International Journal of Research Studies in Science, Engineering and Technology [IJRSSET]
Volume 1, Issue 4, July 2014, PP 10-17
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Part-Based Pedestrian Detection and Tracking for Driver Assistance using two stage Classifier

Arun Kumar HR[1], Nithya E[2]

[1]M. Tech Student, [2]Professor
[1,2]Department of Computer Science and Engineering,
Dr. Ambedkar Institute of Technology, Bengaluru-560056, Karnataka, India

Abstract: *Pedestrian detection is an essential and significant task in any intelligent video surveillance system, as it provides the fundamental information for semantic understanding of the video footages and for improving safety systems for accident prevention. Pedestrian detection and tracking for driver assistance is mainly for the purpose of protecting the pedestrians using the automatic braking. This paper presents a state-of-the-art pedestrian detection system based on a two-stage classifier with Multiple Target Tracking. Candidates are detected and extracted with a Haar-cascade classifier trained with the Daimler Detection Benchmark data set. Then the extracted candidates are 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 Multiple Target Tracking (MTT) system tracking to enhance the recognition robustness and improve the result's stability. Use of MTT in driver assistant systems makes them very efficient and effective in collision avoidance and early warning. 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.*

Keywords: *Multiple Target Tracking, Advanced driver assistance system (ADAS), classifiers, features, pedestrian detection, Kalman filter.*

1. INTRODUCTION

Effective vision systems need to accurately assess situational criticalities from the panoramic surround of a vehicle and simultaneously assess awareness of these criticalities by the driver. Pedestrian detection system can be used in surveillance, Advanced Driver Assistance Systems (ADAS), and many other places. The major 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 high variability in appearance among pedestrians, cluttered background, 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.

The use of Multiple-Target Tracking (MTT) in the pedestrian detection system enhances the affectivity of driver assistance systems to aid drivers in taking correct decisions in critical situations. The purpose of target tracking is to collect data from the sensor Field of View (FOV) containing one or more potential targets of interest and to partition the sensor data into sets of observations, or tracks [1]. Part-based pedestrian 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 are affected 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.

A tracking technique can be introduced to supply missing detection. This paper presents a part based pedestrian detection approach and multiple target tracking for driver assistance, provides the following features:

1) The part based pedestrian detection system with the feature based multiple target tracking for driver assistance.

2) A thorough analysis of the impact of changes in parameters for the part based pedestrian detection system algorithm that goes far beyond what was presented before.
3) This paper uses the INRIA data set as a benchmark data set for pedestrian related training and test sets.
4) The new thing that we implemented in this paper is the Multiple Target Tracking for driver assistance to resolve the problem of occlusion.

2. RELATED WORK

Over the past decades, the essential role of the Pedestrian detection system is to protect the pedestrian and to avoid collision of the vehicles. A survey of the pedestrian detection field and taxonomy of the involved types are provided in [3]. AdaBoost cascade on Haar-like features or HOG+SVM classifiers are used in [2], [3]. In [7], a part-based two-stage pedestrian detector has been presented. It builds on previous work by Geismann and Schneider [2], but extends it by introducing a part-based verification system instead of just full body verification. In [7], the system has been tested on the INRIA dataset and it performs better across the full range of false positives per frame. In [6], [9] 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. Part based pedestrian detection overcome the problem of occlusion and reduces the number of false positive per second. In [8], the multiple target tracking is used in pedestrian detection to reduce the number of false positive per second (FPPS). In [8], [10] the design using ten 4-state filters apart from the other system components. Implementing a fully hardware system with ten 4-state filters would have been simply infeasible. In some, the feature based tracking method is used in pedestrian detection in [6], [7] and feature matcher is also used in this.

3. PROPOSED SYSTEM

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 multiple target 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 non-pedestrians (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.

Fig.1. Different bounding boxes required by Haar cascade and HOG–SVM. The base image is from the Daimler DB data set [6]. The red dashed line is the Haar bounding box and the blue continuous line is the HOG bounding box.

