

A Novel Approach for On-road Vehicle Detection and Tracking

Ilyas EL JAAFARI, Mohamed EL ANSARI, Lahcen KOUTTI, Ayoub ELLAHYANI, Said CHARFI
LabSIV, Department of Computer Science
Faculty of Science, Ibn Zohr University
BP 8106, 80000 Agadir, Morocco

Abstract—On the basis of a necessary development of the road safety, vision-based vehicle detection techniques have gained an important amount of attention. This work presents a novel vehicle detection and tracking approach, and structured based on a vehicle detection process starting from, images or video data acquired from sensors installed on board of the vehicle, to vehicle detection and tracking. The features of the vehicle are extracted by the proposed GIST image processing algorithm, and recognized by the state-of-art Support Vectors Machine classifier. The tracking process was performed based on edge features matching approach. The Kalman filter was used to correct the measurements. Extensive experiments were carried out on real image data validate that it is promising to employ the proposed approach for on road vehicle detection and tracking.

Keywords—Vehicle detection; Vehicle tracking; GIST; SVM; Edge features; Kalman filter

I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) play an important role in enhancing car safety and driving comfort. One of the most important difficulties that ADAS face is the understanding of the environment and guidance of the vehicles in real outdoor scenes. These systems aim to alert a driver about driving environments, possible collision with other vehicles, or take control of the vehicle to enable collision avoidance and mitigation. Vehicle accident statistics disclose that the main threats drivers are facing are from other vehicles. Consequently, robust and reliable vehicle detection system is a required critical and important task not only for ADAS, but also for other real-world applications including urban scene understanding, automated driving, self-guided vehicles, etc. The most common approach to vehicle detection is using active sensors such as lidar, millimeter wave radars, or lasers [1]. Prototype vehicles employing active sensors have shown promising results. However, active sensors have several drawbacks, such as low spatial resolution, slow scanning speed, and are rather expensive. Moreover, when there is a large number of vehicles moving simultaneously in the same direction, these sensors may interfere with each other, which can cause many problems. On the other hand, passive sensors, such as optical cameras, offer a more affordable solution and can be used to track more effectively cars entering a curve or moving from one side of the road to another. Moreover, visual information can be very important in a number of related applications,

such as lane detection, traffic sign recognition, or object identification (e.g., pedestrians, obstacles), without requiring any modifications to road infrastructures.

In general, vehicle detection using optical sensors is a very challenging task due to several factors. For example, vehicles come into the view with different speeds and may vary in shape, size, and color Fig. 1(a). Air pollution and weather conditions (e.g., rain, snow, fog, shadows, and clouds) may affect the visibility of vehicles Fig. 1 (b), In addition, its appearance depends on its pose and can be affected by obstacles, such as other vehicles, pedestrians, etc. Fig. 1 (c), Outdoor lighting conditions varying from day to night may also affect the visibility on the road scenes Fig. 1(d). Moreover, real-time constraints make this task even more challenging.

Almost every visual vehicle detection system follows two basic steps: Hypothesis Generation (HG), which hypothesizes the locations in images, where vehicles might be present. Hypothesis Verification (HV), which verifies the hypothesis.

In this paper, we present a new vehicle detection and tracking approach using vision-based techniques. For the detection, first we refer to hypothesis generation and verification using the sliding window and GIST as a features descriptor along with a well trained linear SVM. For the tracking, we refer to an edge based features matching method. The rest of the paper is organized as follows. Section 2 presents an overview of past work on vehicle detection and tracking. Section 3 details the proposed approach to vehicle detection and tracking. Experimental results are illustrated in Section 4. Section 5 concludes the paper.

II. RELATED WORK

Many different approaches to vehicle detection and tracking have been proposed and it is difficult to compare between these approaches since they are based on different data. Regarding the detection problem, different approaches have been proposed. In the older studies, e.g. [2], [3], as well as in many recent ones, e.g. [4], [5], authors refer to monocular vision or stereo-vision based techniques.

