A solution for probabilistic inference and tracking of obstacles classification in urban traffic scenarios

Ion Giosan, Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca, Romania
{Ion.Giosan, Sergiu.Nedevschi}@cs.utcluj.ro

***Abstract*—Obstacles classification plays an important role in driving assistance systems. Any classification system should accurately distinguish, in real-time, between a set of well-known object classes such as pedestrians, cars and poles and other obstacles. If the object class is determined then the driving assistance system may take the right decision, in case of an imminent impact, in correlation to the vulnerability of the class that object belongs to. An object detection module based on both 2D and 3D information is considered for the obstacles segmentation. Preliminary classification results are obtained, at each image frame, for each detected object. The classification result should be approximately the same for an object that is tracked across frames. We described some methods for accomplishing this issue. First a Bayesian inference is considered for obtaining the class probability of the tracked objects from frame to frame. Then the tracking and filtering of the object's class is realized by applying a k-NN classification on the previously computed class values over the last few frames. These methods improve the stability and accuracy of tracked objects' classification across multiple frames.**

Keywords - obstacle classification; obstacle tracking; Bayesian inference; k-NN classification; classification tracking

I. INTRODUCTION

A very important aspect for driving assistance systems is the objects' detection, tracking and classification. The large variety of different traffic scenarios with different kinds of scene objects makes these tasks more difficult, complex and hard to obtain high accuracy results.

There are different technologies such as RADAR, LIDAR, ultra-sound sensors, piezo-electrical sensors, LASER-scanners and video cameras used for acquiring the environment information. However, the video images acquired with video cameras is the most similar acquisition manner to the human vision system and it's a passive and clean way because it does not imply any source of pollution and does not affect the environment and people. In comparison to a single video camera, a pair of two stereo-cameras provides much more information crucially needed for the algorithms used in solving the driving main tasks. The stereo sensor offers the possibility to determine the depth distance value for any image point but also the possibility to compute the motion vector for any pixel. The combined use of the 3D position and motion information significantly improves the object detection, the discrimination between static and dynamic components, the dynamic object tracking and offers then the possibility of achieving a good classification.

Determining the classes for all objects that appear in the environment or specifically in the traffic scenarios is a requirement that every safety driving assistance system must implement. There are different traffic scenarios where the driving assistance systems may be used. In the highway scenarios, the scene is relatively simple and the obstacles that appear in the traffic are limited to cars, trucks and road-side fences. Opposite to highway scenarios, in the urban traffic scenarios, the classification problem becomes much more difficult due to the environment complexity and the presence of different objects types.

We have identified four main classes used in relevant obstacles classification: cars, pedestrians, poles and other objects. Pedestrians are the most vulnerable participants involved in traffic, so their recognition and protection is a major requirement in urban traffic scenarios. Cars' recognition is also important for a further detection of the occupant cell of the collision car, in order to avoid the injury to its passengers in case of an imminent accident. Poles recognition is important in order to avoid possible collisions with them, to infer the position of the drivable lane knowing that the poles are on the road-sides and also to further detect the traffic road signs and traffic lights.

The classification system should be able to recognize as accurately as possible, in real-time, each of the four classes of objects. It should be also capable of working in different weather conditions and traffic scenarios. The classification algorithms are not yet powerful enough to determine accurately the obstacles' classes considering just the current frame. The features values used in the classifier input for the same object change from frame to frame and this leads to an inaccurate classification. There are consecutive frames in which the obstacle is correctly classified, but there may appear some intercalated frames where the object is wrongly classified.

This drawback is covered in this paper by describing a robust classification tracking technique. The preliminary results of a classifier that classifies all the objects in the current frame are merged together with the results of a tracking module that tracks the obstacles between frames. Our contribution consists in developing a probabilistic method that combines these results. In comparison with other systems, that don't use object classification tracking [3][6] , our proposed method achieves an improvement in the objects classification results.

II. Related Work

There have been a lot of research activities for developing new solutions and systems used for robust object tracking and classification in different surveillance applications or driving assistance systems. The architecture of a stereovision obstacle classification system [1] consists of the following three main modules: object detection based on 3D points grouping [2] and density maps [3] ; motion detection and object tracking [4], [5]; objects classification [6].

