Part-based Pedestrian Detection and Feature-based Tracking for Driver Assistance

Real-Time, Robust Algorithms and Evaluation

Prioletti, Antonio; Møgelmose, Andreas; Grislieri, Paolo; Trivedi, Mohan; Broggi, Alberto; Moeslund, Thomas B.

Published in:
I E E E Transactions on Intelligent Transportation Systems

DOI (link to publication from Publisher):
10.1109/TITS.2013.2262045

Publication date:
2013

Document Version
Early version, also known as pre-print

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Part-Based Pedestrian Detection and Feature-Based Tracking for Driver Assistance: Real-Time, Robust Algorithms, and Evaluation

Antonio Prioletti, *Student Member, IEEE*, Andreas Møgelmose, *Student Member, IEEE*, Paolo Grisleri, Mohan Manubhai Trivedi, *Fellow, IEEE*, Alberto Broggi, *Senior Member, IEEE*, and Thomas B. Moeslund, *Member, IEEE*

***Abstract*—Detecting pedestrians is still a challenging task for automotive vision systems due to the extreme variability of targets, lighting conditions, occlusion, and high-speed vehicle motion. Much research has been focused on this problem in the last ten years and detectors based on classifiers have gained a special place among the different approaches presented. This paper presents a state-of-the-art pedestrian detection system based on a two-stage classifier. Candidates are extracted with a Haar cascade classifier trained with the Daimler Detection Benchmark data set and then validated through a part-based histogram-of-oriented-gradient (HOG) classifier with the aim of lowering the number of false positives. The surviving candidates are then filtered with feature-based tracking to enhance the recognition robustness and improve the results' stability. The system has been implemented on a prototype vehicle and offers high performance in terms of several metrics, such as detection rate, false positives per hour, and frame rate. The novelty of this system relies on the combination of a HOG part-based approach, tracking based on a specific optimized feature, and porting on a real prototype.**

***Index Terms*—Advanced driver assistance system (ADAS), classifiers, features, machine vision, pedestrian detection.**

I. Introduction

OVER the past decade, the essential role of machine vision modules to realize active safety systems for accident prevention is clearly established in academic research [1], [2] and is also reflected in innovative systems introduced by industry [3], [4]. Effective vision systems need to accurately assess situational criticalities from the panoramic surround of a vehicle [5] and simultaneously assess awareness of these criticalities by the driver [6]. One of the major thrusts in situational criticality assessment is that of pedestrian detection, and it still remains an active area of research [7]–[15]. Pedestrian detection has multiple uses, with the most prominent being advanced driver assistance systems (ADASs). The overarching goal is to equip vehicles with sensing capabilities to detect and act on pedestrians in dangerous situations, where the driver would not be able to avoid a collision. A full ADAS with regard to pedestrians would as such not only include detection but also tracking, orientation, intent analysis, and collision prediction.

Pedestrian detection brings many challenges, as outlined by [8]: high variability in appearance among pedestrians, cluttered backgrounds, high dynamic scenes with both pedestrian and camera motion, and strict requirements in both speed and reliability. It follows from this list that there is a high risk of occlusion, and this occlusion might not be present for very long since all objects in the scene are moving relatively to each other. Part-based detection systems seem intuitive to cope well with occlusion as they do not necessarily require the full body to be present to make detection. In addition, many existing systems (see Section II) are plagued by a high false positive per frame (FPPF), something that a part-based system can reduce if requirements of several body parts to be detected are put in place. These two motivations for part-based detection can be somewhat contradictory. Narrowing the classification parameters will reduce the number of false positives but, likewise, the number of true positives. A tracking technique can be introduced to supply missing detection and, thus, counteract this tradeoff.

This paper builds on a part-based staged detection approach (PPD), which was first put forth in [9], providing four major contributions:

1) a thorough analysis of the impact of changes in parameters for this algorithm that goes far beyond what was presented in the initial study;
2) an expansion of the system to a full-fledged ADAS, not just a detection algorithm, and a discussion of the requirements put upon the full system from such an application;
3) the use of more pedestrian-related training and test sets, where the original paper used the INRIA data set [11], which is a more general-purpose person data set;
4) porting of the system to a real prototype vehicle and analysis of critical situations in a real environment, optimizing the system to improve detection and speed performance.

Manuscript received November 12, 2012; revised January 25, 2013 and March 23, 2013; accepted April 22, 2013. This work was supported by Cassa di Risparmio di Parma e Piacenza. The Associate Editor for this paper was Q. Kong.

A. Prioletti, P. Grisleri, and A. Broggi are with the Artificial Vision and Intelligent Systems Laboratory, University of Parma, Parma 43124, Italy (e-mail: antonio.prioletti@studenti.unipr.it).

A. Møgelmose and T. B. Moeslund are with the Visual Analysis of People Laboratory, Aalborg University, Aalborg 9200, Denmark (e-mail: am@create.aau.dk).

M. M. Trivedi is with the Computer Vision and Robotics Research Laboratory, University of California, San Diego, CA 92093-0434 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2013.2262045

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

One of the innovations of this system is the use of histogram-of-oriented-gradient (HOG) features in a PPD; moreover, an optimized kind of a feature has been adopted to decrease as much as possible the computational time; this helps when testing the system on a real prototype. Given the reaction speed of a human, it is clear that a braking assistance system can help in reducing braking distances.

The ADAS is a challenging domain to work within. Reaction times must be fast for driving, where a fraction of a second can be the deciding factor between a collision and a near miss. At the same time, the system must be robust; therefore, no action is erroneously triggered (due to a false detection), which could itself lead to accidents. Further reasoning than just detection is necessary in such a framework, with pedestrian intent estimation being a good example, as presented in [12], or automatic braking, as in [13].

This paper contains an overview of related works in Section II, a description of the implemented pedestrian detection ADAS in Section III, and details of the algorithmic stages in Section III-A–C. A thorough set of experiments follows in Section IV, where the impact of parameter adjustments in the system is investigated. Section V describes the porting of the system to a real prototype car, and Section VI presents a final evaluation of the performance, in comparison with the state of the art of the vision-based detectors, with the full-body approach by Geismann *et al.* and with the final system after the implementation on a real platform [14].

II. Related Works

The purpose of pedestrian detection is first and foremost to protect pedestrians. Pedestrian safety is a large area, including passive solutions, such as car design, and active solutions, such as pedestrian detection. It also involves infrastructure design to a great extent. In [15], a survey of the pedestrian detection field and a taxonomy of the involved system types are provided. Many standard features and learning algorithms have been adapted to pedestrian detection. Common options include an AdaBoost cascade on Haar-like features [16], [17] or HOG+SVM [11], [18], but many other features are also used, such as edgelets [19], variations of gradient maps, or simple intensity images. The cascade classifier based on Haar-like features, which is described in [16], is a very fast algorithm for pedestrian detection. A drawback of this approach is the close link with the appearance of pedestrians and the resulting lack of robustness. An alternative is the solution using HOG and support vector machines (SVMs) presented in [11]. At the cost of speed, this algorithm is much more robust and detects pedestrians in harder situations. The combination of these two algorithms allows the system to benefit from both approaches and obtain a robust system with considerable speedup.