For the monocular vision based techniques, the use of some features to detect vehicles is widely employed in the literature. For example, Histogram of oriented gradient (HOG) features [6] have been used in a number of studies [7]. In [8], the symmetry of the HOG features extracted in a given image

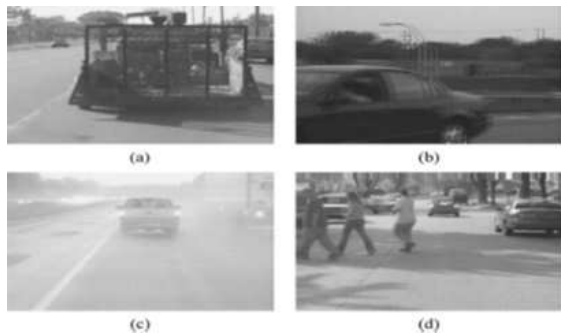


Fig. 1: Examples for difficulties facing the vehicle detection task.

patch, along with the HOG features themselves, was used for vehicle detection. Haar-like features have also been used for vehicle detection in a number of studies [9]–[14]. Haarlike features are popular for two main reasons. First, Haar-like features are well suited to the detection of horizontal, vertical, and symmetric structures. Second, by using the integral image, feature extraction is very fast. While studies that use either HOG or Haar-like features comprise a large portion of recent vehicle detection works, other general image features have been used. In [5], a combination of SURF and edge features are used to detect vehicles, with vehicle parts identified by keypoint detection. In [15], vehicles are detected as a combination of parts, using SIFT features and hidden Conditional Random Field classification. In [5], a combination of speeded-up robust features and edges is used to detect vehicles in the blind spot. In [16] Gabor and Haar features were used for vehicle detection. These features are feed to the classifiers to perform the vehicle detection process. Support vector machines (SVMs) have been widely used with HOG features for vehicle detection [7], [9]. The HOG-SVM formulation was extended to detect and calculate vehicle orientation using multiplicative kernels in [17]. In [9], [18], artificial neural networks were used to classify extracted features for vehicle detection. The combination of Haar-like feature extraction and Adaboost classification has been used to detect rear faces of vehicles in [11], [19], [20]. In [21], Waldboost was used to train the vehicle detector. Motion-based approaches, which require a sequence of images in order to recognize vehicles, are also employed to detect vehicles. In general, these methods has been less common than features-based methods. In [2], [22], adaptive background modelling was used, with vehicles detected based on motion that differentiated them from the background. Optical flow [23], a fundamental machine vision tool, has been used for monocular vehicle detection [24]. In [25], a combination of optical flow and symmetry tracking was used for vehicle detection. In [26], interest points that persisted over long periods were detected as vehicles traveling parallel to the ego vehicle. Ego-motion estimation using optical flow, and integrated detection of vehicles was implemented in [3]. In [27], optical flow was used to detect overtaking vehicles in the blind spot.

On the other hand, in the stereo-vision based techniques, multi-view geometry allows for direct measurement of 3D information, which provides for understanding of scene, motion characteristics, and physical measurements. The ability to track points in 3D, and distinguish moving from static objects, affects the direction of many stereo-vision studies. While monocular vehicle detection often relies on appearance features and machine learning, stereo vehicle detection often relies on motion features, tracking, and filtering. In [28], a histogram of depths, computed from stereo matching, was used to segment out potential vehicles. Clustering was used for object detection in [29]. In [4], clustering was implemented using a modified version of iterative closest point, using polar coordinates to segment objects. In [30], the occupancy grid is populated using motion cues, with particles representing the cells, their probabilities the occupancy, and their velocities estimated for object segmentation and detection.

In the tracking process, many approaches have been used. For example, in [31], the concept of 6D-vision, the tracking of interest points in 3D using Kalman filtering, along with ego-motion compensation, is used to identify moving and static objects in the scene. The authors in [32] have proposed a novel real time traffic supervision approach, which employs optical movement to detect and track vehicles. It uses two new techniques: color contour based matching and gradient based matching. Occupancy grids are widely used in the stereo-vision literature for tracking. In [29], scene tracking and recursive Bayesian filtering are used to populate the occupancy grid each frame, while objects are detected via clustering. In [33], tracked 3D points, using 6D vision, are grouped into an intermediate representation consisting of vertical columns of constant disparity, termed stixels. Stixels are initially formed by computing the free space in the scene, and using the fact that structures of near-constant disparity stand upon the ground plane. The use of the stixel representation considerably reduces the computation expense over tracking all the 6D vision points individually. The tracked stixels are classified as vehicles using probabilistic reasoning and fitting to a cuboid geometric model.