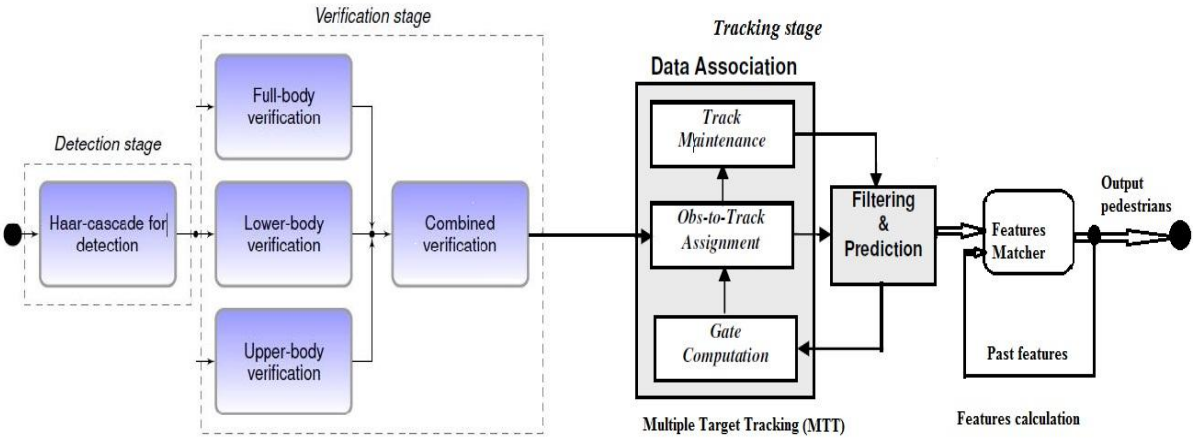

Fig.2. *System Architectural Diagram for the Proposed PPD.*

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.

3.1 System Architecture

The overall system architecture of the Part based Pedestrian detection system shown in Fig.2, consists of 3 stages:

3.1.1 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 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*. 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 (shown in Fig.1). Following the suggestions 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.

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. The PPD was not introduced to the detection stage, as preliminary tests and the work showed a bad performance for this approach. When the bounding boxes of the candidate pedestrians (see Fig.4) have been obtained, they are passed to the verification stage.

3.1.2 Verification stage

a) Part Verification Stage

As opposed to the full-body verification stage, 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.

Fixed ratios 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.

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 take 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 non-pedestrians of the element; the regression SVM provides the estimated function value. A special kind of sparse HOG descriptors is used, whereas our algorithm uses classic dense HOG descriptors. For SVM training, images from several data sets were tested with the goal of analyzing the effects of training sets in 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.

b) 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

$$l_{out} = \{1, \text{if } \Sigma\ i<3 \text{ and } i=0,\ l_i>=1\}$$
$$\{-1, \text{if } \Sigma\ i<3 \text{ and } i=0,\ l_i<1\} \quad (1)$$

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

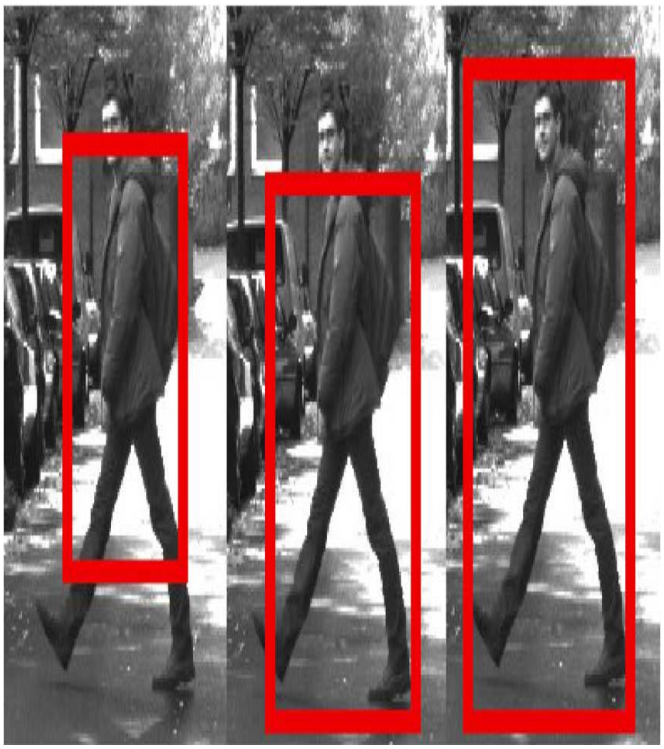

Fig.3. *Example of the degradation of the bounding box varying k from 13 in the last pictures to nine in the second picture and to eight in the first picture.*

