Adaptive Image Sampling and Windows Classification for On–board Pedestrian Detection

David Gerónimo, Angel D. Sappa, Antonio López and Daniel Ponsa

Computer Vision Center, Universitat Autònoma de Barcelona
Edifici O, 08193 Bellaterra, Barcelona, Spain
{dgeronimo,asappa,antonio,daniel}@cvc.uab.es — www.cvc.uab.es/adas

Abstract. On–board pedestrian detection is in the frontier of the state–of–the–art since it implies processing outdoor scenarios from a mobile platform and searching for aspect–changing objects in cluttered urban environments. Most promising approaches include the development of classifiers based on feature selection and machine learning. However, they use a large number of features which compromises real–time. Thus, methods for running the classifiers in only a few image windows must be provided. In this paper we contribute in both aspects, proposing a camera pose estimation method for adaptive sparse image sampling, as well as a classifier for pedestrian detection based on Haar wavelets and edge orientation histograms as features and AdaBoost as learning machine. Both proposals are compared with relevant approaches in the literature, showing comparable results but reducing processing time by four for the sampling tasks and by ten for the classification one.

1 Introduction

Advanced driver assistance systems (ADAS) aim to improve traffic safety and on–board computer vision contributes to that by detecting traffic objects of interest, such as vehicles and pedestrians, using passive sensors. The topics involved are in the frontier of the computer vision state–of–the–art since these tasks require real–time interpretation of outdoor scenarios (uncontrolled illumination) from a mobile platform (fast background changes and presence of objects of unknown movement). In this context, pedestrian detection is even more challenging due to the high variability of pedestrians appearance (different articulated pose, clothes, distance and viewpoint) and because urban environments are quite often cluttered scenarios.

Accordingly, most proposals include 2D pedestrian classification based on feature selection and machine learning [1–3]. A simple way of applying such classifiers to an image would be the folling: for each image pixel below a fixed *ceiling* row, assume that the pixel belongs to the road surface and examine (i.e., apply the classifier) all possible windows with origin at that pixel that could contain a pedestrian according to some pedestrian size constraints (PSC). However, this approach has an implicit assumption: the camera pose does not change, i.e., the horizon line is fixed from image to image. Obviously, due to vehicle movement

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

and road surface irregularities, such assumption is far for being true, specially in urban environments. Therefore, to compensate camera pose changes, many possible windows per pixel should be considered with the mentioned sketched image scanning method, which can imply considering millions of windows. Moreover, due to the high intra–class variability of pedestrians, classifiers tend to use thousands of features. Altogether turns out in a very high processing time.

In this paper we propose to use an on–board camera pose estimation, which will allow to sample the image according to a sparse grid of windows following the PSC (Sect. 2). Notice that such estimation can benefit also other ADAS functionalities (e.g., vehicle detection and road segmentation). Since our system is based on a stereo rig we will use the philosophy of [4], however, the new proposal is four times faster while keeping the estimation accuracy. The proposed technique reduces the number of windows to examine to roughly two thousand. These windows could now be checked by any additional rejecting mechanism, e.g., based on their corresponding depth information. However, at the current stage of our research we address the problem of obtaining a relatively fast pedestrian classifier with state–of–the–art performance or beyond. Then, at the moment, we use all the PSC compliant windows to test such classifier. In particular, [5] has recently been reported as the classifier with the best performance. In this paper we show (Sect. 3) how an alternative based on simple Haar wavelets and edge orientation histograms as features and AdaBoost as learning machine, is able to reach the same performance but being ten times faster. Section 4 presents results from combining the proposed image sampling and classifier, and Section 5 summarizes the main conclusions.

2 Camera Pose Estimation and Image Sampling

The main target at this stage is to define a set of windows (Fig. 1($left$)), regions of interest (ROIs) from now on, by fitting a plane to the road surface, and, at the same time, by estimating the relative camera pose (position and orientation referred to the fitted plane). For that purpose we use a stereo rig and focus on variations produced to the camera height and pitch angle (both referred to the road plane). Variations on camera yaw and roll angles can be neglected [6]. The proposed approach, although similar in philosophy to the one presented in [4], reduces processing time by four.