In this work, the vehicle detection process is performed in only one step. Sliding window is processed by a features descriptor and given to a classifier to decide whether they contain vehicles or not. Then, the regions are considered as true vehicles based on a threshold. For vehicle tracking, edge features are used. A matching edge process is used to find corresponding edges between consecutive frames. These correspondences are used to track the vehicle in the next frame.

III. PROPOSED METHOD

The aim of this work is detection and tracking of the on-road vehicles. The realization of this system was motivated by the crucial need of enhancing the people safety while driving on a road, and providing more comfort when they face other vehicles.

A flow chart of the proposed approach is shown in Fig. 2. The first step of the proposed approach is vehicle detection. Sliding window is processed by a features descriptor and given to a classifier to decide whether they contain vehicles or not. The

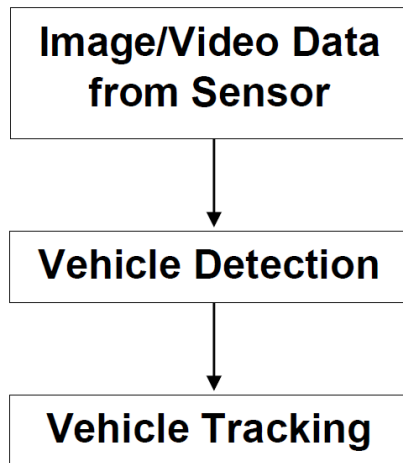


Fig. 2: Flow chart of the proposed approach.

positive windows are classified as true vehicle or false alarms based on a threshold. The second step is vehicle tracking which is based on edge features. Edge curves matching process is applied to find corresponding edges between two consecutive frames. These correspondences are used to find the tracked vehicles in the next frame. Kalman filter is used then to correct the measurements or to predict the new positions if there is no measurements.

A. Vehicle Detection

On-board vehicle detection systems have high computational requirements as they need to process the acquired images at real-time or close to real-time to save time for driver reaction. Searching the whole image to locate potential vehicle locations is prohibitive for real-time applications. The majority of methods reported in the literature follow two basic steps: 1) HG where the locations of possible vehicles in an image are hypothesized and 2) HV where tests are performed to verify the presence of vehicles in an image. Various HG approaches have been proposed in the literature [34]–[36], which can be classified into one of the following three categories:

- 1) knowledge-based: Knowledge-based methods employ a priori knowledge to hypothesize vehicle locations in an image.
- 2) stereo-based: These kind of approaches take advantage of the Inverse Perspective Mapping (IPM) to estimate the locations of vehicles and obstacles in images.
- 3) motion based : Motion-based methods detect vehicles and obstacles using optical flow.

The hypothesized locations from the HG step form the input to the HV step, where tests are performed to verify the correctness of the hypothesis.

In this work, sliding window is used to detect vehicles. The use of the concept sliding window spans a broad variety of domains, such as information technology where it has been widely used in signal processing for analysis of frequent items in packet streams [37]. In informatics, it has been applied to detect a variety of object such as faces, pedestrians, traffic

signs and vehicles etc. Sliding windows have also been applied within the field of natural language processing for collocation detection [38]. We applied the concept of the sliding window to detect relevant information from the on-road images captured by a camera mounted on a car.

The sliding_window function requires three arguments. The first is the image that we are going to loop over. The second argument is the stepSize.

The stepSize indicates how many pixels we are going to 'skip' in both the (x,y) direction. Normally, we would not want to loop over each and every pixel of the image (i.e. stepSize=1) as this would be computationally prohibitive if we were applying an image classifier at each window.

Instead, the stepSize is determined on a per-dataset basis and is tuned to give optimal performance based on your dataset of images. In practice, it's common to use a stepSize of 4 to 8 pixels. Remember, the smaller your step size is, the more windows you will need to examine. The last argument windowSize defines the width and height (in terms of pixels) of the window we are going to extract from the image. Regarding this work we have performed several experiments on the on-road image scenes using different parameters for the sliding window.

stepSize = 4,8,12 in both directions, and we keep skipping the same number of pixels in (x,y) axes for each experiments.

To detect vehicles in multi scale, the window size we have chosen varies between 20 x 20 and 100 x 100.

For more precision we only consider regions where more than 10 detections were found as vehicle. This constraint will reduce the number of false alarms.

We validate the windows based on an analysis of a set of features in the original domain. We propose a set of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) that represent the dominant spatial structure of a scene. These dimensions may be reliably estimated using spectral and coarsely localized information (GIST Descriptor) according to [39].

