Urban traffic dense-stereo obstacle classification using boosting over visual codebook features

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

***Abstract*—Every driving assistance system should have an obstacle classification module. Its main role is to accurately classify obstacles within a set of predefined classes. This paper presents a real-time dense-stereo based obstacle classification system that integrates visual codebook features like HOG, LBP and texton descriptor types in a powerful classifier. The system classifies the obstacles in four main classes: cars, pedestrians, poles/trees and other obstacles. The system acquires the image scenes using a pair of gray level stereo video-cameras. A combined approach using both 2D intensity and 3D depth information is firstly used for accurate obstacle segmentation. Then, the visual codebook features are extracted for a large set of obstacles with manually labeled classes and used for training a robust boosting classifier. The comparative classification results with an approach based on a random forest classifier trained on a relevant feature set show a considerable improvement, especially for the class of other obstacles.**

Keywords—dense stereo system; visual codebook features; boosting classifier; obstacles classification

I. Introduction

The technological development from nowadays offered the constructors the possibility of increasing the number of intelligent vehicles on the market. Each such vehicle has at least a driving assistance system that alerts the driver in case of dangerous situations for avoiding an imminent accident. Some vehicles integrate more than just a driving assistance system. They have protection parts such as automatic triggered external airbags and automatic avoidance maneuver controllers for protecting the pedestrians.

The design of a driving assistance represents a challenge for research engineers. It should provide accurate obstacle detection, tracking and finally classification. In urban traffic scenarios, the frequency of scene background variation, the multitude of environmental conditions, the variety of scene objects and their distance from the acquisition video-cameras, the vibrations of the ego-vehicle and scene cluttering, alltogether make the obstacles detection, tracking and classification more complex and more difficult to achieve high accuracy results.

A variety of different technologies such as LASER-scanners, RADAR, LIDAR, ultrasound sensors, piezoelectric sensors, and video cameras are widely used for acquiring traffic scene information. However, the scene image acquisition using video cameras offers a large amount of information and it is the most similar with the human eyes-vision system, being a passive and clean acquisition modality, with no pollution and no dangerous factors for people and environment. Two stereo-cameras provide richer information than a single camera. This is recommended for all methods developed for solving driving assistance tasks.

The stereo acquisition system offers us the possibility to compute the depth map corresponding to each intensity image and the possibility to determine the motion vector for each scene point. This paradigm of using both 3D positioning (localization through intensity and depth) and motion vectors significantly improves obstacle detection.

Many driving assistance systems have a classification module for assigning a class to every obstacle that appears in the traffic scene. There are at least two different traffic scenarios in which the driving assistance systems are used. In highway environments, the traffic scene is simple and the obstacles that appear are just trucks, cars, and road-side fences. On the other hand, in urban traffic scenes, the classification process is much more difficult, due to the variety of obstacle types that increase the scene complexity.

We only considered four classes that are used for obstacle classification: cars, pedestrians, poles/trees and other obstacles. The most vulnerable traffic participants are pedestrians. In urban traffic scenarios, they should be accurately recognized and protected. Car detection is also important, for occupant cell localization, in order to avoid the passengers' injury in case of an inevitable accident. Pole detection is required in order to avoid an imminent collision using an automatic maneuver and to compute the drivable lane based on the fact that poles are situated on the road-sides. Moreover, it is useful for detecting traffic lights and road signs. The class referring to other obstacles contains every scene object that is not a pedestrian, car or pole/tree.

Unfortunately, the other obstacles class may also contain parts from pedestrians, cars or poles/trees. Using a classic schema with specific features for the first three classes and a trained classifier, it may misclassify the instances from the other obstacles' class. In this paper we present a full

This work was supported by the Romanian Ministry of National Education, UEFISCDI, project number PNII-PCCA 18/2012, SmartCoDrive

classification system that shows an improvement of the classification compared with our previous approach.