The identification and tracking the blobs of an object bounding box in 2D space could be considered a solution for object tracking. Its disadvantages refer to the issues of occlusion that cannot be solved in dense situations. The regions that are grouped together will form a combined blob and cause tracking errors. A Kalman filter that estimates the pedestrians' parameters in this manner is presented in [7]. Defining the 3D geometry of a moving object may solve partially the occlusion problem, but it has the drawback that it is time consuming so it can't be used on detailed geometric object models. In [8] the problem of partial occlusion is solved by considering 3D models. A specification of vehicle models makes it possible to take advantage of the a-priori knowledge about the shape of typical objects in traffic scenes [9].

Features based objects tracking is used by considering relevant features of moving objects (corners or invariant features). These features are continuously tracked for each object. This method overcomes the objects' partial occlusion problem for a small time interval because it may work correctly even with a subset of features extracted from the visible objects' parts. However, it is difficult to identify those features which belong to the same object during the tracking process. In [10] the considered features for tracking are represented by the Harris corners. Each corner position and other attributes are used in a classification procedure to determine if the tracker has worked correctly.

The objects' size and velocity are selected for computing the motion correspondence [11] and then the size and position are used with the Kalman predictors [12] for correctly estimating as much as possible the real trajectories. If the objects' types are known, e.g. the type of an object to be tracked being a pedestrian, then more appearance models of the body-silhouettes can be used [13]. Some probabilistic object appearance models have been used [14] in order to detect and track pedestrians that belong to a group and may occlude each other [15].

There are also other approaches on object tracking. In [16] a tracking method based on wavelet analysis is presented. The wavelet transform is applied for decomposing the image. A particular frequency band is selected then as the input into the neural network for vehicle recognition. All the vehicles are tracked by using their own position coordinates and the wavelet feature differences for identifying the correspondences between vehicle regions. Other methods have been developed to avoid using Kalman filtering. A new stochastic algorithm for robust tracking which is superior to previous Kalman filter based approaches is presented in [17].

Probabilistic tracking methods [18] decompose the human dynamics in order to learn and recognize human beings in video sequences. In [19] a simple tracking based on a mixture of temporal differencing and image template matching is presented. It achieves highly tracking performance in the presence of partial occlusions and achieves good classification.

A large number of algorithms for obstacle classification have been proposed in literature but the problem of achieving a good objects classification in complex traffic scenes is still far from being solved. In the classification step, the type of the previously tracked object is determined. In [20] a feed-forward neural network is used in order to distinguish between persons, vehicles, and background clutters. A Support Vector Machine [21] may also be used for classifying vehicles, humans and animals. A classifier based on error correction output is proposed in [22] and used for distinguish between bikes, cars, trucks, persons and people groups. In [23] an algorithm that does not need to be trained with test sequences of the objects is used for object classification.

Objects' classification based on pattern matching is sometimes limited to 2D image intensity information [24] or it may have additional 3D information. An approach that aimed at pedestrian detection used the dense 3D information but only as a validation method [25] due to the fact that 3D data generated by the dense stereo reconstruction devices is still noisy and has lower confidence than intensity data. In [26] techniques such as Adaboost with illumination independent features are considered for boosting the obstacle classification.

III. Classification system

The whole objects' classification system architecture with all its components and data flow is depicted in Figure 1.

Gray-levels left and right intensity images (512x383 pixels) of the scene are acquired with the stereo vision system. A hardware machine is used then for computing the stereo reconstruction of the two intensity images, resulting in a 3D range image (depth image). In the depth image each scene-point encodes the distance from the stereo cameras, so we have obtained a 3D set of points (2D intensity levels and the distance).

An object detection algorithm is applied on the 3D range image which finds the 3D bounding boxes of all objects in the scene image. All the background points are removed by using depth information; the only remaining 3D points belong to the objects.

A tracking algorithm [27] for objects, based on information supplied by dense stereo and optical flow, is used. It defines an advanced probabilistic cuboidal model for objects and use a dynamic model based on objects' classes. The algorithm is able to deal with hierarchical objects and it shows that tracking greatly improves the performance of object detection.