First introduced in [39]. GIST is a low dimensional representation of the scene, which does not require any form of segmentation according to [40].

Intuitively, GIST summarizes the gradient information (scales and orientations) for different parts of an image, which provides a rough description (the gist) of the scene.

The algorithm presents the advantage of being biologically plausible and of having low computational complexity, sharing its low-level features with a model for visual attention that may operate concurrently on a vision system.

Given an input image, a GIST descriptor is computed by:

- Convolve the image with 32 Gabor filters at 4 scales, 8 orientations, producing 32 feature maps of the same size of the input image.
- Divide each feature map into 16 regions (by a 4 x 4 grid), and then average the feature values within each region.
- Concatenate the 16 averaged values of all 32 feature

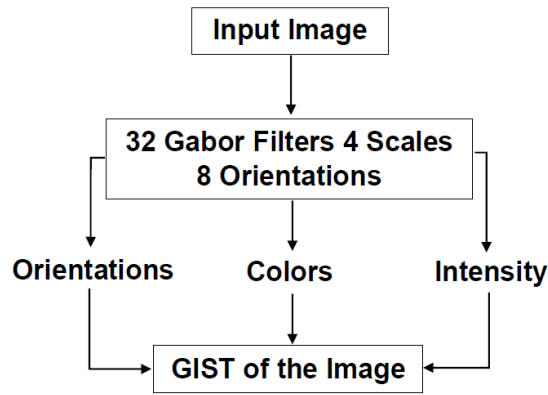


Fig. 3: Framework of GIST.

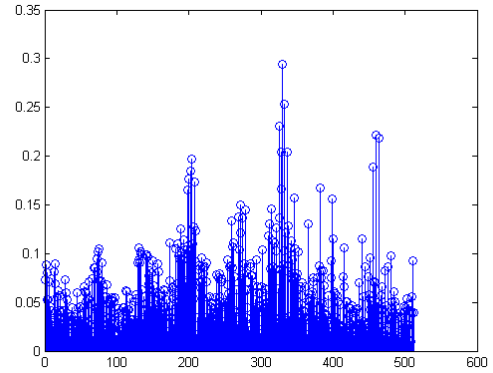


Fig. 5: The GIST histogram of the positive image shown in Fig 4.



Fig. 4: Positive image (left) with its gist descriptor (right).

maps, resulting in a $16 \times 32 = 512$ GIST descriptor.

The Fig. 3 presents a framework of the GIST descriptor and Fig. 4 shows an image from the database with its GIST descriptor.

The GIST descriptor features vector containing 512 values (see Fig. 5) is introduced to a state of the art classifier. Support Vector Machine with linear kernel in a version implemented in Matlab software, and trained by Levenberg Marquardt algorithm.

A new database created in the [41] containing 4000 positive vehicle images and 4000 negative vehicle images is used to train and test the classifiers.

The database consists of images of resolution 64×64 acquired from a vehicle-mounted forward-looking camera. Each image provides a view of the rear of a single vehicle.

A cross-validation procedure is used to test the method. Specifically, 75% of the images are randomly selected for the training set and the remaining 25% are used for the testing set. This process is repeated three times and the average is computed.

B. Vehicle Tracking

In this section, we describe the proposed method for vehicle tracking summarized in Fig. 6. The algorithm is based on edge

features. To extract edges, we had used canny edge operator [42] for the reason that it yields continuous edge curves which are vital to the proposed method and for its detection precision. The algorithm of association proposed in [43] is used to find relationships between consecutive frames. Edge curve matching process is applied based on these relationships to find correspondences between edge curves of two consecutive images. The matched edge curves in an image k of the edge curves of any detected vehicle in the image $k-1$, are used to measure the position of that vehicle in the image k . Kalman filter is used to correct the current position based on the new measurement and the past states.

1) *Edge curves matching process:* Here, we describe the algorithm that matches edge curves between two consecutive images I_{k-1} and I_k . We begin by describing the association algorithm used to find relationship between consecutive images. More details can be found in [43]. Let us consider two edge points P_{k-1}^L and Q_{k-1}^L belonging to the edge $C_{k-1}^{L,i}$ in I_{k-1}^L and their corresponding ones P_k^L and Q_k^L belonging to the edge $C_k^{L,j}$ in I_k^L (see Fig. 7). The associate point to the point Q_k^L is defined as the point belonging the edge $C_{k-1}^{L,j}$ with the same y-coordinate as of Q_k^L (e.g. as shown in Fig. 7 the associate of the point Q_k^L is the point P_{k-1}^L) [43], [44].

