Computer Vision and Image Understanding 114 (2010) 583–595

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

2D–3D-based on-board pedestrian detection system

David Gerónimo [*], Angel D. Sappa, Daniel Ponsa, Antonio M. López

Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:
Received 29 December 2007
Accepted 10 July 2009
Available online 10 February 2010

Keywords:
Pedestrian detection
Advanced Driver Assistance Systems
Horizon line
Haar wavelets
Edge orientation histograms

ABSTRACT

During the next decade, on-board pedestrian detection systems will play a key role in the challenge of increasing traffic safety. The main target of these systems, to detect pedestrians in urban scenarios, implies overcoming difficulties like processing outdoor scenes from a mobile platform and searching for aspect-changing objects in cluttered environments. This makes such systems combine techniques in the state-of-the-art Computer Vision. In this paper we present a three module system based on both 2D and 3D cues. The first module uses 3D information to estimate the road plane parameters and thus select a coherent set of regions of interest (ROIs) to be further analyzed. The second module uses Real AdaBoost and a combined set of Haar wavelets and edge orientation histograms to classify the incoming ROIs as pedestrian or non-pedestrian. The final module loops again with the 3D cue in order to verify the classified ROIs and with the 2D in order to refine the final results. According to the results, the integration of the proposed techniques gives rise to a promising system.

1. Introduction

Nowadays, traffic accidents represent one of the major causes of death worldwide. According to the World Health Organization, everyday 3000 people die as a result of a road accident [1]. Concretely, in the vehicle-to-pedestrian accidents case the Economic Commission for Europe reported almost 150,000 injuries and 7000 killed pedestrians only in the European Union in 2003, representing the second source of fatalities just after vehicle-to-vehicle accidents [2]. However, contrary to the socially accepted view of traffic accidents as a random and unpredictable consequence of road transportation, these fatalities can be tackled by prevention and sensible measures. As a result, in the last decades such a problem is gaining more attention from both governments and industry, which invest big efforts in traffic safety research.

In last decade, in addition to the improvements in the road infrastructures (e.g., visibility enhancements, roundabouts, speed controls, better signposting, etc.), a new area of research has received a special focus: the Advanced Driver Assistance Systems (ADAS). ADAS are intelligent on-board systems that aim at anticipating and preventing accidents, or at least, minimizing their effects when unavoidable. Examples of ADAS are the Adaptive Cruise Control, which adjusts the own vehicle speed in order to keep a safe gap with the preceding vehicle, or the Lane Departure Warning, which warns the driver in case that the vehicle leaves the lane inadvertently. One of the most complex ADAS applications are the Pedestrian Protection Systems (PPSs), focus of this paper. In this case, the aim is to detect and localize static or moving people in a defined area in front of the vehicle in order both to provide information to the driver and to perform evasive or braking actions. Fig. 1 illustrates the typical risky areas to be tackled by a PPS. In regular conditions, the vehicle stopping distance is about 5 m at 30 km/h , increasing up to 12 m at 50 km/h, thus the systems must intelligently focus their techniques on the danger of detecting a pedestrian in these areas.

Computer Vision, by the use of passive sensors like cameras, plays a key role in most of these systems. For instance, cameras are used in PPSs in order to detect the traffic objects of interest (i.e., pedestrians) taking advantage of their rich amount of cues and high resolution. The topics involved in ADAS are in the frontier of the state-of-the-art since they require real-time interpretation of outdoor scenarios (uncontrolled illumination) from a mobile platform (fast background changes and presence of objects of unknown movement). Furthermore, in the PPSs context, pedestrian detection is even more challenging due to the high variability of their appearance (i.e., different articulated pose, clothes, distance and viewpoint) and the cluttered scenarios usually found in urban environments. It is worth to mention that the moving nature of ADAS makes some well-established techniques from other human detection areas, like background subtraction methods for surveillance, not applicable in our case.

In this paper we present a pedestrian detection system that makes use of Computer Vision cues, specially taking advantage of 3D information to enrich the classification, which is typically based on 2D. The system is divided in three steps. First, 3D data

[*] Corresponding author. Fax: +34 935811670.
E-mail addresses: dgeronimo@cvc.uab.es (D. Gerónimo), asappa@cvc.uab.es (A.D. Sappa), daniel@cvc.uab.es (D. Ponsa), antonio@cvc.uab.es (A.M. López).
URL: http://www.cvc.uab.es/adas (D. Gerónimo).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.07.008

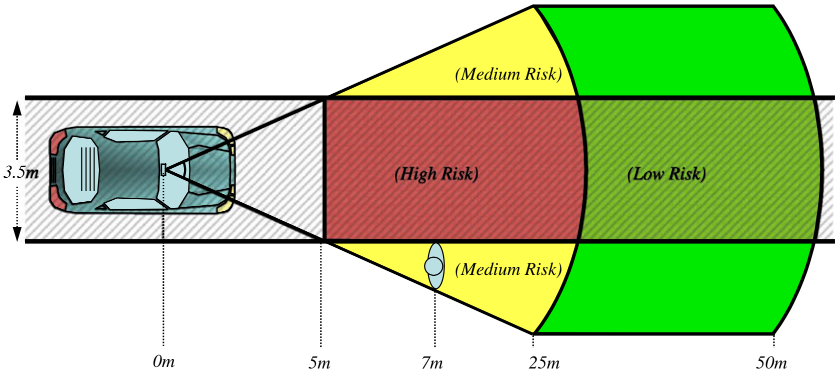

Fig. 1. The different areas of risk when driving. High risk area, in red, corresponds to a big danger of collision with pedestrians, always depending on the speed of the vehicle. Pedestrians in the medium risk area, in yellow, are likely to cross the front road, so typically no imminent is expected but the system must be aware of them. The low risk area, in green, contains pedestrians with no danger of imminent collision but that must be detected in advance since they stand in the vehicle's path. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

computed from a stereo rig are used to estimate the road pose, which is needed to adjust pedestrian sized windows in 3D. These windows, regions of interest (ROIs from now on), are then projected onto the 2D image plane where they are labeled as pedestrians or non-pedestrians by our proposed classifier: Real AdaBoost learning algorithm with Haar wavelets (HW) and edge orientation histogram (EOH) features. The final stage of the system verifies each positive labeled ROI by checking its 3D position and size. A final refinement stage is used to group overlapped redundant detections in 2D.

The remainder of this paper is as follows. After overviewing the related research in Section 2, an introduction to the proposed system is described in Section 3, fitting it to a general PPS architecture presented in [3]. Then, the modules of the current system, which make use of the aforementioned techniques, are placed in this architecture context. The first module, described in Section 4, makes use of the 3D-based adaptive image sampling technique. Section 5 presents the 2D classification module. Section 6 presents the last module, consisting of the 3D verification and the final 2D detections grouping. Finally, Section 7 presents experimental results of each of the three modules and of the whole system. Conclusions are summarized in Section 8.

2. Related research

By having a look at the literature [3] it is seen that most of the systems are based on feature selection and machine learning to perform 2D pedestrian classification. Some examples are the symmetry and binary template based approach by Broggi et al. [4], SVM on gradient images approach by Grubb et al. [5], the hierarchical template matching (*Chamfer System*) and neural networks by Gavrila et al. [6] or the parts-based SVM and AdaBoost approach by Shashua et al. [7]. In fact, PPSs can take advantage from a growing number of general people detection approaches proposed in recent years. For instance, Dalal and Triggs [8] propose *histograms of oriented gradients* (HOG) features and SVM. In [9], Leibe et al. perform the detection in two steps. First, image patches are extracted around difference of Gaussians keypoints [10]. Then, these patches are matched to a pedestrian model, which provides their spatial distribution, later used to cast votes to an hypotheses map. Finally, these hypotheses are verified and refined using template matching inspired in [6]. Tuzel et al. [11] base their classifier on the covariance of different measures (position, first and second order derivatives, gradient module, gradient orientation) in subwindows as features and boosting using Riemannian manifolds. Wu et al. [12] propose a parts-based scheme consisting of four body parts and three view categories to train a boosting-like classifier. They use short edge segments as features. Felzenszwalb et al. [13] use HOG and SVM in a parts-based approach, too. In this case, six different dynamic parts (not constrained to a fixed position in the hypothesis) are used.

Given that these methods are based on processing 2D images, a simple way of applying them is to classify windows of all the possible positions and sizes in the incoming image, which is often referred to as exhaustive window scanning (Fig. 3a). However, although widely used in general human detection approaches [14,8], this procedure not only is too expensive in terms of computational time (millions of windows should be classified) but also potentially increases the number of false positives by providing not relevant ROIs (e.g., sky areas). As a result, prior knowledge of the scene is generally considered to reduce this large amount of windows. For instance, since the system looks for pedestrians, only windows on the road surface should be taken into account for classification. Hence, an intuitive technique often used in ADAS literature is to fix an image row corresponding to the horizon and then assume that all pixels below this row belong to the road surface. As a result, a window laying on each pixel can be generated according to some mean pedestrian size constraints and the geometry of image formation. This approach, used by Gavrila et al. [6], has an implicit assumption: the relative position and orientation between the camera and the road do not change, i.e., the horizon line row is defined for the first frame and kept constant through the whole video sequence. They refer to this constraint as *flat world assumption*. However, due to vehicle movement, road slope and even road surface irregularities, there are many cases where such assumption is not fulfilled, specially in urban scenarios. Therefore, in order to compensate camera changes, many possible different windows per pixel should be considered, which would translate again in a very high processing time and potential false positives.