A large amount of features are extracted for each obstacle. The features comprise 3D attributes, motion attributes, 2D visual attributes. The most relevant attributes considered for the generic classification task are: object dimensions (width and height), aspect ratio, area to distance ratio, lateral speed, longitudinal speed, histogram of oriented gradients used for pedestrians and poles, a contour template matching score used for pedestrians, the vertical texture dissimilarity used for distinguish between pedestrians and poles.

A large database containing about 100000 objects together with their features and their class (obtained by manual label assignment) was built. In order to train the classifier, based on the dataset objects, the WEKA environment is used. The result is a random-forest classification model. This model is applied at each frame on the entire list of the previously detected objects. A classification result in an image frame is shown in Figure 2.

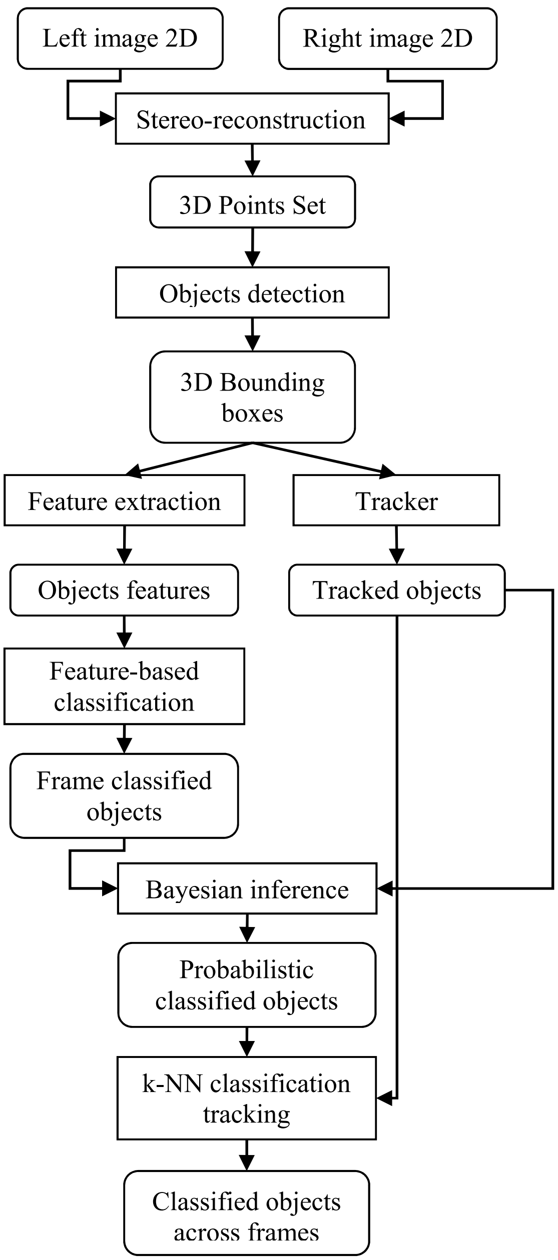


Figure 1. Obstacle classification system architecture

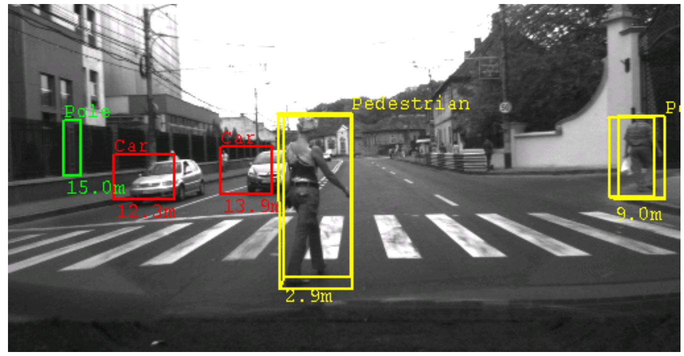


Figure 2. Example of classified objects in an image frame

The issue that occurs refers to the fact that the classification result (the assigned object's class for an object by the classifier) isn't stable across frames. This problem is related to the fact that the features values used in the classifier input for the same object change from frame to frame. There may appear consecutive frames in which the obstacle is correctly classified (having a correct and constant class assignment) but there may appear some intercalated frames where the object is wrongly classified.