Based on these associate edge points computed used the association algorithm described above, we match the edge curves of each two consecutive images. Let $ASS(C_{k-1}^i)$ be the set of associate edge points of the edge curve C_{k-1}^{i-1} . We find the matched edge curve C_k^j of the edge curve C_{k-1}^i by looking for the edge curve which contains the maximum number of edge points in $ASS(C_{k-1}^i)$.

2) *Kalman filter:* In advanced driver assistance systems (ADAS) applications, the fps is high (the time interval between consecutive frames is too small), which allows to assume that the vehicle velocity is constant. In the tracking process, we use the position and the velocity to describe characteristics of

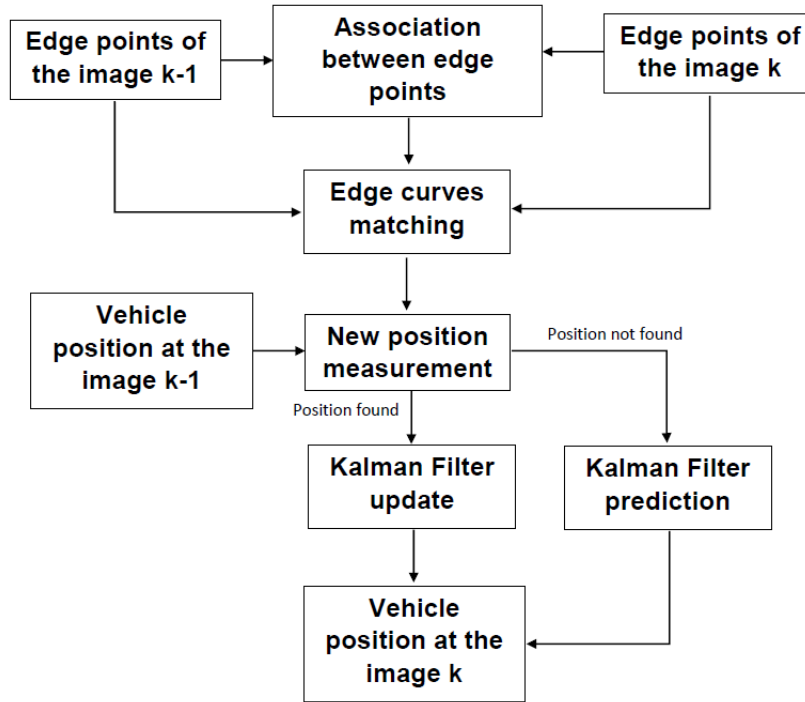


Fig. 6: Main steps of the proposed tracking algorithm.

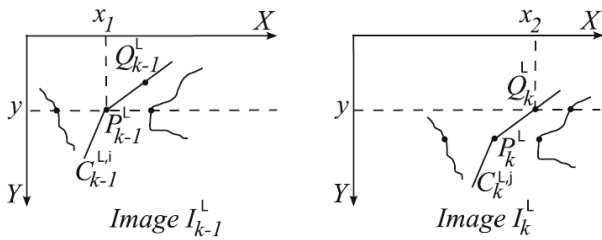


Fig. 7: I_{k-1}^L and I_k^L two consecutive left images. The point P_{k-1}^L in image I_{k-1}^L is the associate of the point Q_k^L in the image I_k^L .

vehicles motion. The system state is defined by the vector:

$$X_k = (x, y, v_x, v_y) \quad (1)$$

The description of the state and the measurement at the step k are defined by the following equations:

$$X_k = F \times X_{k-1} + W_k \quad (2)$$

$$V_k = H_k \times X_k + V_k \quad (3)$$

Which implies that the structure of F is something like:

$$F = \begin{pmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We measure only the position of the vehicle $z=(x,y)$, which implies that the structure of H is something like:

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Kalman tracking process is divided into two steps, prediction and correction. In the step of prediction, the process computes a predicted system state \hat{X} and an error covariance \hat{P} . In the step of correction, the Kalman filter computes the Kalman gain based on error covariance \hat{P} . Then it corrects the state using Kalman gain and the measurement value. In this work, we had used the kalman filter implemented in OPENCV.