Some strategies to avoid the *flat world assumption* have been proposed. For instance, Soga et al. [15] propose a dense stereo based candidate window selection step that avoids an exhaustive searching of the whole image. Candidate windows are defined in those regions that contain solid objects (i.e., vertical surfaces) with a height in between 70 cm and 250 cm. Broggi et al. [16] propose first to identify vertical objects using a kind of v-disparity image [17] obtained from a stereo head. Then, further classification stages are focused only on those vertical objects.

Some systems propose a further step to reinforce detections. Gavrila et al. [6] make use of a disparity consistency test from a calibrated stereo rig to validate silhouette-based hypotheses. Ess et al. [18] propose a multi-frame scheme that jointly estimates scene geometry and verifies hypotheses by using a graphical model. Instead of using a dense depth map, Leibe et al. [19] propose a real time Structure from Motion (SFM) based geometry estimation for continually estimating the camera pose and scene's ground plane at every frame. In this case this online calibration is not used for reducing searching space but for refining each hypothesis (detected pedestrian) under a 3D location prior. Main challenge on SFM-based scene geometry estimation approaches lies on a robust feature point extraction and matching, in particular when the scene contains large amount of moving object.

3. 2D–3D system

The literature overview leads us to two important points, which are taken as keypoints for the current proposal. First, it is difficult to think of a perfect classifier just using 2D cues, thus we bet for combining it with 3D information. Second, a common methodology can be inferred from such proposals when tackling the development of a PPS. In fact, in a recent survey, Gerónimo et al. [3] propose a general architecture for ADAS pedestrian detection. It consists of six modules in which a complete system can be divided. The name and target of all the modules is described next:

- *Preprocessing*: It is the very first computation made with the image, aimed at preparing it for further processing. An example of preprocessing is to perform distortion rectification or contrast adjusting.
- *Foreground segmentation*: It extracts ROIs to be sent to the classification module. The key is to avoid as many background ROIs as possible but not discarding the ones containing pedestrians (Section 4).
- *Object classification*: It labels the selected ROIs as pedestrians or non-pedestrians (Section 5).
- *Verification and refinement*: It provides additional checks for the ROIs classified as pedestrians. It is focused on filtering false positives by using criteria not overlapped with the object classification ones (Section 6).
- *Tracking*: It follows pedestrians along time both to filter out spurious detections and predict their future position and direction.
- *Application*: It consists of all the high level warnings and actions taken by making use of the information of the previous modules. Some examples are acoustic warnings, automatic deployment of airbags or automatic braking.

In the current proposal we focus on three modules, that we understand as the core ones for a PPS: foreground segmentation, object classification and verification/refinement. Next sections describe the proposed solutions within the framework of these three modules as the scheme in Fig. 2 introduces.

In the current work, the richness of stereo vision information is first exploited in a foreground segmentation module according to the following scheme. Initially, it is used to automatically compute current horizon line without assuming a predefined set of constrains such as those used by monocular based systems. The underlying methodology is to fit a surface to 3D road data. Hence, since 3D data are referred to camera coordinate system, the camera position and orientation related to the fitted surface are easily obtained. Nedevschi et al. [20] propose to fit a clothoid model of the road surface using a lateral projection of the 3D points. A similar approach is presented by Danescu et al. [21] in the context of guardrails and fences detection. Both approaches are intended for extracting 3D points above the road, which are later on clustered into objects. In both cases all the processing is performed using only stereo vision information. The main drawback of these approaches lies on the use of edge points mainly coming from the road lane markings; hence they become useless in those areas where lanes are not well defined as often happens in urban scenarios, which are natural ones for pedestrian detection applications. A different approach was presented by Sappa et al. [22]. Although it uses a simple planar model of the road surface, it has been shown useful to cope with uphill/downhill driving, as well as dynamic pitching of the vehicle. In the current paper, a new approach based on 3D data provided by a stereo vision system is presented. It fits a plane to the 3D data points and places uniformly distributed pedestrian sized windows on the estimated road surface. With this adaptive image sampling, the amount of 2D ROIs to classify is reduced in three orders of magnitude, i.e., from millions to thousands (Fig. 3b). Notice that the road estimation technique could also benefit other ADAS functionalities (e.g., vehicle detection and road segmentation).

Once the reduced ROI set is selected, the classifier is aimed at labeling the selected ROIs as pedestrians or non-pedestrians. In this paper we propose to use a combination of Haar wavelets (HW) and edge orientation histograms (EOH) as features and Real AdaBoost as learning machine to provide a linear classifier. These simple and fast-to-compute features, together with the fast and

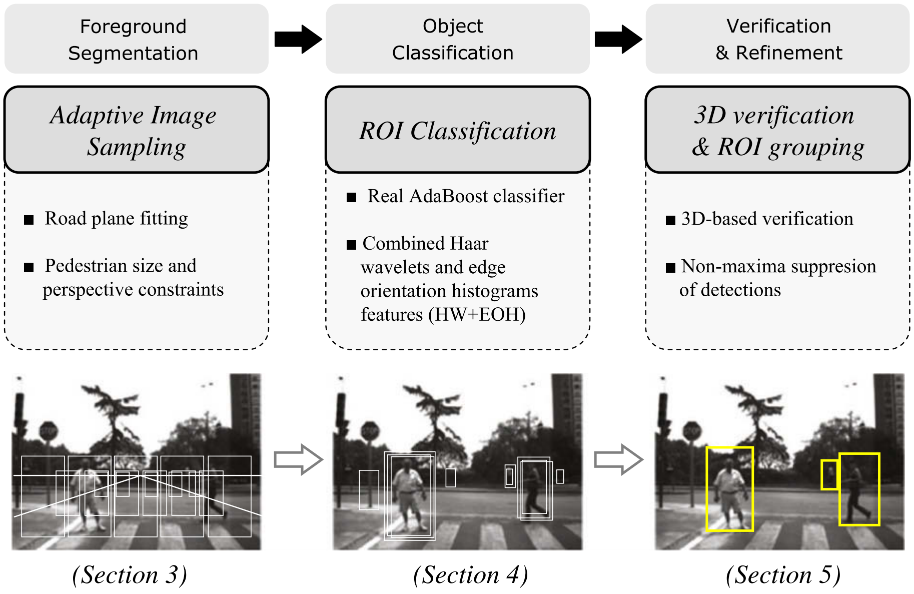
Foreground Segmentation
Object Classification
Verification & Refinement
Adaptive Image Sampling
ROI Classification
3D verification & ROI grouping
■ Road plane fitting
■ Pedestrian size and perspective constraints
■ Real AdaBoost classifier
■ Combined Haar wavelets and edge orientation histograms features (HW+EOH)
■ 3D-based verification
■ Non-maxima suppresion of detections
(Section 3)
(Section 4)
(Section 5)

Fig. 2. The three core modules of the system architecture.

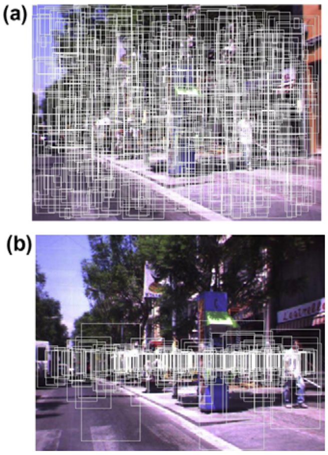

Fig. 3. (a) Exhaustive scan, showing just 0.1% of the total ROIs. (b) Adaptive image sampling, showing only 10% of the ROIs.

effective classifier represent a very convenient option given the computational time restrictions of the problem. We show that this classifier can improve the results of the people classification approach proposed by Dalal et al. [8], which up to our knowledge represents the state-of-the-art in human classification.

Finally, after the 2D based classification stage, stereo information is used again but this time for verifying results obtained with the classifier when possible. In this case, aiming at discarding as many false positives as possible, it is checked that the 3D values of the detected object in each positive ROI match the expected values to the ROI position and size. A final refinement stage groups the overlapped redundant 2D detections by using the mean-shift mode selection method proposed in [23] in order to provide one single detection per pedestrian in the scene.

This novel and more elaborated strategy of combining 2D/3D information results on a robust approach in the sense that every stage is implemented being aware of limitations of 2D/3D data. Hence, 3D is initially used for scene geometry estimation avoiding common problems related with poor stereo data. Then, pedestrians are detected by means of an efficient 2D classification over a reduced set of ROIs. Finally, 3D information is used for validating obtained results, and at the same time for clustering redundant 2D detections. The aim of this stage is to provide refined detections to the tracking module, not included in the current system, which would make use of temporal coherence to feed tracked detections to the application level.

4. Adaptive image sampling

The main target at this stage is to define a set of ROIs, by a uniform sampling of the road surface that results in an adaptive sampling of the image plane (Fig. 4c). It works by fitting a plane to the road surface using a RANSAC based least squares fitting.

In order to acquire the 3D information of the region in front of the host vehicle (Fig. 4b), a commercial stereo vision system (Bumblebee from Point Grey (http://www.ptgrey.com) has been used (Fig. 4a). The baseline of the stereo head is 12 cm and it is connected to the computer by a IEEE-1394 interface. Right and left color[1] images were captured at 5 fps at a resolution of 640 × 480 pixels. Camera control parameters were set to automatic mode to compensate global changes in the light intensity. After capturing these right and left images, 3D data were computed by using the provided 3D reconstruction software.

