Robust Vehicle Detection and Distance Estimation Under Challenging Lighting Conditions

Mahdi Rezaei[1], *Member IEEE*, Mutsuhiro Terauchi[2], *Member IEEE*, and Reinhard Klette[3]
[1] Department of Computer Science, The University of Auckland, New Zealand
[2] Department of Psychology, Hiroshima International University, Japan
[3] School of Engineering, EEE Department, Auckland University of Technology, New Zealand

***Abstract*—Avoiding high computational costs and calibration issues involved in stereo-vision based algorithms, this article proposes real-time monocular-vision based techniques for simultaneous vehicle detection and inter-vehicle distance estimation, in which the performance and robustness of the system remain competitive, even for highly challenging benchmark datasets.**

The paper develops a collision warning system by detecting vehicles ahead, and by identifying safety distances to assist a distracted driver, prior to occurrence of an imminent crash. We introduce adaptive global Haar-like features for vehicle detection, tail-light segmentation, virtual symmetry detection, inter-vehicle distance estimation, as well as an efficient single-sensor multi-feature fusion technique to enhance the accuracy and robustness of our algorithm. The proposed algorithm is able to detect vehicles ahead both at day or night, and also for short- and long-range distances. Experimental results under various weather and lighting conditions (including sunny, rainy, foggy, or snowy) show that the proposed algorithm outperforms state-of-the-art algorithms.

I. Introduction

REAR-END crashes mainly occur due to driver distraction, drowsiness, or fatigue when a driver fails to keep a safe distance from the lead vehicle(s). According to statistics published in 2012 about traffic safety in the USA, a significant percentage of all traffic accidents involves rear-end crashes [1]. The cited study considers 19 categories of crashes such as rear-end, head-on, guard-rail, crash with animal, crash with pedestrians, or rollover, plus their rate of contribution in terms of total number of accidents, fatalities, injuries, and property loss. Although rear-end collisions show a moderate rate of 5.6% fatalities compared to the other 18 types of crashes, it represents the highest rate of injuries (30.9%), and also the highest percentage of property loss (32.9%) among all types of road accidents in the USA at the reported time.

By maintaining early vehicle detection and warning, it is possible to provide more time for a distracted driver to take an appropriate action to resolve driving conflicts, and consequently to decrease the possibility of rear-end crashes.

Various active safety systems and algorithms have been developed by using computer-vision techniques [2], [3], in particular stereo-vision based techniques [4] which still have some remaining accuracy issues [5], Lidar [6], [7] which can provide accurate range information (however, cost and sparse data collection still appears as a critical issue), or a fusion of multiple sensors such as radar and vision [8], [9], [10] to combine the strength of individual sensors.

Monocular vision-based solutions are a strategy-of-choice if stereo vision, Lidar, or a radar fusion is not possible or not cost-effective, for example in consumer-level mobile devices such as smart phones [11].

Regarding vision-based methodologies [12], current research addresses subjects such as vehicle detection based on analysing shadow underneath a vehicle [13], [14], stereo vision to estimate distances between the *ego-vehicle* (i.e. the car the system is operating in) and obstacles [15], [16], optical flow-based methods to detect moving objects and vehicles [17], application of local binary patterns (LBP) [18], [19], or of Haar-like features [20], [21], [22]. Haar-like features are named after the wavelets of the Haar transform [23], and hereafter we call them *Haar features* in this paper.

In the next section, we briefly describe vision-based techniques when reviewing related work. Although we use only a monocular vision sensor for the research reported in this paper, we introduce an accurate, real-time, and effective vehicle-detection algorithm to prevent imminent accidents under various conditions (described in [24] as *situations*; e.g. day, night, rain, and so forth), also dealing successfully with image noise.

Our algorithm is designed by following two fundamental hypothesizes: **(A)** the idea that despite of vehicles' make, model, or colour, all vehicles at a far distance (Fig. 1, left) have similar features and appearances in common, including occlusion edges between vehicle and road background, different light reflectance patterns on the rear wind-shield compared to the body of a vehicle, a tendency towards a rectangular shape of the vehicle, and a visible shadow-bar under the vehicle's rear bumper; **(B)** for short distances, the situation is different (Fig. 1, right); here, a vehicle shows more details and higher resolution which can be a significantly different appearance to other vehicles, different design and body style, different shape of bumpers, or different tail-light shapes.

To the best of our knowledge, there is not yet any research reported on monocular vision for the detection of vehicles being very close in critical traffic scenes; for example, where a vehicle suddenly joins in at an intersection, or due to a previous occlusion. For such close-distance cases, despite the wide variety of vehicle appearances, all those vehicles still adhere to some common features:

1) a high likelihood of a tail-light pairing;
2) a constrained geometrical relationship between the size and the distance of light pairs;
3) red-colour spectrum range for brake- and tail-lights.

Fig. 1. *Left:* Far vehicles appearing as plain rectangles, with a shadow underneath the vehicle. *Right:* A close vehicle with multiple edges, shadows, and complicated features and details.

The paper proposes a novel hierarchical algorithm that is capable of detecting vehicles both at far and close distances, with a substantial improvement in terms of true-positive detection rate, and a lower false-positive alarm rate. The paper is a significant extension of the conference publication [25].

Figure 2 outlines the main idea of our approach. An "adaptive global" boosted classifier, using a novel type of Haar features, provides initial regions of interest which are further analysed by "feature detection" and "data fusion" techniques for eliminating false-positives as well as retrieval of false-negatives.

The paper is organized as follows: Section II reviews the state-of-the-art and related works done so far. A recently proposed variant of Haar features is adapted for vehicle detection in Section III. Section IV discusses line and corner feature analysis for improvement of the detection accuracy. In Section V, a virtual symmetry detection method is introduced for tail-light pairing. In Section VI, a comprehensive multi-data fusion solution model is provided for final vehicle detection based on the Dempster-Shafer theory. The paper continues with using detection results for distance estimation in Section VII. Section VIII provides experimental results and Section IX concludes.

II. Related Work and Contributions

Computer vision-based methods for detecting and localizing vehicles on the road often suffer either from lack of robustness for complicated road scenes, or from very expensive computational cost. This makes many of the introduced driver assistance system to be non-realistic and impractical.

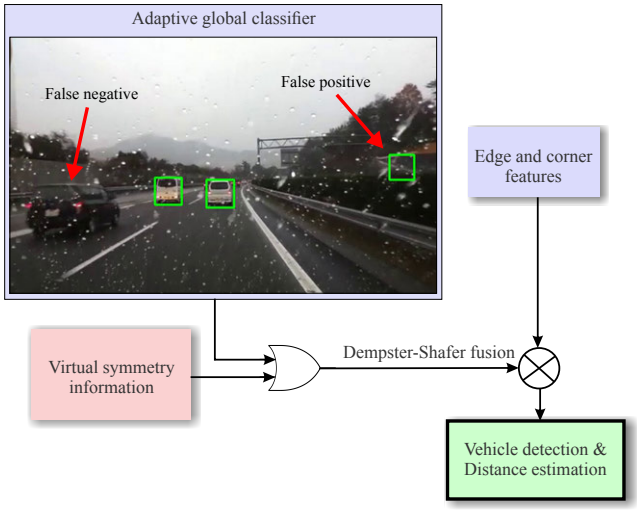


Fig. 2. Brief outline of the proposed detection algorithm, which combines a detection of candidate regions using new types of Haar features with a subsequent analysis of those regions.

A. Related Work

Santos and Correia [26] use a vehicle detection technique after background subtraction based on an estimated (initial) background using a static surveillance camera. This approach is effective in cases such as a parking lot with already analysed parking background; it is not suitable for unknown environments or real-world roads.

O'Malley et al. [27] use simple thresholding for red and white colours to detect taillights. They assume that tail and brake lights in darkness tend to appear as white spots in the video output, surrounded by a red halo region. We consider that this assumption is not necessarily true, as current cameras have auto-exposure control, so they do not capture a white central spot in case of a red light. A second weakness is that this approach only works for night conditions, and a third weakness is that the method only works for the detection of lead vehicles which are levelled to the ego-vehicle; a tilted vehicle (e.g. due to a road ramp, road surface at a curve, or when turning at a round-about) cannot be detected by this approach.

Choi [17] proposes an optical flow-based vehicle-detection method; however, there are many missing detections if the relative speed between ego-vehicle and the observed vehicle becomes close to zero, or the road has a plain texture.

Very recent work by Garcia et al. [9] proposes a fusion technique using radar and optical flow information. While the radar sensor can have multiple detections for the same vehicle, the optical flow technique can only detect overtaking vehicles with considerable velocity differences compared to the ego-vehicle, thus there is the same weakness as in the proposed method in [17].

Haselhoff et al. [10] use a radar sensor to minimize the *region of interest* (ROI) for a detection based on standard Haar features. This can lead to less false-positives; however, there appear to be many weaknesses such as time synchronization issues for radar and vision sensor, or the increasing cost for the system.

Another work by Haselhoff et al. [28] introduces a technique using Haar and triangular features. Reported results indicate improvements compared to a standard detector using Haar features only. Nonetheless, no validation tests and experiments have been considered for night conditions or for other challenging lighting situations.

Huang and Barth [29], and Premebida et al. [6] fuse Lidar data with vision sensor data. The Lidar sensor provides high-resolution but sparse range information with limited performance for object recognition; regarding the vision sensor, vehicle detection is based on Haar features within a predefined ROI calculated in the Lidar data. Such a fusion approach can increase the certainty of the detection compared to a single vision sensor. However, a classifier based on standard Haar features may easily fail in dark, noisy, or other non-ideal lighting situations. Therefore, for such cases, Lidar data could also not help for proper detection in the reported work.

Ali and Afghani [13], and Han et al.[14] provide shadow-based vehicle detection. However, shadows alone are not credible indicators for the existence of a vehicle. A vehicle's shadow varies in size and position; low sun may cause a long shadow, often much longer than the vehicle's actual width, and it falls to the side of the vehicle. This defines challenges for the use of shadows underneath a vehicle. Figure 3 illustrates an example of inaccurate vehicle detection biased by a shadow which is falling to the left. On uneven roads (e.g. up-hill) the shadow underneath a vehicle is often not visible at all.

Nguyen et al. [4] use stereo vision and a genetic algorithm; Toulminet et al. [15] use stereo vision and 3-dimensional (3D) features. Both methods take the advantage of depth information and apply inverse perspective mapping. However, the reported feature detection does not support accurate distinguishing of vehicles from other objects (i.e. false-positives) at night or in complicated road scenes.

Vargas et al. [30] provide a vehicle detection system using sigma-delta-based background subtraction to separate moving vehicles (foreground) from the road (background). The recording camera is fixed (not on a mobile platform). The method is simple and computationally cost effective. It appears to be well-suited for traffic density monitoring. However, the method is not able to identify individual vehicles.

The state-of-the-art general purpose object detection algorithm based on deformable part models (DPM) [31], [32] suffers from inaccurate bounding-box calculation (supposed to indicate the object's actual boundary), and from very high computational costs (typically about 2 seconds per image on a current powerful PC-platform). Both weaknesses can not be ignored in an ADAS application. Figure 4 shows examples of inaccurate bounding box detection based on the DPM.

B. Contributions

In Section IV we discuss the importance of accurate bounding-box calculations as an essential requirement for proper inter-vehicle distance estimation. Also, the high dynamic nature of driving, and permanent risks of a crash, with possible injuries or fatalities, require processing times of only few milliseconds per frame. Although offline or delayed processing is acceptable in many computer vision based applications, implementation of a real-time, feasible, and robust approach is a basic essential for any ADAS.

Fig. 3. Inaccurate vehicle detection based on underneath shadow. Source:[13].

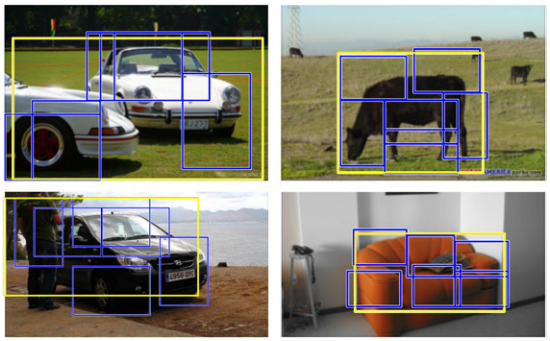


Fig. 4. Unavoidable inaccuracy for objects' boundary calculation based on DPM method [32].

The *histogram of oriented gradients* (HoG) is a common way to derive a descriptor for a bounding box of an object candidate [33], and has also been tested for rear-view vehicle detection by Arróspide et al. [34]. The authors have claimed for inefficiency of symmetric-based approaches for vehicle detection, which is in consistence with our research (that is why we propose the *virtual symmetry* technique in Section V-E as an alternative for symmetric-based approaches). The authors have also reported a detection rate of up to 90% for an HoG-based algorithm under daylight condition. This is far below than what we propose in this paper in terms of both detection rate and the complexity of our experimental database under multi-weather and various lighting conditions.

One of the other important points that has been neglected in almost all related works is that the appearance of a vehicle can highly vary depending on the distance between the observer and the detected vehicle (Fig. 1). This challenge cannot be solved by rotation-invariant or scale-invariant methods, as the appearance of a vehicle at close distance (i.e. a few meters) look completely different to a vehicle's appearance at a distance of e.g. 100 m. Thus, relying on a generic solution for vehicle detection for both short- and long-distances appears to be hard to achieve and non-realistic.

As discussed, there are many publications on general object detection or tracking approaches that are based on *local binary patterns* (LBP) or Haar wavelet classification; however, not many of them can be suitable for highly dynamic and real-time applications such as vehicle surveillance or monitoring. We actually need to incorporate domain specific information from road conditions or vehicles' characteristics to prevent false alarms or missing true detections.

