Transactions of the Institute of Measurement and Control 33, 7 (2011) pp. 783–805

On-board image-based vehicle detection and tracking

Daniel Ponsa, Joan Serrat and Antonio M. López

Computer Vision Center and Computer Science Department,
Universitat Autònoma de Barcelona, Barcelona, Spain

In this paper we present a computer vision system for daytime vehicle detection and localization, an essential step in the development of several types of advanced driver assistance systems. It has a reduced processing time and high accuracy thanks to the combination of vehicle detection with lane-markings estimation and temporal tracking of both vehicles and lane markings. Concerning vehicle detection, our main contribution is a frame scanning process that inspects images according to the geometry of image formation, and with an Adaboost-based detector that is robust to the variability in the different vehicle types (car, van, truck) and lighting conditions. In addition, we propose a new method to estimate the most likely three-dimensional locations of vehicles on the road ahead. With regards to the lane-markings estimation component, we have two main contributions. First, we employ a different image feature to the other commonly used edges: we use ridges, which are better suited to this problem. Second, we adapt RANSAC, a generic robust estimation method, to fit a parametric model of a pair of lane markings to the image features. We qualitatively assess our vehicle detection system in sequences captured on several road types and under very different lighting conditions. The processed videos are available on a web page associated with this paper. A quantitative evaluation of the system has shown quite accurate results (a low number of false positives and negatives) at a reasonable computation time.

Key words: Adaboost; advanced driver assistance systems; Kalman filter; RANSAC; ridge

1. Introduction

With the continuously growing number of motor vehicles over the last few decades, traffic accidents have become an important cause of fatalities. Approximately 1.2

Address for correspondence: Daniel Ponsa, Computer Vision Center, Edifici O, Universitat Autònoma de Barcelona, 08193 Cerdanyola, Spain. E-mail: daniel@cvc.uab.es
Figures 3, 10 and 12 appear in colour online: http://tim.sagepub.com

10.1177/0142331209103039

million people are killed and as many as 50 million people are injured worldwide every year as a result of road traffic accidents. The estimated economic loss in hospital expenses, damaged property and others is on average 1–2% of the world's gross national product (Peden *et al.*, 2004).

One of the initiatives to cope with this serious problem, launched by governments, automotive manufacturers and universities, are the so-called advanced driver assistance systems (ADASs) (Bishop, 2005; Bertozzi *et al.*, 2000). ADASs are a combination of sensors and algorithms to understand the vehicle environment and, accordingly, to assist drivers or warn them of potential hazards with the aim of preventing accidents or minimizing their consequences when unavoidable. Examples of ADASs are adaptive cruise control (ACC; the automatic adjustment of speed to maintain a safe gap from the vehicle in front), lane departure warning (LDW; to avoid leaving the lane inadvertently) or intelligent headlights control (the automatic adaptation of the front lights beam to the route type and traffic conditions).

The most frequent types of accident in highly motorized countries are by far head-on, lateral or rear-end collisions involving two or more vehicles (UNECE, 2007). ADASs directly or indirectly addressing collision warning are thus of great importance. A key component in these systems is the automatic on-board detection of surrounding vehicles, and the estimation of their three-dimensional locations relative to the system's vehicle. The perception of the environment of vehicles has been addressed mainly using active sensors, eg, radar and lidar. However, the richness of information that cameras can provide (intensity, colour, texture, etc), as well as their larger vertical and horizontal resolution, has stimulated a lot of research in developing vision-based solutions, either standalone or in combination with active sensors. Furthermore, camera systems offer some additional advantages which are of special interest for the automotive industry, such as their low cost and low power consumption. Taking these factors into consideration, our work is focused on the detection and three-dimensional localization of surrounding vehicles in daytime conditions using merely a monocular camera system.

Desirable properties of a vehicle detection system include robustness with regards to different lighting conditions and road types, reliability in the sense of a low number of false positives and negatives, and a fast computation time on a standard computer. We propose a vehicle detection system that tries to satisfy these requirements by the integration of vehicle detection, lane-markings estimation and the temporal tracking of both vehicles and lane markings. In the following section, we briefly review related work and summarize the novelties of our approach. Sections 3 and 4 describe our vehicle and lane-markings detection and tracking procedures, respectively. Quantitative and qualitative results are discussed in Section 5. Finally, Section 6 details the main conclusions of our work.

2. Related work and contributions

Our recent research (Ponsa *et al.*, 2005; Ponsa and López, 2007a,b) has been oriented towards the development of an on-board vehicle detection and localization system, relying exclusively on the sequences provided by monocular acquisition. Our goal in this paper is to improve the robustness of this system and reduce its computational cost, by integrating a lane-markings estimation module into it. Before providing the details of our proposal, we first review the state of the art in vehicle and lane-markings detection, as well as the different ways to combine them for the sake of performance improvement. Then, we present the most relevant features of our approach and the details of our principal contributions.

2.1 Vehicle detection

Detecting vehicles in images acquired from a moving platform is a challenging problem. On the one hand, vehicles form a very heterogeneous class of objects, showing a wide inter-class variability with respect to their appearance and dimensions. On the other hand, they have to be detected in an uncontrolled environment with a high variability (background, illumination), and in a reduced processing time.

Many approaches addressing this task are based on the application of heuristic criteria, which presumably check for the common denominator of vehicle appearances. Different authors have proposed the use of different low-level image features to identify vehicles owing to their shadows and lateral edges (Maurer *et al.*, 1996), symmetry (Liu *et al.*, 2005), a combination of shadows, symmetry and edges (Broggi *et al.*, 2004; Hilario *et al.*, 2005), their perceived motion (Betke *et al.*, 2000; Lefaix *et al.*, 2002), texture (Kalinke *et al.*, 1998; ten Kate *et al.*, 2004), etc. These methods are rather too naive to reach reliable detection rates, and commonly generate many false detections, as well as misdetections under challenging illumination conditions. However, they have a very low computational cost, and as a result many authors have used them to preselect image windows where it is worth applying a further more sophisticated and costly detector.

In order to obtain a more discriminant detection, many authors have proposed to follow a machine learning approach. Given a training set composed of vehicle and non-vehicle examples, the idea is to automatically build a classifier able to distinguish between these classes. An essential point in this task is the feature space used to make the classification decision. There have been many different proposals, based on texture descriptors (Kim, 2006), edge orientation histograms (Gepperth *et al.*, 2005), Gabor filters (Sun *et al.*, 2002a), Legendre moments (Zhang *et al.*, 2006), Haar filters (Sun *et al.*, 2002b), etc. Once the feature space is decided, the vehicle classifier can be trained, a task commonly performed using support vector machines (SVMs) (Kim, 2006; Sun *et al.*, 2002a,b; Zhang *et al.*, 2006) and neural networks (NNs) (Gepperth *et al.*, 2005;

Kalinke *et al.*, 1998; Zhang *et al.*, 2006). The detection of vehicles can then be performed by just scanning the image with the learnt classifier. However, since this has a significant computational cost, in practice these classifiers are used to refine detections generated by a faster (but rougher) *predetector*.

2.2 Lane-markings detection

Lane-markings extraction on road sequences is a difficult and not yet completely solved problem owing to shadows, large contrast variations, vehicles occluding the marks, worn markings, vehicle ego-motion etc. It has been the focus of many research efforts in the last 20 years, since lateral guidance has been one of the first driver assistance applications investigated. Recent reviews can be found in Jung and Kelber (2005); McCall and Trivedi (2006). Most lane detection systems are composed of the following three steps.

(1) Collect cues on where the lane markings can be, typically in the form of image points labelled as lane-markings candidates.
(2) Fit a certain lane model to them.
(3) Perform some sort of tracking in order to impose temporal continuity and speed up computations.

Concerning the first point, most approaches determine lane-markings candidates from edges detected in the image. With respect to the model of the projected lane markings, multiple geometrical models have been proposed, from simple straight lines to quadratic, spline and other polynomial curves. The parameters of these models are estimated over time commonly by using a Kalman-based filter.