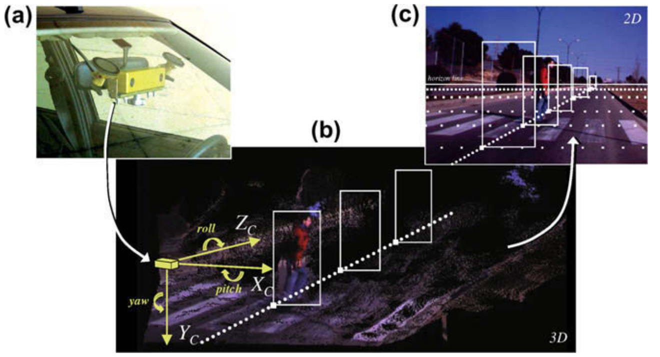

Fig. 4. (a) Front view of the camera setup. (b) Snapshot of the 3D data points obtained with the forward-facing stereo rig (notice that the image contains a large amount of holes due to occlusions and homogeneity in the image texture). (c) Desired sparse sampling: ROIs size in the 2D image space is automatically defined by the corresponding depth and the average pedestrian size.

The camera focal length, 6 mm, provides a horizontal field of view (HFOV) of 43°, and a vertical of 32.97°, which allows to detect pedestrians at a minimum distance of 5 m. The aforementioned reconstruction software provides 3D information until 50 m, which fits our requirements as seen in Fig. 1.

A world coordinate system (X_W, Y_W, Z_W) is defined for every acquired stereo image, in such a way that: the $X_W Z_W$ plane is contained in the current road fitted plane, just under the camera coordinate system (X_C, Y_C, Z_C); the Y_W axis contains the origin of the camera coordinate system; the $X_W Y_W$ plane contains the X_C axis and the $Z_W Y_W$ plane contains the Z_C axis. Due to that, the six extrinsic parameters (three for the position and three orientation angles) that refer the camera coordinate system to the world coordinate system reduce to just three, denoted in the following as (Π, Φ, Θ) (i.e., camera height, roll and pitch). Fig. 5 illustrates the world and camera coordinate systems.

From the (Π, Φ, Θ) parameters, in most situations the value of Φ (roll) is very close to zero. This condition is fulfilled as a result of a specific camera mounting procedure that fixes Φ at rest and because in normal urban driving situations this value scarcely varies [24].

The proposed approach presents a new method that, although similar in philosophy to the one presented in [22], reduces processing time by more than four. It consists of two stages : (i) 3D data point projection and cell selection and (ii) road plane fitting and ROIs setting. Both stages are detailed below.

4.1. 3D data point projection and cell selection

Let $D(r, c)$ be a depth map provided by the stereo pair with R rows and C columns, where each array element (r, c), is a scalar that represents a scene point of coordinates (x_C, y_C, z_C), referred to the camera coordinate system (Fig. 5). The aim at this first stage is to find a compact subset of points, ζ, containing most of the road points. To speed up the whole algorithm, most of the processing at this stage is performed over a 2D space. Initially, 3D data points are mapped onto cells in the $(Y_C Z_C)$ plane, resulting in a 2D discrete representation $\psi(o, q)$; where $o = \lfloor D_Y(r, c) \cdot \sigma \rfloor$ and $q = \lfloor D_Z(r, c) \cdot \sigma \rfloor$, σ representing a scale factor that controls the size of the bins according to the current depth map (Fig. 6). The scaling factor is aimed at reducing the projection dimensions in respect to the whole 3D data in order to both speed up the plane fitting algorithm and be robust to noise. It is defined as: $\sigma = ((R + C)/2)/(\Delta X + \Delta Y + \Delta Z)/3)$; $(\Delta X, \Delta Y, \Delta Z)$ is the working range in 3D space. Every cell of $\psi(o, q)$ keeps a reference to the original 3D data points projected onto that position, as well as a counter with the number of mapped points.

[1] For interpretation of color in Fig. 4, the reader is referred to the web version of this article.

From that 2D representation one cell per column (i.e., in the Y-axis) is selected relying on the assumption that the road surface is the predominant geometry in the given scene. Hence, it picks the cell with the largest number of points in each column of the 2D projection. Finally, every selected cell is represented by the 2D barycenter $\left(0, \left(\sum_{i=0}^{n} y_{C_i}\right)\Big/n, \left(\sum_{i=0}^{n} z_{C_i}\right)/n\right)$ of its n mapped points. The set of these barycenters defines a compact representation of the selected subset of points, ζ. Using both one single point per selected cell and a 2D representation, a considerable reduction in the CPU time is reached during the road plane fitting stage.

4.2. Road plane fitting and ROIs setting

The outcome of the previous stage is a compact subset of points, ζ, where most of them belong to the road. As stated in the previous subsection, Φ (roll) is assumed to be zero, hence the projection is expected to contain a dominant 2D line corresponding to the road together with noise coming from the objects in the scene.

The plane fitting stage consists of two steps. The first one is 2D straight line parametrisation, which selects the dominant line corresponding to the road. It uses a RANSAC based [25] fitting applied over 2D barycenters intended for removing outlier cells. The second step computes plane parameters by means of a least squares fitting over all 3D data points contained into inlier cells and finally places a set of ROIs uniformly distributed over the fitted plane. Both steps are described next.

Initially, every selected cell is associated with a value that takes into account the amount of points mapped onto that position. This value will be considered as a probability density function. The normalized probability density function is defined as follows: $pdf_i = n_i/N$; where n_i represents the number of points mapped onto the cell i and N represents the total amount of points contained in the selected cells.

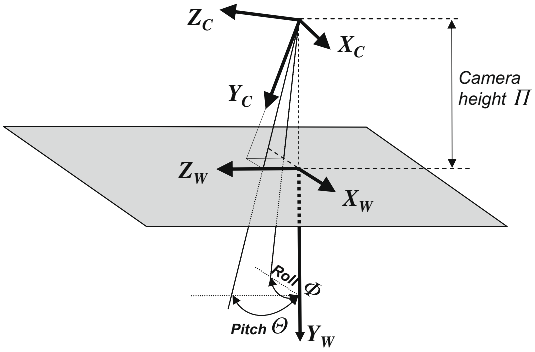

Fig. 5. Camera coordinate system (X_C, Y_C, Z_C) and world coordinate system (X_W, Y_W, Z_W).

Next, a cumulative distribution function, F_j, is defined as: $F_j = \sum_{i=0}^{j} pdf_i$; If the values of F are randomly sampled at n points, the application of the inverse function F^{-1} to those points leads to a set of n points that are adaptively distributed according to pdf_i.

4.2.1. Dominant 2D straight line parametrisation

At the first step a RANSAC based approach is applied to find the largest set of cells that fit a straight line, within a user defined band. In order to speed up the process, a predefined threshold value for inliers/outliers detection has been defined (a band of ±10 cm was enough for taking into account both data point accuracy and road planarity); an automatic threshold could be computed for inliers/outliers detection, following robust estimation of standard deviation of residual errors [26]. However, it will increase CPU time since robust estimation of standard deviation involves computationally expensive algorithms (e.g., sorting functions).

Repeat the following three steps L times (e.g., $L = 80$)

(1) Draw a random subsample of two different barycenter points (P_1, P_2) according to the probability density function pdf_i using the above process.
(2) For this subsample, indexed by l $(l = 1, \ldots, L)$, compute the straight line parameters $(\alpha, \beta)_l$,
(3) For this solution, compute the number of inliers among the entire set of barycenter points contained in ζ, as mentioned above using a ±10 cm margin.

4.2.2. Road plane parametrisation and ROIs setting

At the second step plane parameters are computed by using all 3D data points contained into inlier cells (Fig. 6).

(1) From the previous 2D straight line parametrisation choose the solution that has the highest number of inliers.
(2) Compute $(a, b, c)_i$ plane parameters by using the whole set of 3D points contained in the cells considered as inliers, instead of the corresponding barycenters. To this end, the least squares fitting approach [27], which minimizes the square residual error $(1 - ax_C - by_C - cz_C)^2$ is used.

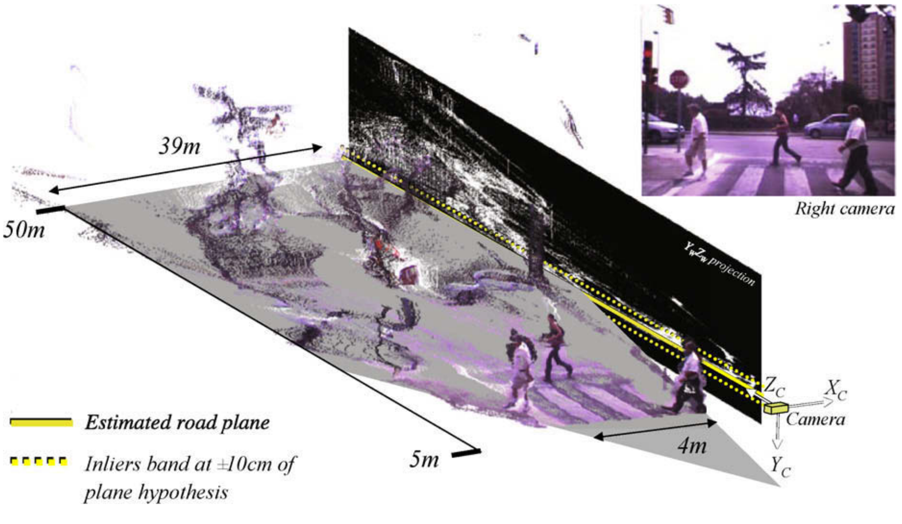

Fig. 6. YZ projection and road plane estimation.