In order to detect vehicles ahead, we use a monocular forward-facing camera that is deployed on the back of the rear-view mirror. The objective is to detect multiple vehicles in a road scene using multiple data clues captured by a single camera. We also propose a solution for distance estimation using a monocular camera.

Challenges that need to be carefully considered are variation in illumination, transition from a sunny scene into shade or a tunnel, light reflections, vehicle occlusions, various lights at night, and the diversity of vehicle types, makes, and models. This creates a high level of complexity which makes feature extraction and vehicle detection as an extremely difficult and unstable task; if the developed methodologies are designed for ideal indoor conditions [35].

Different to other work that puts more efforts into a single solution for vehicle detection, we propose a data fusion approach using edge and corner features in conjunction with our novel boosted classifier called *adaptive global Haar classification* (AGHaar) that is able to detect far-away vehicles at low resolution, and also high-detail vehicles, altogether in a range of about 15 to 100m (like Fig.1, left).

We also fuse temporal and dynamic intensity information and a complementary technique called *virtual symmetry detection* (VSD), that covers vehicle detection at very short distances (as close as 1m) to the ego-vehicle, even when the recorded vehicle occupies the major area of the input sequence (like Fig.1, right).

After the vehicle-detection phase, we perform monocular distance estimation based on a hybrid method combining options inherent to a *bird's eye view* with *pose-based trigonometry*, to be discussed in Section VII.

III. Adaptive Global Haar Classifier

In this section we introduce two techniques to improve the performance of traditional cascaded classifiers. By extending our previous work, originally developed for face and eye classification [37], [38], we propose a vehicle classifier which is *adaptive* with respect to fast intensity changes and extreme lighting conditions to ensure successful vehicle detection at day or night, also under challenging lighting. Furthermore, we develop a new training phase to create a boosted cascade of weak classifiers based on recently proposed *global Haar features*[36], as an efficient complement for standard Haar features. Both contributions together lead to a *faster* and *more accurate* classifier, outperforming the standard classifiers.

A. Global Haar Features

Inspired by ideas in [39], Haar features are widely used for solving various object-detection problems (e.g., [40], [22]). The value of such a local feature is defined by a weighted difference of image values in *white* or *black rectangular windows*, efficiently calculated by using an integral image [41].

We introduce the new concept of *global Haar features*, to be used in conjunction with local features. For any given local Haar feature f, we define two global Haar features as follows. Let w_i and b_i be integral values in white and black regions of a given Haar classifier, respectively; thus, $f = w_i - b_i$ is the Haar-feature's value.

We define global Haar features by $F_w = F - w_i$ and $F_b = F - b_i$, where F is the integral value of the whole sliding window in the search image (Fig. 5).

With the global features we extract global intensity information for the sliding window, which can represent, for example, nearly uniform intensities on a road surface (e.g. when there is no other object shown in the reference window), or a nearly constant intensity of a vehicle (e.g. if a vehicle overlaps the reference window).

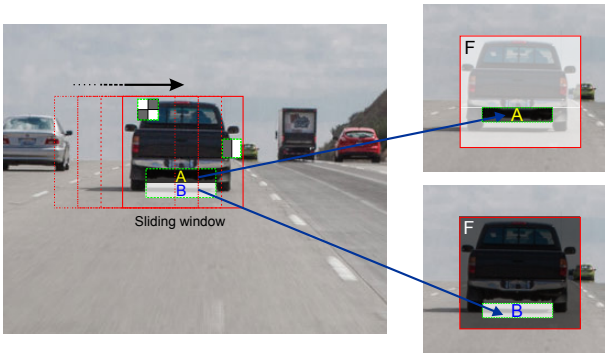


Fig. 5. *Left*: A sliding window with three local Haar features. *Right*: At a given window position, a local Haar feature (defined by white and black regions A and B) extends into two global Haar features by comparing with the sum of image values in the whole reference window.

B. Dynamic Global Haar Features

Based on the method proposed in the previous section, the current local feature is now accompanied by two global feature values, to be used in a weak classifier of the cascade for a given sliding window. In the *dynamic version* of global Haar features, we update F by

$$F_d = F + \sum_{i=1}^{j<=n} (w_i - b_i) \quad (1)$$

where n is the total number of local Haar-like features in the given weak classifier, and j is the current index of the global feature being assessed. Based on the Equation 1, as the input windows progress through the cascade, the value of F is updated to F_d using the global features. We call those *dynamic global Haar-like* features.

Using a boosting algorithm, we can train a cascaded classifier in which each weak classifier can also be accompanied by corresponding global Haar features.

Considering a 50% rejection rate for each stage (each weak classifier), 98.4% of non-objects are rejected within the first six stages ($\sum_{n=1}^{6} 0.5^n = 0.984$). This means, having a minimized number of features in the first six stages plays a crucial rule (i.e. the smaller the number of features in the first six stages, the faster the classifier).

Figure 6-bottom illustrates the case where we trained our classifier with both standard and global Haar features. The figure shows a 33.5% decrease in the total number of features used in comparison to a standard trained classifier (i.e. this also means a 33.5% speed benefit). Similarly, we experienced a 32.2% faster performance within the first six stages.

Positive effects are not limited on saving computation time. Surprisingly (at a first glance), we also experienced more accurate vehicle detection and non-vehicle rejection results. Analysing the situation, this actually makes sense: The new trained classifier is now more confident by using both local and global intensity analysis within the sliding window.

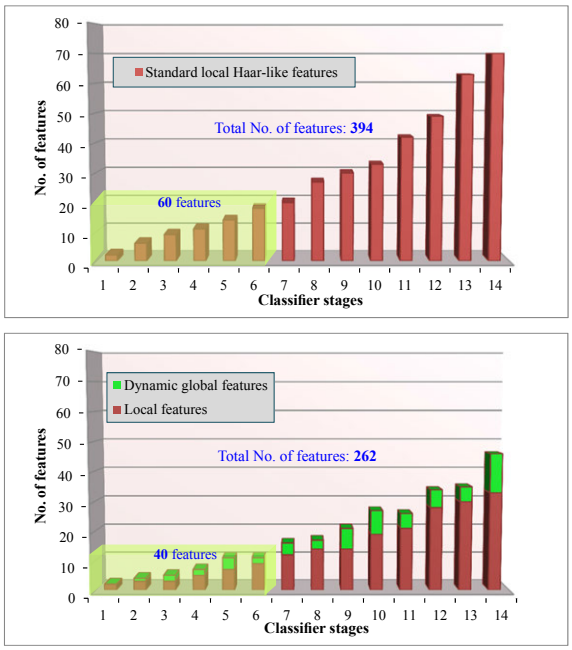


Fig. 6. Total number of features used in a standard Haar-like classifier versus a trained classifier based on both standard and global Haar-like features.

C. Boosted Cascade

In this section we provide the details of the training algorithm. It is a common practice that every stage of a cascaded classifier should reject 50% of the negative samples which passed the previous stage of the cascade, while the true-detection rate remains close to optimal. For the used local Haar features we decide on a case-by-case basis whether we also include the corresponding global Haar features.

One approach is preservation of the current global feature and a search for the next local feature, without considering the effect of the current global feature. If a candidate global feature shows a better rejection rate then it is efficient to choose the candidate feature as the desired global feature, then search for the next local feature. Also, if rejection rates become equal or near to equal, global features are preferred.

For pseudocode for learning a cascade, see Algorithm 1. Applying the learning process, the following weak classifiers are obtained, where the pairs (ϕ_b^k, ϕ_w^k) denote global features:

$$(\theta_l^k, (\phi_b^k, \phi_w^k)), \ldots, (\theta_l^n, (\phi_b^n, \phi_w^n)) \quad (2)$$

We observed that when not using dynamic global features, the number of global features selected during a cascade design is insignificant, so their effect is not noticeable. By using the dynamic global features, the number of global features selected was noticeable and significantly improved the performance of our detector in terms of detection rate, average numbers of features in a window, and early false-alarm rejection. Consequently, this improvement also results in a speed-up.

D. Classifier's Parameter Adaptation

In addition to parameters that affect the training phase of a classifier, there are also parameters which need to be defined during the execution (running step) of a classifier. The main parameters are: initial sliding window size (SWS), scale factor (SF) which specifies the rate by which SWS increases in each new iteration of the search, and the minimum number of neighbours (MNN) which is required to confirm multiple neighbour detections as a single object.

In a recent study for eye and face detection under challenging conditions [38], it is mentioned that even for the same trained classifier there are no constant and optimal parameters; the parameters can be highly different depending on the mean intensity of a scene and the nature of the query object. We apply a similar approach for dynamically analyzing the intensity of the road and the sky to pursue efficient vehicle detection both in day or night; see Fig. 7 for an illustration.

Instead of assigning fixed values for SWS, SF and MNN, we decide for having those parameters to be time variant and adaptive depending on the overall intensity of the current input frame and temporal information. For example, for low light conditions the MNN should have a smaller value than for ideal lighting conditions, because a classifier has a reduced chance of multiple object detections under dark conditions than for day (ideal) light conditions. The question to be answered remains that what should be our reference for measuring the overall light intensity in an input frame?

Considering a dynamic and complex road scene with different intensities due to sun, street lights, vehicles' lights, driving below trees, traffic lights, as well as shadows from moving vehicles, trees, or traffic signs, we need to determine the condition we are driving e.g., on a sunny day, in a tunnel, at night, or in the shade.

To assess the road intensity, we cannot simply use the mean

Algorithm 1 Learning weak classifiers by using local and dynamic global Haar features.

Input: N_p positive samples; N_n negative samples.
Initialisation: Let $F_w = F_b = F$, where F is the sum of intensities in the whole window. Let $k = 1$.
Output: $(\theta_l^k, (\phi_b^k, \phi_w^k)), \ldots, (\theta_l^n, (\phi_b^n, \phi_w^n))$.

1: Find the k^{th} local weak classifier θ_l^k with threshold $T_l^k = \sum_{i=1}^{m_k}(w_i - b_i)$; where m_k is the total number of local features in the k^{th}classifier.
2: Find the next $(k+1^{th})$ weak classifier θ_l^{k+1};
3: Find the k^{th} pair of global weak classifiers ϕ_b^k and ϕ_w^k, corresponding to the black and white parts of the local Haar feature, respectively; set $T_b^k = \sum_{i=1}^{m_k}(F_b - b_i)$, and $T_w^k = \sum_{i=1}^{m_k}(F_w - w_i)$;
4: Decide to choose best classifier(s) among (ϕ_b^k), (ϕ_w^k), and θ_l^{k+1};
5: **if** a global classifier is selected **then**
6: update the values of F_w and F_b as: $F_w = F_w + w_i$, $F_b = F_b - b_i$;
7: Set $k = k + 1$, find the next local weak classifier θ_l^k;
8: Go to Step 3;
9: **else**
10: $k = k+1$; add θ_l^k to the cascade and search for next local weak classifier θ_l^{k+1} ;
11: Go to Step 3;
12: **end if**

Fig. 7. Intensity measurements for road and sky regions under day or night conditions.

intensity of the input sequences. Figure 7 illustrates how we deal with intensity analysis by segmenting a road scene into two parts- "sky" and "road".

After analysing mean intensity and standard deviation of 1680 road and sky segments from 280 diverse samples of road scenes, taken under different weather and lighting conditions, we noticed that the top 5% of the sky region, and the bottom 5% of the road region normally provide acceptable intensity information about the whole scene, which also falls within the given intensity standard deviation.

Therefore, for any unknown scene, we apply 4-point intensity sampling at the expected sky and road regions, as per sampling regions S_l and S_r, and R_l and R_r shown is Figure 7. We use $20\times h/20$ and $w/20\times 20$ patches where w and h denote the width and height of the input sequence, respectively. Then, depending on the identified lighting situation (e.g. day, night), we can adaptively adjust the classifier parameters for more efficient vehicle detection.

Since a strong reflection spot, street lights, or a very dark shadow may fall in one or a few of those four patches, we applied a hybrid intensity averaging including standard *mean* and *mode* ($\mathtt{Mo}$) to make sure we are measuring a balance of actual intensity in the whole scene as per below:

$$I_s(\lambda) = \frac{1}{2}\left[\left(\lambda\cdot\mathtt{Mo}(S_l)+\frac{(1-\lambda)}{m}\sum_{i=1}^{m}S_l^i\right) + \left(\lambda\cdot\mathtt{Mo}(S_r)+\frac{(1-\lambda)}{n}\sum_{j=1}^{n}S_r^j\right)\right] \quad (3)$$

where $I_s(\lambda)$ is the hybrid intensity value of the *sky* region, and m and n are the total numbers of pixels in S_l and S_r.

Figure 7, on the right, shows the obtained segments of the sky and road. Dark-blue and light-blue segments are detected based on mean intensity measurements of S_l and S_r, with a variation of ± 10. Similarly, the green segments show the road surface based on R_l and R_r.

In the shown example of a night scene (Figure 7 bottom left), despite an expectation of dark pixels, some bright pixels fall into the S_l region; this influenced our mean-intensity measurement via the left patch of the sky; consequently, a dark blue segmentation (bottom, right) shows regions around the street lights, instead of being light-blue as the sky in general.

On the other hand, the measurement in S_r supports accurate segmentation of the sky shown as the light-blue segment.

The *mode pixel value* (i.e. the pixel value with the highest frequency of repetition in $S_l \cup S_r$) determines which of the resulting segments (i.e. light-blue or dark-blue) is a better representative of the sky intensity. By assigning λ a value of 0.66, we consider a *double importance factor* for the detected mode intensity compared to a standard mean; this consequently reduces the negative impact of any inappropriate segmentation. In other words, for the night scene shown at the bottom of Fig. 7, the final value of $I_s(\lambda)$ is automatically much closer to the intensity of light-blue segments rather than to that of the dark-blue segments. A similar approach is applied for road background intensity evaluation, $I_r(\lambda)$, which is shown by dark- and light-green segments.