Decomposing the pedestrian shape into parts is gaining great interest in this area, particularly for increased tolerance of occlusion. Interesting dilemmas are how many and which parts of pedestrians to use, and how to integrate all the part-based detectors in a final detector; an example is shown in [20] where, in the first stage, head, arm, and leg detectors were trained in a fully supervised manner and are then combined to fit a rough geometric model . Other two-stage approaches are shown in [21] and [22]. Several feature types and different environment kinds can be used. In [23], a system is developed based on Viola's Adaboost cascade framework, using edgelet features in addition to Haar-like features, to improve the detection of the pedestrians contour; moreover, the concept of interfering objects is introduced, i.e., objects similar to a human body on a feature level. Before detecting pedestrians, they remove this type of an object. In [19], multiple part detectors based on edgelets are combined to form a joint likelihood model that includes cases of multiple possibly interoccluded humans. Due to the high difficulty of detecting interest points at low resolutions, unsupervised part-based approaches that do not rely on key points have been proposed. An example is multiple-instance learning, which determines the position of parts without part-level supervision [24]. In [25], one of the most successful PPD that models unknown part positions as latent variables in an SVM framework is proposed. In [26], this method switching to a part-based system only at sufficiently high resolutions is improved. Detecting highly variable objects, such as pedestrians, is essentially the use of a tracking module. Tracking a variable number of elements in complex scenes is a challenging process. To cope with this kind of problem, a tracking-by-detection approach is commonly [19], [27] used, i.e., pedestrians are detected in individual frames and then associated between frames. The main challenge to this regards discontinuous detection in conjunction with possible false positives and missing detection; this problem makes use of a Kalman filter hard, due to the continuous detection that it needs to give accurate results. Several multiobject tracking systems [28], [29], such as our system, use a large temporal window to make the association; in this way, a pedestrian not detected in two subsequent frames but in more frames can be also included in the tracking system with a temporal delay. Another interesting approach that can be investigated in the future is to represent the uncertainty of a tracking system with a particle filter [30] in a Markovian manner. Using a stereo-based approach is possible to reduce the searching area and, consequently, the elaboration time, as described in [31] and [32]. Examples of detection that is not based on images but instead on time-of-flight (TOF), such as radars and lidars, are put forth in [33]–[35]. These systems very often combine the TOF sensor with a camera as in [13], with a combination of a near-infrared camera and a lidar. Furthermore, they use a scenario-driven search approach where they only look for pedestrians in relevant areas. Further reading on pedestrian protection systems can be found in [15], and comprehensive surveys on vision-based detection systems are found in [7], [8], [36], and [37].

A. Public Data Sets

Several data sets are publicly available. The two best known are the Massachusetts Institute of Technology data set [38] and the INRIA data set [11]. Recently, more comprehensive data sets have been put forth. These include the ETH [39], TUD-Brussels [40], Caltech [41], and Daimler Detection Benchmark (DaimlerDB) [36] pedestrian data sets. Note that the DaimlerDB set should not be confused with the older and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
KEY STATISTICS ABOUT MAJOR PUBLIC PEDESTRIAN DATA SETS, COURTESY OF DOLLÁR *ET AL.* [7]

	Training			Testing			Pedestrian height		
	# pedestrians	# negative images	# positive images	# pedestrians	# negative images	# positive images	10% percentile	median	90% percentile
MIT	924	-	-	-	-	-	128	128	128
INRIA	1208	1218	614	566	453	288	139	279	456
ETH	2388	-	499	12k	-	1804	50	90	189
TUD-Brussels	1776	218	1092	1498	-	508	40	66	112
Daimler-DB	15.6k	6.7k	-	56.5k	-	21.8k	21	47	84
Caltech	192k	61k	67k	155k	56k	65k	27	48	97

smaller Daimler Classification Benchmark, which is often wrongly abbreviated as DaimlerDB. Key statistics about the data sets are presented in Table I and also presented in [7]. While the INRIA data set was used in the first presentation of this system [9], this paper deals mainly with the DaimlerDB since that is a much larger data set created with focus on in-car detection systems. All testing is done against the DaimlerDB (see Section IV for further details), and we compare the training with the DaimlerDB and the INRIA data set.

B. *Performance of the State of the Art*

To know what the performance target for a vision-based system is, we turn to the evaluation of the state-of-the-art performance in [7]. Two results are interesting: the detection rate versus the false positives per frame (FPPF) and the detection speed (frame rate). As this paper uses the DaimlerDB pedestrian data set, we compare our performance with the state-of-the-art detectors on this database, as reported in [7]. Ten different systems have been tested on the data set and detection rates are available at a false positive rate of 0.1 FPPF. The results are shown in Table II. Apart from the ten systems that were tested on the DaimlerDB data set, we have included the fastest detector of all. No detection results were reported for this detector on the DaimlerDB, but on the other sets, it achieved detection rates of around 0.4.

TABLE II
DETECTION RATES AND SPEEDS FOR STATE-OF-THE-ART PEDESTRIAN DETECTION SYSTEMS AT 0.5 FPPF ON THE DAIMLERDB DATA SET, COURTESY OF DOLLÁR *ET AL.* [7]. THE PAPER CONTAINS AN EXPLANATION OF EACH OF THE SYSTEMS. THESE PERFORMANCES ARE DIRECTLY COMPARABLE WITH THE RESULTS OBTAINED IN THIS PAPER. THE FASTEST SYSTEM IS ALSO LISTED, ALTHOUGH DETECTION RATES FOR THE DC DATA SET ARE UNKNOWN. ABBREVIATIONS ARE THE SAME WITH THAT DESCRIBED IN [7]

Algorithm	Part-based	Detection rate	Speed
MultiFtr+Motion	no	0.75	0.004 FPS
LatSvm-V2	yes	0.69	0.164 FPS
MultiFtr+CSS	no	0.68	0.005 FPS
HogLbp	no	0.59	0.014 FPS
HikSvm	no	0.5	0.036 FPS
MultiFtr	no	0.49	0.017 FPS
LatSvm-V1	yes	0.48	0.098 FPS
HOG	no	0.42	0.054 FPS
Shapelet	no	0.10	0.010 FPS
VJ	no	0.09	0.089 FPS
FPDW	no	N/A for DC dataset	2.670 FPS

III. PART-BASED PEDESTRIAN DETECTION SYSTEM OVERVIEW

A two-stage system based on the combination of Haar cascade classifier and a novel part-based HOG–SVM will be presented here; an innovative features-based pedestrian tracking approach will be also described.

A monocular vision system is used since a simple onboard camera is present in many new high-end cars already. A Haar detector is used to reduce the region of interest (ROI) (*detection stage*), providing candidate pedestrians to the HOG detector, which classifies the windows as pedestrians or nonpedestrians (*verification stage*). To increase the robustness of the system and reduce the number of false positives, a PPD is used in the verification stage. The full body, the upper body, and the lower body are each verified using an SVM. These three results are then combined to obtain the final response for the ROI. Two ways were investigated to combine results in the verification stage:

- a simple majority vote, where at least two of three SVMs must classify the window as a pedestrian;
- a more advanced way, where another SVM classifies the window based on the estimated function value from an SVM regression performed on each part.

Due to the high variability in pedestrian appearance, a robust system with strict thresholds for detection may not detect the same pedestrian in subsequent frames and, thus, reduce the detection rate considerably. To counter this, a stage of feature-based tracking was introduced, significantly increasing the number of true positives.

A. *Detection Stage*

An AdaBoost cascade on Haar features is used in the detection stage. Several weak classifiers are combined into a strong classifier; the final classifier is formed with the combination of

Fig. 1. Different bounding boxes required by Haar cascade and HOG–SVM. The base image is from the DaimlerDB data set [36]. The red dashed line is the Haar bounding box and the blue continuous line is the HOG bounding box.

several layers of these strong classifiers. The cascade structure removes most false positives in the first stages, increasing the speed of the classifier and not having to calculate these in the following stages. In the following, we denote the number of cascade stages as k. In [42], a comprehensive description of the algorithm is presented. Unlike HOG features, Haar-like features do not benefit from having much background included. Training images need to be closely cropped around the annotated human shape (e.g., see Fig. 1). Following the suggestions in [42] about the optimal image size for the Haar cascade approach, the training images are resized to 20×40 pixels. Another interesting element in the training phase is the choice of data sets used to train the cascade classifier. Most of the older systems were trained with the INRIA data set, containing general environments and not specifically pedestrians. To show how the change results with different training data sets, the system has been trained with the INRIA data set alone, with the DaimlerDB data set alone, and with a combination of the two sets.

Since the detection stage defines the upper bound of detection for the entire system, it is fundamental to choose the best value for the number of the stages. A lower value of k means not only a high detection rate but also a high number of false positives. Initially, it might seem logical to choose the number of stages as low as possible, to ensure a high number of detections. That will, however, result in inaccurate bounding boxes (and many of them), as shown in Fig. 2; thus, the final results will be incorrect. The PPD was not introduced to the detection stage, as preliminary tests and the work in [43] showed a bad performance for this approach. When the bounding boxes of candidate pedestrians (e.g., see Fig. 3) have been obtained, they are passed to the verification stage.

Fig. 2. Example of the degradation of the bounding box varying k from 13 in the last pictures to nine in the second picture and to eight in the first picture.