(3) In case the number of inliers is smaller than 40% of the total amount of points contained in ζ (e.g., severe occlusion of the road by other vehicles), those plane parameters are discarded and the ones corresponding to the previous frame are used as the correct ones.

Finally, by using the fitted plane parameters, a set of ROIs sampling the whole road plane is defined. The ROIs are rectangular boxes orthogonal to the (X_W, Z_W) plane with their shortest edge parallel to the X_W axis. These ROIs are placed every 0.5 m, in both X_W and Z_W axes (see points in Fig. 4b). Actually, in order to cope with the different pedestrian dimensions, for every dot in that grid a set of five ROIs with the following width and height (in meters) is defined: $\{(0.75 \times 1.5), (0.80 \times 1.6), (0.85 \times 1.7), (0.90 \times 1.8), (0.95 \times 1.9)\}$. These ROIs, around 2000 in total, are projected to the 2D image plane to be classified in the next stage. The projection process consists in computing the four corner coordinates (v_i, u_i) of each 2D ROI as: $v_i = v_0 - fy_{C_i}/z_{C_i}$, $u_i = u_0 - fx_{C_i}/z_{C_i}$; where f is the focal length in pixels, (v_0, u_0) are the coordinates of the principal point and $(x_{C_i}, y_{C_i}, z_{C_i})$ are the corner coordinates of the given ROI in the 3D camera coordinate system. Then, each ROI can be represented in the image as (v_R, u_R, w_R, h_R), where (v_R, u_R) are the left-bottom coordinates and (w_R, h_R) the width and height.

5. ROI classification

Once the list of ROIs laying on the ground has been generated, this stage is aimed at labeling them as pedestrians or non-pedestrians, now by using just one of the cameras. Generally, in PPSs, object classification approaches can be broadly divided into two categories: silhouette matching and appearance based. The papers laying in the former one (e.g., head-and-shoulders binary silhouette [4] or the Chamfer System [6]) have been proven not to be robust enough to carry out the classification task. Thus, appearance based methods must be attached to improve robustness. Some examples of appearance based methods [8,9,6,11–13] have been described in Section 2.

In the current proposal we exploit an appearance based method by making use of a combination of two simple and fast-to-compute sets of features. They are scale and contrast invariant features which are efficiently computed thanks to the integral image representation. Due to the big number of features to learn, among all the possible learning algorithms, Real AdaBoost [28], a well-known fast and robust machine learning algorithm, is used to train the model with simultaneous feature selection. In our case, Real AdaBoost provides a linear combination of threshold-based weak classifiers. These ingredients are specially suitable for such a time and robustness demanding system. Next, these components are described in detail.

5.1. Feature sets

Although a recent paper [8] presents poor results with Haar wavelets (HW) for human detection when used in a single scale basis, they still represent a very fast and efficient approach when using the original formulation [14], i.e., using overcomplete dictionary (overlapped filters) over multiple scales. Moreover, in this paper we combine these features with edge orientation histograms (EOH), which provide complementary information by capturing geometric properties that are difficult to extract with HW. For instance, it is difficult for HW to represent the orientation of pedestrian legs whilst EOH are specially suitable for this case.

5.1.1. Haar wavelets

HW are widely used in other object detection systems [14,29]. A feature of this set is defined by a filter that computes the gray level difference between two defined areas (white and black; Fig. 7a):

$$\text{Filter}_{\text{Haar}}(x, y, w, h, type, R) = E_{\text{white}}(R) - E_{\text{black}}(R), \tag{1}$$

where x, y is the bottom-left position of the given filter in the ROI R; w, h represent its width and height; *type* is one of the filter configurations listed in Fig. 7b, and $E_{\text{area}}(R)$ is the sum of the pixels intensity in the filter region area.

Due to the perspective, the size of ROIs to be classified can vary significantly. Hence, some kind of size normalization is required to establish equivalence between features in the different ROIs. Some authors propose to resize all the ROIs to some standard dimensions (e.g., 64 × 128 in [8]) and then compute features using the rescaled window. This procedure, however, would result in too small windows when used in ADAS applications, i.e., the smallest ROIs hardly measure 30 pixels high. Therefore, in this proposal, ROIs are not resized since this would result in a big information loss. Instead, feature position and dimensions are calculated according to a *canonical window* (in our case, 12 × 24 pixels, which corresponds to a ROI at 50 m), but mapped to the original ROI dimensions when computing their value, thus not losing resolution (Fig. 7c). In addition, thanks to the use of the integral image representation [29], the time required to compute features is constant, independently of the size.

5.1.2. Edge orientation histograms

EOH are proposed by Levi and Weiss for face detection in [30]. They rely on the richness of edge information, so they differ from the intensity area differences of Haar wavelets. In this case, the features are illumination invariant by themselves since gradient orientations do not change, as long as lighting changes in the image are monotonic.

First, the gradient image is computed by a Sobel mask convolution (contrary to the original paper, no edge-thresholding is applied in our case). Then, gradient pixels are distributed into K

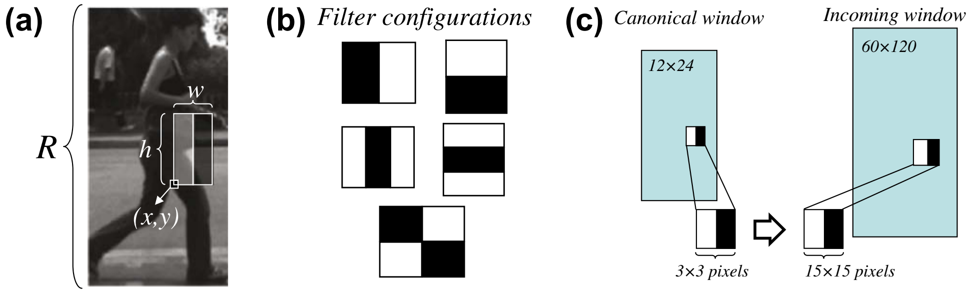

Fig. 7. Computation of Haar wavelet features: (a) Haar feature placed in a sample image; (b) some filter configurations; (c) filter normalization according to the incoming ROI size.

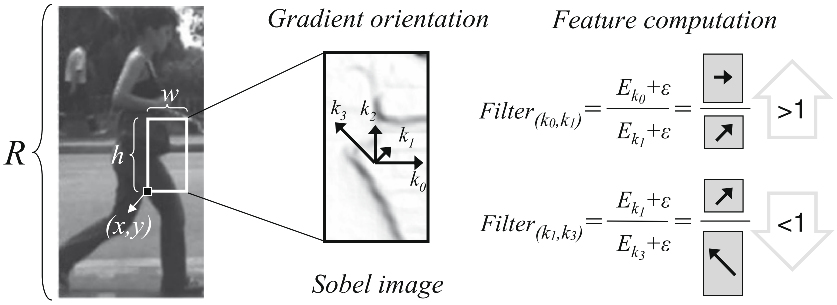
R
w
h
(x,y)
Gradient orientation
k_3 k_2 k_1 k_0
Sobel image
Feature computation
$Filter_{(k_0,k_1)} = \frac{E_{k_0}+\varepsilon}{E_{k_1}+\varepsilon} =$
>1
$Filter_{(k_1,k_3)} = \frac{E_{k_1}+\varepsilon}{E_{k_3}+\varepsilon} =$
<1

Fig. 8. Computation of edge orientation histograms. The feature can be viewed as the ratio of two orientations.

images (in our case we have tested $K = \{4, 6, 9\}$) corresponding to K orientation ranges (also referred as *bins*). Therefore, a pixel in bin $k_n \in K$ contains its gradient magnitude if its orientation is inside k_n's range, otherwise is null. Integral images are now used to store the accumulation image of each of the edge bins. At this stage, we improve the original Levi's algorithm by adding a bin interpolation step that distributes the gradient value into adjacent bins. This step is used in SIFT [10] and HOG [8] features, and in our case it improves the performance of the original EOH formulation in a 1% Detection Rate (DR) at a 1% False Positive Rate (FPR). Finally, the feature value is defined as the relation between two orientations, k_1 and k_2, of the filter in ROI R as:

$$\text{Filter}_{EOH}(x, y, w, h, k_1, k_2, R) = \frac{E_{k_1}(R) + \varepsilon}{E_{k_2}(R) + \varepsilon}. \tag{2}$$

If this value is above a given threshold, it can be said that orientation k_1 is dominant to orientation k_2 in the subregion of R defined by (x, y, w, h). The small value ε is added to the factors for smoothing purposes. Fig. 8 illustrates the previous process.

5.2. Learning algorithm

From the many available options in the literature, the learning algorithm to choose must fulfill the following two requirements. First, given that the set of features is large (over 120,000), it must be able to select a representative subset in an effective manner. Second, the computation in testing time must be low according to the requirements of the application. As a result, Real AdaBoost [28] has been chosen.

6. 3D verification and ROI grouping

Although the classification module provides satisfactory results when classifying state-of-the-art pedestrian databases as illustrated in Section 7, the number of false positives (FP) is still high to fulfill the requirements of ADAS. In addition, as a result of road sampling technique and a desired shift tolerance of the classifier, a number of overlapped ROIs containing a pedestrian are expected to be labeled as positive. Hence, two requirements are expected to be fulfilled by this module: to discard most of the false positives without discarding true positives and to provide one single detection window per pedestrian in the scene. Consequently, this module is divided into two stages. The first one, verification, aims at filtering out the false positives received as a result of 2D misclassification. Thus, the filtering is based on 3D information. The second, refinement, groups overlapped detected windows by using a 2D approach with the aim of providing one single detection per target.