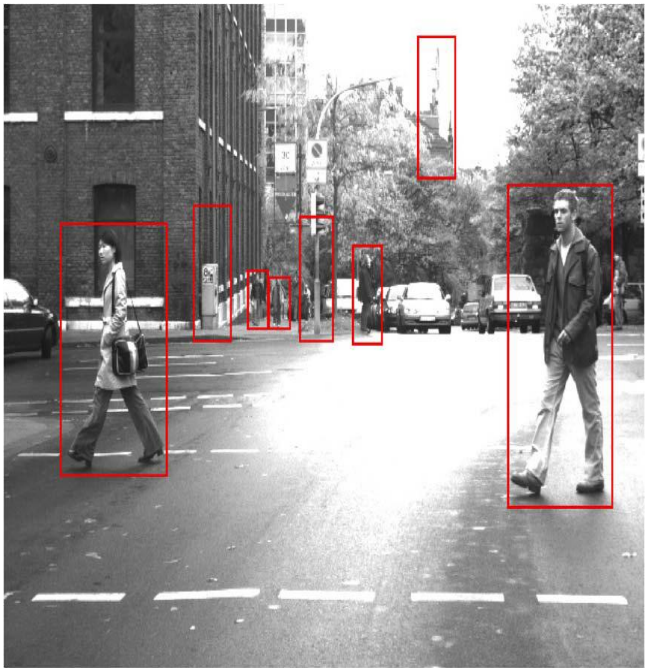

Fig.4. *Detection stage output. Several false positives are contained, but these will be removed in the verification stage.*

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.

3.1.3 Tracking Stage

A feature-based tracking was used to enhance the detection rate. The tracker is introduced to 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. The tracker labels pedestrians increase the number of true positives due to the higher stability of the detection in the case of, for to supply possible missing detection due to mistakes of the classifier in the verification stage.

MTT building blocks

A simplified view of Implementation of Multiple Target Tracking (MTT) system is given in figure 5. The system can broadly be divided into two main functions namely *Data Association* and *Filtering & Prediction*. The two functions work in a close loop. The data association function is further divided into three sub-functions; "*Track maintenance*", "*Observation-to-Track Assignment*" and "*Gate Computation*".

4. IMPLEMENTATION

One of the main contributions of this paper is a thorough evaluation of the algorithm's parameters. Daimler DB was primarily used, with elements from the INRIA data set in a few tests. Unless otherwise specified, images from the training part of Daimler DB were used for training, i.e., both the detection stage and the part verification stage. The test part of Daimler DB 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 here.

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 has been used to determine the detection rates. Therefore, the results should be directly comparable. The PASCAL measure evaluates to true if the overlap is more than50%, i.e.

$$ao \equiv \frac{\text{area}\,(BBdt \cap BBgt)}{\text{area}\,(BBdt \cap BBgt)} > 0.5 \quad (2)$$

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 1500images 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 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. 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 here; therefore, the parameter is relevant for optimization. An example of padding, 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.

$$p_{\text{pixels}} = \frac{w_{\text{ROI}}}{w_t} . p \quad (3)$$

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. 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 analysed and more mistakes happen.

C. 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. 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 positive.

D. Speed Evaluation

This test evaluates the speed of the system at various settings for the detection stage for given hardware. Changing *k*, which is the number of Haar cascade stages, has a large impact on the system speed since it directly influence show many candidates the next stages must 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 trade-off between speed and detection rate, taking the FPPF into account. The largest contribution in processing time is the full-body verification, whereas the contribution of the last stage is practically 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 1.32frame/s.

E. Tracking stage

For the purpose of parallelized implementation we organized the application into sub-modules as shown in figure 4. The functioning of the system is explained as follows. Assuming recursive processing as shown by the loop in figure 4, tracks would have been formed on the previous radar *scan*. When new observations are received from the processing loop is to be executed. Incoming observations are first considered by the "*Gate checker*" for updating of the existing tracks. Gating tests determine which possible "*observation-to-track*" pairings are reasonable, by attributing a cost to each pairing. The costs are calculated as the statistical distance between the *predictions* of the target states given by the filters and the *observe state* coordinates received from the radar. These costs are put together in a *cost matrix* which is then passed on to the *assignment solver* to determine the finalized pairings. The pairings are made in a way to ensure minimum total cost for all the pairings. The finalized observation-track pairings are passed on to the tracking filters which use them for estimating the current states of targets and predicting the next states as well the *error covariance* associated with these predictions. The predicted states and predicted error covariance are used by the "*Gate compute*" function to define probability *gates* or windows around the predicted states. The dimensions of the gates being dictated by the prediction error covariance, these gates demarcate the probability boundaries for the next state coordinate measurements. The "*Gate Compute*" sub-function can be viewed as a first level of "*screening out*" the unlikely target-track associations in case of multiple observations falling close to a single prediction or vice versa. In the second level of "*screening*", namely observation-to-track assignment, a strictly one-to one coupling is established between observations and tracks. The "*Track Maintenance*" sub-function consists of three blocks. The *"obs-less Gate*