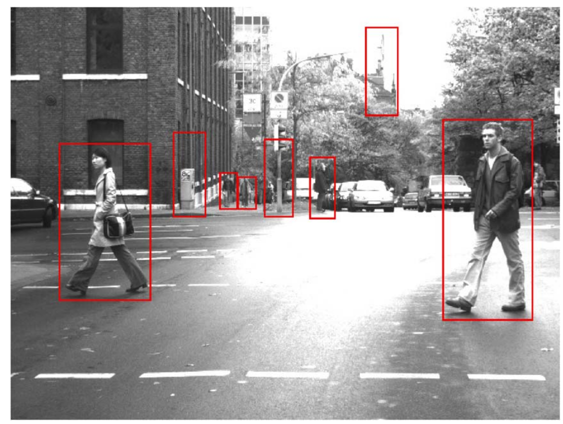


Fig. 3. Detection stage output. Several false positives are contained, but these will be removed in the verification stage.

B. Part Verification Stage

As opposed to the full-body verification stage in [14], a PPD scheme is used in this paper. Two different compositions of body parts have been tested:

- a full body, an upper body, and a lower body;
- a full body, a head, a torso, and legs.

A fixed ratio between them have been used. The upper body and the lower body are obtained by dividing the shape into two equal parts. When we split the shape into three parts, instead, it was assumed a ratio of 16% for head and neck and 34% for torso, whereas legs are considered to occupy 50% of the entire body. These numbers are taken from standard human body ratios. Before passing the ROIs to the SVMs, preprocessing to add background and to resize the image is needed to ensure good performance by HOG–SVMs, which takes some background into account. Then, the individual part verification and the combined verification form the verification stage. SVM regression based on dense HOG descriptors is calculated for each part in the ROIs given by the detection stage. Two different types of SVMs were tested: a linear SVM and a nonlinear SVM. Each was tested in two variants, i.e., a binary SVM or a regression SVM. The binary SVM provides only the classification (pedestrian or nonpedestrian) of the element; the regression SVM provides the estimated function value. In [14], a special kind of sparse HOG descriptors is used, whereas our algorithm uses classic dense HOG descriptors. Integral images were used to speed up the descriptor calculation, as described in [44]. For SVM training, images from several data sets were tested with the goal of analyzing the effects of training sets in

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 4. Example of part boundaries for the two-part and three-part verification.

the verification stage. The process of training the SVMs for the different parts of the body are almost identical; the only changes being the portion of images used to calculate the HOG features. Examples of parts are shown in Fig. 4.

C. Combined Verification Stage

For this last stage, two different approaches have been implemented: *majority vote* and *regression output classification*. The majority vote approach performs the final labeling without further classifiers, and the regression output classification uses one more classifier to label the window. There is a philosophical difference between the voting-based combination methods and the others. Voting-based combination requires only a subset of body parts to be visible and detectable and can deal well with occlusion. The other requires all body parts to be visible, at least to some extent; therefore, they will handle occlusion somewhat worse but reduce the number of false positives. A possible compromise is to use occluded pedestrians in the data set, training the classifier to detect pedestrians partially visible; obviously, this also means an increase in FPPF.

The majority vote approach uses the binary outputs from the SVM. The value will be 1 if the classifier detects the specific part of the body or -1 if the part is not detected. A window is classified as correct detection if at least two out of three classifiers label the window as a pedestrian. The formula used for the majority voting is

$$l_{\mathrm{out}} = \begin{cases} 1, & \text{if } \sum_{i=0}^{i<3} l_i >= 1 \\ -1, & \text{if } \sum_{i=0}^{i<3} l_i < 1 \end{cases} \tag{1}$$

where l_{out} is the final decision, and l_i is the output from one of the three part-based detectors.

Regression output classification uses the three-float value coming from SVMs of the verification stage to train a new classifier. Several types of classifiers were tested: a linear SVM, a nonlinear SVM, and a Bayesian classifier; in the results, the different performances of each one will be shown.

D. Tracking Stage

A feature-based tracking was used to enhance the detection rate. The tracker is introduced to increase the number of true positives due to the higher stability of the detection in the case of, for example, occlusion, and to decrease the number of false positives since only the stable detection will be considered pedestrians. The core of the tracking system is the feature matcher, using the matching approach in [45]. The tracker labels pedestrians to supply possible missing detection due to mistakes of the classifier in the verification stage; a more detailed description of the tracking is presented in Section V. An overview of the flow through the algorithm is shown in Fig. 5.

IV. EXPERIMENTS

One of the main contributions of this paper is a thorough evaluation of the algorithm's parameters. Here, we describe the experiments to determine the best detector, which is then quantitatively and qualitatively tested in Section V. DaimlerDB was primarily used, with elements from the INRIA data set in a few tests. Unless otherwise specified, images from the training part of DaimlerDB was used for training, i.e., both the detection stage and the part verification stage. The test part of DaimlerDB was split into two.

- One portion of 1500 images was used for the parameter optimization here.
- One portion of 500 images was used for the final test presented in Section V.

This ensures that the final performance measures are fully independent of the training images. The experiments are laid out as follows.

1) The best detection stage training is determined, and then, the optimal value of k in the detection stage is decided.
2) The part-based verification is tackled with a comparison of the two-part and three-part approaches. They are compared with a simple detector without a part, similar to the original version of the algorithm proposed by Geismann *et al.* Furthermore, the significance of each part is evaluated.
3) The combined verification stage is tested with various methods.
4) The system speed is tested, and the time is broken down into individual stages.

A. PASCAL Detection Evaluation

For all the following experiments, the PASCAL measure [46] has been used to determine the detection rates. This is also used in [7]; therefore, the results should be directly comparable. The PASCAL measure evaluates to true if the overlap is more than 50%, i.e.,

$$a_o \equiv \frac{\mathrm{area}(BB_{\mathrm{dt}} \cap BB_{\mathrm{gt}})}{\mathrm{area}(BB_{\mathrm{dt}} \cup BB_{\mathrm{gt}})} > 0.5 \tag{2}$$

where BB_{dt} and BB_{gt} are the bounding boxes of the detection and the bounding box of the ground truth, respectively. Each detection is compared with the ground truth of the 1500 images and is counted as a true positive if a_o is true and as a false positive, otherwise. All tests in the following are run

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

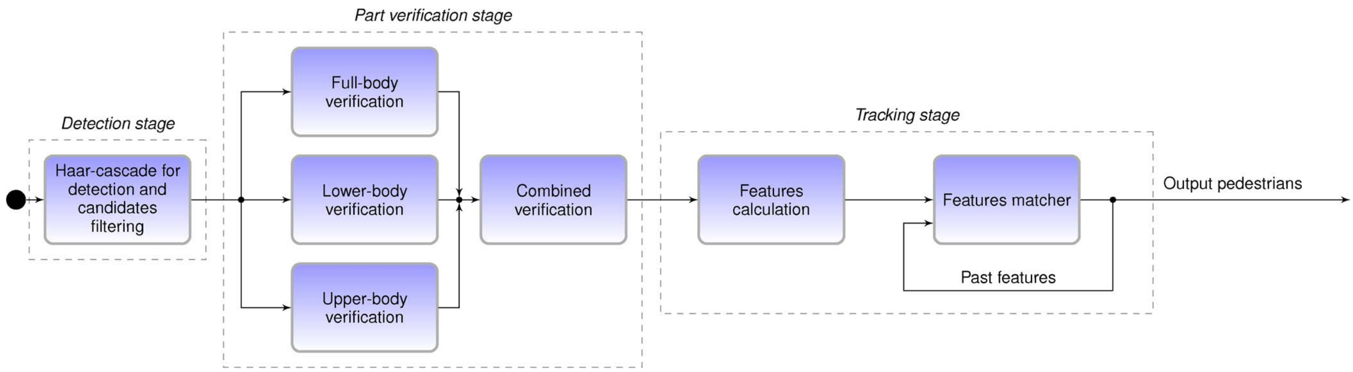


Fig. 5. Flowchart for the proposed PPD and feature-based tracking modules. The output of the detection stage and the following stages are in bounding boxes.