We propose to first perform the 3D verification and the clustering, thus the algorithm is based on the original selected regions and then refinement is done just on the correct ROIs. Otherwise, if refinement came first, the verification could provide innaccurate results since it would be based on windows grouping both true and false positive ROIs, which would be useless.

6.1. 3D verification

The ROI verification is based on the fact that the 3D data corresponding to the image pixels contained in a 2D ROI should be consistent with the 3D ROI position and that the ROI content fulfills the pedestrian size constraints. The algorithm is divided into the following steps:

(1) Fill in points that lack of 3D information (due to occlusions or poor texture and illumination) by weighted interpolation with the pixel neighborhood (Fig. 9b). The filling does not provide any improvement by itself but it is mandatory for the next processing step.
(2) Apply a region growing algorithm using a single seed in the center of the ROI and depth as grouping criterion (Fig. 9c).
(3) Finally, a ROI is verified as a pedestrian if the dimensions of the contained object are similar to the ones of a standard pedestrian, and if the computed object depth (z) matches the defined depth of the ROI (Fig. 9d–f). Formally, this translates into fulfilling the following three constraints:

$$|h_S - \widetilde{h}| < \epsilon_h + e, \tag{3}$$

$$|w_S - \widetilde{w}| < \epsilon_w + e, \tag{4}$$

$$|d_S - \widetilde{d}| \leqslant \epsilon_d, \tag{5}$$

where $h_S,\ w_S,\ d_S$ are the silhouette height, width and depth to the camera; $\widetilde{h},\ \widetilde{w},\ \widetilde{d}$ are the height, width and depth of a standard pedestrian in the given ROI; $\epsilon_h,\ \epsilon_w,\ \epsilon_d$ are the maximum allowed errors for the previous parameters; and e is the error of the stereo computation, which exponentially increases according to the distance to the camera. All these parameters were either set by using standard dimensions (a pedestrian is about 1.70 m height) or manually tuned according to quantitative evaluations (e.g., the error margins in distance and dimensions).

6.2. ROI grouping

Once the ROIs are verified they are referred as detections. The refinement process is aimed at grouping multiple overlapped detections that contain only one pedestrian.

In this case, the non-maxima suppression algorithm proposed by Dalal in [23] is used. The algorithm first represents the set of detections as a kernel density estimate [31] and then searches for the local modes using mean shift [32]. First, a hard clipping function [23] discards detections with small confidence (in our case, lower than 3). Then, mean shift is iteratively computed for each detection until each one converges to a mode (Fig. 10c). As

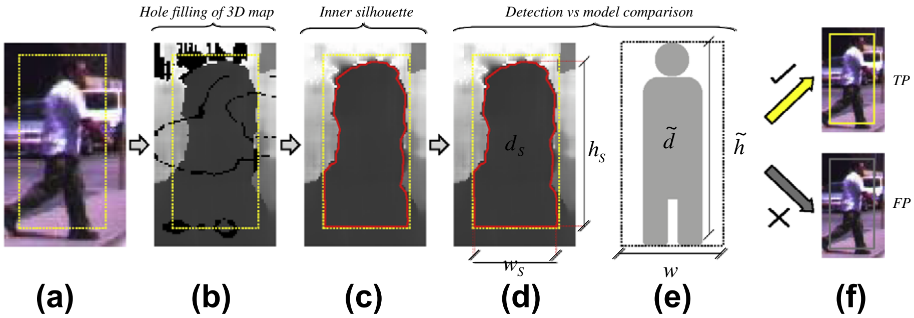

Fig. 9. Verification stage. (a) ROI with a positive confidence coming from the classifier. (b) Holes are filled in by neighbor interpolation to a certain extent. (c) Silhouette segmentation. (d) Silhouette height, width and depth to the camera. (e) Standard pedestrian dimensions. (f) Final decision as true positive or false positive by comparing detected object and standard pedestrian dimensions.

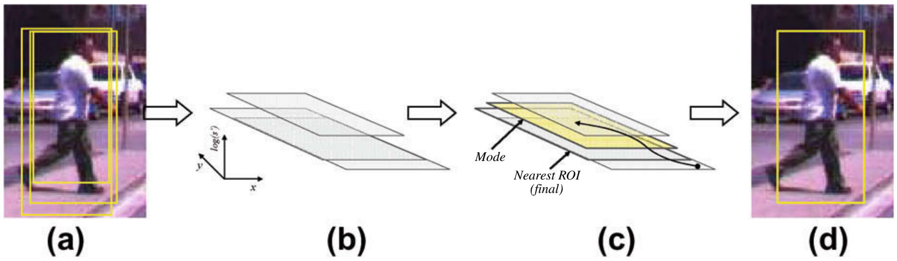

Fig. 10. Detections grouping. (a) Verified ROIs containing a pedestrian. (b) and (c) Mapping of detections to $\tau = [v, u, s]$ space, mean-shift algorithm selects the mode of the detections and selection of the nearest detection to the mode in τ space. (d) The verified ROIs which overlap are grouped to provide a single detection window.

a result, the overlapped detections containing the same pedestrian should be represented by the same mode, i.e., each mode is a final detection (Fig. 10d).

Final detections contain a label indicating the distance to the pedestrian in the final detection, which is computed by averaging the distance of the pixels in the segmented silhouette.

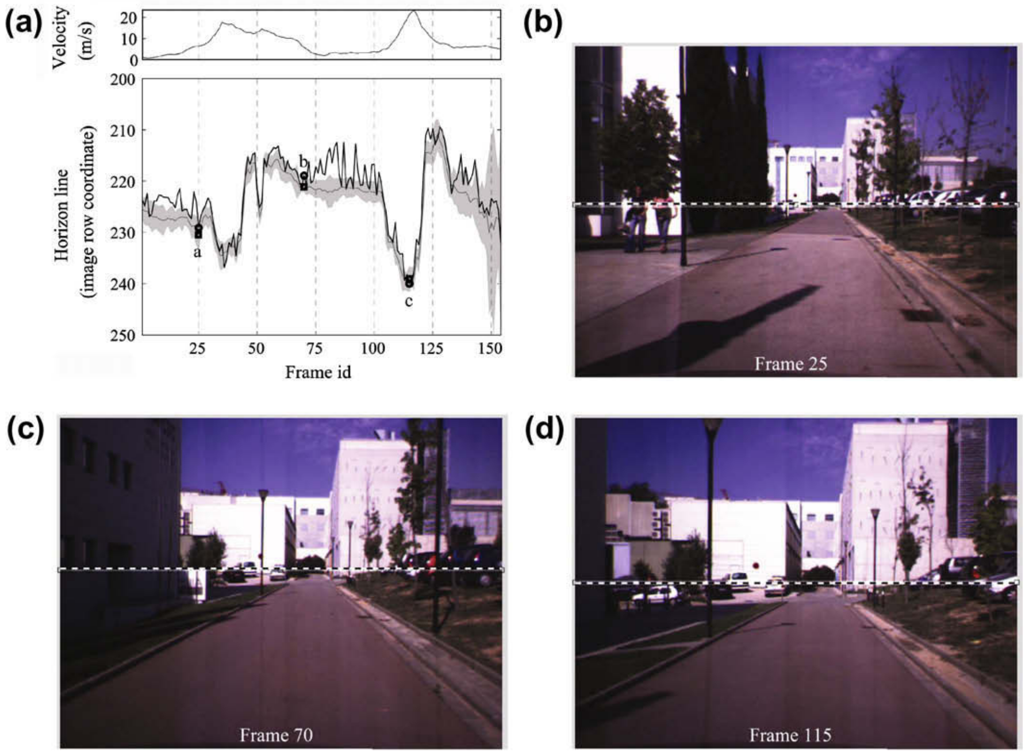

Fig. 11. Horizon line variation along a short video sequence. (a) Plot displaying for each frame the estimated velocity (top-most plot), and the 95% confidence region of the annotated horizon line position (gray-shaded area), together with the horizon line estimated by the proposed method (solid line). Labeled squares and circles highlight respectively the annotated and the estimated horizon line position in some selected frames. These frames are displayed next ((b)–(d)) to illustrate that the estimation provided by the proposed approach (dashed line) is always within the confidence region (brighter image region).

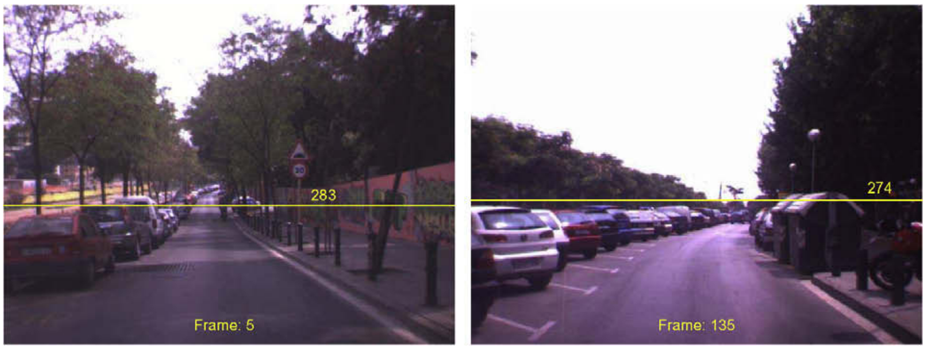

Fig. 12. Estimated horizon line (white line) in frames acquired in a narrow urban road. The robustness of the proposed technique can be qualitatively appreciated, in spite of the car-cluttered street.