As a final stage for defining the adaptive Haar-feature based detector, we experimentally adjusted 10 sets of optimized values for classifier parameters SWS, SF, and MNN based on 10 intensity values of $I_s(\lambda)$ and $I_r(\lambda)$ for the upper (sky) and lower (road) part of the test video sequences.[1] This parameter adaptation is then extended for the whole intensity range of 256 values based on a cubic interpolation [42].

IV. Line and Corner Features

Using the same training dataset, we created three vehicle classifiers using LBP, Standard Haar, and AGHaar features. The training dataset contained 4,637 rear-view annotated vehicles from 1932 frames extracted from Caltech dataset [43], MIT CBCL database [44], EPFL dataset [45], and KITTI dataset [46]. Samples of vehicle detections are shown in Figure 8. The proposed AGHaar classifier provides more accurate vehicle detection, clearly outperforming LBP and standard Haar classifiers. However, we still consider those initial detections by AGHaar as being *vehicle candidates* or *ROIs* only. In order to have even more accurate results (i.e.

[1]Instead of 10, it could be a larger number of sets. The more sets the better the interpolation results. The number 10 proved to be sufficient for obtaining a smooth and acceptable interpolation.

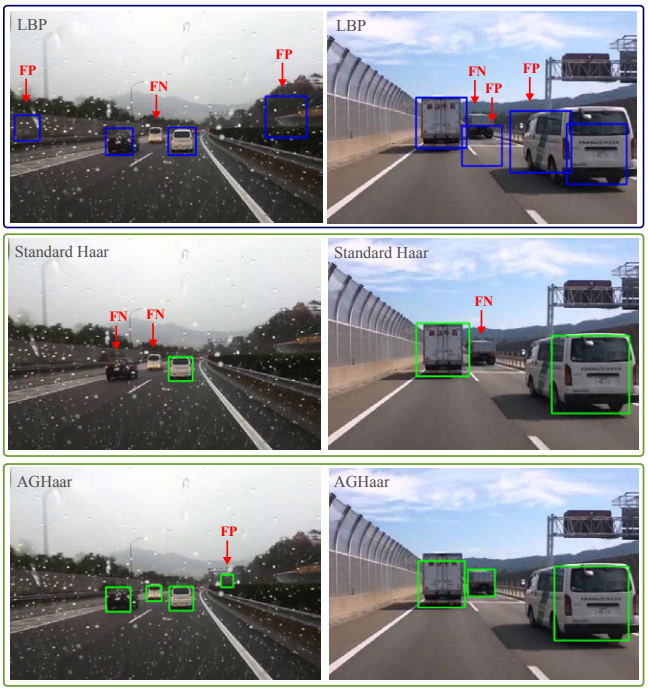


Fig. 8. Samples of vehicle detection based on LBP, Standard Haar, and AGHaar classification.

less false-positives) we continue our evaluation by analysing line and corner features before confirming a ROI is a vehicle.

A. Horizontal Edges

Instead of (e.g.) shadow analysis as illustrated in Fig. 3, we take *parallel horizontal edges* into account as a more credible feature for pointing to a possible existence of a vehicle in a ROI. We hypothesise that horizontal edge features can be perceived due to depth differences between bumper and body of a vehicle, edges around a vehicle's registration plate, or horizontal borders of wind-shields.

We apply the *progressive probabilistic Hough transform* (PPHT) [47] for fast and real-time detection of horizontal lines only. The PPHT was designed following the *standard Hough transform* (SHT) as introduced by Duda and Hart [48]. Detected edge pixels in xy-space are transformed into curves in the *Hough space*, in its discrete version known as *accumulator space*.

In case of the PPHT, a voting scheme is applied to tackle the high computational cost of the SHT. While in the SHT all edge pixels are mapped into the accumulator space, the PPHT only votes based on a fraction of randomly selected pixels. There is one voting bin for each line candidate, and a minimum number of pixels (i.e. of votes) is considered as a threshold for detecting a line. For shorter lines a higher spatial density of supporting pixels is required, while for longer lines less spatial density of supporting pixels is sufficient. Overall, the PPHT ensures much faster line detection while results are almost equal in accuracy with those obtained by SHT [49].

Figure 9 shows a real sample of an accumulator space for a road scene. The figure illustrates that high accumulator values (red regions) are close to the leftmost or rightmost border at around $-90°$ or $+90°$. This confirms for a road scene that

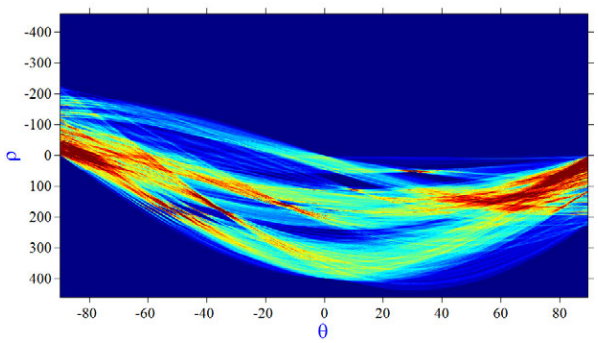


Fig. 9. Edge pixels of a sample road scene mapped into the $\theta\rho$-space. The accumulator values are shown using a colour key where dark-blue is for zero, red is for high values, and light-blue for low positive values.

the number of horizontal lines is considerably higher than the number of lines into other directions. For detecting horizontal lines $y \approx const$ we define two *ranges of interest* for θ:

$$1. \quad 90° - \tau \leq \theta \leq 90°$$
$$2. \quad -90° < \theta \leq -90° + \tau$$

Note that because ρ is considered in PPHT for positive and negative values, θ is only in the range between $-90°$ to $+90°$.

Mapping back from Hough space to Cartesian space, Figure 10-right shows detected horizontal lines for the road scene already used for Fig. 9. As illustrated, we can expect one or more horizontal lines for every visible vehicle in a road scene.

B. Feature-Point Detection

Figure 10, right, also illustrates that there might be a few more horizontal lines which do not belong to vehicles, for example due to shadows (of vehicles or trees), clouds, or rectangular traffic signs (e.g. large boards). However, shaded regions or traffic signs usually have a plain or simple texture. In order to prevent false detections, we also considered analysing corner feature-points in the scene.

Our experimental studies indicate that vehicle regions have typically a much higher density of corner-points comparing to the road, sky, or other background regions (Fig. 11). The visual complexity of a car's rear-view is defined by combinations of a registration plate, tail-lights, a bumper, and the vehicle body. This complexity defines typically significant corners for a vehicle, especially at regions below the back wind-shield.

Among developed feature point detectors such as FAST [51], ORB [52], or FREAK [53], we obtained the best performance with the *Shi-Tomasi* method [54] for detecting "appropriate" corner points in our road scene application context.

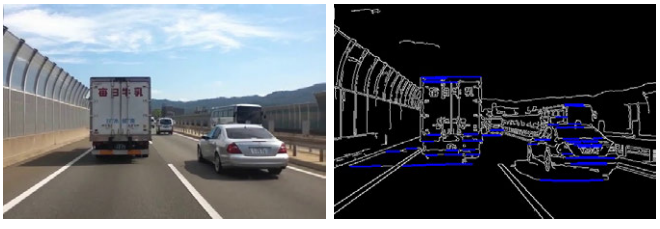


Fig. 10. Horizontal line detection by our customized PPHT.

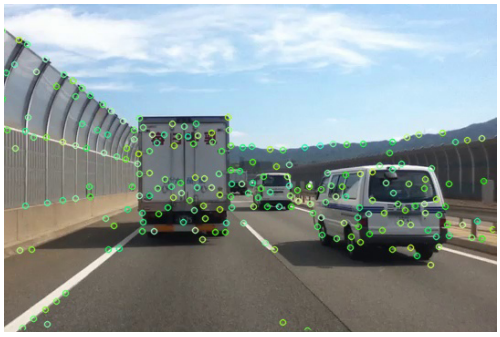


Fig. 11. Detected corner points are considerably more dense in vehicle's rear regions.

Fig. 12. Failed detections or false positive detections for close-up vehicles. *Left:* Haar-based detections. *Right:* LBP-based detections.

Figure 11 shows the detected feature points. This method provides the expected results of higher feature point densities in lower parts of a vehicles' rear view, especially around the registration plate, the bumper, tail-lights, or tires.

So far we discussed three possible clues needed to confirm an ROI as a vehicle: An initial AGHaar detection, horizontal edges, and corner features in lower body part of a vehicle. We use all these clues, In Section VI, to prevent false positives, hence more accurate detections.

V. Detection Based on Tail-lights

In contrast to the previous section that mainly focused on methods for preventing false-positives, this section proposes a method to retrieve missing true detections when applying the AGHaar method. Any classification technique not only needs to be robust for detecting vehicles being at a medium to far distance to the ego-vehicle, it also needs to deal with cases where a vehicle suddenly appears very close to the front of the ego-vehicle (e.g. at a road intersection, or after a temporary occlusion). Tail-light features provide very robust support for very close to mid-range distance vehicle detections.

A. Discussion

Generally a trained classifier can detect the vehicles which have similar appearance-features to the training database images. A strong classifier which is made up of cascade of weak classifiers tries to learn a limited number of (e.g. gray-level Haar) features among a training database in order to detect the given object (e.g. a vehicle).

That is why the classifiers can be more efficient for medium- and far-distances because all vehicles at such distances look like similar as a rectangular patch with a few "limited" and "common" features, like windscreen rectangle part at top, rectangular lights on the vehicle leftmost and rightmost sides, or a bumper in the lower body part.

If we look at vehicles at relatively far distances, due to missing resolution all vehicles look like a plain rectangle with few common features. For such cases, our adaptive global Haar-feature based classifier (Section III) is "highly successful" in general, especially when combined with the described filter using line and corner features.

However, for vehicles at close distance we have a different situation. Figure 12 shows a close-up scene of two vehicles as well as missing detections and false detections by Haar and LBP classifiers. Due to the high diversity in vehicle makes and models, a close vehicle provides much more detail, which can be completely different from one vehicle to another vehicle. Such a huge diversity in details and resolution, and inconsistency in vehicles' shape cannot be learned efficiently or handled by a classifier; there would be numerous false positives and false negatives.

Despite of the diversity in appearances of close vehicles, we hypothesize that there are some common "geometrical features" that all vehicles adhere to. Such geometrical features can not fit as few binary features, template, or pattern, so are not applicable for training in e.g. a Haar-feature or LBP based classification; however, we use them for the next step of our approach- *virtual symmetry detection* (VSD).

Visible features for vehicles at close distance are

1) Tail-light colours (all vehicles use a orange-to-red colour spectrum for tail- and brake-light);
2) Tail-light symmetry (tail-lights in a vehicle are symmetric with the same size and shape);
3) Geometric relations (there are some inherent relationships between the size of a vehicle, the size of its lights, and the distance between the light-pairs).

There are few publications about the analysis of tail-lights for vehicle detection. For example, O'Malley et al. [55] propose a method to detect vehicles based on the symmetry of rear red lights using cross correlation for symmetry detection. However, their method is specifically developed to detect vehicles under night conditions, and symmetry detection using cross correlation only works if the recording camera in the ego-vehicle is exactly behind the target vehicle to ensure visible symmetry of the lights.

This should not be neglected that vehicles in other lanes appear at different poses and angles to the recording camera; therefore the rear lights of the same vehicle cannot be necessarily seen symmetric. Similarly, for many vehicle poses (e.g., Figure 12, bottom), the width of the left tail-light is not visually equal to the width of the right light.

In consequence, methods that rely on "actual symmetry detection" often fail in real-world scenarios. In order to cope with this issue, we apply our VSD approach.

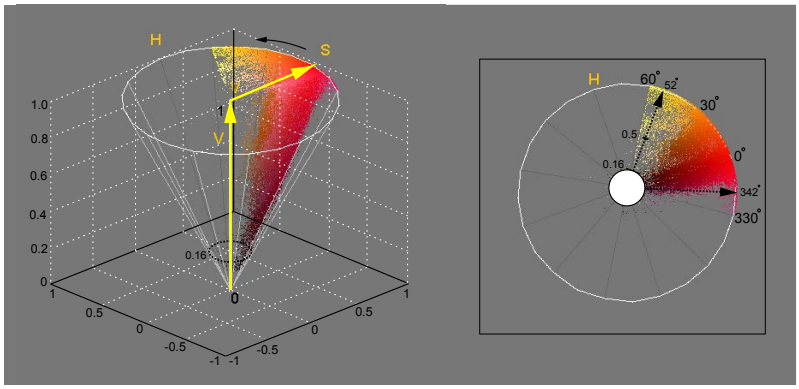


Fig. 13. Extracted colour pixels from the vehicle taillight database. *Left*: HSV conical scatter plot. *Right*: Top view on this 3D plot.

B. Colour Spectrum Analysis

Pursuing the idea of virtual symmetry detection, we created a database of 482 images from tail-lights, brake-lights, indicator-lights, all either in the status of being on or off, under day or night conditions. We converted images from RGB to HSV colour space for a better representation of rear-light pixel colour characteristics. Figure 13 illustrates that the vehicles' rear-light pixel values are scattered in a wide range, from light orange to dark red. This indicates a need for careful considerations, in order to prevent mis-segmentation.

Due to noise in the database images, some pink, black, and yellowish pixels can be seen in the scatter plot (Fig. 13, top view) which are actually do not belong to tail-light pixels. Considering a Gaussian function for the colour pixel distribution in the scatter plot, and excluding the tailed-distribution pixels smaller than -2σ or greater than $+2\sigma$ we remove noisy pixels that have very low density in the scatter plot. Figure 13, right, shows an optimized diagram that excludes noisy pixels with

1) a hue value $H \geq 52°$ (i.e. light yellow pixels),
2) a hue value $H \leq 342°$ (i.e. pink pixels), or
3) an intensity value $V \leq 0.16$ (i.e. nearly black pixels).