2.3 Combining lane and vehicle detection

Several authors have proposed the combination of lane-markings segmentation with vehicle detection, with the aim of improving the performance of one or both components. Basically, three different approaches can be distinguished. The first approach consists of performing the two tasks independently of each other and later combining their results. This is the idea followed by Aufrère *et al.* (2001), in order to identify the closest vehicle on the same lane as that representing the greatest potential danger.

The second strategy consists of using the lane-markings information to assist the vehicle detection process. For instance, Clady *et al.* (2003) used lane markings in the middle of the vehicle detection process. First a simple heuristic is applied to determine image windows likely to contain vehicles. These windows are then filtered according to the coherence of their position and dimensions with respect to lane markings, before evaluating them with a more sophisticated detector. Alternatively, Chih-Hsien and Yung-Hsin (2006) and Sotelo *et al.* (2005) used lane markings to guide the search for vehicles, processing frames in a more efficient and coherent manner.

A third option is to make the two tasks mutually supporting: lane markings help to locate vehicles and vehicles are cues for lane markings, since they are commonly centered in them (Chan-Yu *et al.*, 2007; Dellaert *et al.*, 1998). Conceptually, this is a very appealing strategy, and as shown by Dellaert *et al.* (1998) can be formulated in a principled theoretical way, through an extended Kalman filter incorporating both vehicle and lane parameters.

2.4 Our proposal

The on-board vehicle detection and localization system developed in our recent work is based on scanning frames with a classifier, then checking whether the appearance of different rectangular regions (windows) established on the image matches that of vehicles. In order to scan frames properly, taking into consideration the different three-dimensional locations where the vehicle may be present, both the three-dimensional scene and the acquired images have to be put into relation. Since the acquisition system is monocular, establishing this relation requires some assumptions to be made. In our case, it is assumed that the observed road is planar, and that the camera pose with respect to it is known. With that, we are able to delimit roughly the image zone to be inspected, checking just windows with its base below the horizon line, with dimensions established according to their hypothetical position in the three-dimensional world. Once a vehicle is consistently detected in several adjacent frames, its three-dimensional dynamics is determined, and then is tracked in the rest of the sequence by means of an unscented Kalman filter. Unlike many approaches in the literature, our system characterizes the behaviour of vehicles in three-dimensional world coordinates, instead of just providing information in terms of their image coordinates. Figure 1 sketches the main modules of the proposed system.

Although the performance of our system is quite remarkable, it has a weak point; while the observed road can be correctly assumed to be planar in most cases, the camera pose with respect to it cannot be considered known *a priori*. Owing to the action of the vehicle suspension system, it can vary at each frame. As a consequence of that, in some frames an inaccurate camera pose is used, which means that an inappropriate inspection of frames is done, which can lead to misdetections. Furthermore, even if, in spite of using a wrong pose, vehicles are still detected, the estimation of their three-dimensional locations (and posteriorly of their dynamics) will be biased, and this can lead to the wrong identification of potential hazards.

All of these drawbacks can be avoided if we are able to identify some structure on the road that allows the camera pose to be determined with respect to it. In this paper we propose to use the road lane markings with this objective, following an approach similar to Chih-Hsien and Yung-Hsin (2006) and Sotelo *et al.* (2005). Lane-markings information allows us to delimit more precisely the zone of the image to be inspected, restricting the vehicle search inside the present and neighbouring lanes. This not only reduces the computational cost of vehicle detection, but also helps to reduce the rate of false positives generated. This comes from the fact that outside the road limits there are commonly structured and/or highly symmetric textured objects, that spuriously can be confused as vehicles by the classifiers. Since by using the lane-markings information the inspection of this region is avoided, the generation of these false positives is prevented.

The proposal we present in this paper differs from others following a similar approach in many aspects.

- Our system takes the acquisition geometry explicitly into account to process images, which leads to a more rigorous scanning of frames.
- Vehicles are detected using classifiers trained with the Real Adaboost algorithm (Schapire and Singer, 1999), achieving a high detection rate in spite of lightning conditions and vehicle appearance variability (cars, vans, trucks, etc).
- We provide information about detected vehicles in terms of their three-dimensional locations on the road ahead, instead of just their image coordinates.

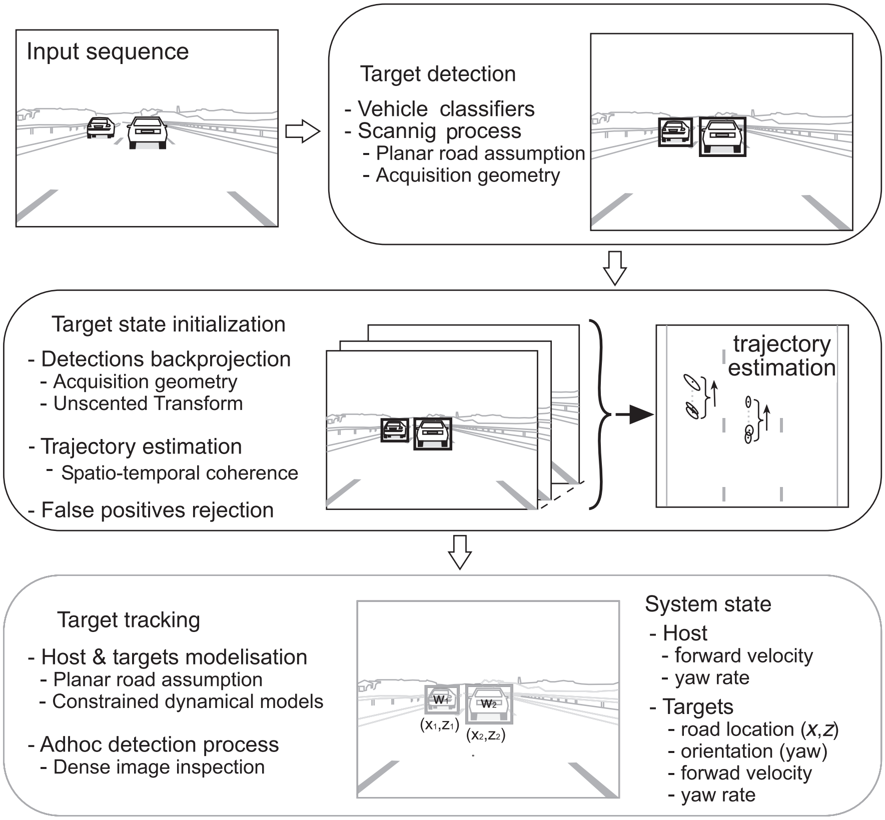


Figure 1 Main modules of our proposal to detect and track vehicles from a mobile platform

With respect to lane-markings detection, our approach differs from previous systems in:

- the use of a better suited image cue to perform this task (image ridges);
- the use of a robust fitting method (RANSAC) to adjust a three-dimensional lane-markings model to them in spite of clutter.

3. Vehicle detection and three-dimensional localization

Our proposal to detect vehicles combines the evaluation of heuristic rules with classifiers trained using Adaboost. In the following both the detection criteria, as well as the procedure of evaluating them on images, are detailed. The estimation of the three-dimensional locations of vehicles and their tracking are also described briefly.

3.1 Vehicle detection

Under normal lighting conditions, a simplistic description of a vehicle can be stated as an approximately rectangular region, with prominent edges in at least one of its sides, and a significant darker region on its bottom. This heuristic accounts for the saliency of a vehicle chassis in images, and the low intensity of tyres and the vehicle's own shadow. We propose to use this criterion, since it allows the homogeneous regions of the image, which in our application context are very significant, to be discarded with a very low computational cost (ie, most of the image displays the road pavement). Our particular implementation of this test is sketched in Figure 2, which is functional for vehicles in front of the camera, or in lateral lanes.