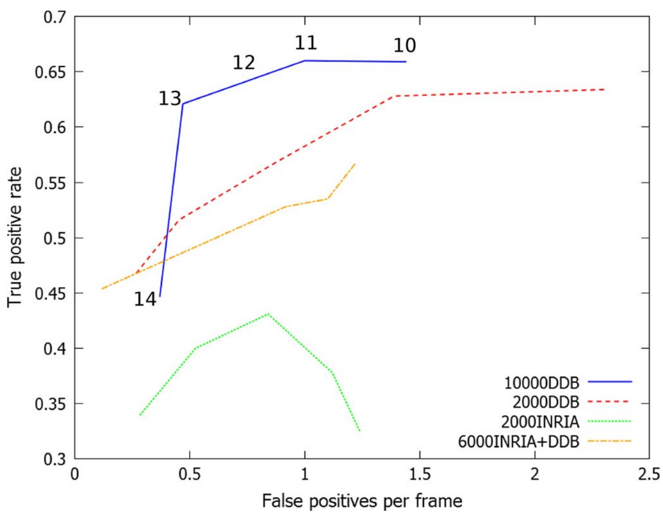


Fig. 6. Comparison of different training sets for the detection stage. The system trained with the DaimlerDB data set performs significantly better, remarking the excessive generality of INRIA. Chosen for having the best training sets, it was analyzed, for the system training with this data set, with the best value of k . As described in Section IV-C, 13 is the best value, obtaining a good tradeoff between true positives and false positives.

on the complete system. For each test, all parameters are held fixed, except for the one in question. Thus, the results cannot necessarily be compared across tests, but the results are always comparable relative to each other within the tests.

B. Training of the Detection Stage

This test pitted different training setups of the Haar cascade. Four versions were tested:

- 2400 DaimlerDB images;
- 2400 INRIA images;
- 6000 images composed of 2400 INRIA and 3600 DaimlerDB images;
- 10 000 DaimlerDB images.

The results are presented in Fig. 6 and show that performance is improved using more images. Fig. 6 is a receiver operating characteristic (ROC) curve created by plotting the fraction of true positives out of the positives (tpr $=$ true positive rate) versus the fraction of false positives out of the negatives (fpr $=$ false positive rate), at various threshold settings. Note the bad performance of the system when trained with the INRIA data sets; this show how the INRIA are too general, being developed for the human detection. The big influence of this kind of data set is also clearly visible in the system trained with 4000 DaimlerDB images and 2000 INRIA images; the system with less images (2000 DaimlerDB), but only from the DaimlerDB data set, performs better than this one with more images.

C. Choice of k in the Detection Stage

This test determines how many stages k the Haar cascade should have. As there are two verification stages after this, the detection stage should be tweaked so that it returns as many true positives as possible, whereas the number of false positives is less important; they will be removed later. Still, there is a point where raising the number of false positives does not provide a better detection performance; therefore, the only effect will be a slowdown of the system since more ROIs must be inspected by the verification stages. Fig. 6 shows the ROC curve for different values of k. Few stages should mean raise the number of both false positives and true positives, but at some point, the quality of the bounding boxes provided by the detection stages degrade to a level where the verification stage only verify a few candidates.

D. Part Verification Padding

Padding p is the amount of area added to the ROIs returned by the detection stage. The HOG–SVM approach is sensitive to the amount of free space around the subject as described in [11]; therefore, the parameter is relevant for optimization. An example of padding is shown in Fig. 3, where the bounding box for the Haar cascade is much closer to the subject than the rest. We express p as a fraction of the width of the ROI found by the detection stage, i.e.,

$$p_{\text{pixels}} = \frac{w_{\text{ROI}}}{w_t} \cdot p \tag{3}$$

where p is the padding value, w_{ROI} is the width of the found ROI, w_t is the width of the training images, and p_{pixels} is the padding measured in pixels. Fig. 7 shows the performance of different padding values. It is evident how less padding means worse images to the verification stage. At the same time, too much padding makes the verification more difficult for the HOG detector since more items are analyzed and more mistakes happen.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

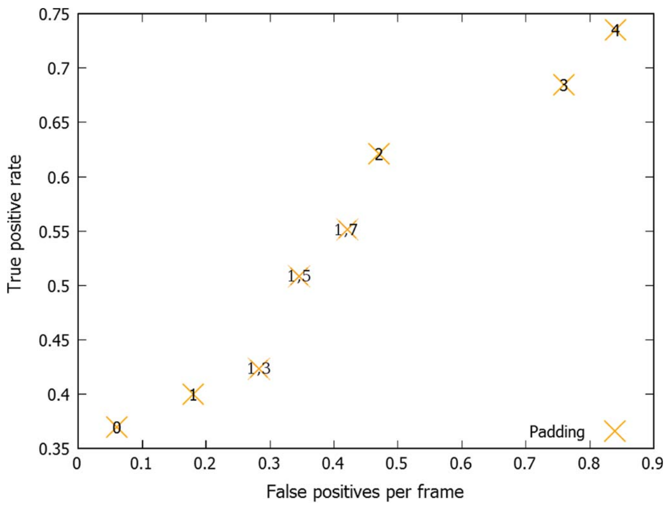


Fig. 7. Choosing the padding values to be put on the ROIs from the detection stage before passing them to the verification. The DaimlerDB training set with 10 000 images and a k value of 13 were used.

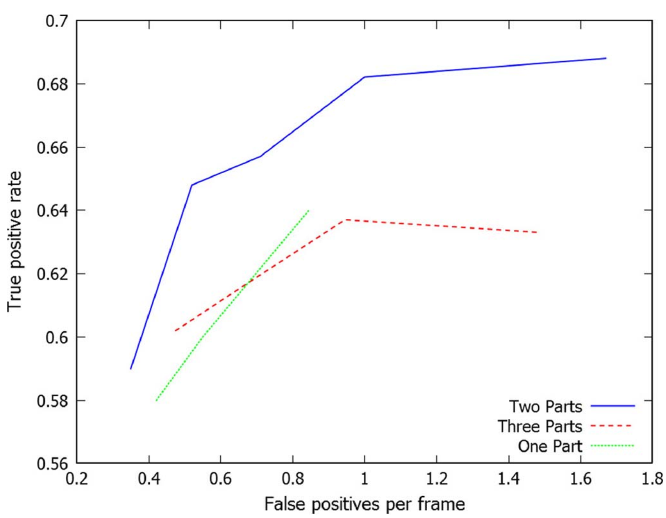


Fig. 8. Detection performance with varying numbers of parts. Note that two-part verification performs well, whereas three-part verification is just as bad as one-part verification due to the low quality of the images and the high difficulty in identifying small areas, such as the head. Considering the results shown in the previous charts, the DaimlerDB training set with 10 000 images, a k value of 13, and a padding value of 2 were used.

E. Number of Parts

The performance of one-, two-, and three-part verification is compared (with one-part verification obviously not being part-based at all). Illustrations of the part boundaries for both the two- and three-part detectors are shown in Fig. 4. The performance of various part numbers is shown in Fig. 8.

Two-part verification is the best choice and three-part verification performs better than having one-part verification at the lower FPPF. These results can be attributed to the quality of the images; the three-part detector needs to detect the head, which is a comparatively small element and too hard to detect in an image with low resolution. With higher resolution images, it is likely that the three-part approach would provide the best results, but at the same time, the speed of the system would suffer.

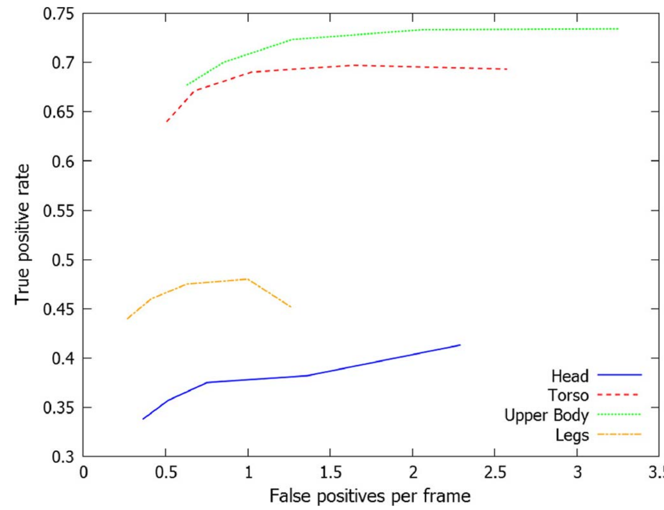


Fig. 9. Detection performance with single parts, showing the reliability of each part type. The graph confirms the assumptions regarding the difficulty to detect the head. The same configuration parameter system of the last pictures was used in this experiment.