The main contributions of the paper are: the visual codebook over HOG, LBP and texton features extraction from a large set of labeled traffic scene obstacles; training a robust boosting classifier; integrating this classification schema in a dense-stereo based system used for accurate obstacle detection. The final results prove that there is a significant improvement in the overall classification accuracy of the entire system.

II. RELATED WORK

Extensive research activities has been done for developing better and better solutions for obstacle detection, tracking and classification used in driving assistance systems. There are many types of vision-based driving assistance systems. However a stereo-based video camera obstacle classification system acquires more and accurate information from traffic scenes. Its architecture [1] contains four principal modules: image acquisition (left and right cameras), obstacle detection [2] based on both 2D intensity and 3D depth points grouping [3] and density maps [4]; motion field based obstacle tracking [5]; obstacle classification [6], [7].

After the requirement of having a good stereo-cameras acquisition system for acquiring high quality images, the obstacle detection plays a very important role in further processing: tracking, accurate feature extraction and classification. The localization and tracking of the obstacle ROI in 2D space could be a solution for further obstacle tracking. However, in dense situations, it has a set of disadvantages referring to the possibility of being occluded by other obstacles. The ROIs that are grouped together form a large blob and cause obstacle tracking errors for sure.

A solution is using a Kalman filter that estimates the parameters [8]. If the 3D geometry of an obstacle is defined, then it may solve the occlusion issue, keeping the disadvantage of the execution time. This approach isn't likely to be used in driving assistance systems especially for obstacles with complex geometry. The issue of obstacle partial occlusion is mainly solved by defining the 3D models [9].

Obstacle tracking takes advantage of the invariant relevant features of moving obstacles (SIFT, SURF or other invariant features). The tracking procedure solves the obstacles' partial occlusion issue, because an obstacle may correctly be tracked even if it only has one part that is visible. However, the difficulty resides in identifying those relevant features that are used for keeping track even if an object is partially occluded. Simple Harris corners [10] are used for obstacle tracking. The tracker validation is achieved with a classifier that uses the corner position and other attributes in the feature set. If the obstacle type is known (e.g. the tracked obstacle is a pedestrian), then body-silhouette models may successfully be used [11]. Other probabilistic appearances models are used in order to track each pedestrian inside a cluttered group [12], [13], [14].

Wavelets are also used for obstacle tracking and they are presented in [15]. First, the wavelet transform is applied for image decomposition. A neural network that takes a selected particular frequency band is used for vehicle detection. Then, every detected vehicle is tracked using its own location and the wavelet differences identify the correspondences between vehicle regions. A robust stochastic algorithm used for obstacle tracking, better than Kalman filter approaches, is described in [16]. In [17], a probabilistic tracking method in presented. It decomposes the human motion with the goal of recognizing basic human attitudes. A simple tracking using a mixture of temporal derivatives and template matching is developed in [18]. The results show a high tracking performance and classification in conditions of partial occlusions.

Obstacle classification is supported in literature by a large number of algorithms and methods. However, the issue of obtaining high accuracy obstacle classification results in complex traffic scenarios is not completely solved yet. A classification module mainly has the objective of assigning a class to each previously tracked object. A neural network is used in [19] in order to classify people, vehicles, and other background clutters. In [20], SVMs are used for classifying animals, humans and vehicles. Bikes, trucks, cars and humans classification based on appearance error correction technique is presented in [21].

A pattern matching approach using 2D image intensity information is used for obstacle classification [22]. Pedestrian detection using dense 3D information as a validation method is described in [23]. The reconstructed dense stereo 3D points is used just for validation due to the fact that they are noisy than the intensity data which has higher confidence. In [24], robust illumination independent features are combined in a boosting technique for building a fast Adaboost classifier. Multiple obstacle features are usually extracted in order to train a classifier for obstacles classification in every frame from the video sequence. The classifier is individually applied in each frame on every detected obstacle.