3) *Position measurement and update:* The measurement of the new position of any vehicle is performed by looking for the edge curves of that vehicle that have corresponding edge curves in the next image. Let $MEC(V_i)$ be the set of the matched edge curves in the next image of the edge curves of the vehicle V_i . The center of the box containing all the matched edge curves in the set $MEC(V_i)$ is considered as the new position of the vehicle V_i in the next image. The new position is given to the Kalman filter discussed above to update the state. If an observation is unavailable for some reasons (i.e. occlusion), the update may be skipped and multiple prediction steps are performed.

TABLE I: Classification accuracy rates of GIST compared with CR-HOG using SVM in %.

	CR-HOG	GIST
Acc	94.98	96.0

IV. EXPERIMENTAL RESULTS

In this section, we present the results obtained by the new method. We note that the hardware used in our experiments is a HP Pavilion dv3 Intel(R) Core(TM) Duo CPU 2.10GH running under Windows 7.

A. Vehicle Detection

In the following we present results from the classification applied. The database consists of 4000 vehicle images and 4000 non vehicle images of resolution 64×64 acquired from a vehicle-mounted forward-looking camera. Each image provides a view of the rear of a single vehicle. And in order to confirm the performance of the proposed scheme, we compare it with the approach proposed in [41].

The original images are manually labeled to provide the ground truth. The image containing vehicle is labeled as positive sample; otherwise, it is labeled as a negative sample. In order to prevent over-fitting of the classification results, we exploited three-fold cross-validation for all the classification experiments.

An example of each type of images from the database that was used for the experiment is shown in Fig. 8. The accuracy of correctly classified rate of samples is provided in Table I. Table I shows the classification results of the SVM in the original color space. From these results we can conclude that the proposed approach outperforms the [41] scheme and shows a superior performance, with an improvement of 1.02% in term of accuracy.

The results obtained can be explained by:

CR-HOG feature extractor is based on the HOGs that evaluate local histograms of image gradient orientations in a dense grid. The underlying idea is that the local appearance and shape of the objects can often be well characterized by the distribution of the local edge directions, even if the corresponding edge positions are not accurately known. The CR-HOG is a new configuration of HOG composed of concentric rectangular cells. In addition to the distribution of the local edge directions, the GIST descriptor represents the scale information. Consequently, more variability is found in the gradient orientation map. Thus better discrimination ability in the features extractor. The results are improved to an overall accuracy of **97.5%** when using the MLP neural network as a classifier instead of the linear SVM.

This can be explained by the few advantages that MLP over other classifiers [45] such as better generalization ability, robust performance and less training data.

B. Vehicle Tracking

In order to assess its performances, the new method has been tested on an image sequence acquired by a sensor mounted



Fig. 8: Samples from the database.

aboard a moving car, the velocity of the car was 90km per hour and the sensor provides 10 images per second. The images size is 377×286 pixels.

Fig. 9 shows the results obtained by the proposed method on the frames #4132, #4136 and #4140 of the sequence described above. it shows the vehicles detected and tracked over time by the proposed method. We had successfully detected the two vehicles in the scene but we have also some false alarms. The two detected vehicles had been successfully tracked over time. The method treats 13 images per second and the sensor used in our experiments provides 10 images per second which makes the new method a real time method.

V. CONCLUSION

In this work, a novel vehicle detection and tracking method was proposed. This method is robust in the context of real traffic scenes. As regards vehicle detection, the GIST descriptor has been utilized based on the extraction of gradient features (scales and orientations) for different parts of an image. The descriptor has proven to have good discriminating properties using a reduced number of features in a simple linear kernel SVM. Thus resulting in a rough description of the scene and ideally suited for real time applications. For vehicle tracking, a new method based on edge curves matching between consecutive frames was proposed. The matched edge curves between consecutive frames are utilized to find the new vehicle's position in the next frame, then Kalman filter was applied to correct the measurements. In addition to its cost effectiveness, the over all system exhibits a good functionality on real world scenarios. For further studies, more complicated scenes and environmental factors must be considered to make the approach more robust. We will combine the GIST descriptor with an other descriptor such as HOG to improve the detection process. We will also use the feature detector SURF together with the matching process used in this paper to improve the tracking process.