Windows satisfying this first criterion are then verified by a trained vehicle classifier. Differently to previous approaches, our choice has been learning this classifier using the Real Adaboost algorithm (Schapire and Singer, 1999). A very appealing

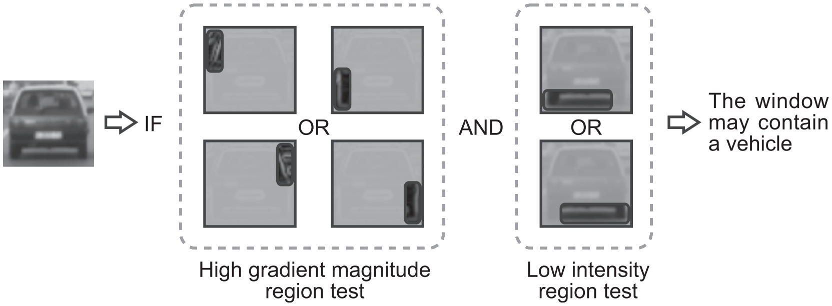

IF
OR
AND
OR
The window may contain a vehicle
High gradient magnitude region test
Low intensity region test

Figure 2 Sketch of the heuristic rule used, which results in a very significant reduction in the number of windows that need further processing

characteristic of this algorithm is that it builds a classifier based on image features identified as particularly discriminant in the training process. Given a training set $\mathbf{T} = \{(\mathbf{H}_1, l_1), \ldots, (\mathbf{H}_{n_r}, l_{n_r})\}$, where $\mathbf{H}_i = \{f_i\}_1^N$ is a set of N features describing the ith example, and l_i a boolean flag indicating whether this example is a vehicle or not, the Adaboost algorithm selects a subset of $n_f \ll N$ features $\mathbf{F} = \{f_i\}_1^{n_f} \subset \mathbf{H}_i$, each with an associated weak classifier r_i, that when combined correctly classify the training examples. The resultant classifier follows the expression

$$R(\mathbf{F}) = \sum_{i=1}^{n_f} r_i(f_i)$$

where r_i is a decision stump on f_i that returns a positive (v_i^+) or negative (v_i^-) value according to its classification decision. That is,

$$r_i(f_i) = \begin{cases} v_i^+ & \text{if } f_i \leq (\text{or } \geq) \text{ threshold} \\ v_i^- & \text{otherwise} \end{cases}$$

Given features $\mathbf{F}$ computed in an image region, $R(\mathbf{F})$ returns a value whose sign provides the final classification decision (positive for vehicles, negative for non-vehicles).

To develop our vehicle detector, we have represented training examples by an over-complete set of Haar-like features (Figure 3(a)), consisting on the responses of the filters proposed by Viola and Jones (2001) for face detection, and also their absolute values. We use this type of features because they can be computed very efficiently, at a cost that is constant whatever the size of the region where they are extracted.

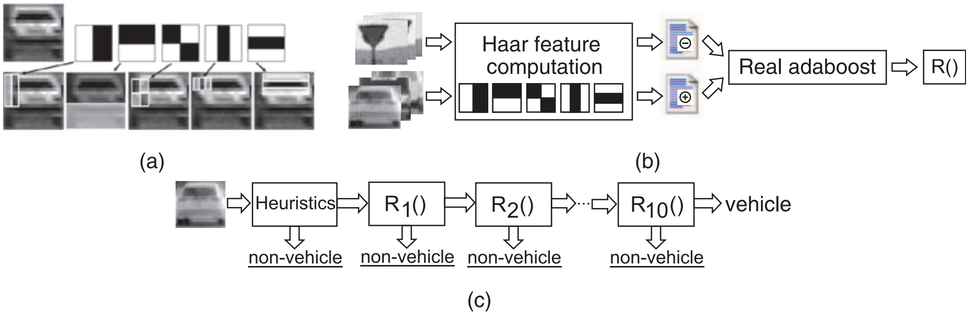


Figure 3 (a) Haar-like filters used, and regions at particular pixels where the filters give a response of high magnitude. (b) Procedure to train the classifier. (c) Window evaluation process. Once a window is labelled as non-vehicle, it is no longer processed in the rest of the cascade

For the purposes of vehicle detection, this is a very interesting characteristic, because they are imaged in a wide range of scales (in our case, from 24 × 18 up to 334 × 278 pixels), and by using Haar-like features we can avoid an explicit size normalization of the inspected windows. Moreover, these features can be computed to be invariant to monotonical changes on the intensity of regions, thus remaining useful even in low-contrast lighting conditions. By using these features, each example is described by a vector of 176 214 Haar filter responses, from where the Real Adaboost will select an adequate discriminant subset.

Instead of learning a single classifier (Figure 3(b)), we train a set of classifiers, each considering a set of counterexamples with increasing difficulty, in terms of the amount of features required to be distinguished from vehicles. Following this strategy, we have trained a cascade of classifiers (COC) of 10 layers. Given an image window, these classifiers are evaluated in sequence, in order to take a classification decision. If the window is classified as non-vehicle in a given COC layer, it is no longer evaluated in the rest of layers, and a negative classification decision is taken (Figure 3(c)). This way to proceed is interesting, since it reduces the computational cost of rejecting non-vehicle windows, which are the majority in a given frame. Overall, 1830 positive examples and 11 820 negative examples have been processed to learn the COC, extracted from sequences acquired in high-speed roads, under sunny and cloudy midday and evening conditions. A more detailed account of the COC learning procedure, as well as a quantitative study of the performance achieved can be found in Ponsa and López (2007a).

3.2 Frame scanning process

Now we have described how we check the presence of a vehicle in a given image window, we have to specify how we determine the image windows inspected at each frame. To do that in a principled way, we model the acquisition geometry of our system. It consists of a monochrome camera, mounted inside the cabin near the rear-view mirror, facing the road ahead with a slight inclination. The camera has a 1/3 inch CMOS sensor and a fixed premounted optics with focal length around 7 mm. Images are digitized at a resolution of 640 × 480 pixels.

We define a world coordinate system (WCS) relative to the camera position, placed externally just below the camera, on the road that sustains the host vehicle. This coordinate system gives the reference to relate the camera with the three-dimensional world. Since the procedure followed to mount the camera in the vehicle sets the camera X^{Cam} axis near parallel to the world X^{W} axis, the camera roll angle is practically zero. The camera yaw angle is zero by the WCS definition, as the axes Z^{W} and Z^{Cam} lie on the same plane (Figure 4). Owing to this, the camera extrinsic parameters are reduced to just the height H of the camera over the ground (ie, the camera origin is by definition $\mathbf{o}^{\mathrm{Cam}} = [0, H, 0]^{\mathrm{T}}$, with $H = 1.6$ in our experiments) and the inclination of the camera with respect to the road surface, ie, the camera pitch angle φ. The camera intrinsic parameters are modelled assuming a pinhole camera with zero skew (Hartley and Zisserman, 2004). This requires estimating the effective camera focal length in the image X and Y directions, namely f^u and f^v, and its center of projection (u^0, v^0). These parameters have been obtained by calibrating the camera with the software provided by Bouguet (2004).

Using that convention, points on the road surface (ie, $(x, 0, z)$, since the road is assumed flat) project onto the image coordinates (u, v) given by

$$u = u^0 + \frac{f^u x}{z \cos\varphi - H \sin\varphi} \tag{1}$$

$$v = v^0 + \frac{f^v (H \cos\varphi + z \sin\varphi)}{z \cos\varphi - H \sin\varphi} \tag{2}$$

We use these mapping expressions between the road and the image plane to define the frame scanning process of our vehicle detection. We define a hypothetical regular grid on the road surface, taking into account road positions up to 70 m away. The grid points determine a discrete set of road positions where we check the hypothesis that a vehicle is present (Figure 4). For each grid point, we determine rectangular windows in the image plane where vehicles of different sizes would project. Each of these windows is then evaluated with the heuristic and the vehicle classifiers, obtaining finally as result a set of windows that presumably contains vehicles.