In connection with this, an analysis of the significance of each part was done. The results show how the detection performance would be, relying on that specific part only. Four parts have been tested: a lower body, an upper body, a head, and a torso. The lower body is used both for the two- and three-part verification, whereas the upper body is only used for the two-part verification, and the head and torso are used for the three-part verification. Results of this analysis are shown in Fig. 9. None of the parts alone perform better than a unified detector, but the upper body and torso provide the major contribution to the detection. These results support the hypothesis that the three-part verification has a worse performance than the two-part verification, i.e., due to the low resolution for the head detection. In this figure, the head detection system is the worst, with a very low detection rate. The combination of the upper body and the legs/lower body is the best combination due to the high detection rate from the upper body and the reduction in false positives provided by the lower body.

F. Combined Verification Step

For the final combined verification step, four options have been investigated: the linear SVM, the radial SVM, and the Bayesian classification for confidence classification and majority vote based on the discrete classification from the part verifiers. The result of this comparison is shown in Fig. 10. The vote-based combination should better deal with occlusion than the other approaches, but at the same time, more false positives are returned by this method. The best performance, i.e., when the goal is a low FPPF, is given by the radial approach. This logically follows from the nonlinearity of the data returned from the part detectors. The plot of the Bayesian approach shows an excellent detection rate but with a high number of false positives. Applying a linear separation on set of nonlinear data, the Bayesian approach classifies more elements as pedestrians but, at the same time, incorrectly classifies a greater number of true negatives. This explains the high detection rate and the raise in false positives.

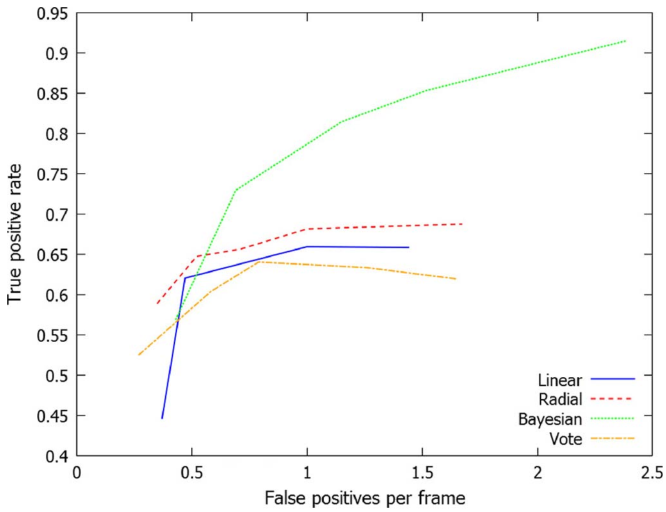
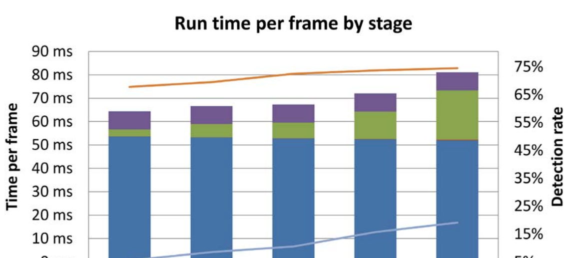


Fig. 10. Comparison of different methods for combined part verification. With a low false positive rate, the radial approach performs better. The system configuration is as follows: DaimlerDB training set, a k value of 13, a padding value of 2, and a two-part-based approach.



	k = 13	k = 12	k = 11	k = 10	k = 9
Tracking [ms]	0,015	0,017	0,019	0,020	0,024
Features calc [ms]	7,720	7,720	7,720	7,720	7,720
Body part calc [ms]	2,980	5,660	6,670	11,720	21,130
Prepare HOG calc [ms]	0,043	0,078	0,091	0,165	0,339
Haar calculation [ms]	53,600	53,200	52,800	52,400	51,900
Detection rate	0,673	0,690	0,720	0,732	0,740
False positives	0,050	0,080	0,100	0,152	0,186

Fig. 11. Speed versus detection rate and FPPF. The time has been measured for each stage, which is denoted as k. We see that the reduction in false positives and the increase in true positives increase the number of stages. The PC is equipped with a 2.20-GHz Intel Core i7-2670QM CPU and 8 GB of DDR2 RAM.

G. Speed Evaluation

This test evaluates the speed of the system at various settings for the detection stage for given hardware. The results are shown in Fig. 11. Changing k, which is the number of Haar cascade stages, has a large impact on the system speed since it directly influences how many candidates the next stages must evaluate. The largest contribution in processing time is the full-body verification, whereas the contribution of the last stage is practically irrelevant. Setting a high k results in lower number of ROIs and a faster system, and in a system capable of detecting fewer targets. The goal here is to choose the system where parameters are set to obtain a tradeoff between speed and detection rate, taking the FPPF into account. Speed has been measured on a run of 1000 images, and the results are the mean of those runs. For the fastest run, a complete calculation can be performed in about 0.757 s, corresponding to 1.32 frame/s.

TABLE III
FINAL SYSTEM PERFORMANCE

	Detection rate	False positive per frame
Basic system	0.69	0.5
With candidate size filtering	**0.673**	**0.046**
With resized image (320x240)	0.55	0.02

	Frame rate	Image size
Unoptimized system	1.32 FPS	640x480
Parallelized system	**16.67 FPS**	640x480
	30 FPS	320x240
Fastest system in [7]	2.6 FPS	640x480

H. Parameterization of the Input Image Geometry

To make the system more configurable, the possibility of choosing the image size has been added; therefore, processing time can be only adjusted by resizing the input image. Camera calibration parameters will automatically change to ensure the correct behavior of perspective and inverse perspective mapping (IPM) functions used when filtering candidates. Resizing the image results in a reduction in processing speed and the true positive rate, as shown in Table III.

I. Key Improvements

Significant improvements were applied to the system described in [9], as shown in Fig. 6, by comparing the blue graph with the green graph. At the FPPF of 0.5, the true positive rate was increased from 0.4 to 0.63 with a speedup of more than 16×. The filtering of candidates and feature-based tracking introduced a significant speedup as the implementation was parallelized.

J. Evaluations With "Real-World" Driving Data

Fig. 12 shows some examples of possible circumstances that may occur in a real environment, varying from simple, medium, and hard situations.

The first two lines represent simple situations with pedestrians crossing the street, riding bicycles, or walking along the sidewalk, and some more critical situations, with pedestrians partially occluded in a structured environment. Line C shows, instead, some case of hard detection as highly occluded pedestrians, pedestrian underexposure, and pedestrians situated in a highly complex scene.

A measure of the maximum distance of recognition is provided by line D, where one can recognize pedestrians at about 45/50 m away.

In addition, our algorithm still has some shortcomings, as shown in the last line. Some "common" errors of classification are shown in the last two pictures; however, these errors can be considered superficial since they are only present in single frames and, therefore, can be detached by our tracking system.

A relevant problem is shown in the first picture of line E. Due to the geometric filter on the size of pedestrians, our system does not detect pedestrians smaller than 1.45 m. A possible

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 12. Line A shows the examples of pedestrians detection with simple environment. Line B shows the examples of behavior of our detector in the presence of small occlusion and structured scenes. The potential of our classifier in the presence of pedestrians strongly occluded, highly structured scenes, and underexposure of the camera is shown in line C. Examples of detection of pedestrians far apart, at about 45 m, are shown in line D. Line E shows the samples of possible detector problems: pedestrians are too small, cyclists at the intersections, and "common" misclassification, such as trees and poles.

solution is to broaden the constraints of the filter, thus obtaining a greater number of false positives. An additional downside of the geometric filter regards the accuracy of the calculation of the IPM if the ground is not flat, as is presupposed by our system. Using techniques of image stabilization described in [47] and [48] could provide significant improvements; an alternative solution would involve introduction of a stereo-based approach to filter candidate pedestrians.

A further case of interest is depicted in the second illustration of the last line, showing the situation where a pedestrian is crossing the street where the car is turning. With cameras situated in the front of the vehicle, it is impossible to detect the pedestrian in time to brake. A solution could be the introduction of cameras that allow looking on the side of the car and detect pedestrians in advance.

V. Porting Part-Based Pedestrian Detection on an Instrumented Vehicle Test Bed

With the aim of testing the developed system in the real world, the original standalone software has been ported first to a prototyping software platform to optimize it in a laboratory

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

setting and then to a real hardware platform. Given on the results obtained on the real platform, a set of additional features have been identified and implemented to improve the detection performance in a number of critical situations. These modifications are described in the following.