REFERENCES

- [1] W. D. Jones, "Building safer cars," *Spectrum, IEEE*, vol. 39, no. 1, pp. 82–85, 2002.

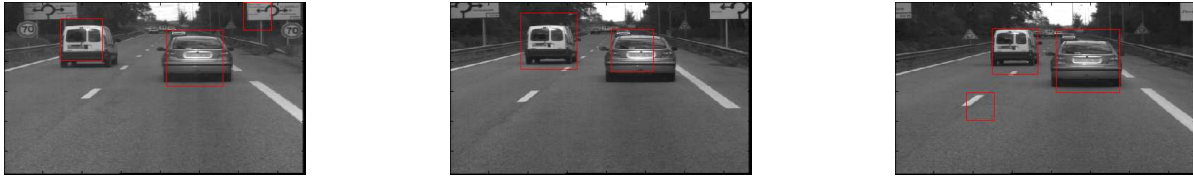


Fig. 9: The results obtained by the proposed method on the frames #4132, #4136 and #4140 of the sequence described above

- [2] J. Wang, G. Bebis, and R. Miller, "Overtaking vehicle detection using dynamic and quasi-static background modeling," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on.* IEEE, 2005, pp. 64–64.
- [3] D. Baehring, S. Simon, W. Niehsen, and C. Stiller, "Detection of close cut-in and overtaking vehicles for driver assistance based on planar parallax," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.* IEEE, 2005, pp. 290–295.
- [4] B. Barrois and C. Wöhler, "3d pose estimation of vehicles using stereo camera," in *Encyclopedia of Sustainability Science and Technology.* Springer, 2012, pp. 10 589–10 612.
- [5] B.-F. Lin, Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, L.-A. Chuang, S.-S. Huang, and M.-F. Lo, "Integrating appearance and edge features for sedan vehicle detection in the blind-spot area," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 737–747, 2012.
- [6] 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*, vol. 1. IEEE, 2005, pp. 886–893.
- [7] W. Liu, X. Wen, B. Duan, H. Yuan, and N. Wang, "Rear vehicle detection and tracking for lane change assist," in *Intelligent Vehicles Symposium, 2007 IEEE.* IEEE, 2007, pp. 252–257.
- [8] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-based vehicle detection system with consideration of the detecting location," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1243–1252, 2012.
- [9] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2019–2034, 2006.
- [10] D. Ponsa, A. López, F. Lumbreras, J. Serrat, and T. Graf, "3d vehicle sensor based on monocular vision," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE.* IEEE, 2005, pp. 1096–1101.
- [11] J. Cui, F. Liu, Z. Li, and Z. Jia, "Vehicle localisation using a single camera," in *Intelligent Vehicles Symposium (IV), 2010 IEEE.* IEEE, 2010, pp. 871–876.
- [12] A. Haselhoff, S. Schauland, and A. Kummert, "A signal theoretic approach to measure the influence of image resolution for appearance-based vehicle detection," in *Intelligent Vehicles Symposium, 2008 IEEE.* IEEE, 2008, pp. 822–827.
- [13] S. Sivaraman and M. M. Trivedi, "Active learning based robust monocular vehicle detection for on-road safety systems," in *Intelligent Vehicles Symposium, 2009 IEEE.* IEEE, 2009, pp. 399–404.
- [14] —, "A general active-learning framework for on-road vehicle recognition and tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 267–276, 2010.
- [15] X. Zhang, N. Zheng, Y. He, and F. Wang, "Vehicle detection using an extended hidden random field model," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.* IEEE, 2011, pp. 1555–1559.
- [16] Y. Zhang, S. J. Kiselewich, W. Bauson *et al.*, "Legendre and gabor moments for vehicle recognition in forward collision warning," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE.* IEEE, 2006, pp. 1185–1190.
- [17] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Learning a family of detectors via multiplicative kernels," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 3, pp. 514–530, 2011.
- [18] O. Ludwig and U. Nunes, "Improving the generalization properties of neural networks: An application to vehicle detection," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on.* IEEE, 2008, pp. 310–315.
- [19] I. Kallenbach, R. Schweiger, G. Palm, and O. Löhlein, "Multi-class object detection in vision systems using a hierarchy of cascaded classifiers," in *Intelligent Vehicles Symposium, 2006 IEEE.* IEEE, 2006, pp. 383–387.
- [20] D. Withopf and B. Jähne, "Learning algorithm for real-time vehicle tracking," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE.* IEEE, 2006, pp. 516–521.
- [21] S. Sivaraman and M. Trivedi, "Real-time vehicle detection by parts for urban driver assistance," in *IEEE Intell. Transp. Syst. Conf.* 2012.
- [22] A. Broggi, A. Cappelunga, S. Cattani, and P. Zani, "Lateral vehicles detection using monocular high resolution cameras on terramax," in *Intelligent Vehicles Symposium, 2008 IEEE.* IEEE, 2008, pp. 1143–1148.
- [23] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 81, 1981, pp. 674–679.
- [24] E. Martínez, M. Díaz, J. Melenchón, J. Montero, I. Iriondo, J. C. Socoró *et al.*, "Driving assistance system based on the detection of head-on collisions," in *Intelligent vehicles symposium, 2008 IEEE.* IEEE, 2008, pp. 913–918.
- [25] J. Arróspide, L. Salgado, M. Nieto, and F. Jaureguizar, "On-board robust vehicle detection and tracking using adaptive quality evaluation," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on.* IEEE, 2008.
- [26] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 2, pp. 583–595, 2011.
- [27] J. D. Alonso, E. R. Vidal, A. Rotter, and M. Mühlberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 5, pp. 2736–2746, 2008.
- [28] T. Kowsari, S. Beauchemin, and J. Cho, "Real-time vehicle detection and tracking using stereo vision and multi-view adaboost," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.* IEEE, 2011, pp. 1255–1260.
- [29] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," in *Intelligent Vehicles Symposium (IV), 2010 IEEE.* IEEE, 2010, pp. 313–318.
- [30] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *Intelligent*

- Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1331–1342, 2011.
- [31] C. Rabe, U. Franke, and S. Gehrig, “Fast detection of moving objects in complex scenarios,” in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 398–403.
- [32] A. A. Ambardekar, *Efficient vehicle tracking and classification for an automated traffic surveillance system*. ProQuest, 2007.
- [33] F. Erbs, A. Barth, and U. Franke, “Moving vehicle detection by optimal segmentation of the dynamic stixel world,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 951–956.
- [34] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi, “Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis,” *Image Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2364–2375, Aug 2006.
- [35] A. Mukhtar, L. Xia, and T. B. Tang, “Vehicle detection techniques for collision avoidance systems: A review,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 5, pp. 2318–2338, Oct 2015.
- [36] X. Wang, J. Tang, J. Niu, and X. Zhao, “Vision-based two-step brake detection method for vehicle collision avoidance,” *Neurocomputing*, vol. 173, Part 2, pp. 450 – 461, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231215012874>
- [37] M. Datar, A. Gionis, P. Indyk, and R. Motwani, “Maintaining stream statistics over sliding windows,” *SIAM J. Comput.*, vol. 31, no. 6, pp. 1794–1813, Jun. 2002. [Online]. Available: <http://dx.doi.org/10.1137/S0097539701398363>
- [38] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.
- [39] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, May 2001. [Online]. Available: <http://dx.doi.org/10.1023/A:1011139631724>
- [40] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, “Evaluation of gist descriptors for web-scale image search,” in *Proceedings of the ACM International Conference on Image and Video Retrieval*, ser. CIVR '09. New York, NY, USA: ACM, 2009, pp. 19:1–19:8. [Online]. Available: <http://doi.acm.org/10.1145/1646396.1646421>
- [41] J. Arrspide, L. Salgado, and M. Nieto, “Video analysis based vehicle detection and tracking using an mcmc sampling framework,” *EURASIP Journal on Advances in Signal Processing*, pp. ene–20, January 2012.
- [42] J. Canny, “A computational approach to edge detection,” *IEEE Trans Pattern Anal Mach Intell*, pp. 679–698, 1986.
- [43] M. E. Ansari, S. Mousset, and A. Bensrhair, “Temporal consistent real-time stereo for intelligent vehicles,” *Pattern Recognition Letters*, pp. 31(11):1226–1238, 2010.
- [44] M. E. Ansari, S. Mousset, A. Bensrhair, and G. Bebis, “Temporal consistent fast stereo matching for advanced driver assistance systems,” in *IEEE intelligent vehicles symposium. San Diego*, 2010, pp. 825–831.
- [45] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.