2.1 3D data point projection and cell selection

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. Let $D(r, c)$ be a depth map with R rows and C columns, where each array element (r, c), is a scalar that represents a scene point of coordinates (X, Y, Z), referred to the sensor coordinate system (Fig. 1($right$)). Initially, 3D data points are mapped onto a 2D discrete representation $P(u, v)$; where $u = \lfloor D_Y(r, c) \cdot \sigma \rfloor$ and $v = \lfloor D_Z(r, c) \cdot \sigma \rfloor$, σ representing a scale

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

factor [4]. Every cell of $P(u, v)$ keeps a reference to the original 3D data point projected onto that position, as well as a counter with the number of mapped points.

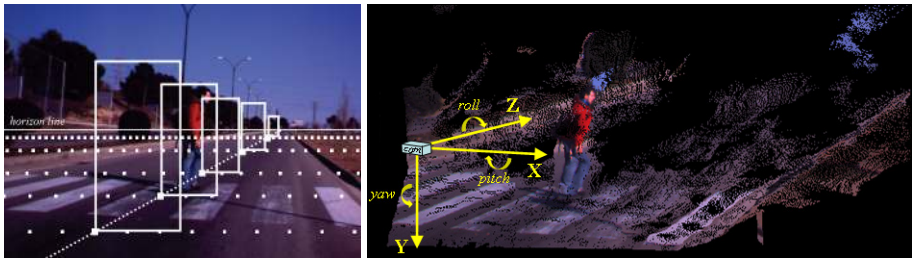

Fig. 1. (*left*) Desired sparse sampling: ROIs size in the 2D image space is automatically defined by the corresponding depth. (*right*) Snapshot of the corresponding 3D data points computed with the forward–facing stereo rig (notice that the image contains a large amount of holes due to occlusions and noisy regions).

From that 2D representation (YZ plane) one cell per column is selected relying on the assumption that the road surface is the predominant geometry in the given scene. Hence, the selection process goes bottom-up, in the 2D projection, through every column, and picks the first cell with more points than an adaptive threshold τ. The value of τ is defined for every column as 80% of the maximum amount of points mapped onto the cells of that column. It avoids the use of a fixed threshold value for the whole 2D space, as it is imposed in [4]. Recall that the density of points decreases with the distance to the sensor, hence the threshold value should depend on the depth. Finally, every selected cell is represented by the barycenter of its mapped points. The set of these barycenters define the sought subset of points, ζ. This data compression step is another difference with [4], where all points mapped into the selected cells were used for the fitting process. Using one single point per selected cell, a considerable reduction in the CPU time is reached.

2.2 RANSAC fitting with a compact set of 3D points

The outcome of the previous stage is a compact subset of points, ζ, where most of them belong to the road. However, since some outliers are also included in that subset of points, a RANSAC based [7] approach is used for computing plane parameters. 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: $f_{(i)} = n_{(i)}/N$; where $n_{(i)}$ represents the number of points mapped onto the cell i (Fig. 2($left$)) and N represents the total amount of points contained in the selected cells.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

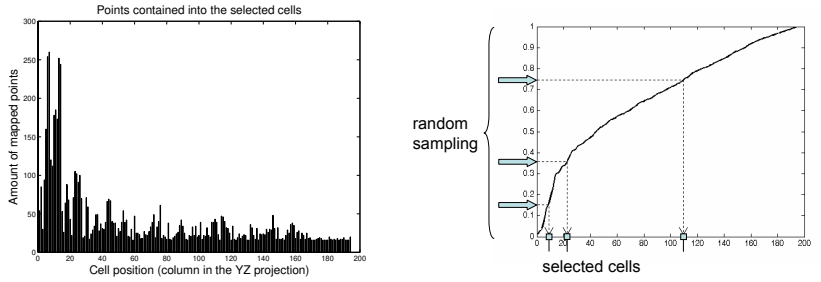

Fig. 2. (*left*) Bar diagram showing the amount of points mapped into the selected cells—recall that only one cell per column is picked up. (*right*) Cumulative distribution function computed from the amount of points mapped into every single cell.