A. Porting and Optimization

The original code has been ported to be an application of the latest version of the GOLD[49] software.

GOLD offers a number of advantages in this phase, allowing the application to deal with virtual devices, instead of using the hardware directly. This allows the system to work in the laboratory on recorded data previously taken and stored on a disk, or on a real platform and taking data from the hardware.

During the porting, a conversion of the Daimler database images has been done, making it possible to read this recording with GOLD and to use this data set as input for the pedestrian detection application. This has been done mainly for the availability of a high-quality per-frame ground truth that can be recovered any time and can be used to check the consistency of the results with those obtained with the standalone application.

GOLD also offers a profiling application programming interface allowing timing of different parts of the application, seeing the time spent in the execution of these parts for every frame, and computing cumulative statistics collected across a playback session.

B. Platform

After reaching an acceptable performance level, the system was transferred to a real prototype vehicle [50].

The platform is equipped with ten cameras, and one of these, looking forward, has been physically connected to the application. The camera used is a PointGrey DragonFly 2, working at 10 Hz and producing images with a resolution of 1024×768. The camera is equipped with a 6-mm micro lens, which provides an acceptable level of distortion for this application. The camera has a firewire interface that is connected to an adapter located in the trunk, in an industrial PC. The PC is equipped with a 2.20-GHz Intel Core i7-2670QM CPU and 8 GB of DDR2 RAM. Using this configuration and downsampling the image to 640×480 pixels, it is possible to keep the processing time below 100 ms for simple scenes generating a reasonably low number of candidates. During the tests on the real platform, some weaknesses of the original system emerged. These weaknesses were mostly due to a lack of robustness in the results observed over long driving periods and include a high number of false positives and two discontinuous recognition processes of the same targets observed along several frames.

C. Candidate Filtering

As a first step, a set of filters was introduced to reduce false positives to remove the candidates with size outside of a selected range of [1.45–2.20] m in real-world measurements. The IPM technique [51], [52] was used to calculate the position of the pedestrian candidate in real-world coordinates; by using the pedestrian baseline, it is possible to determine the ratio of pixels and meters at this distance and estimate the pedestrian height in the world, knowing its height in image coordinates, using the flat road assumption. The application of this filter gives a good reduction in false positives with a small impact on true positives; quantitative results are shown in Section VI.

D. Features

Classification schemes can be enhanced with a tracking system to counteract the high instability of the detector due to the high variability of pedestrians. A feature-based tracking system was used to fix this lack. Features provide a robust base to track people due to their translation and light invariance. A set of features, as detailed in Section V-E and described in [45], based on multiple local convolutions, key points and descriptors, are extracted from two different hash images. Stable feature locations are obtained to filter the input images with 5×5 blob and corner masks, and then, it was applied with nonmaximum suppression (NMS) and nonminimum suppression [53] on the filtered images. Starting from the pedestrian output from the verification stage, features are computed and used to match pedestrians in subsequent frames. The feature-based tracking has the downside of being dependent on the vehicle egomotion. Vehicles moving at high speed, particularly in conjunction with low frame rates, cause a high difference between two subsequent frames and, consequently, a bad match between corresponding features. To cope with this problem, a higher frame rate must be used. Another downside of using features for a tracking system is the difficulty of distinguishing between foreground and background pixels. As a result, some matches could be wrong, but the impact of these errors is very low and decreases pedestrian motion.

E. Tracking

When a candidate pedestrian has been recognized by the SVM for 250 ms (a time limit is used due to the variability of frame rates), it is considered a true pedestrian, and it is introduced in the tracking system. In the following frames, the pedestrian features will be matched with new candidate pedestrians, and their positions and descriptor will be updated with the new one. By using the sum-of-absolute-difference error metric, 11×11 block windows of horizontal and vertical Sobel filter responses were compared with each other. The whole block window with Sobel responses is reduced to 8 bits, and the differences over a sparse set of 16 locations are summed. For further significant speedups, it was matched only to a subset of all features, which are found by the NMS. The features are then assigned to a 50×50 pixel bin of an equally spaced grid and will be computed with the minimum and maximum displacements for each bin. In this way, we reduce the final search space and speed up the system. If no candidate matches the search criteria (missing detection by the SVM), searching for a match will be done across the entire image. If a match is found, a ghost pedestrian will be introduced. It will be updated

TABLE IV
FINAL SYSTEM CONFIGURATION

Training Dataset	10000 Daimler-DB
k-value	13
Padding	2
Number of parts	2
Combined verification	Radial

for up to 0.5 s; after which, it will be removed. A flowchart of the tracking system is shown in Fig. 5.

F. Higher Frame Rate

The best results from the feature-based tracking are obtained in correspondence to a good match between the features extracted from the candidate images in consecutive frames. When working at low frame rates, such as 10 Hz, the high variability between consecutive frames, which are both due to the object movement in the scene and the vehicle egomotion, leads to a bad performance of the feature matcher and, as a consequence, the tracking system. Using the prototype platform, a new set of images has been recorded at 30 Hz from one of the forward-looking cameras. These images have been used offline with tracking enabled, showing significant improvements in the result robustness and reducing the blinking of correct detection caused by missed detection in single frames and by the false positives. Unfortunately, the frame rate of the DaimlerDB data set is lower than 10 Hz, and this is a limiting factor for comparing recognition performance improvement with the tracking system. To get a significant sampling speed in real time, the prototype was altered to acquire images with different sizes. The reduction of the input image to 320 $\times$ 240 pixels leads to a frame rate of 20 frame/s, and offers a level of recognition performance similar to the one obtained at 30 Hz with the offline processing.

VI. PERFORMANCE EVALUATION AND COMPARISONS

After the evaluation of all the parameters, a final system to be tested on the DaimlerDB has been defined. Table IV contains the parameter values used in the final system.

A. Final Test Without Tracking

Fig. 13 shows the performance of the final system. This figure shows results for several values of k to plot ROC curves and gives a detection rate of about 0.69 with an FPPF of 0.5, considering 13 as the best value for k. Despite a high FPPF, our system is directly comparable with others shown in [7]; it shows the same performance of *LatSvm-V2*, which is one of the most successful PPD described in [25], but with a huge speedup of 10$\times$ (not considering the extra speedup described in the following). A better performance is achieved by filtering the candidates as described earlier, reducing the false positive rate from 0.5 to 0.046 with a small reduction of the true positive rate to 0.673, as shown in Fig. 14. These results allow our system to gain a foothold in the state of the arts consolidated by a huge speedup described in the following. The results are summarized in Table III.

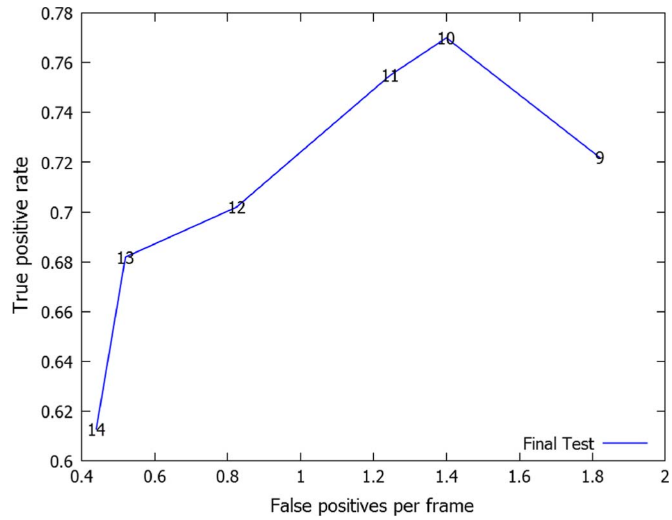


Fig. 13. Final test detection. Evaluation on the performance on the last part of DaimlerDB images without the optimization for the porting on a real prototype.