Since scanned windows overlap and our vehicle classifiers are robust to moderate misalignments between inspection windows and vehicles, multiple detections of a same vehicle can be obtained in neighbouring windows. To provide a single detection per vehicle we cluster detections according to their size and degree of overlap, and describe then each cluster by its mean and covariance. Hence, this process results in a list of Gaussian distributions detailing the image regions where vehicles have been detected.

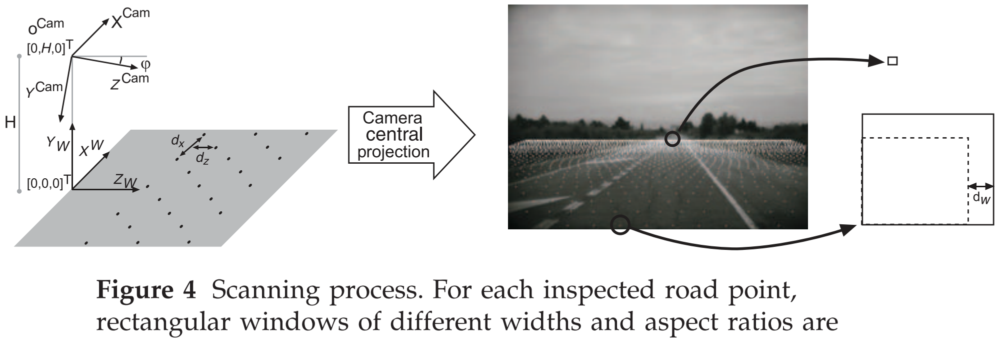


Figure 4 Scanning process. For each inspected road point, rectangular windows of different widths and aspect ratios are evaluated in the image

3.3 Vehicle three-dimensional localization and tracking

For ADAS purposes, detections obtained have to be expressed in terms of the locations of vehicles with respect to the host. Since the mapping between the road and the image plane is assumed an homography, the inverse relation of Equations (1) and (2) can be easily computed, which corresponds to

$$x = \frac{f^{v} H \left(u^{0} - u\right)}{f^{u}\left((v^{0} - v)\cos\varphi + f^{v}\sin\varphi\right)} \tag{3}$$

$$z = -\left(\frac{H\left(f^{v}\cos\varphi - \left(u^{0} - u\right)\sin\varphi\right)}{(u^{0} - u)\cos\varphi + f^{v}\sin\varphi}\right) \tag{4}$$

With these expressions, and using the unscented transform, we can convert the Gaussian distributions of detection windows onto Gaussian distributions of the road coordinates where vehicles may lay (Figure 5). In that way, we locate vehicles taking into account the uncertainty in the image regions where they are detected.

As important as the three-dimensional locations of vehicles is knowing their dynamical behaviour. To characterize it, we propose to connect the road zones from detections in successive frames, constructing a trajectory that encodes this information. This is achieved through a multiple hypothesis tracking algorithm based on the Kalman filter, which is described in detail by Ponsa and López (2007b). This trajectory construction process is also useful in discarding false positives in our detection process since they do not present spatio-temporal consistence and are unable to generate coherent trajectories. When a trajectory is consistently updated in a given number of frames, we compute an initial three-dimensional state of the vehicle, which is given to a vehicle tracking module that updates it in the rest of the sequence.

Our tracking module is based on the proposal of Dellaert and Thorpe (1997), that we have extended to take into account multiple targets. The system state maintains the

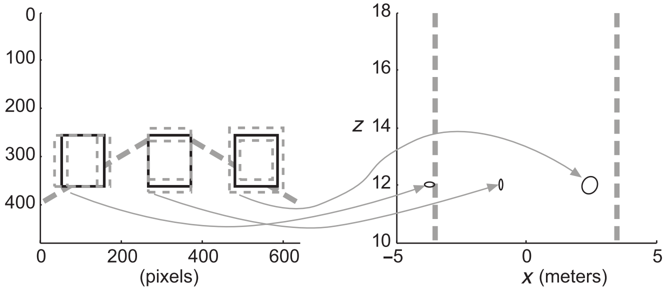


Figure 5 Backprojection of detections into their corresponding road locations. The uncertainty of detections is explicitly taken into account

target locations and dynamics, and also the dynamics of the host vehicle. This is important because it allows a more accurate system modelization, which results in a better description of the changes observed in images in successive frames. Thanks to this, the windows in the image where tracked vehicles should be observed are established more confidently, and we can scan them densely in order to detect vehicles with more accuracy than using the frame scanning process described in Section 3.2. For the sake of clarity and brevity, we omit a detailed description of the multiple target scheme used, as it may be found in Ponsa *et al.* (2005).

The performance of the overall system is obviously dependent on the quality of the vehicle detection functions proposed, but also on our assumptions of how the observed scene is reflected on images. Assuming that the road is planar is not a big problem, since this is correctly fulfilled in most situations. However, to relate it with the image plane we need to know the camera pose (ie, the pitch angle φ) with respect to it. Since this pose varies owing to the effect of the vehicle suspension system, the use of a fixed value of φ would lead to suboptimal performance. To avoid that, we propose to estimate the camera pose at each frame from the detection of the road lane markings. In addition, with the lane-markings localization we will be able to constrain more tightly the zone in the image where vehicles may be found.

4. Lane detection and tracking

In this section we provide details of our lane-markings estimation approach. With respect to previous systems, our method increases the reliability of the lane-markings extraction in two ways. First, we employ a different image feature other than the commonly used edges: ridges, which we claim are better suited to this problem. Second, we have adapted RANSAC, a generic robust estimation method, to fit a parametric model to the candidate lane-marking points. Our model consists of a pair of hyperbolas sharing a common horizontal asymptote, which are constrained to be parallel on the road plane. The use of a better suited feature (ridges) combined with a robust fitting method contributes to improve the reliability of lane-markings detection.

4.1 Lane-markings features

Ideally, lane markings are white lines on a dark road surface. Thus, the first step is usually based on image edges, defined as extrema of the gradient magnitude along the gradient direction. However, the gradient magnitude can be misleading: cast shadows and vehicles may give rise to high gradient values, while worn markings and poor lighting conditions (eg, in tunnels) reduce the contrast of lane markings. We employ a different low-level image feature, namely ridges. This term comes from considering an image as a landscape, with the intensity considered as the height. Accordingly, a ridge is the medial axis of a thick, brighter elongated structure. As such, it is better than an edge at describing what a lane marking is (Figure 6).

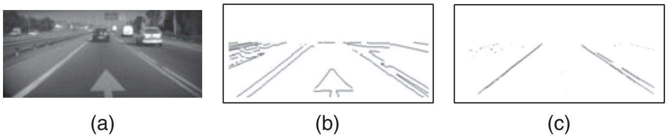


Figure 6 (a) Original frame. (b) Edges computed as zero crossings of the Laplacian of Gaussian $L * \nabla 2G_{\sigma_u,\sigma v}$ for $\sigma_u = 0.5$, $\sigma_v = 3.0$. (c) Ridgeness $\tilde{\kappa}(\mathbf{x})$

Analogously to edges, the detection of ridges in an image can be performed through the computation of a measure, ridgeness, which quantifies how much a pixel neighbourhood resembles such geometric structure.