Next, a cumulative distribution function, $F_{(j)}$, is obtained as: $F_{(j)} = \sum_{i=0}^{j} f_{(i)}$; 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 $f_{(i)}$ (Fig. 2(*right*)).

The fitting process computes plane parameters by means of an efficient RANSAC based least squares approach. In order to speed up the process, a predefined threshold value for inliers/outliers detection has been defined (a band of $\pm$10 cm was enough for taking into account both 3D data point accuracy and road planarity). The proposed approach works as follows:

Random sampling: *Repeat the following three steps K times (e.g., K=100)* (1) Draw a random subsample of 3 different 3D points (P_1, P_2, P_3) according to the probability density function $f_{(i)}$ using the above process (Fig. 2(*right*)); (2) For this subsample, indexed by k $(k = 1, ..., K)$, compute plane parameters $(a, b, c)_k$; (3) For this solution, compute the number of inliers among the entire set of 3D points contained in ζ, as mentioned above using $\pm$10cm.

Solution: (1) Choose the solution that has the highest number of inliers. Let $(a, b, c)_i$ be this solution; (2) Refine $(a, b, c)_i$ by using its corresponding inliers with the least squares fitting approach; (3) In case the number of inliers is smaller than 10% of the total amount of points contained in ζ, 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, the camera pose is directly obtained (recall that 3D data are referred to the camera coordinate system). Then, a set of ROIs sampling the whole road plane is obtained by placing every 0.5 m, in both X and Z axes (see grid points in Fig. 1(*left*)), a set of 5 boxes spanning from (0.75×1.5) m up to (0.95×1.9) m. These boxes, around 2,000 in total, are projected to the 2D image plane and their content is classified in the next stage.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

3 Pedestrian Classifier

Once the system has a list of ROIs laying on the ground, this stage is aimed at labeling them as pedestrians or non–pedestrians. Our proposal is to exploit Haar wavelets and edge orientation histograms as features and Real AdaBoost as learning algorithm. Next we describe the components of the classifier.

3.1 Classifier features

Haar wavelets. Haar wavelets (HW) are simple and fast–to–compute features, reminiscent of Haar basis functions used by *Papageorgiou et al.* [1] for object detection. A feature of this set is defined by a filter that computes the gray level difference between two defined areas (white and black) (Fig. 3*(left)*):

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

where x, y is the bottom-left position of the given image region R; w, h represent rectangle width and height; *type* is one of the filter configurations listed in Fig. 3*(middle)*, and $E_{area}(R)$ is the summatory of the pixels in the region area (white or black). In order to compute E, the *integral image* (ii) representation [8] has been used, where the summed values of a certain region can be efficiently computed by four ii accesses. In this work, we have followed the approach of *Viola and Jones* [8], where filters are not constrained to a fixed size, as proposed in [1], but can vary in size and aspect ratio.

Due to perspective, different ROIs framing a pedestrian can have different sizes, so normalization is required to establish an equivalence between the features computed in each ROI. To achieve that, features are computed following the proposal in [8], obtaining results equivalent to normalizing examples aspect ratio to fit an area of 12×24 pixels (Fig. 3($right$)), which in our acquisition system corresponds to a *standard* pedestrian at about 50m.

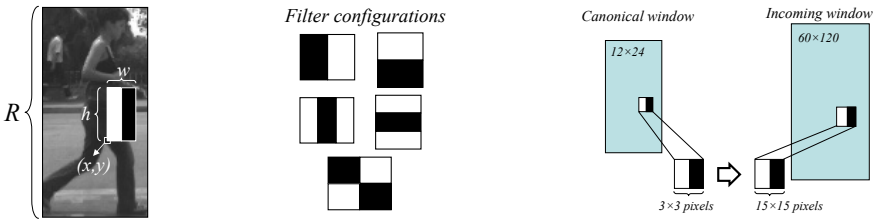

Fig. 3. Computation of Haar wavelet features: ($left$) Haar feature placed in a sample image; ($middle$) Some filter configurations; ($right$) Filter normalization according to the incoming ROI size.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Edge orientation histograms. Edge orientation histograms (EOH)[1] are proposed by *Levi and Weiss* for face detection in [9]. They rely on the richness of edge information, so they differ from the intensity area differences of Haar Wavelets but maintain the same invariance properties to global illumination changes.