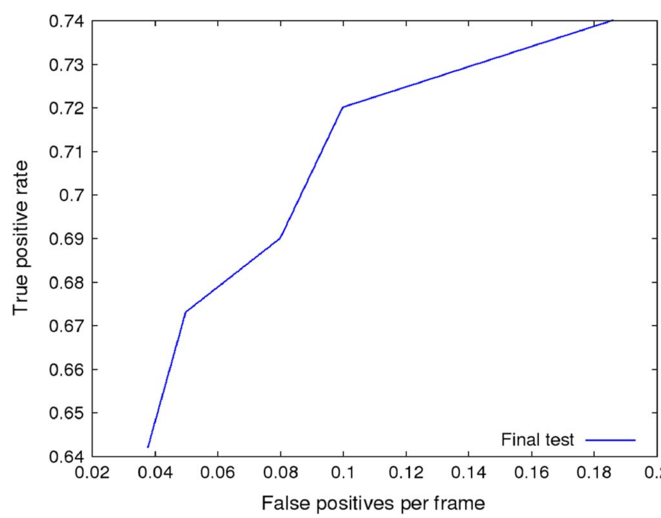


Fig. 14. Evaluation on the performance of the optimized system on the last part of DaimlerDB images.

B. Tracking Improvements

Introducing a tracking system resulted in significant improvements in the number of true positives and a reduction in the number of false positives. The performance improvements due to the introduction of tracking were tested on our own data set (two sequences of 5182 and 11 490 frames, respectively) captured on the real prototype described in Section V. It was not possible to use the DaimlerDB test set due its low frame rate of about 10 Hz, which is too low to ensure a stable tracking. An increase of 27% and 22% in true positives on the two data sets was obtained with a reduction of 5% and 10% in false positives. These results showcase better stability of the system, allowing tracking of the pedestrian in consecutive frames and opening the way for further improvements, such as determining pedestrian direction and orientation [54].

Fig. 15. Examples of detection on a prototype platform. People in different poses are detected, including cyclists or people walking close to a tree that are often hard to detect. A missed detection is shown in the last figure due to overexposure of the pedestrians.

C. Performance on the Prototype Platform

To guarantee real-time performance on the prototype platform (GOLD), a parallelization technique was introduced. Parallelization of Haar-feature and HOG-feature calculation and classification was obtained by compiling OpenCV with thread building blocks (TBBs) enabled. In this way, it is possible to take advantage of multicore CPUs. Further parallelization was obtained by executing the classification of HOG features for the different body parts on separate threads, reducing the verification stage processing time of about 30%. With an image of 640×480 pixels, the processing time changed from 755 to 60 ms, which is a speedup of $12\times$. Thus, our system is running eight times faster than the fastest system presented in [7]; 16.67 versus 2.6 frame/s. A further speedup can be provided by reducing the images size, which results in a processing speed of about 30 Hz on an image of 320×240. This approach, however, has a detrimental effect on detection rates. Examples of detection on a prototype platform are shown in Fig. 15.

VII. Concluding Remarks

Various studies to improve the presented pedestrian detector are currently ongoing. Since feature-based tracking works better at higher frame rates, a low-level reimplementation of the two-stage classifier fully exploiting multicore-processor (or graphics processing units) features may give some significant speedup. The current system relies on OpenCV 2.4 compiled with Intel TBB support. Looking at the CPU utilization, we get values between 60% and 80% for each core, which is a clear indication that some serial piece of code is still present. By reducing the image area, the processor utilization falls, ranging from 80% at 640×480 pixels to 60% for 320×240 pixel images.

Another improvement can be added to the high-level processing, introducing filters on the predicted pedestrian trajectory. In particular, when working with high frame rates, a good tracking of the pedestrian trajectory is produced from the current system. A Kalman filter could provide a prediction of the trajectory that a pedestrian is taking in the future, which could be evaluated to predict dangerous situations.

The vehicle egomotion has intentionally not been used for this system since one of the constraints was to obtain a final system simply relying on vision. Introducing a visual odometry block could supply information on egomotion without breaking this requirement. However, additional computational power would be needed.

In this paper, a novel pedestrian detector system, running on a prototype vehicle platform, has been presented. The algorithm generates possible pedestrian candidates from the input image using a Haar cascade classifier. Candidates are then validated through a novel part-based HOG filter. A feature-based tracking system takes the output of the two-stage detector and compares the features of new candidates with those of the past. Matching is performed with the aim of assigning a consistent label to each candidate and of improving the recognition robustness, by filling false negatives filtered by the previous phases. The whole system has been ported to a prototyping framework and integrated on a platform vehicle, for testing and optimization. A significant performance improvement has been obtained by exploiting the CPU multicore features. As a result, a system working at 20 Hz and offering performance comparable with the state of the art has been obtained. Additional real-world tests have been performed on the platform for finding weaknesses. Although the system is faster compared with the state of the art, its detection performance compares very favorably to the state of the art with a true positive rate of 0.673 at a FPPF of only 0.046.

Acknowledgment

The authors would like to thank their colleagues in the Laboratory for Intelligent and Safe Automobiles, the Computer Vision and Robotics Research Laboratory, and the reviewers for their constructive comments. The work described in this paper has been developed in the framework of the Open intelligent systems for Future Autonomous Vehicles (OFAV) Project funded by the European Research Council (ERC) within an Advanced Investigators Grant.

References

[1] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 108–120, Mar. 2007.

[2] U. Nunes, C. Laugier, and M. M. Trivedi, "Introducing perception, planning, and navigation for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 375–379, Sep. 2009.

[3] T. Dang, J. Desens, U. Franke, D. Gavrila, L. Schafers, and W. Ziegler, *Steering and Evasion Assist*. London, U.K.: Springer-Verlag, 2012.

[4] A. Bartels, M. Meinecke, and S. Steinmeyer, *Lane Change Assistance*. London, U.K.: Springer-Verlag, 2012.

[5] T. Gandhi and M. M. Trivedi, "Parametric ego-motion estimation for vehicle surround analysis using an omnidirectional camera," *Mach. Vis. Appl.*, vol. 16, no. 2, pp. 85–95, Feb. 2005.

[6] K. S. Huang, M. M. Trivedi, and T. Gandhi, "Driver's view and vehicle surround estimation using omnidirectional video stream," in *Proc. IEEE Intell. Veh. Symp.*, 2003, pp. 444–449.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[8] D. Geronimo, A. Lopez, A. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.

[9] A. Møgelmose, A. Prioletti, M. M. Trivedi, A. Broggi, and T. B. Moeslund, "A two-stage part-based pedestrian detection system using monocular vision," in *Proc. 15th IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 73–77.

[10] A. Prioletti, P. Grisleri, M. Trivedi, and A. Broggi, "Design and implementation of a high performance pedestrian detection," presented at the IEEE Intell. Veh. Symp., Gold Coast, Australia, 2013, Paper WePO2T1.25.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. CVPR*, 2005, vol. 1, pp. 886–893.

[12] S. Krotosky and M. Trivedi, "On color-, infrared-, and multimodal-stereo approaches to pedestrian detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 619–629, Dec. 2007.

[13] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. Jung, "A new approach to urban pedestrian detection for automatic braking," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 594–605, Dec. 2009.

[14] P. Geismann and G. Schneider, "A two-staged approach to vision-based pedestrian recognition using Haar and HOG features," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 554–559.

[15] T. Gandhi and M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 413–430, Sep. 2007.

[16] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-6644-8

[17] D. Gerónimo, A. D. Sappa, A. López, and D. Ponsa, *Adaptive Image Sampling and Windows Classification for On-board Pedestrian Detection*. Bielefeld, Germany: Univ. Bielefeld, 2007.

[18] L. Zhang, B. Wu, and R. Nevatia, "Pedestrian detection in infrared images based on local shape features," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[19] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *Int. J. Comput. Vis.*, vol. 75, no. 2, pp. 247–266, Nov. 2007.

[20] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, Apr. 2001.

[21] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Computer Vision–ECCV*, T. Pajdla and J. Matas, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2004, ser. Lecture Notes in Computer Science, pp. 69–82. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24670-1_6

[22] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. Gavrila, "Multi-cue pedestrian classification with partial occlusion handling," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 990–997.

[23] X. Mao, F. Qi, and W. Zhu, "Multiple-part based pedestrian detection using interfering object detection," in *Proc. 3rd ICNC*, 2007, vol. 2, pp. 165–169.

[24] L. Zhe and L. Davis, "Multiple instance feature for robust part-based object detection," in *Proc. Conf. CVPR*, Jun. 2009, pp. 405–412.

[25] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[26] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 241–245.

[27] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[28] M. Andriluka, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[29] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *Proc. 11th IEEE ICCV*, Oct. 2007, pp. 1–7.

[30] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.

[31] D. F. Llorca, M. A Sotelo, A. M Hellín, A. Orellana, M. Gavilán, I. G Daza, and A. G Lorente, "Stereo regions-of-interest selection for pedestrian protection: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 25, pp. 226–237, Dec. 2012, doi:10.1016/j.trc.2012.06.006.