7. Experimental results

As in any complex system, the obtained final result will depend on the success of every single component. In this section, the performance of each module is evaluated. Then, some final detection images illustrate the whole system results. A 3.2 GHz Pentium IV PC with a non-optimized code has been used.

7.1. Adaptive image sampling performance

In this section the performance of the proposed approach is studied by using several stereo video sequences. Final results will depend on the one hand on the accuracy of the estimated camera position and orientation, referred to the world coordinate system; and on the other hand on the accuracy of the fitted plane. For this reason, an initial validation procedure of the camera position and orientation technique is proposed. This validation procedure measures the quality of obtained results by representing them as a single value: *the horizon line*. The horizon line position (v_i) for a given frame i is computed by back-projecting into the image plane a point, $P_i(x_{C_i}, y_{C_i}, z_{C_i})$, lying over the fitted plane, far away from the camera coordinate system. Let $y_{C_i} = (1 - cz_{C_i})/b$ be the y_C coordinate of P_i by assuming $x_{C_i} = 0$. The corresponding y_{C_i} back-projection into the image plane, which defines the row position of the sought horizon line, is obtained as $v_i = v_0 + fy_{C_i}/z_{C_i} = v_0 + f/z_{C_i}b - fc/b$; where v_0 represents the vertical coordinate of the principal point; and z_{C_i} is the depth value of P_i. As mentioned above, since the point is far away from the camera $(z_{C_i} \to \infty)$, the horizon line is finally computed as $v_i = v_0 - fc/b$. The automatically computed horizon line position is compared with a value manually annotated by nine different users. Users were asked to locate a vanishing point in every frame, taking advantage of parallel structures observed on the road region neighboring the vehicle holding the camera (mainly lane borders and lane markings). From the collected annotation, the Gaussian distribution of the most likely horizon line location at each frame was considered.

Fig. 11 shows a short video sequence, where the host vehicle performs quite sudden changes on its velocity on a quite flat road. Fig. 11a depicts both: (*top*) the estimated vehicle velocity at each frame; (*bottom*) the ground truth confidence region of the annotated horizon line and the horizon location computed with the proposed technique. The information plotted in this second graphic is represented graphically in three frames of this short video sequence, in order to provide a better understanding of the method performance. It is clearly observed that the image row coordinate of the horizon line increases notably when the vehicle accelerates, decreasing more abruptly in the decelerations due to the behavior of the vehicle suspension system. This sequence also shows horizon line variations due to a change in the road pavement (around frame 50), and to the crossing of roads with different slopes (final frames). The major difference between the algorithm output and the ground truth annotation, around frame 80, corresponds to the quality of 3D road points provided by the acquisition system, which are notably sparse and noisy (due to the road homogeneity observed in these frames, very few road points can be reliably matched in the images of the acquired stereo pair), causing a less precise plane characterization. However, the difference on the horizon line location is less than 10 pixels. In general (90% of the processed frames) this horizon line location error is smaller than or equal to 4 pixels, which is a remarkable performance.

Robust location of horizon line in narrow urban scenarios can be appreciated in Fig. 12. The proposed approach works even in those frames where the image regions corresponding to the road

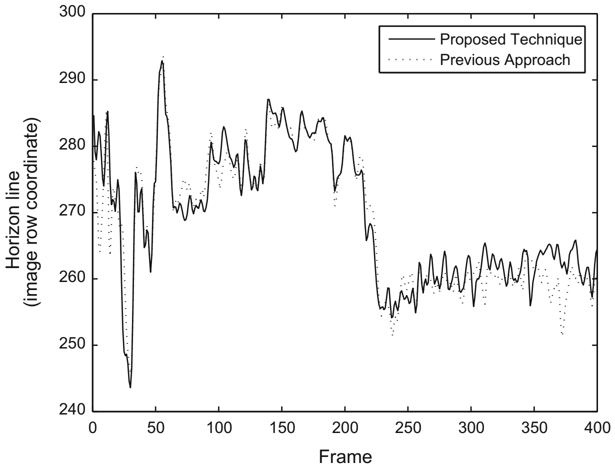

Fig. 13. Comparisons between horizon lines computed with the proposed technique and with a previous approach presented in [22].

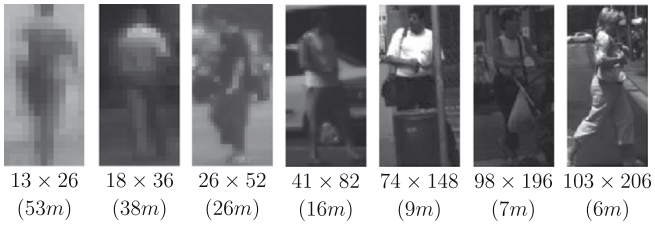

Fig. 14. Some positive samples of the database illustrating the high variability in terms of clothes, pose, illumination, background, and sizes. The size in the image (in pixels) and the approximate distance to the camera is noted below each sample.

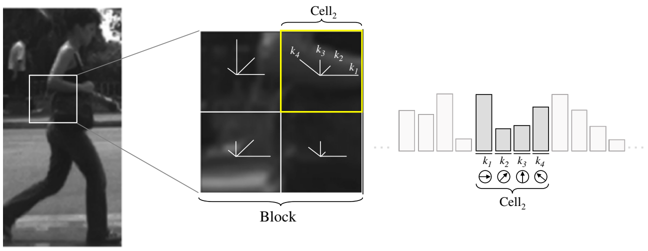

Fig. 15. Computation of histograms of oriented gradients. The classifier receives a vector from each block, representing the histogram of orientations for each cell in the block.

is notably smaller than the one present in the previous testing frames. This scene has been used to compare the proposed approach with a former one presented in [22]. Main difference with the previous approach lies in the road plane fitting stage (Section 4.2). While in [22] the road plane parameters were computed through a RANSAC based approach directly performed over the raw 3D road data points, in the current version firstly a dominant 2D straight line is obtained. This 2D straight line is then used to define the set of 3D points (inliers) considered to compute the plane parameters by means of the least squares fitting approach. Improvements are twofold, on the one hand the one-dimension reduction and on the other hand the proposed probability density based 2D point selection. This allows a reduction in the CPU time. Fig. 13 plots the horizon line position as a function of time by using the proposed technique and [22]. As can be appreciated, similar results are obtained, but more than four time faster with the proposed approach. On average, the new proposal took 78 ms per frame including both 3D points computation and on-board estimation of camera position and orientation.

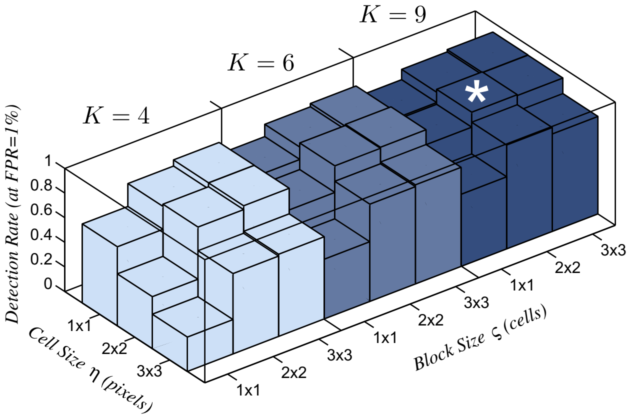

Fig. 16. Detection rate at FPR = 1% for all possible configurations of K, η and ς of HOG features (the best one is marked with a star).

7.2. ROI classification performance

In order to evaluate the performance of the classifier a pedestrian database has been built [33]. Contrary to other non ADAS-oriented databases [8], it contains images at different scales from urban scenarios. In our case, since color information is discarded as an useful cue, samples are transformed to grayscale. The complete database consists of 1000 positive samples, i.e., pedestrians; (Fig. 14) and 5000 negative ones, i.e., ROIs fulfilling the pedestrian size constraints but not containing pedestrians, thus no easy samples containing sky or building facades are likely to be selected. Samples are 1:2 aspect ratio, and maintain the original dimensions (not rescaling).

Each experiment randomly selects 700 positive and 4000 negative samples (training set) to learn a model, and use the remaining (testing set) to measure the classifier performance. The performance rates and plots are the result of averaging four independent experiments.

The proposed classifier is compared with, as far as we are concerned, the current state-of-the-art best classifier for human detection, which uses histograms of oriented gradients (HOG) features and support vector machine (SVM) learning, proposed by Dalal and Triggs [8]. HOG are SIFT-inspired features [10] that rely on gradient orientation information. The idea is to divide the image into small regions, named *cells*, that are represented by a 1D histogram of the gradient orientation. Cells are grouped in larger spatial regions called *blocks* so the histograms contained in a block are attached and normalized.

