time frame $k$ as in (9) and the later comes from the last ("past") estimated posterior probability, i.e. the prior probability is not constant but indeed dynamically updated as

$$P(\Theta_k^i) = P(\Theta^i|\Omega_{k-1}) \quad (11)$$

Knowing the "initial" prior probability for each class, our classification method is based on the Algorithm 1, which outputs the maximum posterior probability for each segment. To decide which is the most "likely" class $q_i$, modelled by $\Theta^i$, for the segment $S_j$ a decision rule is obtained as follows

$$S_j \in q_i \quad if \quad \log(P(\Theta^i|\Omega_k)) = \max(\log(P(\Theta^u|\Omega_k))) \quad (12)$$

where $u = 1, 2, \ldots, N_{\mathscr{C}}$, and $N_{\mathscr{C}}$ is the total number of classes.

The classification result must be stable through a sequence of time frames for assigning a class to an object.

**Algorithm1:** Bayesian classifier

**Initialization:**
$k$: time frame;
$i = 1, 2, \ldots, N_{\mathscr{C}}$: class index;
$\Theta^i$: trained model for each class;
$P(\Theta_{k=0}^i)$: initial prior probability;
**Begin:**
$N_S$ is the number of current detected segments;
$j = 1, 2, \ldots, N_S$: segment index ($S_j$);
Receive the feature-vector $\Omega_k$ of each segment;
**for** $j = 1 : N_S$,
$ML(j, 0) = 0$;
**for** $i = 1 : 3$,
$P(\Theta_k^i)_j = P(\Theta^i|\Omega_{k-1})_j$;
$ML(j, i) = \max[log(P(\Theta^i|\Omega_k)_j), ML(j, i-1)]$;
**end**
**end**
**End**
**Decision:** classify each segment $S_j$ based on rule (12).

## V. EXPERIMENTAL RESULTS

An outdoor scenario was considered in the experimental evaluation of the proposed MTDTS and classification system. The experiments have been taken place in the University campus (see Fig. 3) and the results presented in this section are focused on the classification module.

Experimental data were collected using a laser range finder working up to 8,0 m and mounted on an AGV-like vehicle approximately 64,5 cm above and parallel to the ground. The extracted feature-vectors were used in a supervised training in order to estimate the finite GMM model for each category by means of an EM algorithm. A public toolbox was used for this purpose [1]. The trained models were constrained

Fig. 3. A snapshot of the outdoor environment taken by a CCD-camera mounted above and in the front of vehicle chassis.

TABLE I
THE 3-COMPONENT FINITE GMM FOR PEDESTRIAN-CLASS

| | | |
|---|---|---|
| Feature1 | $\mu_{3\times1}$ | $[471.61 \quad 52.328 \quad 216.81]$ |
| | $\Sigma_{3\times3}$ | $\begin{bmatrix} 7722 & 513.26 & -520.79 \\ 513.26 & 119.92 & -129.55 \\ -520.79 & -129.55 & 9066.6 \end{bmatrix}$ |
| | $\alpha_1$ | 0.47174 |
| Feature2 | $\mu_{3\times1}$ | $[330.97 \quad 46.464 \quad 274.71]$ |
| | $\Sigma_{3\times3}$ | $\begin{bmatrix} 8769.2 & 837.83 & 3795.5 \\ 837.83 & 102.17 & 100.75 \\ 3795.5 & 100.75 & 35962 \end{bmatrix}$ |
| | $\alpha_2$ | 0.32473 |
| Feature3 | $\mu_{3\times1}$ | $[426.08 \quad 62.848 \quad 644.14]$ |
| | $\Sigma_{3\times3}$ | $\begin{bmatrix} 25750 & 2590.8 & -14109 \\ 2590.8 & 473.25 & -1506.8 \\ -14109 & -1506.8 & 97751 \end{bmatrix}$ |
| | $\alpha_3$ | 0.20353 |

to a maximum of 3-component GMM; as an example, the model for Class3 (persons) is presented in Table-I, where a total of 5610 samples were used in the training stage for this category. An example of segmented points which represent some objects in a outdoor semi-cluttered scenario is presented in Fig. 4. The objects of interest are labelled as car, person, tree and post.