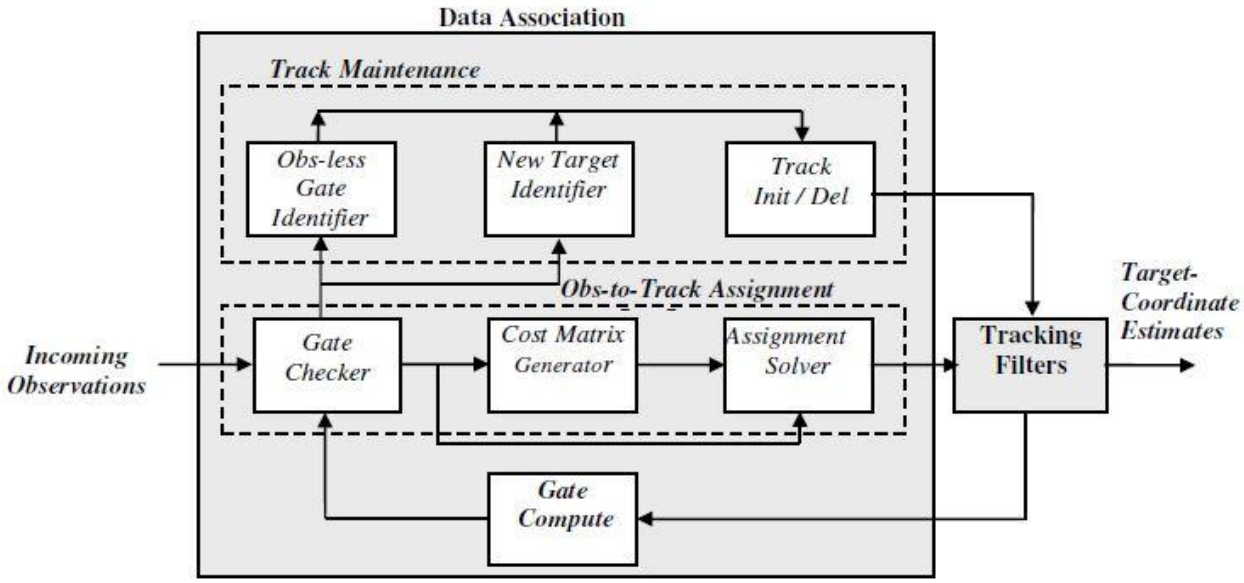

Fig.4. *Implementation of Proposed Multiple Target Tracking..*

Identifier" identifies the gate where no observation falls. This indicates a probable disappearance of an already known target and hence the deletion of its track after confirmation. The *"New Target Identifier*" detects observations that fall outside all the gates. These observations are potential candidates for initiating new tracks after confirmation. The *"Track Init/Del"* block initiates new tracks or deletes existing ones when needed. In context of this work, 3 observations out of 5 scans for the same target initiate a new track while 3 consecutive misses out of 5 scans for an existing target prompts the deletion of its track. The "*Tracking filters*" block in figure 2, is particularly important. We use Kalman filters for this block. The number of filters employed is equal to the maximum number of targets to be tracked. In our current work we have fixed this number at 10. In the final system we will increase it up to 20 as the radar we are using can measure the coordinates of a maximum of 20 targets. Hence this block will use 20 similar filters in final system.

At start up, at most 10 of the *"incoming observations"* would simply pass through the *"Gate Checker", "Cost Matrix Generator"* and *"Assignment Solver"* on to the filters' inputs. The filter takes an observation as an *"inaccurate"* representation of the *"true state"* of the target and the amount of inaccuracy of the observation depends on the

measurement variance of the sensor. The filter then estimates the current state of the target and predicts its next state before the next observation is available. To estimate the true state we need a *process model*, a *measurement model* and an *estimator*.

5. CONCLUSION

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 Multiple Target 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, the pedestrian detection system of Multiple Target Tracking is faster compared with the other detection systems, its detection performance compares very favorably to the state with a true positive rate of more than 0.673 at a FPPF of only 0.046.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Andreas Møgelmose, Antonio Prioletti, "Two-stage Part-Based Pedestrian Detection", Anchorage, Alaska, USA, September 16-19, 2012.

[7] Antonio Prioletti, Andreas Møgelmose, "Part-Based Pedestrian Detection and Feature-Based Tracking for Driver Assistance: Real-Time, Robust Algorithms, and Evaluation", Ieee Transactions On Intelligent Transportation Systems, Vol. 14, No. 3, September2013.

[8] Jehangir Khan, Smail Niar, Atika Menhaj, Yassin Elhillali "Multiple Target Tracking System Design for driver Assistance", Université de Valenciennes et. du Hainaut cambresis, France.

[9] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. Gavrila, "Multi-cuepedestrian classification with partial occlusion handling," in Proc. IEEEConf. CVPR, Jun. 2010, pp. 990–997.

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