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 classified into β images (in our case we have tested $\beta = \{4, 6, 9\}$) corresponding to β orientation ranges (also referred as *bins*). Therefore, a pixel in bin $k_n \in \beta$ contains its gradient magnitude if its orientation is inside β_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 a bin interpolation step has been included in order to distribute the gradient value into adjacent bins. This step is used in SIFT [10] and HOG [5] features, and in our case it improves the EOH performance in 1% at 0.01 False Positive Rate (FPR). Finally, the feature value is defined as the relation between two orientations, k_1 and k_2, of region R as:

$$\text{Feature}_{EOH}(x, y, w, h, k_1, k_2, R) = \frac{E_{k_1}(R)+\epsilon}{E_{k_2}(R)+\epsilon}$$

If this value is above a given threshold, it can be said that orientation k_1 is dominant to orientation k_2 for R. The small value ϵ is added to the factors for smoothing purposes.

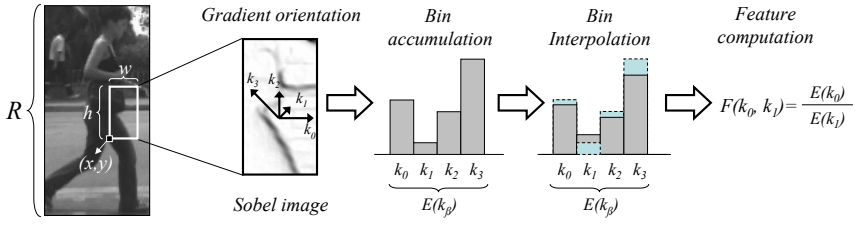

Fig. 4. Computation of edge orientation histograms.

3.2 Classifier learning

We make use of Real AdaBoost [11] as learning machine, of proven usefulness in similar classification works [8]. The idea is to build a *strong* classifier by combining the response of a set of *weak* classifiers, improving the performance that a complex classifier would have alone. In our case, since both HW and EOH features are represented by a real value, each weak classifier corresponds to threshold–like rule on each feature value.

[1] In order to respect the author's work, in this paper we maintain the original name. However, since this can lead to confusion with other similar feature names like the *histograms of oriented gradients* (HOG) in [5], we think that a more convenient name would be *ratios of gradient orientations*.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

4 Results

We illustrate camera pose estimation (represented in a single value as the horizon line position) from a fragment of a video sequence. Fig. 5($left$) shows a single frame with iis corresponding horizon line directly computed from the camera pose estimation. Fig. 5($right$) presents a comparison with [4] where similar results can be appreciated; however, it should be noticed that the current approach is about four times faster than the previous one [4]—i.e., on average, the new proposal took 90 ms per frame.

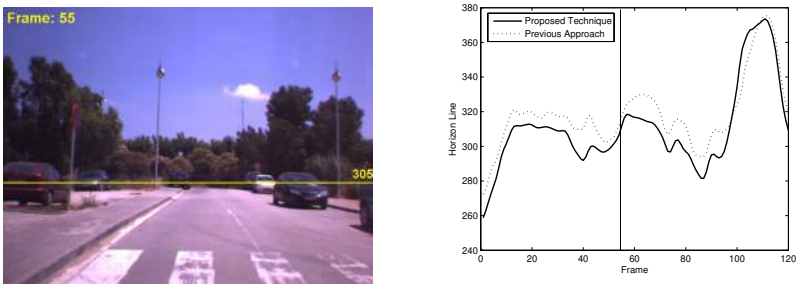

Fig. 5. ($left$) Horizon line position for a given frame. ($right$) Comparison between the proposed camera pose technique and a previous one [4]. The plot also shows the large variability of the horizon line at urban scenarios.

In order to illustrate the performance of the classifier we have built a pedestrian database. Differently to other non ADAS–oriented databases [5], it contains images at different scales from urban scenarios. In our case, color information is discarded as an useful cue, so samples are transformed to grayscale. The complete database consists of $1,000$ positive samples (i.e., pedestrians; Fig. 6) and $5,000$ negative ones (i.e., ROIs fulfilling the PSC but not containing pedestrians). Each experiment randomly selects 700 positive and $4,000$ negative samples (training set) to learn a model, and use the remaining (testing set) to measure the classifier performance. All performance rates and plots are the result of averaging 4 independent experiments.