For each segment the procedures of feature extraction, tracking and data association are performed as discussed in the previous sections. Afterwards the classification process, depicted in Algorithm 1, is called and the classification algorithm outputs the category that fits the segment more likely. Considering the objects whose segments are depicted in Fig. 4, the log-posterior probability values for each category are shown in Fig. 5, 6 and 7, where the round-markers represent Class3 (pedestrian) results, square-markers represent the Class2 (cars), and asterisk-markers represent Class1 (tree/posts) respectively. These results are an example showing the effectiveness of the proposed Bayesian classifier. However in order to improve the performance of the object

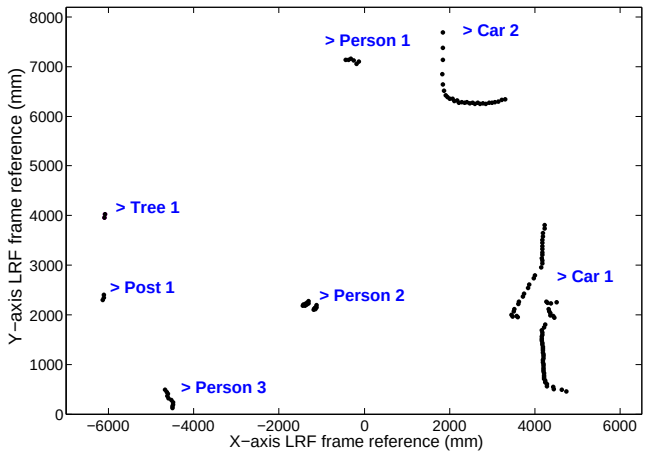


Fig. 4. Scenario with some objects of interest.

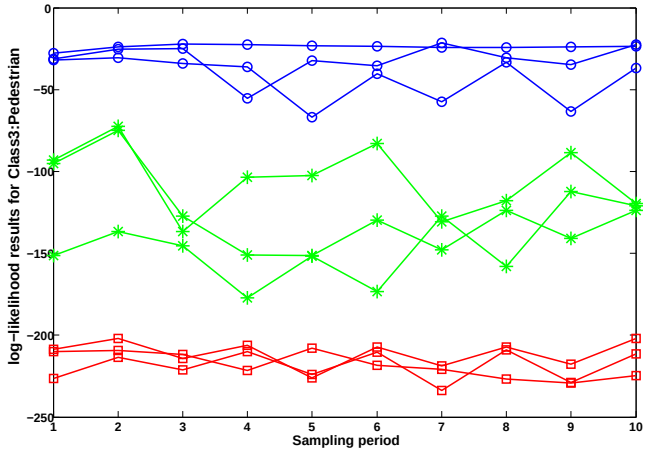


Fig. 5. Classification result for Class3 (Person:1,2 and 3) from Fig.4. (∗) Class1; (□) Class2; (O) Class3.

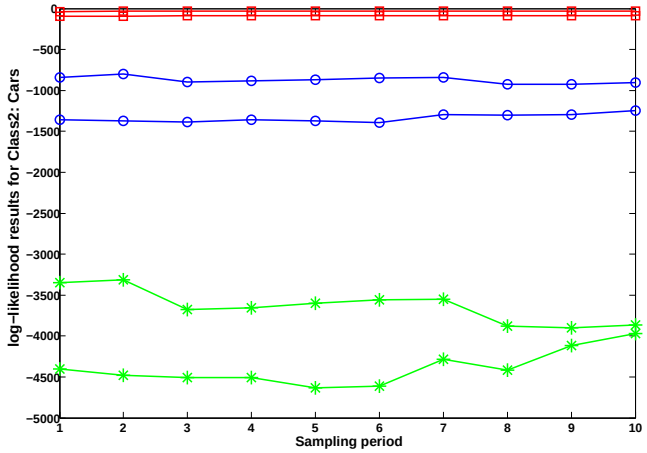


Fig. 6. Classification result for Class2 (Car1 and Car2) from Fig.4. (∗) Class1; (□) Class2; (O) Class3.

classification is indispensable that data association works properly. This is more critical when more than one segment actually constitute a single object and a merging decision has to be done.