We have followed the indications of the authors as strictly as possible, and tuned the best parameters for our database in order to provide a rigorous and fair comparison with our proposal. As the authors suggest, no smoothing is applied to the incoming image, and a simple 1D $[-1,0,1]$ mask is used to extract the gradient information. Next, we have tested the best parameters for our database: number of bins ($K=\{4,6,9\}$ in 0–180°), cell sizes ($\eta=\{1\times1,2\times2,3\times3\}$ pixels) and block sizes ($\varsigma=\{1\times1,2\times2,3\times3\}$ cells), for our 12×24 canonical windows (notice that, similarly to HW and EOH, in this case blocks and cells are also scaled according to the size of the sample, as appreciated in Fig. 15). Although it is not done in the original proposal, we have made use of the integral image representation to speed up the computation of HOG. Block overlapping is set to the maximum possible, i.e., ς-fold coverage for each cell. Bin interpolation is also used here. As a last step, the block histogram is normalized using *L2-Hys*, the best method in the original paper, i.e., L2-normalizing,

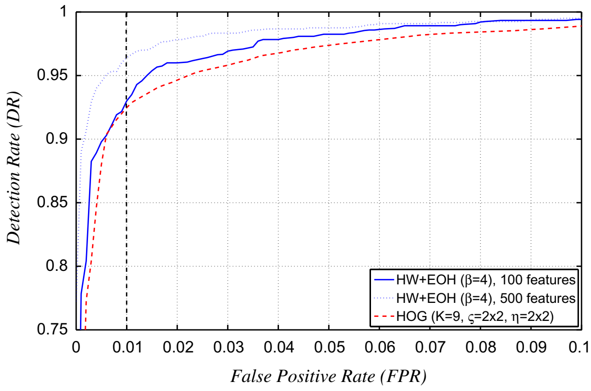

Fig. 17. Comparison between the proposed classifier and the best HOG-based classifier.

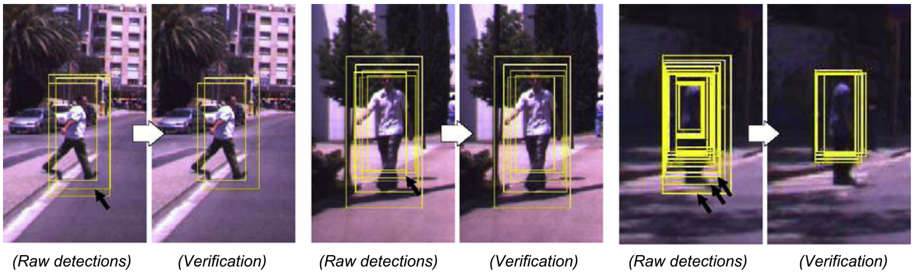

Fig. 18. Three different frames illustrating some false negatives, i.e., discarded positive detections (marked with a black arrow), after the verification stage as a result of slight shifted or oversized ROIs. In typical situations, though, a number of positive detections fitting the same pedestrian is accepted and passed to the refinement.

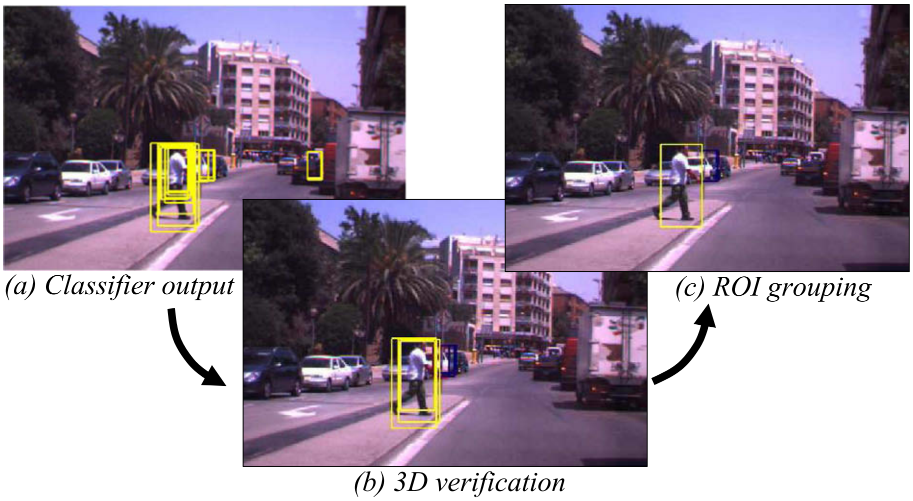

Fig. 19. Verification and refinement example. Note that images are cropped and scaled, so dimensions do not correspond to the ones in Fig. 20. (a) Classifier output has both false detections and overlapped correct detections. (b) 3D verification discards the FP, labels the detections that fulfill the Section 6 constraints as verified ROIs (yellow color), and as low-certainty ROIs the ones that lack of 3D information (red color). (c) Finally, overlapped ROIs are grouped and distance is estimated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

clipping values above 0.2 and then renormalizing. Finally, the features are fed to a linear SVM (following the authors indications, SVMLight[2] with $C = 0.01$ has been used). According to Fig. 16, the optimum parameters are $K = 9$, $\eta = 2 \times 2$ and $\varsigma = 2 \times 2$, which provide a Detection Rate (DR) of 92.5% at a False Positive Rate (FPR) of 1%.

Regarding our proposal, we have also made tests with $K = \{4, 6, 9\}$ for EOH, with very similar results. Hence, we bet for the $K = 4$ bins version since it requires less computational time.

Fig. 17 presents a comparison between our proposal and the HOG-based classifier. As can be seen, with 100 features (i.e., Real AdaBoost weak rules) we reach the same performance as HOG. However, our proposed features are ten times faster to compute (each ROI is classified in 0.015 ms). With 500 features the DR improves 4% (at a FPR of 1%) and it is computed about two times faster than HOG-based classifier.

Zhu et al. [34] have proposed a cascaded version of the HOG-based classifier, achieving similar detection performance to [8] but with a lower computational time. Hence, attending to Fig. 17 our approach is also comparable in terms of detection performance to [34]. However, a study is needed to check which of the algorithms is faster, having in mind that our approach could also take advantage of such a cascaded-scheme. Our immediate future work goes toward the use of such efficient classification scheme.

7.3. 3D verification and ROI grouping performance

In order to evaluate the 3D verification stage, a qualitative analysis has been made by computing verification statistics on the positive ROIs provided by the classifier. A testing set, not overlapped with the classification database, consisting of five urban driving sequences has been used. The classifier output ROIs in these sequences, 13,666 in total, have been manually divided as true positive (2240) and false positive (11,426), and then checked the verification output.

From the 11,426 negatives, 72% of them are discarded by the verification, while 23% are labeled as likely pedestrian and a 5% as pedestrians. This means that this stage is specially useful to filter out a big number of false positives. Regarding the 2240 true pedestrians, the verification stage labels 40% as pedestrians and 38% as likely pedestrians, which happens when there is not enough 3D information available, usually further than 25 m. The drawback of the high performance at discarding false positives lays on the ROIs containing pedestrians which are rejected by the verification, on average 22% of them. However, this percentage of false negatives in this stage does not affect the final results thanks to dense sampling of the scene, so at least one ROI per pedestrian is accepted and sent to the next stage (Fig. 18).

In the case of the refinement, a simple visual inspection of the results shows that the requirement for this stage, to have one single final window per pedestrian, is also fulfilled. In addition, the use of stereo data to compute the pedestrian distance provides two advantages. Contrary to other techniques [4], in our case the final window is not required to be perfectly adjusted to the pedes-

[2] http://svmlight.joachims.org.

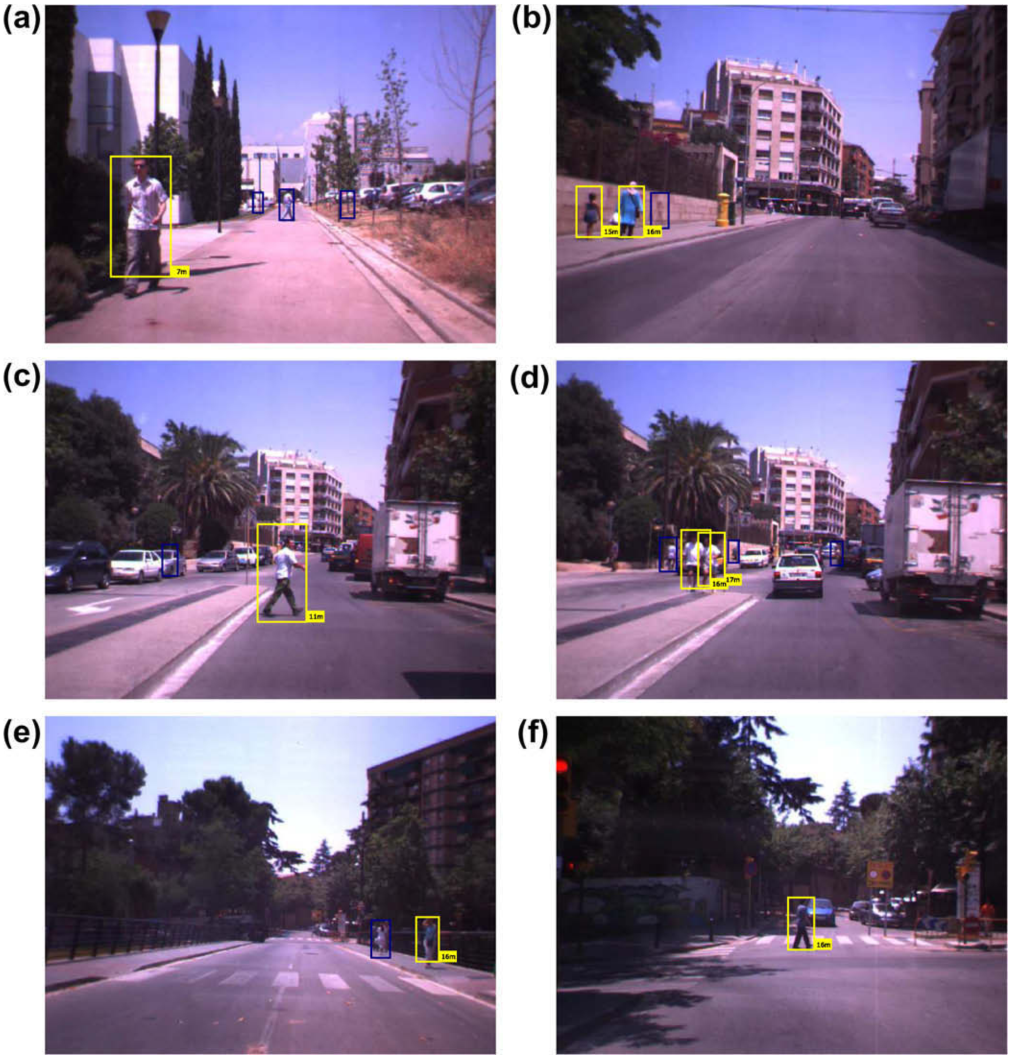

Fig. 20. Some snapshots of the system output when working in real urban scenarios. Notice that the bottom row of the images correspond to 5 m, as seen in Fig. 1. The yellow bounding boxes represent detected pedestrians and the red ones are low-confident detections where the classifier labeled them as pedestrians but the posterior module did not have enough 3D information to verify them. As expected, red detections (see (a) or (e)) correspond to far ROIs whilst yellow ones are the nearest. Next to each final detection it is noted the average distance of the 3D data contained in the ROI. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

trian feet, overcoming problems with occluded or poor-contrasted feet. Second, using the average of distances over all the pedestrian area is more reliable than just using the window bottom position.