Visual codebook based approaches became popular over the last decade. A visual codebook can be trained for any local descriptor type. It consists of a set of possible local descriptor vectors, called codewords. Any local descriptor vector can be matched to one of the codewords (the most similar one, using a specific metric). This way, an image or an image region can be described by the distribution of codewords, by creating histograms over the codewords. These histograms can be further used as features for classification purposes. The codebook based approaches obtained state of the art results on classification challenges such as the Pascal VOC Challenge [25] or the ImageNET Challenge [26].

In this paper we propose boosting over codebook features for classifying obstacles in traffic scenarios. Boosting fuses a set of weak classifiers into a single strong classifier. The weak classifiers are selected from a large set of possible simple classifiers, such as decision stumps over classification features. Boosting was successfully applied for face detection in [27]. Torralba et al. proposed the joint boosting algorithm in [28] that can be used for multiclass classification. Joint boosting is based on feature sharing between multiple binary classifiers that lead to a final multiclass classifier. It was initially used for multiclass object detection [28]. State of the art results were recently obtained for semantic segmentation of images using joint boosting. Each pixel of the image is assigned to a semantic class such as sky, grass, car, etc. Boosting was successfully applied for the multiclass classification of individual pixels in [29],[30],[31]. This is a difficult task considering the large number of training instances (a single training image contains several hundred thousands of training pixels) and a high number of semantic classes (21 in the case of MSRC dataset [29]).

III. OBSTACLE CLASSIFICATION SYSTEM

The obstacle classification system architecture with all its modules is briefly introduced in this chapter and depicted in Figure 1.

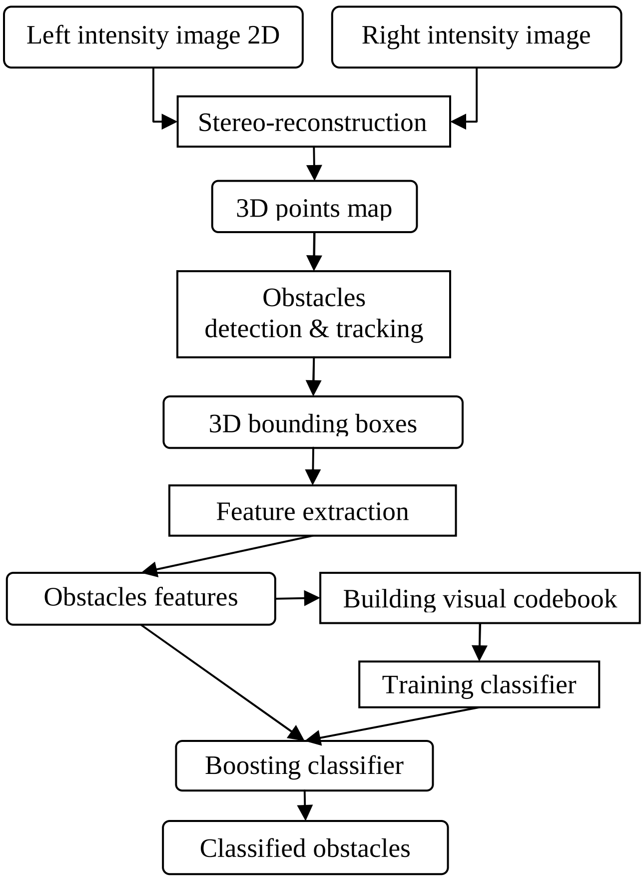

Figure 1. Obstacle classification system architecture

The stereo vision image acquisition system acquires gray level left and right intensity images with resolution of 512x383 pixels. Stereo-reconstruction is performed using a TYZX hardware machine having as input the two intensity images. The result is a depth map that completes the 2D intensity information in a 3D points map. The depth map encodes, in each location point, the distance from the stereo acquisition cameras.

An obstacle detection module takes the 3D information and computes the 3D bounding boxes of all scene obstacles. The background points are filtered considering the information from the depth map. The remaining 3D points are more likely to be obstacle points.