The rest of the pixel distribution in the scatter plot is considered to be valid for the tail-light segmentation procedure.

C. Tail-light Segmentation

Figure 14 shows the steps applied for segmentation and pairing of tail-lights. After conversion from RGB to HSV space (Fig. 14, A), we apply pixel matching for all three channels based on information obtained from Figure 13 followed by binary thresholding (Fig. 14, B).

Figure 14.C depicts detected isolated contours. We use chain coding [56] for keeping the original accuracy of contours compared to techniques using encoded contours [57]. Detections are simply based on 8-connected components in the thresholded image.

Figure 14.D illustrates the applied procedure for filling the holes in the binary (thresholded) image, thus creating connected regions. This aims at detecting the actual region of tail-lights if there are missing pixels due to noise or illumination artefacts.

The shown bounding box illustrates the overall width and height of the detected group of contours. Figure 14.E and Fig. 14.F illustrate tail-light pairing and the approximation of the vehicle region, two procedures to be discussed in the next two subsections.

D. Taillight Pairing by Template Matching

Before describing tail-light pairing based on the VSD method, we first discuss potential weaknesses of other methods such as symmetry detection based on template matching (as used in a recent work by Gu and Lee [58]), for a better highlight of the strength of our VSD method.

Let T be a detected tail-light contour in an $m \times n$ window, called the *template*. We search in the $M \times N$ image I for a contour which is similar in shape to the horizontally flipped image of T. As usual in template matching, for each location (x, y) of T (i.e. the location of the topmost, leftmost point in T in I) we calculate a *cross-correlation score*, defining a matrix R of size $(M-m+1) \times (N-n+1)$. Location (x, y) in R which contains the cross-correlation score

$$R(x,y) = \frac{\sum_{i,j}(T'(i,j) \cdot I'(x_i, y_j))}{\sqrt{\sum_{i,j} T'(i,j)^2 \cdot \sum_{i,j} I'(x_i, y_j)^2}} \tag{4}$$

where $1 \leq i \leq m$, $1 \leq j \leq n$, $x_i = x + i$, $y_j = y + j$,

$$T'(i,j) = T(i,j) - \frac{1}{m \cdot n} \sum_{h,k} T(h,k)$$

$$I'(x_i, y_j) = I(x_i, y_j) - \frac{1}{m \cdot n} \sum_{h,k} I(x_h, y_k)$$

with $1 \leq h \leq m$, $1 \leq k \leq n$, $x_h = x + h$, and $y_k = y + k$. We decided to use this particular cross-correlation method due to its accuracy of matching and time performance in the given context [59].

We slide the template T over image I by one pixel at a time, left to right and top to bottom. For every one-pixel sliding, the matrix $\mathbf{R}$ returns a similarity metric by comparing the *sliding patch* (i.e., the template T over the current sub-image).

Figure 15, upper right, illustrates the matrix R as a correlation map for each position of query template T over I. Position (x, y) in the upper-left corner of the patch corresponds to a matching value in the correlation map. The brighter a pixel is at position (x, y), the higher is the level of similarity of I to T at that position. Normalized R returns values between 0 and 1, and any values greater than 0.95 are considered to indicate a

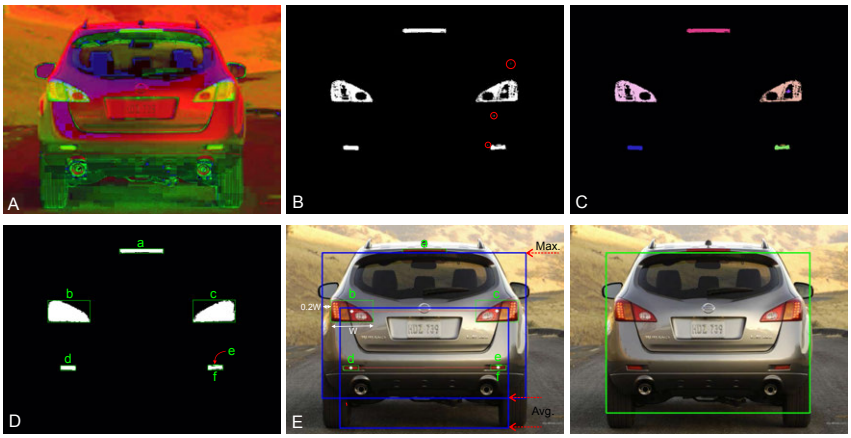


Fig. 14. (A) HSV conversion. (B) Binary thresholded image. (C) Individual contour detection and noise removal. (D) After hole filling. (E) Pairing and vehicle approximation. (F) Final detection.

potential match for tail-light contour pairing. However, Fig. 15, bottom right, illustrates that the result of pairing is not accurate because, due to the camera viewing angle, a pair of lights cannot always be seen as fully symmetric and equal in width and height.

E. Tail-light Pairing by Virtual Symmetry Detection

In this section we discuss on details of the VSD method. For the $a = 6$ detected sample contours in Fig. 14, D, there are $2^a = 64$ different ways for pairing. However, only two of those are correct pairs of tail-lights: $\{b, c\}$ and $\{f, d\}$.

Furthermore, Fig. 14 illustrates that contours C_i and C_j of a pair of tail-lights can be asymmetric, of different width, or of different height. We cannot rely on a strict symmetry; instead, we can define some geometrical rules based on statistical analysis on a rich dataset of tail-light images to manifest a *virtual symmetry* among already detected contours.

Assessing 400 selected vehicle images from KITTI [60] and EPFL [45] dataset, and the measuring baseline size of tail-lights, the ratio of tail-light's width and height, their mean sizes, variances and standard deviations, we identified five optimized rules for virtual symmetry detection.

We consider C_i and C_j as being *virtually symmetric* (i.e. forming a *pair*) if the following conditions are met:

1. $|A(C_i) - A(C_j)| \leq 0.3 \left[\frac{A(C_i) + A(C_j)}{2}\right]$
2. $-15° \leq \alpha(C_k) \leq 15°$, for $k = i, j$
3. $0.9 \cdot (W(C_i) + W(C_j)) \leq |X(C_i) - X(C_j)| \leq 5.3 \cdot (W(C_i) + W(C_j))$
4. $\max(H(C_i), H(C_j)) \leq 1.35 \cdot \min(H(C_i), H(C_j))$
5. $\left|\frac{W(C_i)}{H(C_i)} - \frac{W(C_j)}{H(C_j)}\right| \leq 0.2$

where $A(C)$ is the number of pixels in contour C (i.e. area of C in pixels), $W(C)$ and $H(C)$ are width and height of C in pixels, $X(C)$ is the x-coordinate of the centroid of C, and $\alpha(C)$ is the angle of the main axis of C.

The first condition is only true for contours which have more than 70% similarity in terms of their area. Condition 2 allows a maximum of 15 degrees tilt for each of the two contours (e.g. due to road angle; Figs. 15 and 16). With Condition 3, we make sure that the pair of contours has a baseline distance

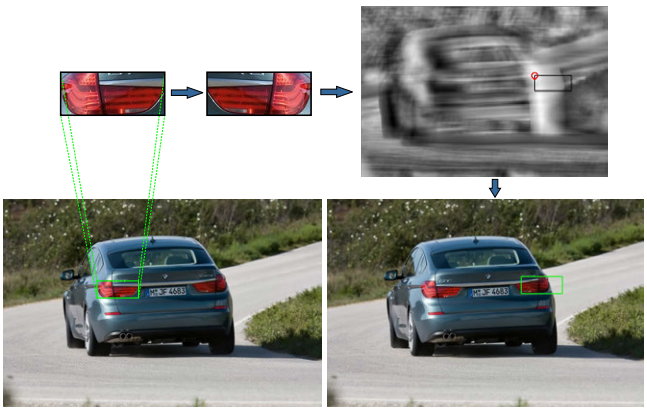


Fig. 15. Query template, correlation map, and template matching over the input image.

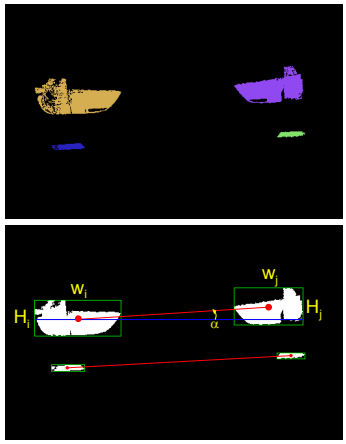


Fig. 16. Virtual symmetry detection (VSD) and tail-light pairing. An example of tilt an variation in size of the same light-pairs, depending on the camera angle and road curvature.

within the measured standard deviation range. By applying Condition 4 we check the height difference between the left and right contour which should be less than 35% (as per the measured mean in the dataset). Finally, Condition 5 compares ratio of width to height of the contour-pair candidates that could not be more than 0.2.

Figure 17 shows experimental results based on tail-light pairing where Haar and LBP classifiers failed to detect those close distance cars. We consider a car region approximation based on the distance between pairs of lights, taking left-most and right-most pixels of detected lights for defining width.

If multiple and parallel tail-lights are detected, such as in Fig. 14, E, a normalization is applied as below:

$$\begin{aligned} X_l &= \min\{x_{l0}, x_{l1}, ..., x_{lk}\} - \gamma \cdot W_l \\ X_r &= \max\{x_{r0}, x_{r1}, ..., x_{rk}\} + \gamma \cdot W_r \\ Y_t &= \min\{y_{t0}, y_{t1}, ..., y_{tk}\} \\ Y_b &= \frac{\sum_{bi=0}^{n} y_{bi}}{k} \end{aligned}$$

where the values x_{li} belong to the left vertical sides of initially detected rectangles, x_{ri} belong to the right sides, y_{ti} belong to the top-horizontal sides, y_{bi} belong to the bottom-horizontal sides, and $\gamma = 0.2$ considers a distance of $\pm 0.2 \cdot W$ as the average left and right margin of the car sides from the tail-light pairs, based on the information obtained from the two datasets discussed above.

Any detection that falls within another detected region is ignored as being a false detection, if its size is much smaller than the larger region (e.g., Fig. 17.D). As per the results shown in Fig. 17, our VSD method outperforms much more accurately and faster than the template matching method discussed in Section V.D.

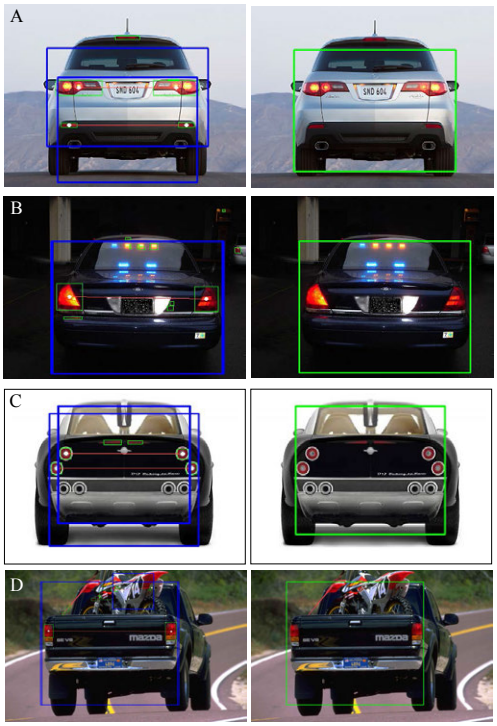


Fig. 17. Experimental results for tail-light segmentation and pairing.

VI. Data Fusion and Temporal Information

This section describes how to combine results obtained by AGHaar classification, horizontal lines, corner features, and virtual symmetry detection. Figure 18 provides an illustration (to be discussed further below). The ultimate goal is accurate vehicle detection. Since the obtained data are derived from a single sensor, obviously time-synchronized, and at the same pose for the different processes, we take benefit of that for the multi-data fusion process. There is no need for time-alignment, data registration, sensor validation, or other challenges that are generally involved in multi-sensor fusion techniques.

We already showed that the novel AGHaar method alone is robust enough in a majority of road scenarios. In order to ensure an even more reliable detection, we apply data fusion for all the available evidences to detect a vehicle, same as what a driver is doing while driving; for example if the full body of a vehicle is not visible in foggy weather, an expert driver may consider looking for a registration plate, tail-lights, or other features of a vehicle to estimate its location and size.

Our fusion approach leads to more accurate results while increasing computation cost only insignificantly. This is not hindering the real-time performance of the whole process.

We considered two possible approaches for data fusion, namely the Bayesian and the *Dempster-Shafer* [61] theory. The Bayesian method interprets weights of input entities as probabilities. The Dempster-Shafer theory (also called *theory of belief*, or *D-S theory* for short) assigns "masses" based on human expertise which only approximate the concept of probabilities. Since the Bayesian approach is based on "pure" statistical analysis, you also need to be "pure" (i.e. very accurate) on providing all statistical data for each source of information. This, consequently, comes with the requirement of a comprehensive initial database analysis among a wide range of recorded videos from different roads scenes. If not doing so, resulting inaccurate weight assignments can cause completely wrong outcomes of data fusion [62].

In contrast to the Bayesian method, the D-S theory is well-known for its effectiveness in expressing uncertain judgements of experts by serving as an alternative method of modelling evidence and uncertainty compared to the Bayesian probabilistic approach. The D-S theory is based on two ideas: *(1)* Defining a degree of belief to identify "subjective probabilities" for a related question, and *(2)* Dempster's rule to combine degrees of belief from independent items of evidence.

By using the D-S theory as a data fusion solution for vehicle detection, we not only consider two categories of "vehicle" and "no-vehicle" but we also assign a degree of belief for an "unknown" status. Considering a mass for the "unknown" status we are adding a safety margin to prevent potentially wrong detections. This automatically takes us to more rational decisions based on a combination of information consensus and human expertise; whereas in the Bayesian technique, we only have two probability values (for "existing" or "not existing"), but not a combination of both.