[32] W. Khan and J. Morris, "Safety of stereo driver assistance systems," in *Proc. IEEE IV Symp.*, Jun. 2012, pp. 469–475.

[33] K. Fuerstenberg, "Pedestrian protection using laserscanners," in *Proc. IEEE Intell. Transp. Syst.*, Sep. 2005, pp. 437–442.

[34] U. Scheunert, H. Cramer, B. Fardi, and G. Wanielik, "Multi sensor based tracking of pedestrians: A survey of suitable movement models," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2004, pp. 774–778.

[35] A. Yoshizawa, M. Yamamoto, and J. Ogata, "Pedestrian detection with convolutional nerual networks," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 224–229.

[36] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2179–2195, Dec. 2009.

[37] T. Gandhi and M. M. Trivedi, "Computer Vision and Machine Learning for Enhancing Pedestrian Safety," in *Computational Intelligence in Automotive Applications*, D. Prokhorov, Ed. Berlin Heidelberg, Germany: Springer-Verlag, 2008, ser. Studies in Computational Intelligence, pp. 59–77. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-79257-4_4

[38] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, Jun. 2000.

[39] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis," in *Proc. 11th IEEE ICCV*, Oct. 2007, pp. 1–8.

[40] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 794–801.

[41] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 304–311.

[42] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2001.

[43] I. Alonso, D. Llorca, M. Sotelo, L. Bergasa, P. de Toro, J. Nuevo, M. Ocaña, and M. Garrido, "Combination of feature extraction methods for SVM pedestrian detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 292–307, Jun. 2007.

[44] F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces," in *Proc. IEEE Comput. Soc. Conf. CVPR*, 2005, vol. 1, pp. 829–836.

[45] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Proc. IEEE IV Symp.*, 2011, pp. 963–968.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[47] A. Broggi, P. Grisleri, T. Graf, and M. Meinecke, "A software video stabilization system for automotive oriented applications," in *Proc. 61st IEEE VTC-Spring*, May–Jun. 1, 2005, vol. 5, pp. 2760–2764.

[48] L. Bombini, P. Cerri, P. Grisleri, S. Scaffardi, and P. Zani, "An evaluation of monocular image stabilization algorithms for automotive applications," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2006, pp. 1562–1567.

[49] M. Bertozzi, L. Bombini, A. Broggi, P. Cerri, P. Grisleri, and P. Zani, "GOLD: A framework for developing intelligent-vehicle vision applications," *IEEE Intell. Syst.*, vol. 23, no. 1, pp. 69–71, Jan./Feb. 2008.

[50] P. Grisleri and I. Fedriga, "The BRAiVE platform," in *Proc. 7th IFAC Symp. Intell. Autonom. Veh.*, Lecce, Italy, Sep. 2010, pp. 497–502.

[51] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biol. Cybern.*, vol. 64, no. 3, pp. 177–185, Jan. 1991. [Online]. Available: http://www.biomedsearch.com/nih/Inverse-perspective-mapping-simplifies-optical/2004128.html

[52] M. Bertozzi, A. Broggi, A. Fascioli, and R. Fascioli, "Stereo inverse perspective mapping: Theory and applications," *Image Vis. Comput. J.*, vol. 16, no. 8, pp. 585–590, Jun. 1998.

[53] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th ICPR*, 2006, pp. 850–855.

[54] T. Gandhi and M. Trivedi, "Image based estimation of pedestrian orientation for improving path prediction," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2008, pp. 506–511.

Antonio Prioletti (S'12) received the Master's degree in computer engineering from the University of Parma, Parma, Italy, in 2012. He is currently working toward the Ph.D. degree with the Artificial Vision and Intelligent Systems Laboratory, University of Parma.

He was a Visiting Scholar with the Computer Vision and Robotics Reasearch Laboratory, University of California, San Diego, CA, USA. His main research interests include computer vision and its applications on advanced driver-assistance systems.

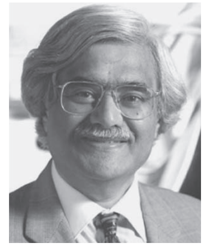
Andreas Møgelmose (S'12) received the Bachelor's degree in computer engineering on the topic of information processing systems and the M.Sc. degree in informatics from Aalborg University, Aalborg, Denmark, in 2010 and 2012, respectively. He is currently working toward the Ph.D. degree in the field of visual analysis of people with the Visual Analysis of People Laboratory, Aalborg University.

He was a Visiting Scholar with the Laboratory for Intelligent and Safe Automobiles, Computer Vision and Robotics Research Laboratory, University of California, San Diego, CA, USA. His main research interests include computer vision and machine learning, particularly in the area of looking at people.

Paolo Grisleri received the Dr.Eng. degree in computer engineering and the Ph.D. degree from the University of Parma, Parma, Italy, in 2002 and 2006, respectively.

In 2002, he was a Researcher with the Department of Information Engineering, University of Parma. His research interests include computer vision, data acquisition techniques, and system architectures for advanced driver assistance systems.

He serves as an Associate Editor and member of the Editorial Board for four issues of the IEEE TRANSACTIONS INTELLIGENT TRANSPORTATION SYSTEMS and is an Associate Editor for the IEEE INTELLIGENT TRANSPORTATION SYSTEM SOCIETY NEWSLETTER.



Mohan Manubhai Trivedi (F'11) received the B.E. (Hons.) degree in electronics from the Birla Institute of Technology and Science, Pilani, India, in 1974 and the M.S. and Ph.D. degrees in electrical engineering from Utah State University, Logan, UT, USA, in 1976 and 1979, respectively.

He served on the Executive Committee of the California Institute for Telecommunication and Information Technologies as the Leader of the Intelligent Transportation Layer at the University of California, San Diego (UCSD) and as a charter member and Vice Chair of the Executive Committee of the University of California Systemwide Digital Media Initiative. He is currently a Professor of electrical and computer engineering and the Founding Director of the Laboratory for Intelligent and Safe Automobiles, Computer Vision and Robotics Research Laboratory, UCSD. He and his team are currently working toward research in machine and human perception, machine learning, human-centered multimodal interfaces, intelligent transportation, and driver-assistance and active safety systems. He serves as a Consultant to industry and government agencies in the U.S. and abroad, including the national academies, major automobile manufactures, and research initiatives in Asia and Europe.

Dr. Trivedi is a Fellow of the International Association of Pattern Recognition (for contributions to vision systems for situational awareness and human-centered vehicle safety) and of the International Society for Optics and Photonics (for distinguished contributions to the field of optical engineering). He was the Program Chair of the IEEE Intelligent Vehicles (IV) Symposium in 2006 and the General Chair of the IEEE IV Symposium in 2010. He has been elected to the Board of members of the IEEE Intelligent Transportation System Society. He is an Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and the *Image and Vision Computing Journal*.

Alberto Broggi (SM'06) received the Dr. Ing. (Master) degree in electronic engineering and the Ph.D. degree in information technology from the University of Parma, Parma, Italy, in 1990 and 1994, respectively.

He is currently a Full Professor with the University of Parma, where he is also the Director of the Artificial Vision and Intelligent Systems Laboratory. He is the author of more than 150 publications in international scientific journals, book chapters, and refereed conference proceedings.

Dr. Broggi served as the Editor-in-Chief of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS for the term 2004–2008. He served as the President the IEEE Intelligent Transportation Systems Society for the term 2010–2011.

Thomas B. Moeslund (M'12) received the M.Sc.E.E. and Ph.D. degrees from Aalborg University, Aalborg, Denmark, in 1996 and 2003, respectively.

He is currently an Associate Professor and the Head of the Visual Analysis of People Laboratory with Aalborg University. He has been involved in ten national and international research projects, as a Coordinator, Work Package leader, and Researcher. He is the author of about 100 peer-reviewed papers (citations: 2655; H-index: 16). His research interests include all aspects of computer vision, with a special focus on automatic analysis of people.

Dr. Moeslund has been a Co-chair of four international workshops/tutorials and a member of the Program Committee for a number of conferences and workshops. He serves as an Associate Editor and as a member of the Editorial Board for four international journals. He received the Most Cited Paper Award in 2009, the Best IEEE Paper Award in 2010, and the Teacher of the Year Award in 2010.