In order to solve this problem, for the objects that were successfully tracked across frames, we propose two stages that refine the frame classified objects. First step consists in a Bayesian inference that is applied for computing the class probability of the tracked objects from frame to frame. Second step is the tracking and filtering of the object's class by taking a k-NN classification on the previously computed class values over the last few frames. These two steps are largely described in the next two chapters.

The objects' classification across frames achieves an improvement compared to the standard classification that doesn't take into account the previously mentioned two steps.

IV. Bayesian Inference

Bayesian probability updating is an important statistical technique which could be used in computing the class probabilities that a classifier assigns to an object in the classification process. It is especially important in the analysis of the classification results viewed as a sequence of data.

Our objects' classifier is generating independent and identically distributed classes C_i according to the class that the candidate object O belongs to. We considered the main four classes – pedestrian, car, pole, other object:

$$n = 4,\ i = \overline{1, n}$$
$$C_1 = pedestrian$$
$$C_2 = car$$
$$C_3 = pole$$
$$C_4 = other$$

The probability distribution for each of them is unknown. The conditional probabilities $P(O \mid C_i)$ are specified to define the models for each class C_i. The value of $P(C_i)$ represents the occurrence probability of class C_i. In the initial step we consider the set of initial class prior probabilities:

$$P(C_i) = \frac{1}{n} = 0.25,\ i = \overline{1,n} \tag{1}$$

In each frame, for each detected object O, the classifier assigns a probability $P(O \mid C_i)$ that the object may appear into one of the four considered classes $C_i,\ i = \overline{1,n}$. This probability represents the likelihood $L(O|C_i)$ used in the Bayesian inference:

$$P(C_i|O) = \frac{L(O|C_i)}{P(O)} \cdot P(C_i),\ i = \overline{1,n} \tag{2}$$

We considered here two approaches in computing the likelihood $L(O|C_i)$ in the current frame:

- First method considers the object's likelihood as being equal to the probability assigned by the classifier in the current frame:

$$L(O|C_i) = P(O|C_i),\ \ i = \overline{1,n} \tag{3}$$

- Second method consists in computing the likelihood as being the average of all likelihoods that the tracked object had been assigned by the classifier across all its appearances in time:

$$L(O|C_i) = \frac{1}{N}\sum_{f=1}^{N} P_f(O|C_i),\ \ i = \overline{1,n} \tag{4}$$

  where N represents the number of frames where the object appeared in the frames sequence and $P_f(O|C_i)$ is the probability assigned by the classifier in the frame *f*.

Remark that the second method is suitable only for objects that were tracked across multiple frames. In case of an object that couldn't have been tracked across multiple frames, the second method of computing the likelihood is similar to the first method:

$$N = 1,\ L(O|C_i) = P_1(O|C_i) = P(O|C_i),\ i = \overline{1,n}$$

The a-posteriori class probability $P(C_i|O)$ can now be computed using (2), having the likelihood already computed by either the first method or second method described above. The Bayesian inference probabilistic classifies the object into the class index W that has the maximum a-posteriori probability from all the a-posteriori probabilities set (5).

$$W = \arg\max_{i=\overline{1,n}} \left(P(C_i|O)\right) \tag{5}$$

The final step consists in updating the a-priori probabilities with the a-posteriori values (6) and then repeating the inference in the next frame where the object is tracked and so on, until it disappears from the scene.

$$P(C_i) = P(C_i|O),\ i = \overline{1,n} \tag{6}$$

A discussion regarding what method is best suitable for computing the object's likelihood in the current frame is presented in the experimental results chapter.

V. K-NN CLASSIFICATION TRACKING

The k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest examples in the feature space. An object is classified by a majority vote of its neighbors by assigning the class most common amongst its *k* nearest neighbors.

We used the k-NN method to filter false classifications of each tracked object along several successive frames. We suppose that the object is right classified in almost all frames where it is tracked, but there are few frames where the Bayesian inference gets a wrong result. The objective is to filter these wrong classifications and convert them to the right class. The k-NN method is suitable for accomplishing this task.