The obstacle tracker is based on both dense stereo and optical flow information and it is used for tracking the obstacles that appear across multiple frames. It introduces a probabilistic cuboidal model for each obstacle and uses a different dynamic model for each obstacle class. The tracker module is able to store hierarchical obstacles. It is proven that the tracking module greatly improves the accuracy and performance of the obstacle detection.

IV. OBSTACLE DESCRIPTION

In order to classify an obstacle, features that can be used for classification have to be computed. In this work we propose the use of visual codebook features. First a set of local descriptors has to be specified for which the codebooks are created. We used three different descriptors: HOG, LBP and texton.

The HOG (Histogram of Oriented Gradients) descriptor is based on the oriented gradients inside a pixel neighborhood. Initially it was used for constructing descriptor vectors for sliding windows in the context of pedestrian detection [32]. In our experiments we computed the HOG descriptors for 16x16 pixel regions in a 2x2 cell configuration (four 8x8 pixel cells) using 9 contrast insensitive orientation bins. A 36 dimensional descriptor vector is obtained.

Local Binary Pattern (LBP) is used in various computer vision applications for local description purposes. State of the art results were used in [33] and [34] based on LBP descriptors. The main idea is that the center pixel of the local region is compared to several neighbor pixels. Each comparison results in a binary number (1 if it is greater or equal, 0 otherwise). In our experiments we used a 7x7 pixel region, i.e. the center pixel is compared to 48 neighbors, resulting in a 48 dimensional descriptor vector. The gray level intensity is used for comparison.

Texton features were used in [35] and [29] for color texture description. A set of linear filters are applied over the input image. The filter set consists of Gaussian filters over color channels, LoG (laplacian of Gaussian) filters, oriented first and second derivative over gray intensity. For grayscale application we use the gray intensity for the Gaussian filters instead of the color channels. The filters are computed at multiple scales. The concatenation of the filter responses at a pixel is used as a local descriptor vector. In our experiments we used the following 10 filters:

- Gaussian filter at $\sigma = 2$ and 4
- LoG filter at $\sigma = 2$ and 4 over gray intensity
- First order derivative of Gaussian at $\sigma = 2$ and 4 and at angles 0 ° and 90 ° over gray intensity

A different visual codebook is trained for each local descriptor type using set of training images for each obstacle. A large set of local descriptors are randomly sampled (in our experiments we used 500000 samples) for K-means clustering.

We used 200 cluster centers for each descriptor type. The resulting cluster centers, which are also descriptor vectors, will represent the words of the codebook. The codebook elements identify the most specific local descriptor responses and the most frequent ones.

Using a visual codebook, a codeword map can be created for each descriptor type. Each pixel location in an image is assigned to the most similar codeword from the codebook based on the local descriptor vector. Similarity is determined using Euclidean distance over the descriptor vectors. The codeword map of the image of an obstacle is shown in Figure 2. Each codeword is represented by a different intensity level.

Figure 2. Codeword map extraction: top – obstacle intensity image; botom – texton codeword map