There are different mathematical characterizations of ridgeness. López *et al.* (2000) proposed a new characterization which compares favourably to others and that we have adapted for the problem at hand. The ridgeness for a grey-level image $L(\mathbf{x})$ with $\mathbf{x} = (u, v)$ can be simply formulated as

$$\tilde{\kappa}(\mathbf{x}) = -\operatorname{div}(\tilde{\mathbf{w}}(\mathbf{x})) \tag{5}$$

where $\operatorname{div}(\mathbf{f}(\mathbf{x}))$ denotes the divergence of a vector field $\mathbf{f}(\mathbf{x}) = (f_u(\mathbf{x}), f_v(\mathbf{x}))$; that is, the sum of the partial derivatives $\partial_u f_u(\mathbf{x}) + \partial_v f_v(\mathbf{x})$ in a certain discrete neighbourhood of $\mathbf{x}$ such as the four nearest neighbours. The vector field $\tilde{\mathbf{w}}(\mathbf{x})$ is computed as follows. First, the gradient of $L(\mathbf{x})$, $\mathbf{w}(\mathbf{x}) = (\partial_u L(\mathbf{x}), \partial_v L(\mathbf{x}))^\top$ is calculated from the finite centred vertical and horizontal differences of $G_{\sigma_\mathrm{d}}(\mathbf{x}) * L(\mathbf{x})$, the original image convolved with a symmetric Gaussian kernel of standard deviation σ_d. Next, the so-called structure tensor of $\mathbf{w}(\mathbf{x})$ is computed, that is, $\mathbf{S}(\mathbf{x}) = G_{\sigma_i}(\mathbf{x}) * (\mathbf{w}(\mathbf{x}) \cdot \mathbf{w}^\top(\mathbf{x}))$, where $*$ stands for the element-wise convolution, now with another Gaussian kernel of standard deviation σ_i. At each position $\mathbf{x}$, $\mathbf{S}(\mathbf{x})$ is a 2×2 semi-positive defined matrix. The normalized eigenvector $\mathbf{w}'(\mathbf{x})$ of the largest eigenvalue of $\mathbf{S}(\mathbf{x})$ is calculated, which represents the local dominant gradient orientation. Finally, we are able to obtain from it the vector field $\tilde{\mathbf{w}}(\mathbf{x}) = sign(\mathbf{w}'(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x}))\mathbf{w}'(\mathbf{x})$.

Positive values of $\tilde{\kappa}(\mathbf{x})$ measure the similarity of a neighbourhood to a ridge structure. In fact, it has been shown (López *et al.*, 2000) that these values lie in the range [0, 2]. The higher the value in a point, the more similar its neighbourhood to a ridge, with points attaining a value of 1.0 meaning an already considerable similarity, and points reaching the maximum of 2.0 being peaks (local maxima). In addition, these values are homogeneously distributed along the centre lines, thus facilitating thresholding. In order to detect lane markings, we only take into account those pixels for which $\tilde{\kappa}(\mathbf{x}) > 0.25$, a value fixed experimentally but with a large margin before the selected pixels change significantly.

Owing to perspective, the width of the imaged lane markings decreases with distance. In order not to miss them when computing $\mathbf{w}(\mathbf{x})$, we want the upper rows of the image to be less smoothed than lower rows, but just along the horizontal direction. This is achieved by an anisotropic Gaussian smoothing, with covariance matrix

$\Sigma = \mathrm{diag}(\sigma_{\mathrm{d}u}, \sigma_{\mathrm{d}v})$ where $\sigma_{\mathrm{d}v}$ is constant and $\sigma_{\mathrm{d}u}$ increases with the row number. Specifically, we subsample acquired images to 320×240 pixels resolution, and set $\sigma_i = \sigma_{\mathrm{d}v} = 0.5$ and $\sigma_{\mathrm{d}u}$ with a value that increases linearly with the row number from 0.5 to 6.0. Examples of the proposed lane-markings detector under challenging conditions are shown in Figure 7.

4.2 Lane-markings modelling, fitting and tracking

Several geometrical models for the projected lane markings have been proposed, and examples can be found in the references. However, few are built on a sound geometrical base like that in Guiducci (1999). There it was shown that, under the assumptions of a flat road and constant curvature, the two lines delimiting a lane are projected onto the image plane as a pair of hyperbolae with one common asymptote. Specifically, a pair of equations is provided relating image coordinates of the left and right lines (u_l, v_l), (u_r, v_r) with several road geometric parameters and the effective camera focal lengths (f^u, f^v):

$$u_l = f^u\left(\frac{\theta}{\cos\varphi} + \frac{\cos\varphi}{Hf^v} x_{\mathrm{c}}(v_l + f^v \tan\varphi) + \frac{f^v H C_0 / \cos^3\varphi}{4(v_l + f^v \tan\varphi)}\right) \tag{6}$$

$$u_r = f^u\left(\frac{\theta}{\cos\varphi} + \frac{\cos\varphi}{Hf^v} (x_{\mathrm{c}} - L)(v_r + f^v \tan\varphi) + \frac{f^v H C_0 / \cos^3\varphi}{4(v_r + f^v \tan\varphi)}\right) \tag{7}$$

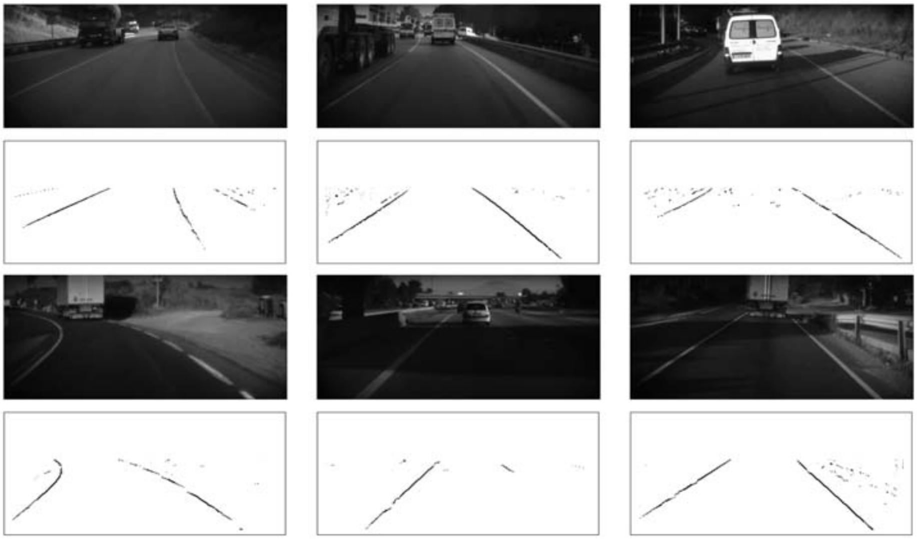


Figure 7 Detection under challenging conditions. From left to right, top to bottom, original and ridge images corresponding to: worn off paint, tire marks, white vehicle, high curvature, inside a tunnel, cast shadows

Equations (6) and (7) follow the expression of a hyperbola with horizontal asymptote at $v = v_a$:

$$u - u_a = a(v - v_a) + \frac{b}{(v - v_a)} \tag{8}$$

Figure 8 illustrates their parameters. WCS and camera coordinate systems share a common origin but have different orientation. For the WCS the Z-axis is parallel to the road tangent, the Y-axis points downward and is orthogonal to the road plane, whereas the X-axis is parallel to the road plane and orthogonal to the road tangent and therefore to the lane markings. The camera coordinate system has its Y-axis coincident with the vehicle's direction and sustains an angle $\theta \ll 1$ rad with the road tangent line, also referred as the yaw angle. It also forms an angle φ with the road plane, the pitch angle. The lane has width L and the camera is located at a horizontal distance of x_{c} metres from the left lane border and at height H above the ground. Of course, L, x_{c}, θ and φ may vary over time, but we assume that the camera height H over the ground is constant (in our experiments, $H = 1.6$). Here C_0 denotes the lane curvature.