In the considered context we experienced that a D-S theory-based fusion approach leads to more acceptable results, especially if we have incompleteness of information and a situation

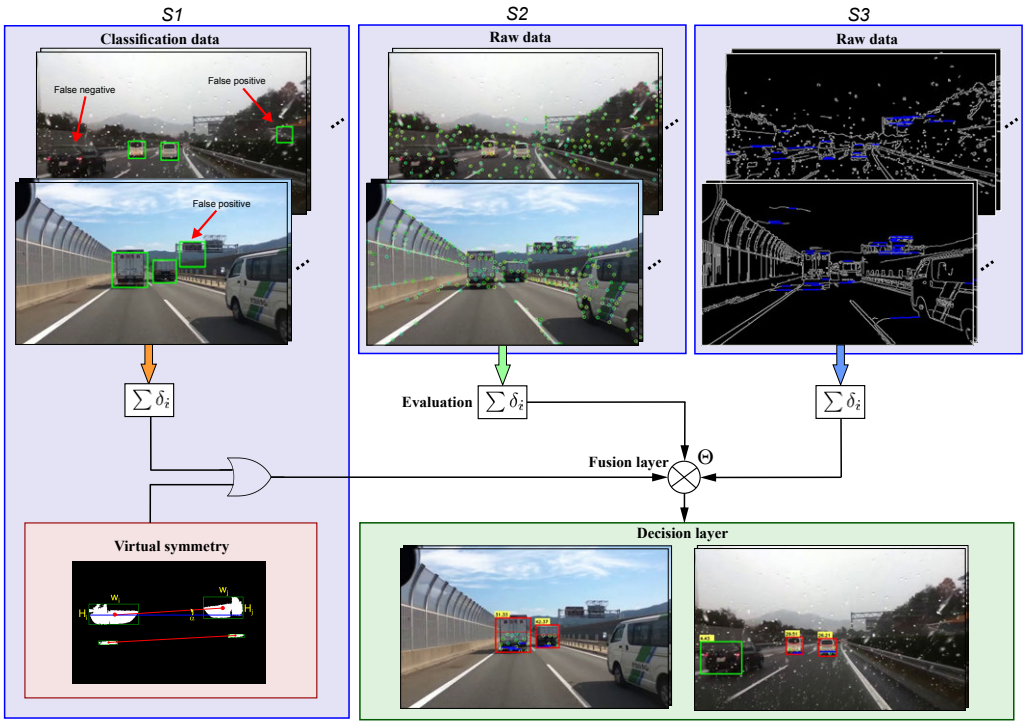


Fig. 18. A single-sensor multi-information fusion framework, showing examples of successful vehicle detection.

where the accuracy of each information source cannot be assured individually.

Let $\Theta = \{T, NT\}$ be the set representing the state of vehicle detection from each of the four available information sources described in Sections III to V (i.e. AGHaar, virtual symmetry, corner features, and horizontal lines) where T represents that a target (vehicle) is detected, and NT stands for non-target (non-vehicles). Each element in the power set $2^{\Theta} = \{ \varnothing, \{T\}, \{NT\}, \{T, NT\} \}$ is considered to be a proposition of the actual state of the vehicle detection system.

Based on the theory of evidence, a mass m_i is assigned for each element in 2^{Θ}, where $1 \leq i \leq 3$ stands for the three main information sources as follows: $i = 1$ is for AGHaar, combined with virtual symmetry (combination details are provided later in this section), $i = 2$ for corner features, and $i = 3$ for horizontal lines. Those three functions m_i are also called *basic belief assignments* for information sources 1, 2, or 3, satisfying

$$m_i : 2^{\Theta} \rightarrow [0, 1] \tag{5}$$

with two properties:

$$\begin{aligned} m_i(\varnothing) &= 0 \\ \sum_{A \in 2^{\Theta}} m_i(A) &= 1 \end{aligned}$$

The mass $m_i(A)$ represents the ratio of all relative and available evidences that support the validity of state A from the i^{th} information source.

For example, considering AGHaar and a VSD combination (AGHaar-VSD) as our main source of vehicle detection (Fig. 18, left), we consider $m_1(T) = 0.75$, $m_1(NT) = 0.15$, and $m_1(U) = 0.1$ which means that we have a belief into the true detection rate by AGHaar-VSD in 75% of all cases, we also have a 15% belief for false detections, and have no opinion in 10% of the cases (unknown assignment) due to lack of knowledge or incompleteness of analysis. Table I summarizes the masses identified based on the accuracy of the AGHaar-VSD classification in our ground-truth test dataset.

Depending on size and distance of rectangular regions selected by AGHaar as vehicle candidates, we expect a number of corners and horizontal lines that fall into the lower part of the ROI if the candidate is actually a true positive (a vehicle).

The closer value to the chosen threshold τ (as defined above), the higher the possibility of being confirmed as a vehicle. In other words, if the numbers of detected corners and horizontal lines is lower than the defined threshold then the D-S framework decreases the level of belief by appropriately decreasing the default masses of m_2(T) and m_3(T), and, on the other hand, it increases $m_2(NT)$ and $m_3(NT)$ to reject false candidates in the fusion process. However, masses m_2(U) and m_3(U) remain always unchanged.

Also, in order to prevent incorrect updates of m_2 and m_3 due to motion blur noise, we apply weighted averaging on the masses by considering the masses allocated for the past n frames (e.g., $n = 30$ in the past second) to use temporal

TABLE I
MASS ASSIGNMENTS FOR THREE SOURCES OF INFORMATION.

Status	Source 1 (m_1) *AGHaar/Sym.*	Source 2 (m_2) *Corner features*	Source 3 (m_3) *Horizontal lines*
T	75%	55%*	65%*
NT	15%	25%	20%
U	10%	20%	15%
Total	100%	100%	100%

* Maximum mass value if features match with threshold τ.

information as well:

$$\overline{m}_i = \frac{\sum_{t=1}^{n} \delta_t \; m_i}{\sum_{t=1}^{n} m_i} \tag{6}$$

Values for n and δ_t may vary depending on the ego-vehicle's speed. n varies between 5 and 30. In low speed scenarios we apply a $5fps$-based weighted averaging and in high speed cases we apply a $30fps$ averaging to ensure we have accurate averaging as the speed increases. δ_t is in the range of $(0, 1]$. In high speed cases due to motion blur effect, we decrease δ_t, and in low speed cases δ_t increases towards 1.

Considering a processing of 30 frames per second, the masses in the past few frames should remain almost close to the actual updated values as per the previous step, or may have only a 'smooth' change. Therefore, if a sudden change happens in the current frame due to considerable noise (e.g. intense light) then the weighted averaging contributes to the masses from temporal information to maintain a moderated mass for the current frame, as well.

Considering the masses m_i as being the confidence value in each element of 2^{Θ}, we measure the combined confidence value $m_{1,2,3}(Z)$ by fusing information from Sources 1 to 3 based on Dempster's rule of combination:

$$\begin{aligned} m_{1,2,3}(Z) &= (m_1 \oplus m_2 \oplus m_3)(Z) \\ &= \frac{\sum\limits_{A \cap B \cap C = Z} m_1(A) \cdot m_2(B) \cdot m_3(C)}{1 - \sum\limits_{A \cap B \cap C = \emptyset} m_1(A) \cdot m_2(B) \cdot m_3(C)} \end{aligned} \tag{7}$$

where $\oplus$ denotes the orthogonal sum which is defined by summing the mass product over all elements in the numerator part whose intersections are $A \cap B \cap C = Z$, and the denominator applies normalization in the range of $[0, 1]$; it shows the amount of conflict when there is no intersection (no agreement) by those individual sources.

Figure 18 shows two examples of fusion results under rainy or sunny conditions based on Dempster's rule of combination.

Detections by AGHaar and VSD are technically independent of each other; however, as discussed earlier, we combine them as information Source 1 in our D-S fusion platform. The combination is represented by the logical symbol of "OR" in Fig. 18 and the same mass m_1 in Table I. In case of an AGHaar failure (missing detections), VSD directly acts along with corner features and horizontal lines. In case of detections by both AGHaar and VSD for the same vehicle, we apply the mean to define only one ROI per vehicle candidate, before going for data fusion with corner and horizontal edge features.

Overall, the defined multi-clue data fusion approach provides more confident detection as well as a reduced rate of false-negatives that may occur due to AGHaar failures.

VII. Inter-Vehicle Distance Estimation

After having vehicles detected, the next step is to label the identified ROIs by an estimate for their distance to the ego-vehicle. Using monocular vision only it is not possible to directly obtain depth and distance information from a road scene. However, after remapping the camera image into a 2D transformed domain we can have a distance estimate based on homogeneously distributed pixel distances in the new 2D transformed imaged. Figures 19 and 20 illustrate our distance measurement techniques.

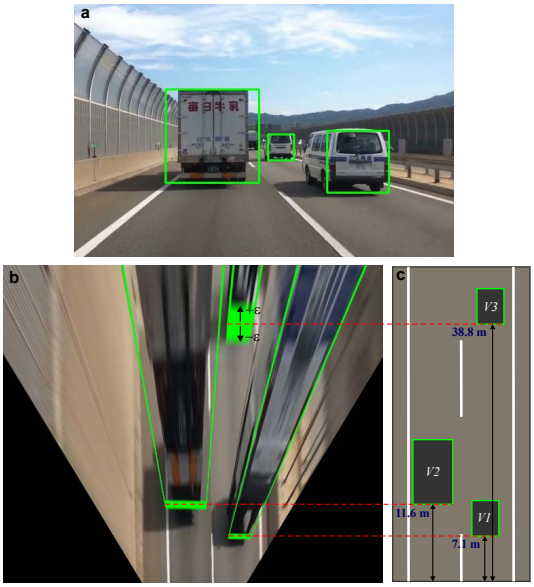


Fig. 19. Distance estimation based on bird's eye view.

Assuming an almost planar road surface, knowing the camera optic parameters, camera high, and camera angle, the *inverse perspective mapping* (IPM) can map the recorded images into a *bird's-eye view* [63], approximating an orthogonal top-down view of the scene. Figure 19 shows a mapping of a recorded image into a bird's-eye view using 4-point calibration [64] and our subsequent distance estimation.

Measuring the pixel-distance from the target vehicle to the ego-vehicle in the bird's-eye view, and comparing it with a ground truth metric for the same camera parameters and camera installation, a distance estimation can be performed as illustrated in Fig. 19, b and c.

Recent work by Tuohy et al. [65] also considers a similar approach for distance estimation; however an important weakness is neglected. We highlight this weakness as per Figure 19, b. Considering the bottom side of a green bonding-box as our distance reference, the bird's-eye view cannot precisely tell where the vehicle is located on the road; especially for distances of more than 30 m such as the vehicle *V3* (Fig. 19, the farther vehicle).

The figure shows that every single pixel in the recorded perspective image needs to be mapped into multiple points in a bird's eye view. This transformation involves interpolation. Our evaluations show that the interpolation errors, as well as errors involved in 4-point calibration stage, cause a distance estimation error up to $\varepsilon = \pm 8\%$. This technique can be suitable for basic driver-assistance systems to prevent imminent crashes; however, as the distance increases, the estimation error can increase exponentially. We aimed at improving this technique such that we have more accurate distance estimates than just using the bird's-eye view.

As illustrated in Fig. 20, we have a forward looking camera

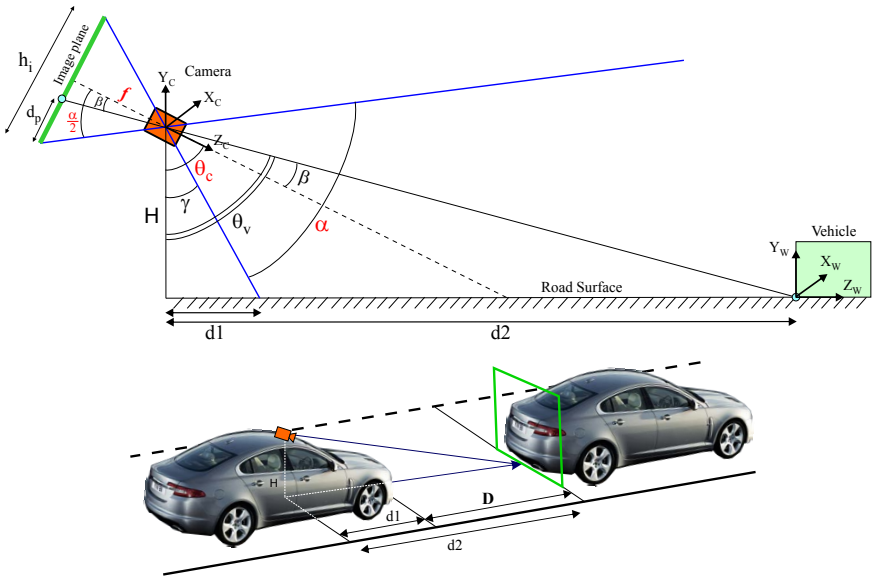


Fig. 20. Real-world inter vehicle distance estimation based on pixel distance information in 2D image plane.

(close to the rear-view mirror), we know the camera field-of-view defined by angle α, the height H of the camera above road level, and the camera angle θ_c in $X_cY_cZ_c$ coordinates.