The proposed classifier is compared with, as far as we are concerned, the current state–of–the–art best classifier for pedestrian detection, which uses *histograms of oriented gradients* (HOG) features and support vector machine (SVM) learning, proposed by *Dalal and Triggs* in [5]. 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

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

13×26	18×36	26×52	41×82	74×148	98×196	103×206
$(53m)$	$(38m)$	$(26m)$	$(16m)$	$(9m)$	$(7m)$	$(6m)$

Fig. 6. Some positive samples of the database illustrating the high variability in terms of clothes, pose, illumination, background, and sizes. Below each sample it is noted its size in the image (in pixels) and the real distance to the camera.

comparison with our proposal. As the authors suggest, no smoothing is aplied 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 ($\beta = \{4, 6, 9\}$[2] in $0 - 180°$), cell sizes ($\eta = \{1 \times 1, 2 \times 2, 3 \times 3\}$ pixels) and block sizes ($\varsigma = \{1 \times 1, 2 \times 2, 3 \times 3\}$ cells), for our 24×12 cannonical windows. As a last step, the block histogram is normalized using *L2-Hys*, the best method in the original paper. Finally, the features are fed to a linear SVM (we have also used SVMLight[3]) with $C = 0.01$. According to Fig. 7($left$), the optimum parameters are $\beta = 9$, $\eta = 2 \times 2$ and $\varsigma = 2 \times 2$, which provide a Detection Rate (DR) of 0.925 at a false positive rate (FPR) of 0.01.

Regarding our proposal, we have also made tests with $\beta = \{4, 6, 9\}$ for EOH, with very similar results. Hence, we bet for the $\beta = 4$ bins version since it requires less computation time. Fig. 7($right$) presents a comparison between our proposal and the HOG–based classifier. As can be seen, with 100 features (i.e., 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 FPR=0.01), and it is computed about two times faster than HOG–based classifier.

Our preliminary tests combining the proposed camera pose estimation and classifier are giving satisfactory results in complex real scenes (Fig. 8). However, since the classifier is built to be tolerant to some amount of displacement each pedestrian gives rise to several detections. Currently, we are developing a method to take the best detection from such multiple ones for further quantitative performance evaluation of the whole system (i.e., in terms of DR and FPR at system level, not only at the classification level).

[2] 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.

[3] `http://svmlight.joachims.org`

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

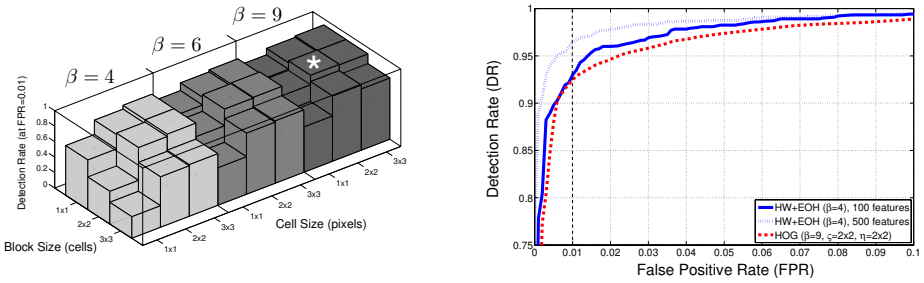

Fig. 7. (*left*) Detection rate at FPR=0.01 for all possible configurations of β, η and ς of HOG features (the best one is marked with a star). (*right*) Comparison between our proposed classifier and the best HOG–based classifier.