During the experiments and under some circumstances

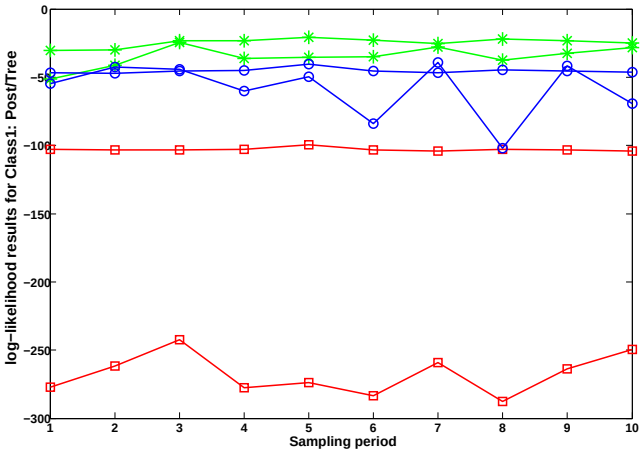


Fig. 7. Classification result for Class1 (Tree1 and Post1) from Fig.4. (∗) Class1; (□) Class2; (O) Class3.

(when all detected objects are motionless) it was evident that a more confident classification result is achieved after some sampling periods, as can be seen from Fig. 7. Therefore to overcome this problem the decision rule is based on the evaluation of the last five results from the classification module before an object under tracking gets classified. In practice, this procedure is even more necessary at the beginning of object's detection, specially because the classification depends on the object's dynamic.

## VI. CONCLUSIONS AND FUTURE WORK

A tracking and data association system working in real outdoor environments has its robustness greatly enhanced if a classification module integrates the overall object detection system. The main result in using a classifier is that thereafter an object gets classified, routines related to modeling, geometrical characterization, tracking and data association are adjusted in accordance with the object's class, turning the overall MTDTS more effective. Interesting experimental results of the proposed classifier and MTDTS, with real-world data, have been verified, encouraging further research on Bayesian GMM-based classifiers and on its applicability to new data processing tasks. Despite the satisfactory results, the modules that constitute the MTDTS are in constant improvement. So far, separated laser-based and vision-based [9], [10] object/pedestrian detection methods have been studied in our laboratory. It is clear for us that a cooperative sensor fusion approach, including various sensor modalities such as vision and lasers, will enhance the effectiveness and robustness of the whole system, being this another direction of our research work.

## REFERENCES

[1] 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/).

[2] G. A. Borges, and M. J. Aldon. "Line extraction in 2D range images for mobile robotics", *Journal of Intelligent & Robotic Systems*, v. 40, n. 3, 2004, pp. 267-297.

[3] L. C. Bento, U. Nunes, F. Moita, and A. Surrecio, "Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments", *IEEE Intelligent Transportation Systems Conference (ITSC)*, Vienna, Austria, 2005.

[4] A. Mendes and U. Nunes, "Situation-based multi-target detection and tracking with laserscanner in outdoor semi-structured environment", *in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004, Sendai, Japan.

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

[6] A. J. Lipton, H. Fugiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video", *IEEE Image Understanding Workshop*, 1998, pp. 129-136.

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

[8] 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.

[9] G. Monteiro, P. Peixoto and U. Nunes, "Vision-based pedestrian detection using Haar-like features", *in Proc. 6th National Festival of Robotics, Scientific Meeting (ROBOTICA)*, Guimaraes, Portugal, 2006.

[10] Q. Yu, H. Araujo and H. Wang, "A stereovision method for obstacle detection and tracking in non-flat urban environments", *in Autonomous Robots*, Vol. 19, No. 2, September 2005, pp. 141-157.