Assume a detected vehicle in the road scene at an (unknown) position (X_w, Y_w, Z_w). Let θ_v be the angle of a projection ray (for the camera) pointing to the intersection of the planar rear-part approximation of the detected vehicle with the planar road surface (Fig. 20, top). The actual distance D between ego-vehicle and preceding vehicle is equal to $d_2 - d_1$ and can be computed as follows:

$$\begin{aligned} D &= H \cdot \tan(\theta_v) - H \cdot \tan(\gamma) \\ &= H \cdot \left[\tan(\theta_c + \beta) - \tan(\theta_c - \frac{\alpha}{2})\right] \end{aligned} \tag{8}$$

Knowing the θ_c and α values, only β is needed to calculate D. On the other hand, we have that

$$\tan(\beta) = \frac{\frac{h_i}{2} - d_p}{f} \tag{9}$$

were h_i is the height of the recorded image plane (in pixel unit), d_p is the distance from the bottom side of the detected vehicle to the bottom of the image plane (also in pixel unit), and f is the camera's focal length. Also we have that

$$f = \frac{h_i}{2 \cdot \tan(\frac{\alpha}{2})} \tag{10}$$

Finally, including β and f in Equ. (8), the distance D is completed as:

$$D = H \cdot \left[\tan\left(\theta_c + \tan^{-1}\left(\frac{\frac{h_i}{2} - d_p}{\frac{h}{2 \cdot \tan(\frac{\alpha}{2})}}\right)\right) - \tan(\theta_c - \frac{\alpha}{2})\right] \tag{11}$$

If the ego-vehicle's shock absorbers vibrate on an uneven road then H and β may slightly change and negatively affect the actual D value. This is the only known to us weakness of this approach. Weighted averaging on the bird's-eye view and the camera-pose-based trigonometric solution is applied to ensure a more reliable distance estimation. We provide further details in the experimental Section VIII.A.

VIII. Experimental Results

In this section we evaluate the performance of the proposed vehicle detection and distance estimation techniques, for various traffic scenarios, weather conditions, as well as difficult lighting conditions.

Unfortunately, there are only a few basic, publicly available datasets useful for comparative performance evaluation, and the data are mainly recorded in daylight only (e.g. KITTI data), or from some elevated positions (such as from traffic surveillance cameras) which is not applicable in this research.

We used the iROADS dataset, Set 10, Part 1 [66], which is recorded with an 0.7-megapixel camera (1280×720), a

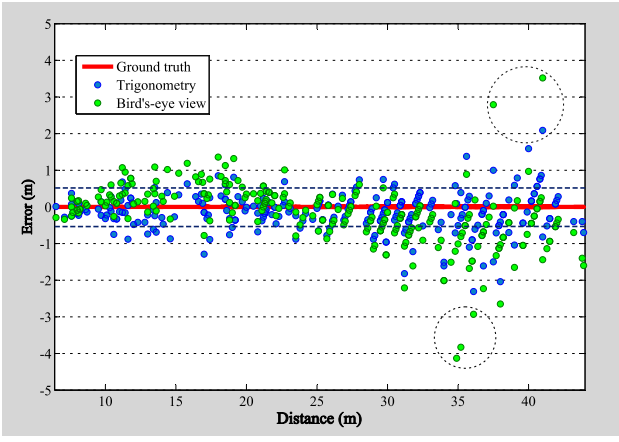


Fig. 21. Distance estimation errors for the bird's-eye view technique or the camera-pose-based trigonometric technique.

Fig. 22. Samples of vehicle detection at close distance with four different approaches.

60° field of view, and 30 fps recording rate, mounted on the back of a rear-view mirror in a car, with a camera tilt angle of $\theta_c = 82^\circ$, and at a height of about $H = 153$ cm above the road surface. Those parameters have been used for comparing ground truth information and distance estimation. We also considered sequences recorded in second part of iROADS dataset [66], with different cameras in different traffic scenarios for vehicle detection.

A. Evaluations of Distance Estimation

We compare distance estimation either based on bird's-eye views, or by the proposed camera-pose-based trigonometric technique.

Japan has one of the highest standards for roads in terms of consistency in road signs and lane markings. We used traffic data recorded in Japan (iROADS dataset, Set 1) [66] to evaluate the accuracy of the distance estimation methods discussed in Section VII. Knowing that the length of any white marking segment in Japan is $8.0m$, and the length of a gap between two white segments is $12.0m$, we extracted ground-truth distance data for about $10km$ of the given road.

Using the proposed fusion classifier for vehicle detection, and knowing the camera assembly and relevant pose parameters, the two distance estimation methods, as discussed in Section VII, have been evaluated.

Figure 21 shows the *distance to vehicles* errors (to vehicles in front of the ego-vehicle), defined by comparing with ground truth represented by the red line. Vehicles are at distances of 6 to $50m$ to the ego-vehicle. We considered a confidence interval of $\pm 60cm$ for ground truth measurements. Distance estimation by the camera-pose-based trigonometric method shows more accurate results compared to the bird's-eye view approach. For the camera-pose-based trigonometric method, the error is mainly within the confidence margin of our ground-truth reference. Interestingly, both approaches show a very similar error behaviour for medium distances (in a range of about 22 to $27m$). The error level increases significantly (up to 9%) for the bird's-eye view technique for far distances.

We identified two common sources of error for both approaches, and a third source of error for the bird's-eye view approach: *(1)* The error of vehicle localization from the vehicle classifier (detector); *(2)* changes in camera height H due to activities of the vehicle shock absorber; and, *(3)* the error for the bird's-eye view technique due to interpolation and 4-point calibration errors as discussed in Section VII.

Dashed circles in Fig. 21 show errors which are considerably different to neighbours. Those cases occurred when the ego-vehicle performed sudden braking, or because of high activities of the shock absorbers (i.e. changes in camera tilt angle and in height H).

Measuring the standard deviation of errors for both techniques, we considered a weighted averaging with coefficient 0.7 for the camera-pose-based trigonometric method, and coefficient 0.3 for the bird's-eye view technique.

B. Evaluations of the Proposed Vehicle Detection

Figures 22, 23, 24, 25, and 27 illustrate our evaluations performed on the combined dataset EISATS, Set 10 (iROADS) [66]. As per the definition of *situations* in paper[24] for variability of traffic or road conditions, a *robust* technique has to perform with reasonable accuracy for different situations. In our experiments we used data for six different situations:

1) *close distance*: up to $1m$ to the ego-vehicle;
2) *day*: Daylight situation;
3) *night*: Evening and night situation;
4) *rainy day*: Rainy weather under daylight condition;
5) *rainy night*: Rainy weather under night condition;
6) *snow*: Snowy situation.

We applied a full analysis on true detection and false-positive rates by comparing LBP classification, standard Haar-like classification, AGHaar classification, and our proposed D-S data fusion approach.

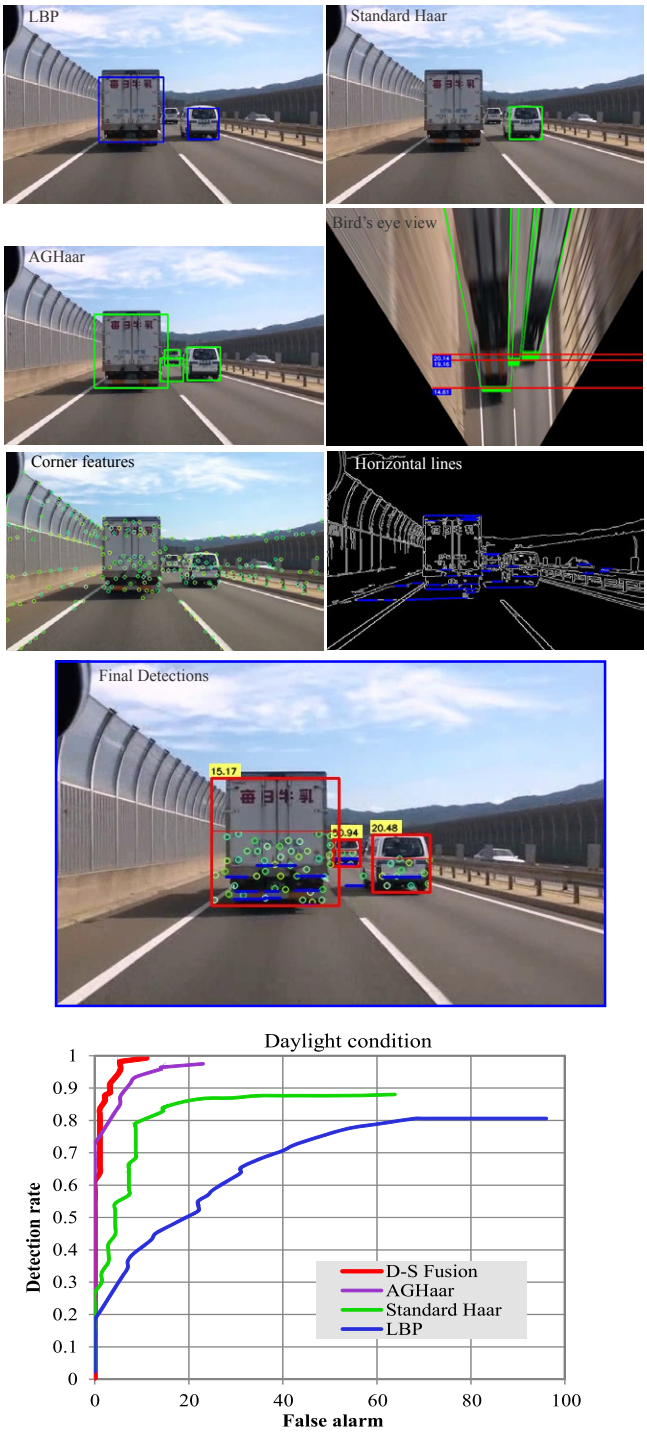


Fig. 23. Vehicle detection, distance estimation, and performance evaluations for *day*. From left to right, top to bottom: The first three images shows detection results for the discussed LBP, HAAR, and AGHaar approaches, the forth image provides bird's-eye view distance estimation for the AGHaar-based image. The fifth image provides corners features, the sixth image illustrates the outcome of horizontal edge detection, and the seventh shows the final results of vehicle detection and distance estimation after the proposed data fusion technique. The estimated distances (in m) are given in the yellow rectangles on the top left side of the red bounding boxes. The bottom image represents the ROC curve and performance evaluation for *day*.

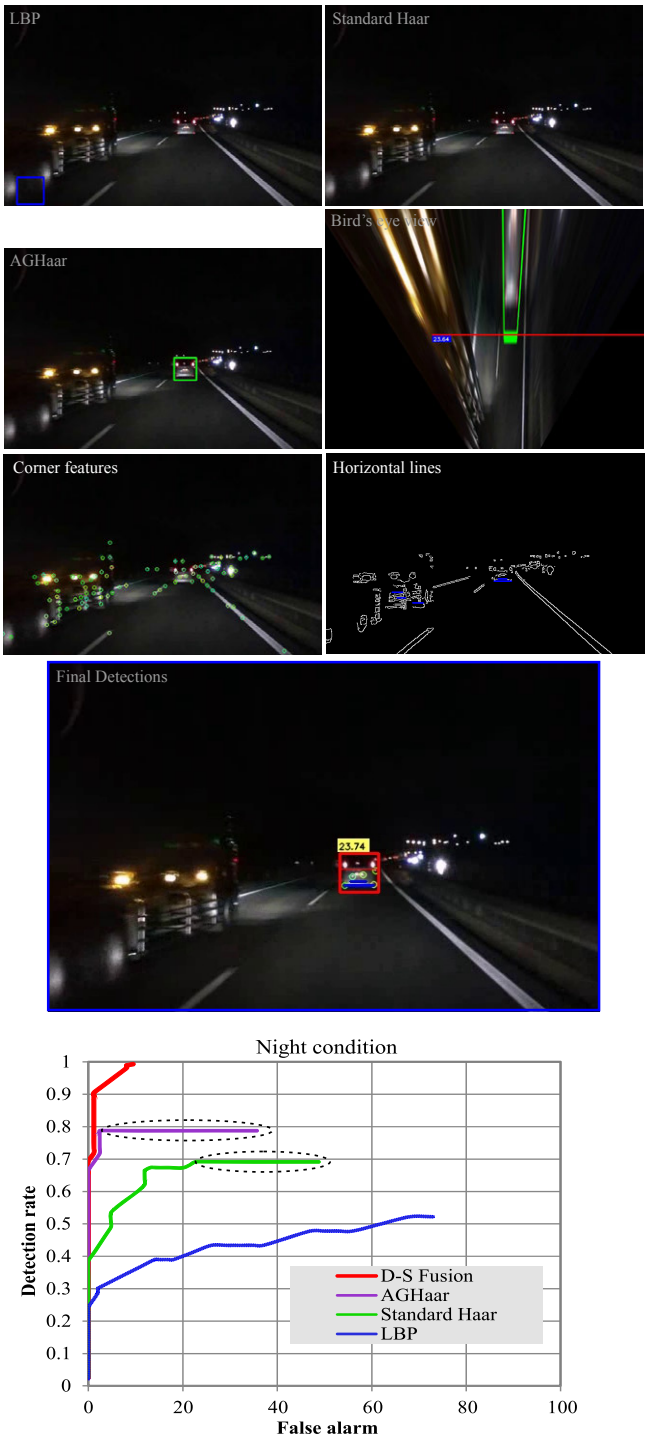


Fig. 24. Vehicle detection, distance estimation, and performance evaluations for situation *night*. Order of images and descriptions as per Fig. 23,

The accuracy and robustness of our detection method has been evaluated on image sequences in the six different situations listed above. First we evaluated close-by vehicle detection based on the VSD approach.

In a database of 500 images, ranging from older to modern models of vehicles, we gained 91.6% true detection, and 1.2% false alarm. Figure 22 shows samples comparing our VSD method to other techniques. As discussed earlier, a weakness of other approaches is that many false alarms and misses of true-positives can be expected. Since the VSD method is only a part of our overall D-S fusion method, we continue with a more detailed discussion of the other five situations.