Given the value of the camera pitch φ, Equations (6) and (7) are linear with respect to the four unknowns θ, x_{c}, L and C_0. They can be compactly rewritten as

$$\begin{bmatrix} 1 & 0 & v_l' & 1/v_l' \\ 1 & -v_r' & v_r' & 1/v_r' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} u_l \\ u_r \end{bmatrix} \tag{9}$$

with $v_r{}' = v_r/f^v + \tan\varphi$, $v_l{}' = v_l/f^v + \tan\varphi$ and

$$\theta = \frac{\cos\varphi}{f^u} a_1, \quad L = \frac{H}{f^u \cos\varphi} a_2, \quad x_{\mathrm{c}} = \frac{H}{f^u \cos\varphi} a_3, \quad C_0 = \frac{4\cos^3\varphi}{f^u H} a_4 \tag{10}$$

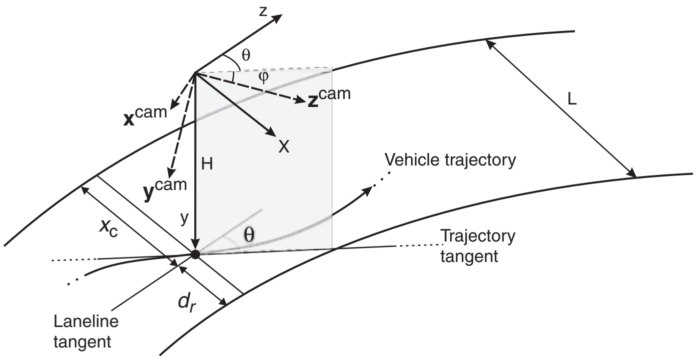

z
θ
φ
$\mathbf{z}^{\mathrm{cam}}$
$\mathbf{x}^{\mathrm{cam}}$
X
L
H
Vehicle trajectory
$\mathbf{y}^{\mathrm{cam}}$
y
x_c
θ
Trajectory tangent
d_r
Laneline tangent

Figure 8 Image acquisition geometry

However, at each frame φ is unknown, except for a range of feasible values that it may take (in our case $\varphi \in [0.6^\circ, 2.6^\circ]$). We propose to ascertain it by solving the system in (9) for several hypotheses of φ inside this range, taking the hypothesis that leads to a better adjustment of the linear model to the lane markings in the image.

In order to solve Equation (9) a minimum of four points are necessary, provided that there is at least one point on each curve. In addition, they correspond to two parallel curves L metres apart, when back projected to the road plane. This implies that we are going to fit *both left and right* lane markings at the same time and enforcing parallelism, that is, consistency in the solution. In addition, the sparsity of candidates on one side of the lane due to occlusions or dashed lane marks can be compensated for by those on the other side.

If more than four points are known, we obtain an over-constrained system that is solved in the least-squares sense. The problem, of course, is the selection of the right points among all candidates from the previous detection step. We need a robust technique in the sense of simultaneously classifying candidate points into lane points (inliers) and not lane points (outliers, at least for that side), and perform the fitting only to the former ones. We use RANSAC, which stands for Random Sample Consensus (Fischler and Bolles, 1981), which is a general estimation technique for such purpose based on the principle of hypotheses generation and verification.

In order to take advantage of the temporal coherence of lane markings parameters in consecutive images (ie, their smooth change from frame to frame), we feed the RANSAC output into a Kalman filter, where they are considered as the observations of the lane-markings state, which varies along time following a Brownian motion model. In this way, incorrect model fittings that may occur in challenging situations are managed properly, obtaining a more stable lane-markings estimation.

Figure 9 shows results of lane-markings detection in quite challenging situations owing to occlusions, shadows, reflections and poor lighting conditions, where nevertheless our method succeeds.

5. Results

The benefit of using lane-markings information in the detection process is clearly shown in Figure 10. On the one hand, the horizon line v_h is better located in the image, which is a consequence of using an accurate camera pitch φ (v_h and φ are related by $v_h = v_0 + f^v \tan \varphi$). On the other hand, the vehicle search region is constrained just to the current and the adjacent lanes. This region is delimited using expressions (6) and (7), assuming a lane width of $3L$.

In quantitative terms, using lane markings we reduce the inspected image area by an average of 28%. Since most of the discarded zones correspond to road locations distant to the camera, this implies that we elude to process a part of the image where a large number of small-scale windows would have been checked. As a result, the final number of inspected windows is reduced on average by 52%. This obviously results in a

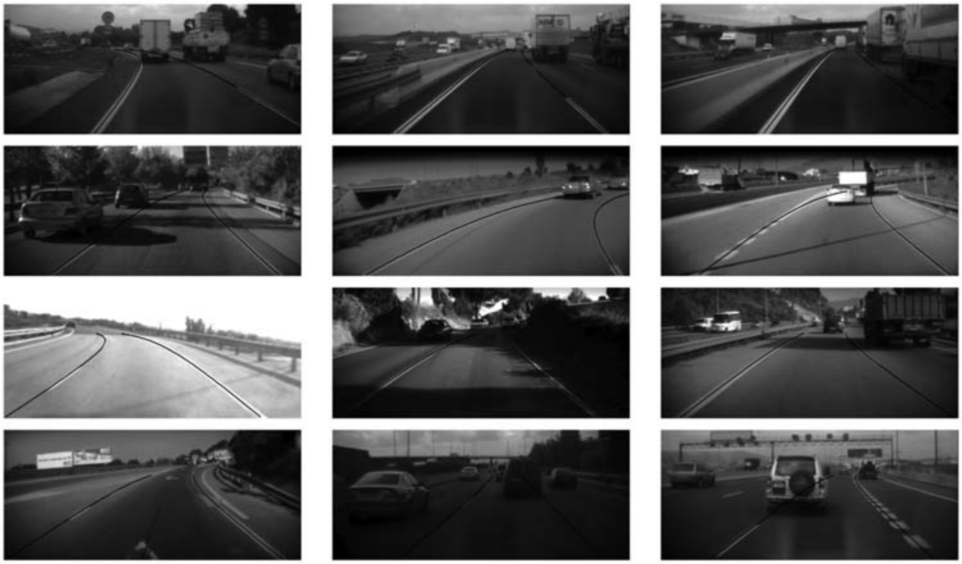


Figure 9 Segmented lane curves on frames acquired by different cameras: dashed lines, occlusion, special road marks, shadows

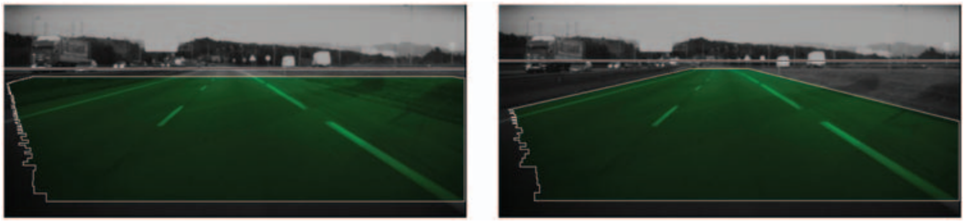


Figure 10 Outlined and in green, detection search region ignoring (left) or using (right) the lane-markings information. The horizontal white–black line represents the horizon line, according to the camera pitch used

reduction in the cost of the detection process. However, now we have the additional cost of the lane-markings estimation. Taking both processes into consideration, we observe that their overall cost is still lower than detecting vehicles without lane-markings information. Figure 11 shows statistics of the processing time per frame required by both approaches. Without lane markings, a frame is processed on average each 191.16 ms in a P4 3.0 GHz PC, which means a frame rate of 5.23 fps. On the other hand, estimating the lane markings and then detecting vehicles requires on average just 150.22 ms per frame, which means a frame rate of 6.66 fps. Note that now, with a smaller computational cost, we obtain more information from the driving environment (ie, the lane markings), which we can use to fulfil other ADAS tasks, such as LDW.