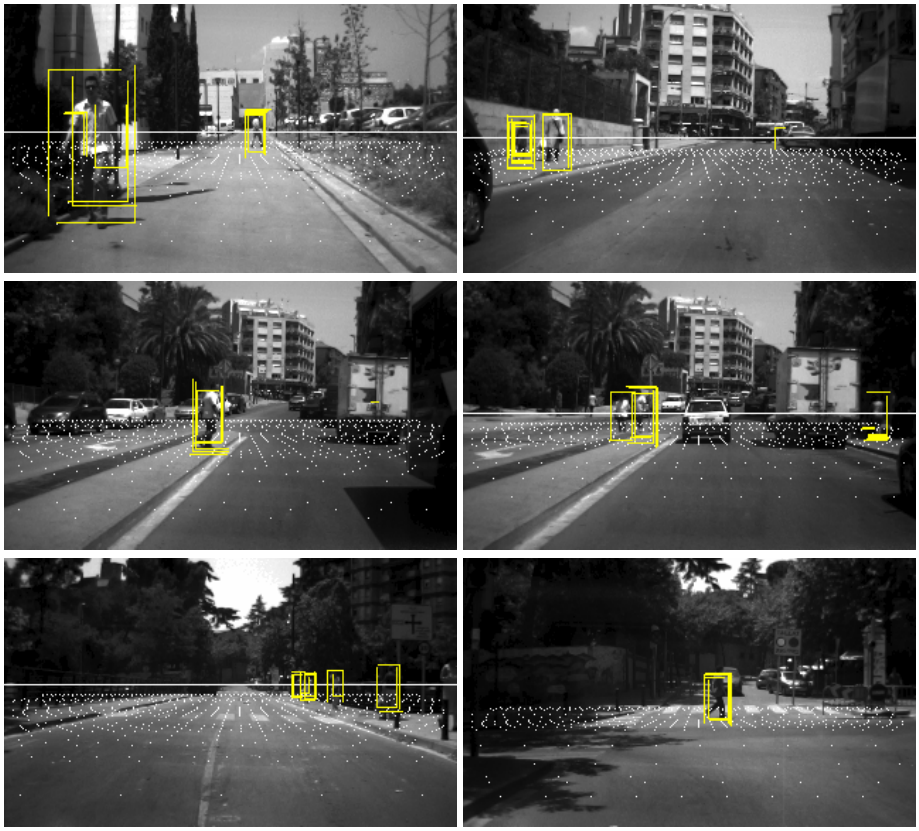

Fig. 8. Some snapshots of the system working in real scenes. The white horizontal line represents the estimated horizon; the white dots compose the ground sampling grid (adjusted frame–by–frame according to the horizon) and the yellow boxes represent positive detections (no postprocessing was applied to filter out multiple detections).

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

5 Conclusions

This paper presents improvements in two crucial steps of pedestrian detection: (1) an adaptive image sampling method based on camera pose estimation is described, also useful for other ADAS tasks as vehicle detection and road segmentation; (2) a pedestrian classifier based on fast–to–compute features, namely Haar wavelets and edge orientation histograms, and Real AdaBoost as learning machine, is presented too. In both cases we have compared our proposal with other relevant methods in the literature showing that we obtain comparable results but with a considerable reduction in processing time (four times faster for camera pose estimation and ten times for classification).

References

1. Papageorgiou, C., Poggio, T.: A trainable system for object detection. International Journal on Computer Vision **38**(1) (2000) 15–33
2. Gavrila, D., Giebel, J., Munder, S.: Vision–based pedestrian detection: The PROTECTOR system. In: Proc. of the IEEE Intelligent Vehicles Symposium , Parma, Italy (2004)
3. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian detection for driving assistance systems: Single–frame classification and system level performance. In: Proc. of the IEEE Intelligent Vehicles Symposium , Parma, Italy (2004)
4. Sappa, A., Gerónimo, D., Dornaika, F., López, A.: On-board camera extrinsic parameter estimation. Electronics Letters **42**(13) (2006) 745–747
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition . Volume 2., San Diego, CA, USA (2005) 886–893
6. Labayrade, R., Aubert, D.: A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision. In: Proc. of the IEEE Intelligent Vehicles Symposium , Columbus, OH, USA. (2003) 31–36
7. Fischler, M., Bolles, R.: 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
8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition , Kauai Marriott, USA (2001)
9. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition , Washington DC, USA (2004) 53–60
10. Lowe, D.: Distinctive image features from scale–invariant keypoints. International Journal on Computer Vision **60**(2) (2004) 91–110
11. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence–rated predictions. Machine Learning **37**(3) (1999) 297–336

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8