Fig. 19 illustrates the module results in a frame. As can be seen, the verification discards false positives, marking two of them as likely pedestrian (undetermined due to poor 3D information at such distance). Then, the refinement provides final detections and distances.

The consumed time to verify and refine a ROI is 1 s using Matlab non-optimized code if the 3D information in the ROI is nearly complete. Future work in this module is focused on optimizing the computational time. Integrating this module with the others in a single C-based framework and optimizing the 3D filling stage could reduce the time consumption to a few milliseconds.

7.4. System results

Fig. 20 presents examples with results of the complete system.[3] The adaptive image sampling module tends to adjust the ROIs correctly, which makes the classification module receive bounded pedestrians, and thus provide many correct overlapped detections. In general, the final detections (yellow) are correct for many different scene illuminations and pedestrian poses. The ambiguity seems to increase with distance, thus several low-certainty ROIs appear in the background as a result of missing 3D information. However, a big distance also involves a longer time to react and prevent a possible accident as appreciated in Fig. 1, hence these ROIs could be tracked and further analyzed in time without danger of a sudden collision.

8. Conclusions

This paper presents a system that detects pedestrians from a moving vehicle in urban scenarios. There are three main contributions presented as independent modules. First, an adaptive image sampling method estimates the relative camera/road plane position in order to distribute pedestrian sized ROIs along the surface. This algorithm is also useful for other ADAS tasks like vehicle detection and road segmentation. Second, a pedestrian classifier based on fast-to-compute features, namely Haar wavelets and edge orientation histograms, and Real AdaBoost as learning machine, is

[3] Complete sequences can be found in http://www.cvc.uab.es/adas/projects/pedestrians/cviu2009.

presented, improving the results of the state-of-the-art in human detection. Third, a final module uses 3D data and window grouping to both verify the positive ROIs and refine the final detections, thus reducing the false positive rate thanks to the verification stage and providing one final detection per pedestrian thanks to the dense sampling and the refinement stage.

In addition, we have proposed a strategy to combine 2D/3D information in a cooperative module scheme where the output results of each step are used as input of next. In this way, 2D and 3D cues are exploited in each step depending on the task to be achieved and taking into account the limitations of the data.

Three main tasks, one for each module, are left as future work. First, non-uniform road scanning schemes have to be tested in order to provide a more sensible image sampling, e.g., a denser and sparser sampling for near and far distances respectively would provide more accurate results and at the same optimized efforts (e.g., a ROI at 40 m and 45 m nearly projects at the same image position, thus the scanning can be sparser at such distances). Second, to improve the classification module by adding cascades to Real AdaBoost and using different classifiers for different range of distances. Finally, a tracking module (e.g., recent proposals by Ess et al. [35] and Zhang et al. [36]) would help to discard spurious intermittent detections of false positives and provide stronger evidence of detections along time providing specific actions for the different areas of risk.

Acknowledgments

This work was been founded by the Government of Spain under Project TRA2007-62526/AUT and research programme Consolider Ingenio 2010: MIPRCV (CSD200700018). D. Gerónimo is supported by Grant BES-2005-8864.

References

[1] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A.A. Hyder, E. Jarawan, C. Mathers (Eds.), World Report on Road Traffic Injury Prevention, World Health Organization, Geneva, Switzerland, 2004.
[2] United Nations, Economic Commission for Europe. Statistics of Road Traffic Accidents in Europe and North America, Volume L, Geneva, Switzerland, 2005.
[3] D. Gerónimo, A. López, A. Sappa, Computer vision approaches to pedestrian detection: visible spectrum survey, in: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Girona, Spain, 2007, pp. 547–554.
[4] A. Broggi, M. Bertozzi, A. Fascioli, M. Sechi, Shape-based pedestrian detection, in: Proceedings of the IEEE International Conference on Intelligent Transportation System, Dearborn, MI, USA, 2000, pp. 215–220.
[5] G. Grubb, A. Zelinsky, L. Nilsson, M. Rilbe, 3D vision sensing for improved pedestrian safety, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy, 2004.
[6] D. Gavrila, S. Munder, Multi-cue pedestrian detection and tracking from a moving vehicle, International Journal on Computer Vision 73 (1) (2007) 41–59.
[7] A. Shashua, Y. Gdalyahu, G. Hayun, Pedestrian detection for driving assistance systems: single-frame classification and system level performance, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy, 2004.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego CA, USA, vol. 2, 2005, pp. 886–893.
[9] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 2005, pp. 878–885.
[10] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal on Computer Vision 60 (2) (2004) 91–110.
[11] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on riemannian manifold, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008).
[12] B. Wu, R. Nevatia, Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet part detectors, International Journal on Computer Vision 75 (2) (2007) 247–266.
[13] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage AK, USA, 2008.
[14] C. Papageorgiou, T. Poggio, A trainable system for object detection, International Journal on Computer Vision 38 (1) (2000) 15–33.
[15] M. Soga, T. Kato, M. Ohta, Y. Ninomiya, Pedestrian detection with stereo vision, in: Proceedings of the IEEE International Conference on Data Engineering, Tokyo, Japan, 2005.
[16] A. Broggi, A. Fascioli, I. Fedriga, A. Tibaldi, M.D. Rose, Stereo-based preprocessing for human shape localization in unstructured environments, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Columbus, OH, USA, 2003, pp. 410–415.
[17] R. Labayrade, D. Aubert, J. Tarel, Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Versailles France, vol. 2, 2002, pp. 17–21.
[18] A. Ess, B. Leibe, L. Van Gool, Depth and appearance for mobile scene analysis, in: Proceedings of the International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007.
[19] B. Leibe, K. Schindler, N. Cornelis, L. Van Gool, Coupled object detection and tracking from static cameras and moving vehicles, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1683–1698.
[20] S. Nedevschi, R. Danescu, D. Frentiu, T. Graf, R. Schmidt, High accuracy stereovision approach for obstacle detection on non-planar roads, in: Proceedings of the IEEE Intelligent Engineering Systems, Cluj Napoca, Romania, 2004, pp. 211–216.
[21] R. Danescu, S. Sobol, S. Nedevschi, T. Graf, Stereovision-based side lane and guardrail detection, in: Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Toronto, Canada, 2006, pp. 1156–1161.
[22] A. Sappa, D. Gerónimo, F. Dornaika, A. López, On-board camera extrinsic parameter estimation, Electronics Letters 42 (13) (2006) 745–747.
[23] N. Dalal, Finding People in Images and Video Sequences, Ph.D. thesis, INRIA Rhône Alpes, France, 2006.
[24] R. Labayrade, D. Aubert, A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Columbus, OH, USA, 2003, pp. 31–36.
[25] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Graphics and Image Processing 24 (6) (1981) 381–395.
[26] P. Rousseeuw, A. Leroy, Robust Regression and Outlier Detection, John Wiley & Sons, New York, 1987.
[27] C. Wang, H. Tanahashi, H. Hirayu, Y. Niwa, K. Yamamoto, Comparison of local plane fitting methods for range data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai Marriot, HW, USA, 2001, pp. 663–669.
[28] R. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336.
[29] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai Marriott, HW, USA, 2001.
[30] K. Levi, Y. Weiss, Learning object detection from a small number of examples: the importance of good features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington DC, CO, USA, 2004, pp. 53–60.
[31] D. Comaniciu, An algorithm for data-driven bandwidth selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2) (2003) 281–288.
[32] D. Comaniciu, V. Ramesh, P. Meer, The variable bandwidth mean shift and data-driven scale selection, in: Proceedings of the International Conference on Computer Vision, Vancouver, Canada, vol. 1, 2001, pp. 438–445.
[33] CVC Pedestrian Database, <http://www.cvc.uab.es/adas/databases/CVC-CER-01>.
[34] Q. Zhu, S. Avidan, M.-C. Yeh, K.-T. Cheng, Fast human detection using a cascade of histograms of oriented gradients, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York City, NY, USA, vol. 2, 2006, pp. 1491–1498.
[35] A. Ess, B. Leibe, K. Schindler, L. Van Gool, A mobile vision system for robust multi-person tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008.
[36] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008.