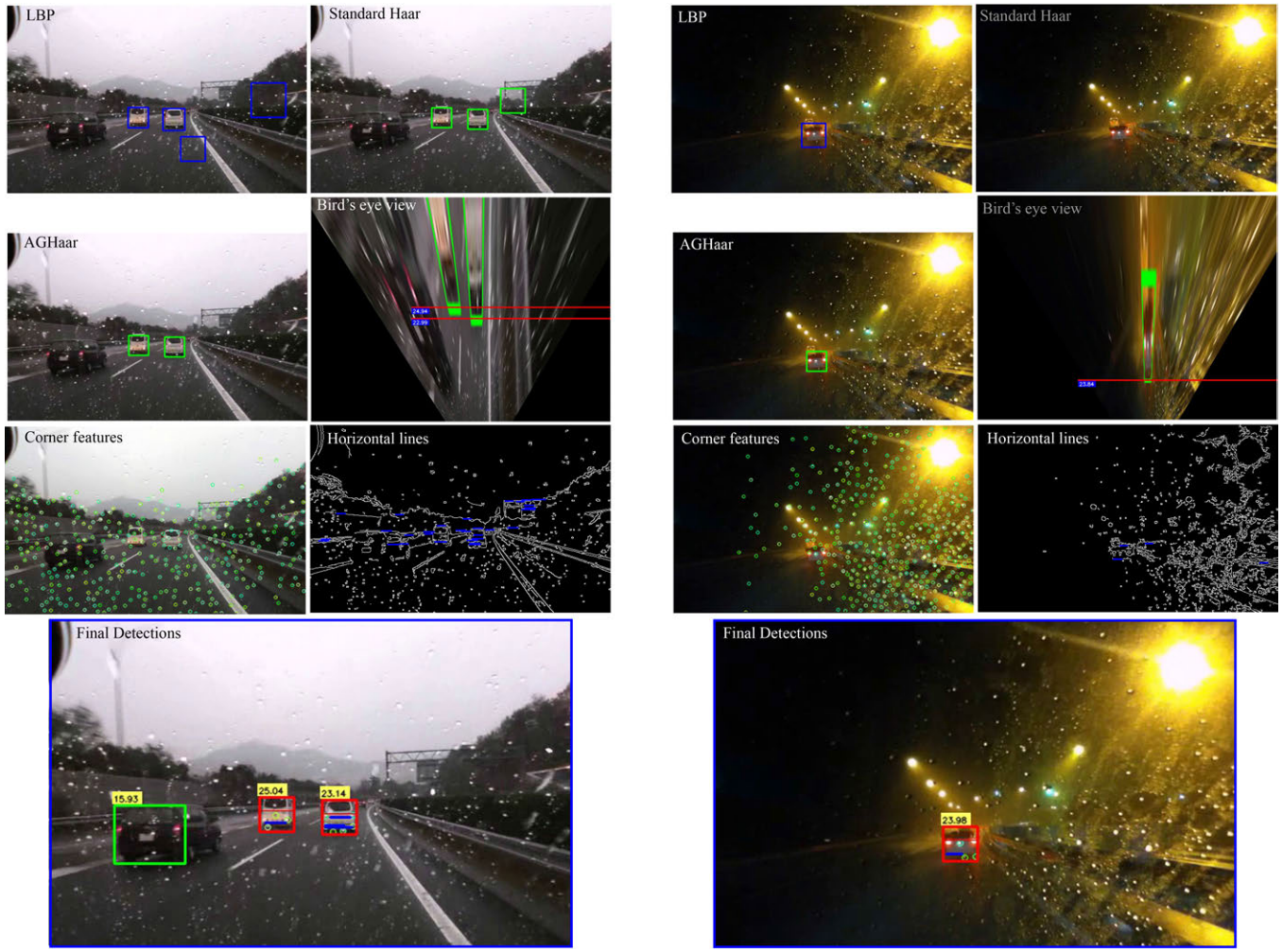


Fig. 25. Vehicle detection, distance estimation, and performance evaluations for *rainy day* and *rainy night*. Order of images and descriptions as per Fig. 23.

Figure 23 shows detection results and *receiver operating characteristic* (ROC) curves for the situation *day*. LBP based classification shows the lowest detection rate and the highest rate of false positives. AGHaar alone and the D-S fusion-based method show relatively similar behavior, better than LBP, while the D-S fusion-based method outperforms the best results with a smaller rate of false alarms. The estimated distances, shown in the bottom image for the proposed approach, are slightly different to those obtained by the bird's-eye view technique, as expected, because of weighted averaging discussed in Section VIII-A.

Figure 24 illustrates experimental results for situation *night*. The figure shows that LBP and the *standard Viola-Jones method* (standard Haar) perform weak under night conditions. Also, the horizontal sub-curves in standard Haar and AGHaar curves (dashed ellipses) in ROC plot show that those algorithms have no success for some parts of the test dataset. Those parts of the curves represent cases where only false alarms or no true detections occur.

The LBP detector shows a detection rate as low as 52% with a considerable number of false detections. Overall, the night-condition graph shows lower performance result for LBP, standard Haar, and AGHaar compared to their corresponding ROC plots for day-light condition. This makes sense as it is less likely that the classifiers can capture the relevant features at low-light or night conditions. However, due to VSD and our D-S fusion techniques, the fusion-based ROC curve shows a very good detection rate (close to 0.99) with a very small rate of false alarms in both situations- *day* and *night*.

Figures 25 and 26 provide samples of results for rainy day and rainy night conditions. Those situations are challenging. For example, Fig. 25 shows that there are many false alarms for LBP and standard Haar methods, as well as some missing detections for AGHaar. However, the bottom image shows perfect detections after incorporating VSD and D-S fusion techniques. The green rectangle shows a detection after tail-light pairing and VSD.

In contrast to results for the situation *day*, for the situation *rainy night* the AGHaar method did not perform visibly better than standard Haar. This is mainly due to reflections of street lights on rain droplets (see Fig. 25, top) constituting strong noise than can consequently lead to false alarms. However, again the D-S fusion method shows still a high true-detection rate, almost as good as for situations *day* or *night*, with only a minor increase in false alarms (raised from 10 to 19) which is a very small portion considering the total number of true detections in our test dataset.

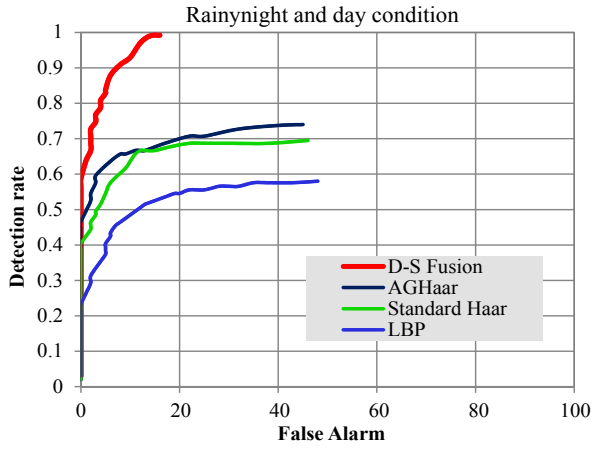


Fig. 26. Performance evaluation for situations *rainy day* and *rainy night*.

Figure 27 shows detection results for situation *snow* and provides the ROC curves for our sixth dataset containing 1,200 frames from snowy road scenes. Under these conditions, LBP shows a significant increase in its false alarm rate while keeping the detection rate just below 70%, which is an average performance. Standard Haar shows purer performance than for *rain* situations, *night*, or *day*. On the other hand, interestingly, AGHaar performs considerably better, showing effectiveness of our adaptive global Haar-like classifier for challenging lighting conditions and dynamic environments.

With the D-S fusion approach we had a detection rate of close to 1.0 in the previous four situations. For situation *snow*, the detection rate stops at 0.88 (almost the same as for AGHaar) but it also shows a reduction in the number of false alarms. This can be due to a significant variation in illumination for the dark grey appearance of the road surface, in contrast to the bright white surrounding covered by snow. This may cause strong sunlight or street light reflections, camera blooming, thus difficulties for a better detection performance.

Table II summarises the precision rate and recall rate for the proposed method on the four discussed individual datasets plus a comprehensive mixed dataset including all-weather conditions, challenging lighting conditions, and close-distance vehicles. Although the standard classifiers can gain up to around 90% recall rate for ideal daylight conditions, their detection rate dramatically decreases to under 60% on a real-world challenging and comprehensive dataset.

Except for close distance datasets, AGHaar shows a visible improvement compared to other standard classifiers. The data fusion method is the best performer with a detection rate of 96.8% for the multi-weather and lighting dataset, with a very high precision rate of 95.1%.

In the case where the vehicle's rear view is occluded by other vehicles or obstacles, it is possible to detect the vehicle as long as the tail-light pairing satisfies *virtual symmetry*, even when two preceding vehicles are near each other and at the same distance to the ego-vehicle.

IX. Concluding Remarks

The research proved that even for a specific rear-view vehicle detection, we need to deal with a high dimensions

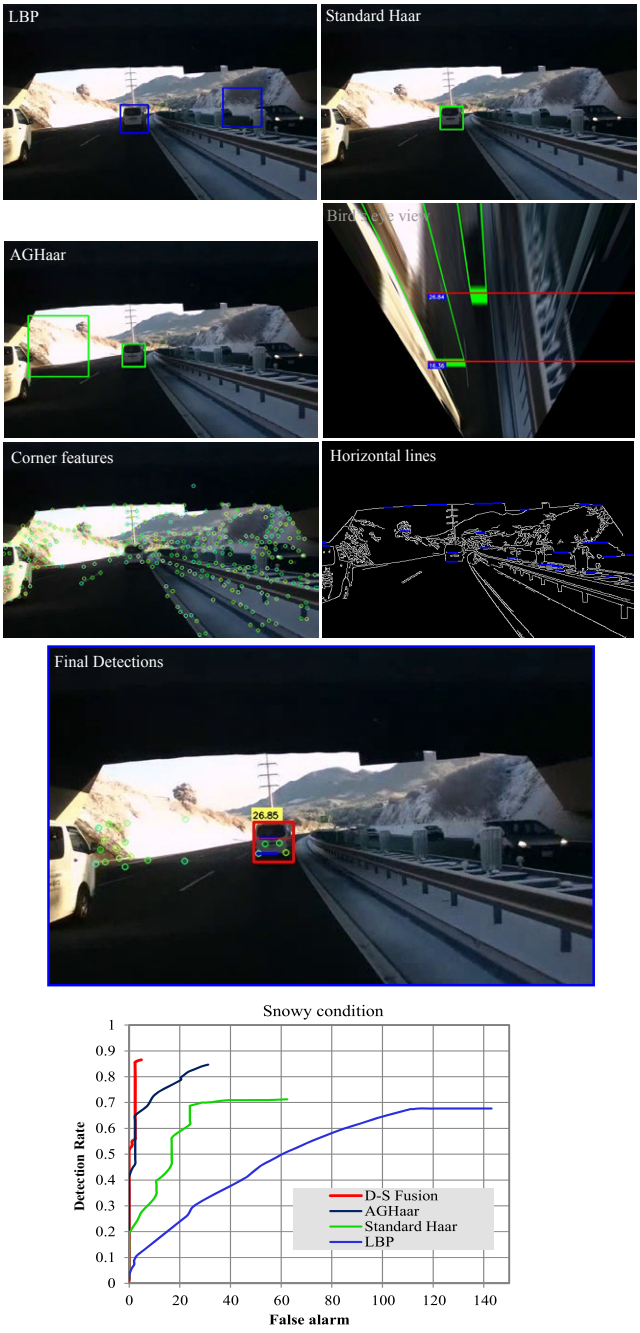


Fig. 27. Vehicle detection and distance estimation in situation *snow*. Order of images and descriptions follow Fig. 23.

of parameters and challenges to obtain a robust result. If the research supposed to approach for a general vehicle detections, such as multi-direction vehicle detections, the results could have not yet been acceptable. For example, the latest achievements and state-of-the-art work listed on the KITTI benchmark website results show very low detection rates ranging from 18.4% to 74.9% for multi-view vehicle detection, even under (ideal) day-light conditions [46]. These results are still far satisfying from needs of real-world applications or industry expectations. We also discussed bounding box inaccuracy and extremely high-computational cost of DPM, the state-of-the-art object detection method proposed by Felzenszwalb et al. [31], [32].

In our research we focused on a specific detection scenario,

TABLE II
PERFORMANCE EVALUATION OF THE PROPOSED METHODS ON FIVE INDIVIDUAL DATASETS AND ONE MIXED COMPREHENSIVE DATASET.

	LBP		**Standard Haar**		**AGHaar**		**VSD & D-S Fusion**	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall* (%)
Day dataset	62.8	81.0	73.4	88.2	89.0	97.5	95.2	99.6
Night dataset	63.2	52.6	73.9	69.5	81.5	79.2	95.7	99.2
Rainy dataset	70.6	57.7	75.6	69.8	78.7	73.8	91.6	99.3
Snowy dataset	48.2	67.0	69.4	71.4	84.1	84.8	97.2	87.5
Short distance dataset	0	0	1.9	3.0	2.1	6.1	96.1*	98.8*
All-weather mixed dataset	54.4	57.4	65.2	66.8	74.3	75.1	**95.1**	**96.8**

*Obtained based on VSD only

namely rear-view vehicle detection, in order to prevent rear-end collisions. However, we aimed at covering a diversity of ideal to difficult weather and lighting conditions. We gained a detection rate of up to 97% with a high precision rate of 95% not only for day conditions, but also for rainy, snowy, foggy, and many other real-world challenging conditions. This is a significant step forward compared to previously reported results. The paper provided a detailed proposal for vehicle detection and distance estimation, using only monocular vision, and a real-time data stream from a mounted camera. The experiments showed a superior performance of the newly proposed AGHaar vehicle detector, compared to common LBP or standard Haar-like classifiers. It also became apparent that the proposed virtual-symmetry detection is important for detecting vehicles at very close distances in addition to medium or far distances, that can also be covered by the AGHaar approach.