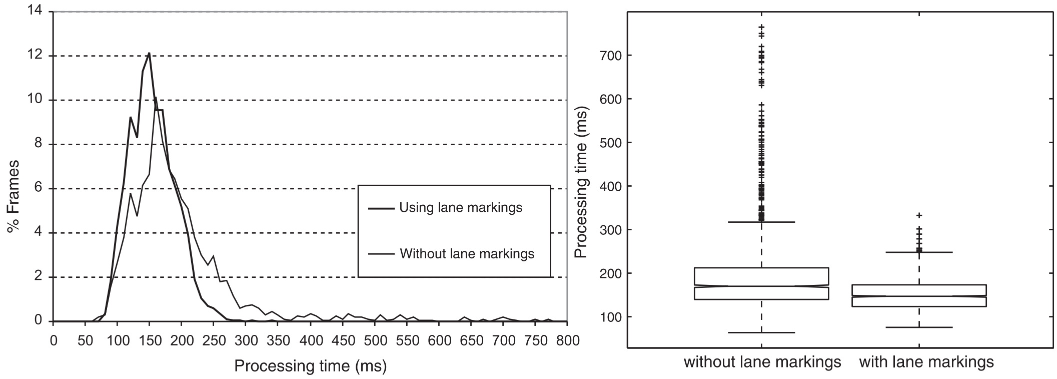
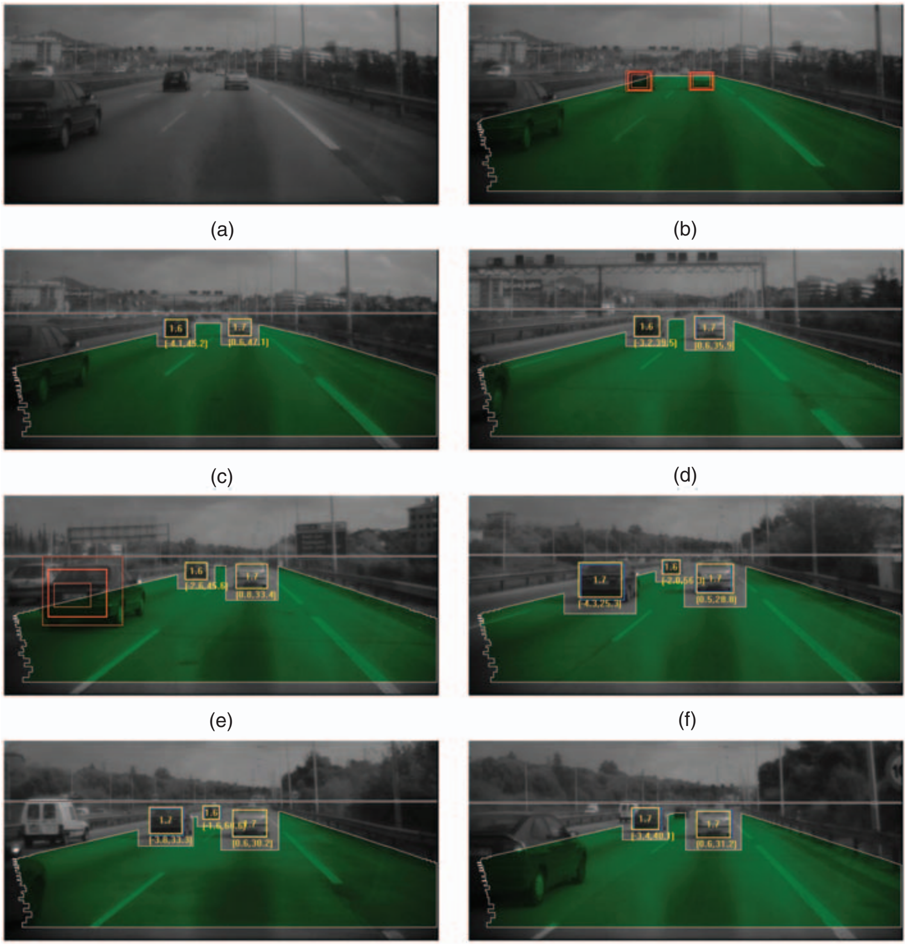


Figure 11 Statistical comparison of the frame processing time required with and without the use of lane markings. Left: Histogram of the processing time per frame. Right: Processing time boxplots, where an statistical significant difference between both methods is observed

In addition to the reduction in processing time, the use of lane markings also leads to qualitatively better detection and tracking of vehicles. Figure 12 gives an indication of how our system performs, showing how several vehicles are detected and tracked over time. However, still images do not allow one to appreciate the dynamic nature of the results, that is, how long it takes to verify a given vehicle in the image, how well it is tracked over time, when and why it is lost, etc. Hence, we have built a web page (www.cvc.uab.es/adas/projects/vehicledetection/TIMC/) where several processed video sequences can be viewed. Quantitative results of the system performance in these sequences are presented in Tables 1 and 2, organized according to the driving environment (highways and local roads).

On the highway sequences (Table 1), the traffic in the (potentially) two lateral lanes is all travelling in our same direction, thus all vehicles show their rear or lateral sides. All but one car and van are at least detected, and most cars are tracked successfully. However, trucks are more difficult to detect. The reason for this is that there is more variability in their appearances than with other types of vehicles, which furthermore changes significantly with the camera viewpoint. Hence, this suggests that we should consider a larger number of truck examples in the classifier learning process. It is remarkable that there are no false positives and only four objects are spuriously mistaken as vehicles. They are fences or shadows exhibiting marked vertical and horizontal line textures. Fifteen vehicles were not tracked because they were not detected for a sufficiently long time to trigger the tracker due to occlusions or because they moved further than the system operative distance.

Local roads (Table 2) are more challenging. On these roads the traffic in the left lane moves in the opposite direction to the camera and oncoming vehicles are far more frequent that those moving in the same direction as the camera. In fact, the



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 12 Example of the system performance. (a) First frame of the sequence. After several consecutive detections (b) new targets are added to the tracking module (c). This is represented by a yellow rectangle outlining the vehicle, where the estimation of its width and road location is numerically indicated. These targets are properly tracked in the following frames (d). New vehicles entering in the scene are properly detected (e) and tracked (f). Vehicles outside the inspection region are ignored (g). At far distances, very occasionally mistracking events occur (h)

vehicle in front may be the same during one whole sequence. This implies that most vehicles are forward facing, a point of view under-represented in the training set (less than the 10% of examples) originally devised for highways. Moreover, their relative velocity is larger, thus reducing the time that they are visible (detectable)

Table 1 Detection and tracking results for seven highway video sequences, 9 min long in total. Detected obstacles are false positives, not detected vehicles are false negatives

		Detected	
Vehicle	Not detected	Not tracked	Tracked
Car	1	7	42
Van	1	–	3
Truck	4	8	13
Non-vehicle	–	4	–

Table 2 Detection and tracking results for five local roads sequences, 5 min long in total. Detected or tracked obstacles are false positives, not detected vehicles are false negatives. Up and down arrows denote oncoming and outgoing (moving in the same direction as the camera) vehicles, respectively

			Detected			
	Not detected		Not tracked		Tracked	
Vehicle	↓	↑	↓	↑	↓	↑
Car	8	1	38	3	5	8
Van	–	–	6	–	–	–
Truck	1	–	5	–	4	1
Non-vehicle	–		6		1	

and in consequence the probability to trigger the tracker. In spite of it, only a few cars are completely missed, most of them are detected during a few frames and even tracked.

6. Conclusions

In this paper we have presented a monocular computer vision system devoted to detecting and tracking vehicles observed from a forward-facing camera mounted in a host vehicle. The most relevant features of the proposed system are as follows.

- Vehicles are detected through a frame scanning process, that takes the acquisition geometry into account to determine a set of windows in the image to be inspected.
- These windows are evaluated by a proposed heuristic criterion and a cascade of classifiers learnt with Real Adaboost, discerning the presence of vehicles in frames.
- Unlike many approaches in the literature, we estimate the three-dimensional locations of vehicles and track them along time.

The proposed system requires knowledge of the camera pose with respect to the road, which is assumed to be planar. Since this pose can vary at each frame, we propose to estimate it taking advantage of the road lane markings. Concerning that point:

- a novel method has been presented, based on fitting a three-dimensional lane-markings model to ridges in the images;
- this is performed combining the RANSAC algorithm with a Kalman filter, achieving a significant robustness to image clutter.

Estimated lane markings allow us to ascertain the camera pose, but also delimit more precisely the image zone where the vehicles may be found. Experiments show that these two aspects contribute in improving the vehicle detection and tracking performance, and reducing the overall computational cost of the system.