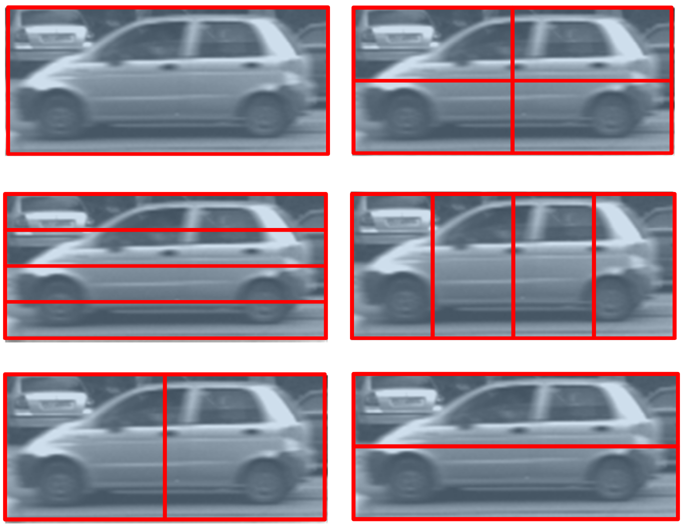

Figure 3. The 17 regions resulting from four different image partitioning

The image or any region in the image can be described by the distribution of codewords. We used 17 regions resulting from 4 different image partitionings, presented in Figure 3. For each region, each codeword is counted and normalized to the region size. A total of 17x3x200 features result that can be used for classification. Because the regions are defined relatively to the height and width of the object, the number of features does not depend on the size and aspect ratio of the object.

V. Obstacle Classification

In order to recognize the obstacles in a traffic scene, each detected obstacle is classified based on visual codebook features. In this work we show how boosting can be applied over these features.

We employ the Joint Boosting algorithm of [28] that proved to be efficient in different complex applications for multiclass classification as presented in Chapter II. A strong classifier is built iteratively using weak classifiers based on decision stump. Joint boosting exploits the fact that the same weak learner can be used for multiple classes. This way the algorithm focuses on weak classifiers that generalize well and improves generalization for the strong classifier.

The visual codebook features can be easily used for boosting. We define a weak classifier by a decision stump over the normalized codeword count in one of the 17 regions (from the 4 partitionings). The training procedure consists of multiple boosting rounds. Each boosting round searches exhaustively for the weak classifier that provides the smallest classification error rate over the training set. After finding the best performing weak classifier, the training instances are reweighted. Instances that are misclassified are penalized with a higher weight. This will affect the classification error rate computation in the next boosting round. Before the first round, each training instance is weighted equally.

Within boosting the classification of the image of an obstacle consists of $\boldsymbol{K}$ decision stumps, i.e. comparison of individual normalized codeword counts to a threshold, where $\boldsymbol{K}$ represents the number of training boosting rounds. For faster training speed and reduction of overfitting [29] recommend the use of a larger number of boosting rounds and a smaller random subset of the training features in each round. In our experiments we used 20000 rounds and 0.01 sampling rate of training features. This way, during training, each individual feature was used 200 times on average.

VI. Experimental Results

In this chapter we present our experimental results obtained with the proposed classification system based on the visual codebook. A comparison between this approach and a previous method is presented as well.

We evaluate the boosting classifier trained with the visual codebook over HOG, LBP and texton features considering 25000 instances from each class (car, pedestrian, pole/tree, other obstacle). The evaluation was done using a 10 fold cross-validation procedure. The confusion matrix and the performance metrics are presented in TABLE I and TABLE II respectively.

TABLE I. THE CONFUSION MATRIX FOR THE VISUAL CODEBOOK CLASSIFICATION SYSTEM

		Prediction			
		Car	**Pedestrian**	**Pole/ Tree**	**Other Obstacle**
Ground truth	**Car**	23640	50	90	1220
	Pedestrian	49	23911	269	771
	Pole/Tree	99	262	23159	1480
	Other Obstacle	1629	1282	1680	20409

TABLE II. THE PERFORMANCE PARAMETERS FOR THE VISUAL CODEBOOK CLASSIFICATION SYSTEM

Class	TP Rate	FP Rate
Car	0.95	0.07
Pedestrian	0.96	0.06
Pole/Tree	0.93	0.08
Other Obstacle	0.82	0.14
OVERALL	**0.915**	**0.087**

In an older approach we developed a random forest classifier based on a large feature set containing obstacle dimensions, speed, pattern matching, texture dissimilarity etc. The performance measurements of that classifier evaluated using the same procedures (25000 instances from each class, 10 fold cross-validation) are presented in TABLE III.