Considering the value *k* as being the number of last frames where the object appeared in its tracks, and knowing the class W_f that was assigned for the object at frame *f* (see Figure 3. !), we can vote for each class appearance in all *k* frames.

…	W_1	W_2	W_3	…	W_k
previous frames	last *k*-frames				

Figure 3. Class assigned to the object at each frame in the last *k* frames

$$V(C_i) = \sum_{f=1}^{k} W_f(C_i),\ i = \overline{1,n}$$
$$W_f(C_i) = \begin{cases} 1,\ \textit{if object has class } C_i \textit{ in frame } f \\ 0,\ \textit{otherwise} \end{cases} \tag{7}$$

The object's classification was performed by using the Bayesian inference, so the class indexes in the above formula were computed using (5).

After computing all the votes in the last *k* frames with (7), the class index that is reassigned to the object in current frame is given by that class having the maximum number of votes (8) (*k*-NN method).

$$W = \arg\max_{i=\overline{1,n}} \left(V(C_i)\right) \tag{8}$$

Remark that the filtering result depends on the number *k* of last frames that is considered in the voting process. If *k* it is chosen to be small, the classification tracking filter tends to perform a light filtering, with similar results with the process when only the Bayesian inference is used. In the opposite way, if *k* is large then the filtering process is very rough having the undesired possibility of affecting the classification in a wrong way. A discussion regarding what is best value for *k* is presented in the experimental results chapter.

VI. EXPERIMENTAL RESULTS

This chapter presents comparative classification results obtained without any probabilistic inference or classification tracking and those achieved by using both of them. There is also a comparison between computing the likelihood as being the value in the current frame (instant likelihood) (3) or as being the mean over several frames (average likelihood) (4).

The simple frame-based classifier is developed using WEKA environment, with a random forest model used in the classification process. We have built a database containing about 100000 objects together with their features and their class (pedestrians, cars, poles, others). We used about 80% of the dataset for training and the remaining 20% for testing the classifier. Our simple classification results are presented in TABLE I.

TABLE I. SIMPLE FRAME-BASED CLASSIFICATION RESULTS

Class	TP Rate	FP Rate	ROC value
Pedestrian	0.908	0.032	0.9025
Car	0.877	0.101	0.8408
Pole	0.853	0.094	0.8255
Other	0.692	0.085	0.6804

A. *Classification tracking with instant likelihood*

The classification results considering the last *k* frames for k-NN classification and instant likelihood in the Bayesian inference are depicted for each class in next four tables (TABLE II. -TABLE V.).

TABLE II. PEDESTRIANS K-NN CLASSIFICATION TRACKING WITH INSTANT LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.0874	0.0404	0.0492	0.0417	0.029	0.059	0.048
TP Rate	0.9254	0.921	0.9314	0.9314	0.942	0.9186	0.9248
ROC value	0.8850	0.9112	0.9155	0.9197	0.9351	0.8994	0.9107

TABLE III. CARS K-NN CLASSIFICATION TRACKING WITH INSTANT LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.1236	0.1316	0.1252	0.1169	0.106	0.1189	0.1115
TP Rate	0.8982	0.897	0.9037	0.9036	0.912	0.8972	0.884
ROC value	0.8398	0.8328	0.8420	0.8484	0.8622	0.8428	0.8391

TABLE IV. POLES K-NN CLASSIFICATION TRACKING WITH INSTANT LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.0992	0.1058	0.0904	0.074	0.0873	0.1088	0.0852
TP Rate	0.8618	0.872	0.8866	0.89	0.8826	0.8902	0.8606
ROC value	0.8298	0.8339	0.8549	0.8674	0.8536	0.8454	0.8366

TABLE V. OTHERS K-NN CLASSIFICATION TRACKING WITH INSTANT LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.081	0.098	0.1144	0.0843	0.075	0.1002	0.0951
TP Rate	0.6734	0.6811	0.6874	0.6936	0.7	0.6838	0.6708
ROC value	0.6635	0.6663	0.6671	0.6822	0.6907	0.6683	0.6573

There are variations in ROC values for all object classes when *k* is varied from 3 to 12 last frames. From all graphics, we can see that these variations depend on the different values of *k* (see Figure 4. !). However, it could be remarked that we have a local maximum on almost all ROC curves in the neighborhood of *k*=8. A value of *k*=8 is optimal for best tracking classification results considering instant likelihood in Bayesian inference.