Acknowledgements

This work has been partially funded by grants TRA2007-62526/AUT of the Spanish Education and Science Ministry, and Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

References

Aufrère, R., **Marmoiton, F.**, **Chapuis, R.**, **Collange, F.** and **Dérutin, J.** 2001: Road detection and vehicle tracking by vision for adaptive cruise control. *The International Journal of Robotics Research* 20(4), 267–86.

Bertozzi, M., **Broggi, A.** and **Fascioli, A.** 2000: Vision-based intelligent vehicles: state of the art and perspectives. *Robotics and Autonomous Systems* 32, 1–16.

Betke, M., **Haritaoglu, E.** and **Davis, L.S.** 2000: Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications* 12(2), 69–83.

Bishop, R. 2005: *Intelligent Vehicle Technology and Trends*, Artech House Publishers, Boston.

Bouguet, J. 2004: Camera calibration toolbox for Matlab, www.vision.caltech.edu/bouguetj/calib_doc/

Broggi, A., **Cerri, P.** and **Antonello, P.C.** 2004: Multi-resolution vehicle detection using artificial vision. *IEEE Intelligent Vehicles Symposium*, Parma, Italy, 310–4.

Chan-Yu, H., **Shih-Shinh, H.**, **Yi-Ming, C.**, **Li-Chen, F.** and **Pei-Yung, H.** 2007: Driver assistance system using integrated information from lane geometry and vehicle detection. *Proceedings IEEE Intelligent Transportation Systems Conference*, Seattle, WA, USA, 986–91.

Chih-Hsien, Y. and **Yung-Hsin, C.** 2006: Development of vision-based lane and vehicle detection system via the implementation of a dual-core DSP. *Proceedings IEEE Intelligent Transportation Systems Conference*, Toronto, Canada, 1179–84.

Clady, X., **Collange, F.**, **Jurie, F.** and **Martinet, P.** 2003: Cars detection and tracking with a vision sensor. *Proceedings IEEE Intelligent Vehicles Symposium*, Columbus, Ohio, USA, 593–8.

Dellaert, F., **Pomerleau, D.** and **Thorpe, C.** 1998: Model-based car tracking integrated with a road follower. *International Conference on Robotics and Automation*, Leuven, Belgium, 1889–94.

Dellaert, F. and **Thorpe, C.** 1997: Robust car tracking using Kalman filtering and bayesian templates. *Proceedings of SPIE: Intelligent Transportation Systems*, Pittsburgh, Pennsylvania, USA, 3207, 72–83.

Fischler, M.A. and **Bolles, R.C.** 1981: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), 381–95.

Gepperth, A., **Edelbrunner, J.** and **Bücher, T.** 2005: Real-time detection and classification of cars in video sequences. *IEEE Intelligent Vehicles Symposium,* Las Vegas, NV, 625–31.

Guiducci, A. 1999: Parametric model of the perspective projection of a road with application to lane keeping and 3D road reconstruction. *Computer Vision and Image Understanding* 73(3), 414–27.

Hartley, R.I. and **Zisserman, A.** 2004: *Multiple View Geometry in Computer Vision,* 2nd edition, Cambridge University Press, Cambridge.

Hilario, C., **Collado, J.**, **Armingol, J.** and **de la Escalera, A.** 2005: Pyramidal image analysis for vehicle detection. *IEEE Intelligent Vehicles Symposium,* Las Vegas, NV, 88–93.

Jung, C. and **Kelber, C.** 2005: Lane following and lane departure using a linear–parabolic model. *Image and Vision Computing* 23(13), 1192–202.

Kalinke, T., **Tzomakas, C.** and **von Seelen, W.** 1998: A texture-based object detection and an adaptive model-based classification. *IEEE International Conference on Intelligent Vehicles, Stuttgart, Germany*, Vol. 1, 143–48.

Kim, Z. 2006: Realtime obstacle detection and tracking based on constrained Delaunay triangulation. *IEEE Intelligent Transportation Systems Conference,* Toronto, ON, 548–53.

Lefaix, G., **Marchand, E.** and **Bouthemy, P.** 2002: Motion-based obstacle detection and tracking for car driving assistance. *International Conference on Pattern Recognition,* Québec, QC, 74–7.

Liu, T., **Zheng, N.**, **Zhao, L.** and **Cheng, H.** 2005: Learning based symmetric features selection for vehicle detection. *IEEE Intelligent Vehicles Symposium,* Las Vegas, NV, 124–9.

López, A., **Lloret, D.**, **Serrat, J.** and **Villanueva, J.** 2000: Multilocal creaseness based on the level-set extrinsic curvature. *Computer Vision and Image Understanding* 77(2), 111–44.

Maurer, M., **Behringer, R.**, **Fürst, S.**, **Thomanek, F.** and **Dickmanns, E.** 1996: A compact vision system for road vehicle guidance. *International Conference on Pattern Recognition,* Vienna, Austria, Vol. 3, 313–7.

McCall, J. and **Trivedi, M.** 2006: Video-based lane estimation and tracking for driver assistance: survey, system and evaluation. *IEEE Transactions on Intelligent Transportation Systems* 1(7), 20–37.

Peden, M., **Scurfield, R.**, **Sleet, D.**, **Mohan, D.**, **Hyder, A.**, **Jarawan, E.** and **Mathers, C.** 2004: *World Report on Road Traffic Injury Prevention,* WHO Presswww.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/index.html

Ponsa, D. and **López, A.** 2007a: Cascade of classifiers for vehicle detection. *Advanced Concepts for Intelligent Vision Systems,* Delft, the Netherlands (*Lecture Notes in Computer Science*, Vol. 4678, Springer, Berlin, 980–89.

Ponsa, D. and **López, A.** 2007b: Vehicle trajectory estimation based on monocular vision. *3rd Iberian Conference on Pattern Recognition and Image Analysis,* Girona, Spain (*Lecture Notes in Computer Science*, Vol. 4477:I), Springer, Berlin, 587–94.

Ponsa, D., **López, A.**, **Serrat, J.** and **Lumbreras, F.** 2005: Multiple vehicle 3D tracking using an unscented Kalman filter. *8th International IEEE Conference on Intelligent Transportation Systems,* Viena, Austria, 1108–13.

Schapire, R.E. and **Singer, Y.** 1999: Improved boosting using confidence-rated predictions. *Machine Learning* 37(3), 297–336.

Sotelo, M., **Nuevo, J.**, **Bergasa, L.** and **Ocaña, M.** 2005: Road vehicle recognition in monocular images. *Proceedings IEEE International Symposium on Industrial Electronics,* Duvrobnik, Croatia, Vol. 4, 1471–76.

Sun, Z., **Bebis, G.** and **Miller, R.** 2002a: On-road vehicle detection using Gabor filters and support vector machines. *International Conference on Digital Signal Processing,* Santorini, Greece, Vol. 2, 1019–22.

Sun, Z., **Miller, R.**, **Bebis, G.** and **DiMeo, D.** 2002b: A real-time precrash vehicle detection system. *Sixth IEEE Workshop on Applications of Computer Vision,* Orlando, FL, 171–6.

ten Kate, T., **van Leewen, M.**, **Moro-Ellenberger, S.**, **Driessen, B.**, **Versluis, A.** and **Groen, F.** 2004: Mid-range and distant vehicle detection with a mobile camera. *IEEE Intelligent Vehicles Symposium,* Parma, Italy, 72–76.

United Nations Economic Commission for Europe (UNECE). 2007: *Statistics of Road Traffic Accidents in Europe and North America,* www.unece.org/trans/main/wp6/transstatpub.html#accidents

Viola, P. and **Jones, M.** 2001: Rapid object detection using a boosted cascade of simple features. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition,* Kauai, Hawaii, USA, Vol. 1, 511–8.

Zhang, Y., **Kiselewich, S.J.** and **Bauson, W.A.** 2006: Legendre and Gabor moments for vehicle recognition in forward collision warning. *IEEE Intelligent Transportation Systems Conference,* Toronto, ON, 1185–90.