TABLE III. OLDER APPROACH: THE CLASSIFIER PERFORMANCE METRICS

Class	TP Rate	FP Rate
Car	0.90	0.12
Pedestrian	0.94	0.03
Pole/Tree	0.89	0.07
Other Obstacle	0.69	0.08
AVERAGE	**0.855**	**0.075**

We achieved better classification results on almost all classes. However the significant improvement is obtained on the class of other obstacles. In Figure 4 is shown an example of a comparative urban traffic obstacle classification (cars with red, pedestrians with yellow and poles/trees with green) using both the older approach vs. the new approach. The remark is that in the older approach the truck and the rightmost pole were detected by the obstacle detection module, but they were classified as being other obstacles. The new classifier correctly classifies them as a car (truck) and a pole/tree respectively.

VII. CONCLUSIONS

We developed a fast real-time dense-stereo based obstacle classification system used in urban traffic scenes. The system integrates a visual codebook over features like HOG, LBP and texton. A boosting schema is selected for building a powerful classifier. Scene obstacles are classified in four main classes: cars, pedestrians, poles/trees and other obstacles. The image acquisition system is composed of a pair of gray levels stereo video-cameras. A multi-paradigm that uses both 2D intensity and 3D depth information is used for accurate obstacles segmentation. The visual codebook is extracted from a large set manually labeled obstacles. They are further used for training a robust boosting classifier. The comparative classification results with an older approach based on a random forest classifier show a considerable overall improvement, especially for the class of other obstacles.

The classification system achieves real-time execution at about 20 fps on a PC with an Intel Core i7 processor at 3.4 GHz frequency.

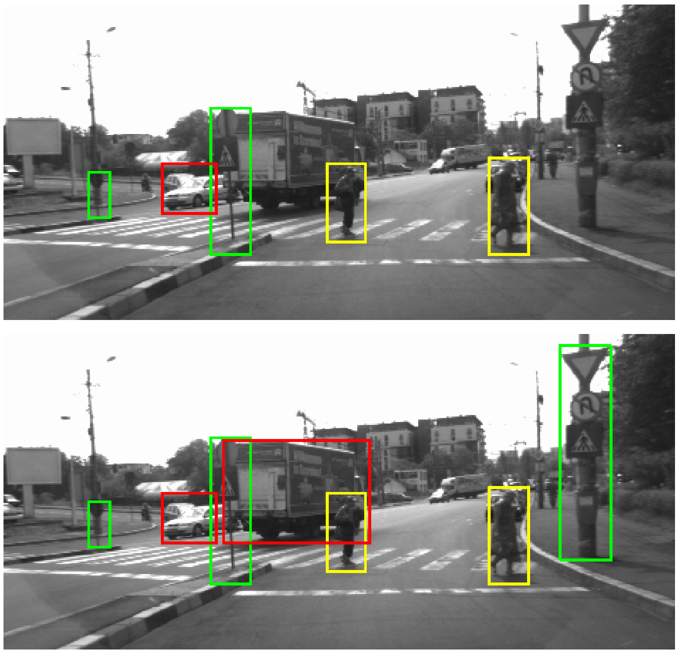

Figure 4. Obstacle classification improvement: top - old approach; bottom - new approach

REFERENCES

[1] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, *et al.*, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision," in *Intelligent Vehicles Symposium, 2007 IEEE*, 2007, pp. 276-283.

[2] D. F. Llorca, M. A. Sotelo, A. M. Hellín, A. Orellana, M. Gavilan, I. G. Daza, *et al.*, "Stereo regions-of-interest selection for pedestrian protection: A survey," *Transportation research part C: emerging technologies*, vol. 25, pp. 226-237, 2012.

[3] C. Pocol, S. Nedevschi, and M. A. Obojski, "Obstacle Detection for Mobile Robots, Using Dense Stereo Reconstruction," in *Intelligent Computer Communication and Processing, 2007 IEEE International Conference on*, 2007, pp. 127-132.