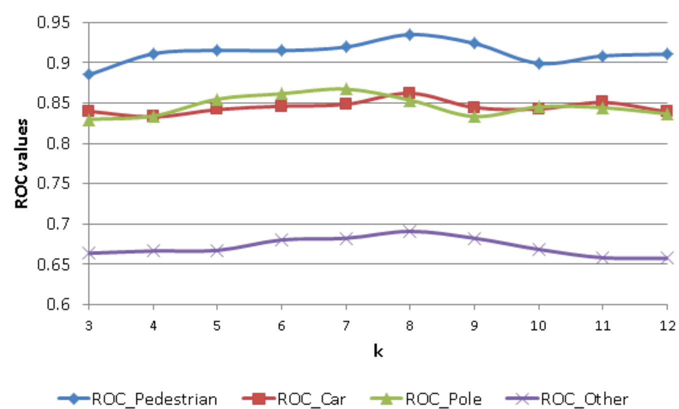


Figure 4. ROC curves for all four object classes, instant likelihood

B. *Classification tracking with average likelihood*

The classification results considering the last *k* frames for k-NN classification and average likelihood in the Bayesian inference are depicted for each class in next four tables (TABLE VI. -TABLE IX.).

TABLE VI. PEDESTRIANS K-NN CLASSIFICATION TRACKING WITH AVERAGE LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.0343	0.034	0.029	0.0398	0.0407	0.0414	0.0437
TP Rate	0.9321	0.9355	0.94	0.9298	0.923	0.918	0.918
ROC value	0.9239	0.9270	0.9333	0.9193	0.9129	0.9081	0.9070

TABLE VII. CARS K-NN CLASSIFICATION TRACKING WITH AVERAGE LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.1121	0.1156	0.108	0.1186	0.1144	0.124	0.135
TP Rate	0.8972	0.8977	0.9	0.8995	0.892	0.878	0.866
ROC value	0.8479	0.8456	0.8528	0.8445	0.8426	0.8260	0.8097

TABLE VIII. POLES K-NN CLASSIFICATION TRACKING WITH AVERAGE LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.0795	0.076	0.0778	0.0805	0.0858	0.0802	0.1201
TP Rate	0.8882	0.89	0.8865	0.885	0.8756	0.869	0.872
ROC value	0.8628	0.8662	0.8623	0.8596	0.8488	0.8463	0.8244

TABLE IX. OTHERS K-NN CLASSIFICATION TRACKING WITH AVERAGE LIKELIHOOD FOR BAYESIAN INFERENCE

k	3	4	5	7	8	10	12
FP Rate	0.0906	0.0867	0.08	0.0849	0.0942	0.0874	0.1195
TP Rate	0.686	0.6891	0.695	0.6876	0.69	0.6806	0.6712
ROC value	0.6731	0.6772	0.6846	0.6762	0.6760	0.6688	0.6501

We applied the same procedure as in the previous sub-chapter; the value *k* is varied from 3 to 12 last frames. We can see that there are relative small variations among different values of *k* (see Figure 5. !) in comparison with the previous method. However, it could be remarked that we have a local maximum on almost all ROC curves in the neighborhood of *k*=5. A value of *k*=5 is optimal for best tracking classification results considering average likelihood in Bayesian inference.

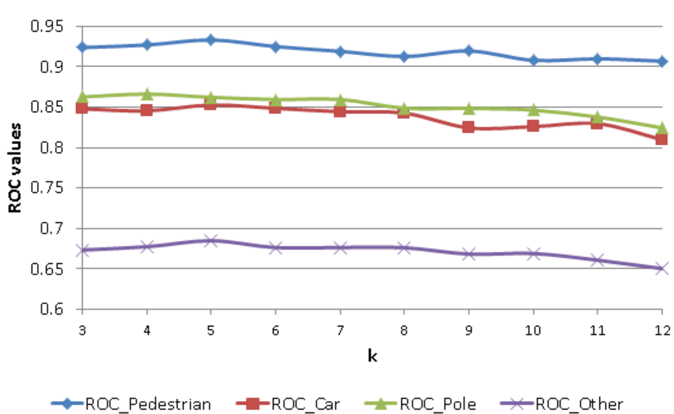


Figure 5. ROC curves for all four object classes, average likelihood

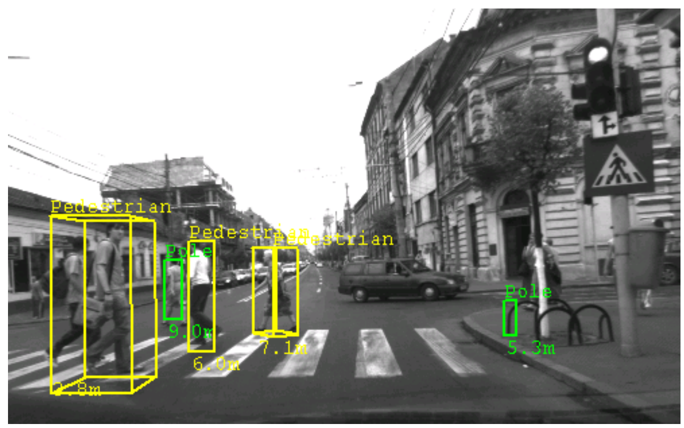
VII. CONCLUSIONS

We developed a real-time classification tracking system that may be used with success in improving obstacle classification results (pedestrian, car, pole, other) in each image frame, for each detected object. First, a probabilistic Bayesian inference is considered for obtaining the class probability of the tracked objects from frame to frame. Then the tracking and filtering of the object's class is realized by applying a k-NN classification on the previously computed class values over the last few frames.

We described two methods of computing the likelihood for the Bayesian inference. The experimental results show that both of them have improved the preliminary simple frame-based classification (in terms of TP rate, FP rate and ROC value). However, choosing the second method of computing the averaged likelihood is better than the first one because it achieves almost the same performance but considering just *k*=5 frames for inferring the result (increasing the decision speed).

Statistically, the classification of almost all obstacle classes was improved in a range of maximum 5% in TP and FP rates. The exception is the class "Other" that didn't get an improvement neither in FP rate nor TP rate. This resides from the fact that the "Other" class may contain obstacle parts from the other three classes and they are very hard to be tracked between frames.

An example of a frame obstacles classification result, before and after applying the probabilistic classification tracking technique, is shown in Figure 6. Remark that the probabilistic classification tracking method managed better to assign classes to all the obstacles from the scene.



a) without classificaton tracking (some objects are wrongly classified)

b) with classificaton tracking (the objects are correctly classified)

Figure 6. Comparison between frame obstacles classification result with and without classificaton tracking

REFERENCES

[1] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, and M. A. Obojski, "A sensor for urban driving assistance systems based on dense stereovision," in Proceedings of Intelligent Vehicles 2007, Istanbul, June 2007, pp. 276–283.

[2] C. Pocol, S. Nedevschi, "Obstacle detection for mobile robots, using dense stereo reconstruction," in IEEE International Conference on Intelligent Computer Communication and Processing, September 2007, pp. 127–132.

[3] C. Tomiuc, S. Nedevschi, "Real-time pedestrian classification exploiting 2d and 3d information," in Intelligent Transport Systems, IET, vol. 2, September 2008, pp. 201–210.

[4] R. Danescu, S. Nedevschi, M. M. Meinecke, and T. Graf, "Stereovision based vehicle tracking in urban traffic environments," in Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC 2007), 2007, pp. 400–404.

[5] S. Bota, S. Nedevschi, "Tracking Multiple Objects in Urban Traffic Environments Using Dense Stereo and Optical Flow" in Proceedings of the 14th International IEEE Conference On Intelligent Transportation Systems, Washington DC, USA (ITSC 2011), pp. 791–796.