[4] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-Based Pedestrian Detection for Collision-Avoidance Applications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 380-391, 2009.
[5] R. Danescu, S. Nedevschi, M. M. Meinecke, and T. Graf, "Stereovision Based Vehicle Tracking in Urban Traffic Environments," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, 2007, pp. 400-404.
[6] S. Bota, S. Nedevschi, and M. Konig, "A framework for object detection, tracking and classification in urban traffic scenarios using stereovision," in *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, 2009, pp. 153-156.
[7] I. Giosan and S. Nedevschi, "A solution for probabilistic inference and tracking of obstacles classification in urban traffic scenarios," in *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on*, 2012, pp. 221-227.
[8] O. Masoud and N. P. Papanikolopoulos, "A novel method for tracking and counting pedestrians in real-time using a single camera," *Vehicular Technology, IEEE Transactions on*, vol. 50, pp. 1267-1278, 2001.
[9] D. Koller, K. Danilidis, and H.-H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vision*, vol. 10, pp. 257-281, 1993.
[10] B. Mccane, B. Galvin, and K. Novins, "Algorithmic Fusion for More Robust Feature Tracking," *Int. J. Comput. Vision*, vol. 49, pp. 79-89, 2002.
[11] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 809-830, 2000.
[12] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, pp. 1151-1163, 2002.
[13] H. K. Galoogahi, "Tracking Groups of People in Presence of Occlusion," in *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, 2010, pp. 438-443.
[14] B. Lau, K. O. Arras, and W. Burgard, "Tracking groups of people with a multi-model hypothesis tracker," presented at the Proceedings of the 2009 IEEE international conference on Robotics and Automation, Kobe, Japan, 2009.
[15] J. B. Kim, C. W. Lee, K. M. Lee, T. S. Yun, and H. J. Kim, "Wavelet-based vehicle tracking for automatic traffic surveillance," in *TENCON 2001. Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology*, 2001, pp. 313-316 vol.1.
[16] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," presented at the Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I, 1996.
[17] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 568-574.
[18] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, 1998, pp. 8-14.
[19] D. Toth and T. Aach, "Detection and recognition of moving objects using statistical motion detection and Fourier descriptors," in *Image Analysis and Processing, 2003.Proceedings. 12th International Conference on*, 2003, pp. 430-435.
[20] E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, and S. Lepchev, "A real-time system for classification of moving objects," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2002, pp. 688-691 vol.3.
[21] Z. Lun, S. Z. Li, Y. Xiaotong, and X. Shiming, "Real-time Object Classification in Video Surveillance Based on Appearance Learning," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1-8.
[22] D. Gavrila, "Pedestrian Detection from a Moving Vehicle," *Proceedings of the 6th European Conference on Computer Vision-Part II*, pp. 37-49, 2000.
[23] D. M. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: the PROTECTOR system," in *Intelligent Vehicles Symposium, 2004 IEEE*, 2004, pp. 13-18.
[24] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and AdaBoost classification," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, 2005, pp. 66-71.
[25] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303-338, 2010.
[26] D. Jia, D. Wei, R. Socher, L. Li-Jia, L. Kai, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248-255.
[27] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *Int. J. Comput. Vision*, vol. 57, pp. 137-154, 2004.
[28] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. II-762-II-769 Vol.2.
[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *Int. J. Comput. Vision*, vol. 81, pp. 2-23, 2009.
[30] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 739-746.
[31] P. Kohli, Ladicky, x, L., and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.
[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 886-893 vol. 1.
[33] W. Jianxin and J. M. Rehg, "CENTRIST: A Visual Descriptor for Scene Categorization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 1489-1501, 2011.
[34] S. Banerji, A. Verma, and C. Liu, "Novel color LBP descriptors for scene and image texture classification," in *15th International Conference on Image Processing, Computer Vision, and Pattern Recognition*, Las Vegas, Nevada, 2011, pp. 537-543.
[35] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *Int. J. Comput. Vision*, vol. 43, pp. 7-27, 2001.