[6] S. Bota, S. Nedevschi, "A framework for Object Detection, Tracking and Classification in Urban Traffic Scenarios Using Stereovision" in Proceedings of the 5th Conference on International Conference on Intelligent Computer Communication and Processing, 27-29 August 2009, Cluj-Napoca, Romania, pp. 153-158.

[7] O. Masoud, N. P. Papanikolopoulos, "A novel method for tracking and counting pedestrians in real-time using a single camera", in Proceedings of IEEE Transactions on Vehicular Technology, vol. 50, pp. 1267-1278, 2001.

[8] D. Koller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes" in Int. Journal Computer Vision, 1993, vol. 10, pp. 257–281.

[9] A. Cavallaro, F. Ziliani, R. Castagno, T. Ebrahimi, "Vehicle extraction based on focus of attention, multi feature segmentation and tracking", in Proceedings of European signal processing conference EUSIPCO 2000, Tampere, Finland, pp. 2161-2164.

[10] B. McCane, B. Galvin, and K. Novins, "Algorithmic fusion for more robust feature tracking" in International Journal Computer Vision, 2002, vol. 49, pp. 79–89.

[11] O. Javed, M. Shah, "Tracking and object classification for automated surveillance" in Proceedings of European Conference on Computer Vision, 2002, pp. 343–357.

[12] C. Stauffer, W. Grimson, "Learning patterns of activity using real-time tracking" in Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, pp. 747–757.

[13] D. Harwood, L. Davis, "W4: Real time surveillance of people and their activities" in Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000,

[14] A. Elgamal, R. Duraiswami, D. Harwood, L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", in Proceedings of IEEE, 2002, pp. 1151–1163.

[15] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, H. Wechsler, "Tracking groups of people" in Proceedings of Computer Vision and Image Understanding, 2000, pp. 42–56.

[16] J. B. Kim, C. W. Lee, K. M. Lee, T. S. Yun, H. H. Kim, "Wavelet-based vehicle tracking for automatic traffic surveillance" in Proceedings of IEEE International Conference TENCON'01, Singapore, 2001, vol. 1, pp. 313–316.

[17] M. Isard, A. Blake, "Contour tracking by stochastic propagation of conditional density", In Proceedings of European Conference of Computer Vision, Cambridge, pages 343–356, 1996.

[18] C. Bregler, "Learning and recognizing human dynamics in video sequences" in Proceedings of IEEE International Conference CVPR, 1997, pp. 568–574.

[19] A. J. Lipton, H. Fujiyoshi, R. S. Patil, "Moving target classification and tracking from real-time video", in Proceedings of the IEEE Workshop Application of Computer Vision, 1998, pp. 8-14.

[20] D. Toth, T. Aach, "Detection and Recognition of Moving Objects Using Statistical Motion Detection and Fourier Descriptors," in Proceedings of the 12th International Conference on Image Analysis and Processing, 2003, pp. 430–435.

[21] E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, S. Lepchev, "A Real-Time System for Classification of Moving Objects" in 16th International Conference on Pattern Recognition, 2002, vol. 3, pp. 688–691.

[22] L. Zhang, S. Z. Li, X. Yuan, and S. Xiang, "Real-time Object Classification in Video Surveillance Based on Appearance Learning," in IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[23] O. Javed, M. Shah, Computer Vision - ECCV 2002, Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2002, vol. 2353/2002, chapter Tracking And Object Classification For Automated Surveillance, pp. 439–443.

[24] D. M. Gavrila, "Pedestrian detection from a moving vehicle," in Proceedings of the European Conference on Computer Vision, 2000, pp. 37–49.

[25] D. M. Gavrila, J. Giebel, S. Munder, "Vision-based pedestrian detection: the PROTECTOR system," in Intelligent Vehicles Symposium, 2004, pp. 13–18.

[26] A. Khammari, F. Nashashibi, Y. Abramson, C. Laurgeau, "Vehicle detection combining gradient analysis and adaboost classification," in Intelligent Transportation Systems Proceedings, 2005, pp. 66–71.

[27] S. Bota, S. Nedevschi, "Vision Based Obstacle Tracking in Urban Traffic Environment", in Proceedings of IEEE Conference On Intelligent Computer Communication and Processing, Cluj-Napoca, 2011, pp. 231–238.