The results of the proposed trigonometric distance estimator was sufficiently accurate to warn a distracted driver at the appropriate time, before an imminent rear-end collision occurs.

The time-effectiveness of the proposed methods and the implemented D-S fusion technique allows us a real-time processing of 25 to 28fps for the entire multi-data fusion system using only one monocular camera. The results obtained on a Corei5 2.7 GHz PC platform with 8GB of RAM and Windows 7.

Comprehensive experimental studies for our all-weather database illustrate the robustness of the proposed method across various situations. To the best of our knowledge, such a diverse set of challenging situations has neither been used in published benchmarks, nor is available in form of a public dataset. As part of this research, we also made our accumulated dataset publicly available as EISATS Set 10, Parts 1 and 2, on [66] for the use of interested researchers.

REFERENCES

[1] National Highway Traffic Safety Administration: Traffic safety facts. U.S. Dep. Transportation, 2012.

[2] H. Bai, T. Wu, and C. Liu: Motion and Haar-like features based vehicle detection. In Proc. *Int. Conf. Multi-Media Modeling* pp. 356–359, 2006.

[3] Z. Sun, G. Bebis, and R. Miller: On-road vehicle detection: A review. *IEEE Trans. Pattern Analysis Machine Intelligence*, 28:694–711, 2006.

[4] V.D. Nguyen, T.T. Nguyen, D.D. Nguyen, and J.W. Jeon: Toward real-time vehicle detection using stereo vision and an evolutionary algorithm. In Proc. *Vehicular Technology Conf.*, pp. 1–5, 2012.

[5] Heidelberg Vision Challenge at ECCV: hci.iwr.uni-heidelberg.de/Static/challenge2012, 2012.

[6] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto: A LIDAR and vision-based approach for pedestrian and vehicle detection and tracking. In Proc. *IEEE Intelligent Transportation Systems Conf.*, pp. 1044–1049, 2007.

[7] W. Yao and U. Stilla: Comparison of two methods for vehicle extraction from airborne LiDAR data toward motion analysis. *IEEE Geoscience Remote Sensing Letters*, 8:607–611, 2011.

[8] Distronic Plus with Steering Assist: techcenter.mercedes-benz.com/_en/distronic_plus_steering_assist/detail.html, 2013.

[9] F. Garcia, P. Cerri. A. Broggi, A. Escalera, and J.M. Armingo: Data fusion for overtaking vehicle detection based on radar and optical flow. In Proc. *IEEE Intelligent Vehicles Symposium*, pp. 494–499, 2012.

[10] A. Haselhoff, A. Kummert, and G. Schneider: Radar-vision fusion for vehicle detection by means of improved Haar-like feature and AdaBoost approach. In Proc. *European Signal Processing Conf.* pp. 2070–2074, 2007.

[11] P.K. Zoratti: Automotive driver assistance systems: Using the processing power of FPGAs. White Paper: XA Spartan-6 Automotive FPGAs, XILINX, WP399 (v1.0), 2011.

[12] R. Klette: Concise Computer Vision. Springer, London, 2014.

[13] A. Ali and S. Afghani: Shadow based on-road vehicle detection and verification using Haar wavelet packet transform. In Proc. *IEEE Int. Conf. Information Communication Technologies*, pp. 346–350, 2005.

[14] S. Han, Y. Han, and H. Hahn: Vehicle detection method using Haar-like features on real time system. In Proc. *World Academy Science Engineering Technology*, pp. 455–459, 2009.

[15] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi: Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. *IEEE Trans. Image Processing*, 15:2364–2375, 2006.

[16] S. Hermann and R. Klette: Iterative semi-global matching for robust driver assistance systems. In Proc. *Asian Conf. Computer Vision*, LNCS 7726, pp. 465–478, 2012.

[17] J. Choi: Realtime on-road vehicle detection with optical flows and Haar-like feature detectors. TR, CS Department, Univ. Illinois Urbana-Champaign, 2006.

[18] T. Ojala, Pietikäinen, and T. Mäenpää: Multiresolution grey-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24:971–987, 2002.

[19] Z. Qian, H. Shi, and J. Yang: Video vehicle detection based on local features. *Advanced Materials Research*, 186:56–60, 2011.

[20] F. Moutarde, B. Stanciulescu, and A. Breheret: Real-time visual detection of vehicles and pedestrians with new efficient AdaBoost features. In Proc. *Workshop Planning Perception Navigation Intelligent Vehicles*, pp. 70–75, 2008.

[21] T.T. Nguyen, H. Grabner, H. Bischof, and B. Bruber: On-line boosting for car detection from aerial images. In Proc. *IEEE Int. Conf. Research Innovation Vision Future*, pp. 87–95, 2007.

[22] X. Wen, H. Yuan, C. Yang, C. Song, B. Duan, and H. Zhao: Improved Haar wavelet feature extraction approaches for vehicle detection. In Proc. *IEEE Conf. Intelligent Transportation Systems*, pp. 1050–1053, 2007.

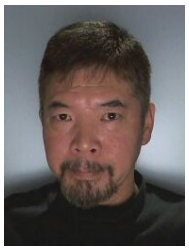
[23] A. Haar: Zur Theorie der orthogonalen Funktionensysteme. *Math. Annalen*, 69:331–371, 1910.

[24] R. Klette, N. Krüger, T. Vaudrey, K. Pauwels, M. van Hulle, S. Morales, F. Kandil, R. Haeusler, N. Pugeault, C. Rabe, and M. Lappe: Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database. *IEEE Trans. Vehicular Technology*, 60:2012–2026, 2011.

[25] M. Rezaei and R. Klette: Look at the driver, look at the road: No distraction! No accident! In Proc. *Computer Vision Pattern Recognition* (CVPR 2014), pp. 129–136, 2014.

[26] D. Santos and P.L. Correia: Car recognition based on back lights and rear view features. In Proc. *IEEE Workshop Image Analysis Multimedia Interactive Services*, pp. 137–140, 2009.
[27] R. O'Malley, M. Glavin, and E. Jones: Vehicle detection at night based on taillight detection. In Proc. *Int. Symp. Vehicular Computing Systems*, pp. 128–135, 2008.
[28] A. Haselhoff and A. Kummert: A vehicle detection system based on Haar and triangle features. In Proc. *IEEE Intelligent Vehicles Symposium*, pp. 261–266, 2009.
[29] L. Huang and M. Barth: Tightly-coupled LIDAR and computer vision integration for vehicle detection. In Proc. *IEEE Intelligent Vehicles Symposium*, pp. 604–609, 2009.
[30] M. Vargas, J. M. Milla, S. L. Toral, and F. Barrero: An enhanced background estimation algorithm for vehicle detection in urban traffic scenes. *IEEE Trans. Vehicular Technology*, 59:3694–3709, 2010.
[31] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester: Cascade object detection with deformable part models. In Proc. *Computer Vision Pattern Recognition*, pp. 2241–2248, 2010.
[32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan: Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Analysis Machine Intelligence*, 32:1627–1645, 2010.
[33] N. Dalal and B. Triggs: Histograms of oriented gradients for human detection. In Proc. *Computer Vision Pattern Recognition*, pp. 886–893, 2005.
[34] J. Arróspide, L. Salgado, and J. Marinas: HOG-like gradient-based descriptor for visual vehicle detection. In Proc. *IEEE Intelligent Vehicles Symposium*, pp. 223–228, 2012.
[35] A. Jazayeri, H. Cai, J.Y. Zheng, and M. Tüceryan: Vehicle detection and tracking in car video based on motion model. *IEEE Trans. Intelligent Transportation Systems*, 12:583–595, 2012.
[36] M. Rezaei, H. Ziaei Nafchi, and S. Morales: Global Haar-Like features: A new extension of classic Haar features for efficient face detection in noisy images. In Proc. *Pacific-Rim Symp. Image Video Technology*, LNCS 8333, pp. 302–313, 2014.
[37] M. Rezaei and M. Terauchi: Vehicle detection based on multi-feature clues and Dempster-Shafer fusion theory. In Proc. *Pacific-Rim Symp. Image Video Technology*, LNCS 8333, pp. 60–72, 2014.
[38] M. Rezaei and R. Klette: Novel adaptive eye detection and tracking for challenging lighting conditions. In Proc. *Asian Conf. Computer Vision Workshops*, LNCS 7729, pp. 427–440, 2012.
[39] P. Viola and M. Jones: Rapid object detection using a boosted cascade of simple features. In Proc. *Computer Vision Pattern Recognition*, pp. 511–518, 2001.
[40] M. Rezaei and R. Klette: Simultaneous analysis of driver behaviour and road condition for driver distraction detection. *Int. J. Image Data Fusion*, 2:217–236, 2011.
[41] F. Crow: Summed-area tables for texture mapping. *Computer Graphics* 18:207–212, 1984.
[42] Wikipedia, Bicubic interpolation: en.wikipedia.org/wiki/Bicubic_interpolation, 2013.
[43] B. Philip and P. Updike: Caltech Vehicle Dataset, http://www.vision.caltech.edu/html-files/archive.html, 2001.
[44] C. Papageorgiou and T. Poggio: A trainable system for object detection. *Int. J. Computer Vision*, 38:15–33, 2000.
[45] EPFL University, Computer Vision Laboratory: EPFL Multi-view car dataset. http://cvlab.epfl.ch/data/pose, 2012
[46] KITTI Benchmark website: CAR Detection Benchmark. www.cvlibs.net/datasets/kitti/eval_object.php, 2013.
[47] J. Matas, C. Galambos, and J.V. Kittler: Robust detection of lines using the progressive probabilistic Hough transform. *Computer Vision Image Understanding*, 78:119–137, 2000.
[48] R.O. Duda and P.E. Hart: Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972.
[49] N. Kiryati, Y. Eldar, and A.M. Bruckstein: A probabilistic Hough transform. *Pattern Recognition*, 24:303–316, 1991.
[50] Z. Xu and B.-S. Shin: A statistical method for peak localization in Hough space by analysing butterflies. In Proc. *Pacific-Rim Symp. Image Video Technology*, LNCS 8333, Springer, Heidelberg, pp. 111–123, 2013.
[51] E. Rosten and T. Drummond: Machine learning for high-speed corner detection. In Proc. *European Conf. Computer Vision*, pp. 430–443, 2006.
[52] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski: ORB: an efficient alternative to SIFT or SURF. In Proc. *Int. Conf. Computer Vision*, pp. 2564–2571, 2011.
[53] A. Alahi, R. Ortiz, and P. Vandergheynst: FREAK: Fast Retina Keypoint. In Proc. *Computer Vision Pattern Recognition*, pp. 510–517, 2012.
[54] J. Shi and C. Tomasi: Good features to track. In Proc. *Computer Vision Pattern Recognition*, pp. 593–600, 1994.
[55] R. O'Malley, E. Jones, and M. Glavin: Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions. *IEEE Trans. Intelligent Transportation Systems*, 11:453–462, 2010.
[56] H. Freeman: On the encoding of arbitrary geometric configurations. *IRE Trans. Electronic Computers*, 10:260–268, 1961.
[57] Y. Kui Liu and B. Žalik: An efficient chain code with Huffman coding. *Pattern Recognition*. 38:553–555, 2005.
[58] H.-Z. Gu and S.-Y., Lee: Car model recognition by utilizing symmetric property to overcome severe pose variation. *Machine Vision Applications*, 24:255–274, 2013
[59] K. Briechle and U.D. Hanebeck: Template matching using fast normalized cross correlation. In Proc. *Optical Pattern Recognition*, pp. 95–102, 2001.
[60] A. Geiger, P. Lenz, and R. Urtasun: Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proc. *Computer Vision Pattern Recognition*, pp. 3354–3361, 2012
[61] G. Shafer: *A Mathematical Theory of Evidence.* Princeton University Press, 1976.
[62] D. Koks and S. Challa: An introduction to Bayesian and Dempster-Shafer data fusion. TR, DSTO-TR-1436, DSTO Systems Sciences Laboratory, 2005.
[63] R. Jiang, R. Klette, T. Vaudrey, and S. Wang: New lane model and distance transform for lane detection and tracking. In Proc. *Computer Analysis Images Patterns*, LNCS 5702. pp. 1044–1052, 2009.
[64] Z. W. Kim: Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intelligent Transportation Systems*, 9:16–26, 2008.
[65] S. Tuohy, D. O'Cuslsin, E. Jones, and M. Glavin: Distance determination for an automotive environment using inverse perspective mapping in OpenCV. In Proc. *Signals Systems*, pp. 100–105, 2010.
[66] M. Rezaei, and M. Terauchi: iROADS Dataset (Intercity Roads and Adverse Driving Scenarios). Enpeda image sequence analysis test site-EISATS, Set 10, http://ccv.wordpress.fos.auckland.ac.nz/eisats/, 2014.

Dr. Mahdi Rezaei Received his PhD degree in Computer Science with the "Best Thesis Award 2014" from the University of Auckland, New Zealand. His main research interests are advanced driver-assistance systems, driver distraction detection, and vehicle safety, based on computer vision, machine learning, and data fusion techniques. He is a member of the Environment Perception and Driver Assistance (*.enpeda..*) research project, in collaboration with Daimler A.G. (Mercedes Benz), Germany, for the development of next generation intelligent vehicles.



Dr. Mutsuhiro Terauchi is an associate professor at the Department of Psychology, Hiroshima International University, Japan. His research interests include driver assistance systems, road safety, and human audio/visual information processing and sensory substitution for disabled.

Dr. Reinhard Klette is a Fellow of the Royal Society of New Zealand. He is a professor in the School of Engineering, Auckland University of Technology, and director of the Centre for Robotics and Vision (CeRV). He directs research projects which examine how to use multiple video streams (recorded in cars, in drones, or on boats) to understand dynamic 3D environments. He (co-)authored about 500 publications, presented keynotes at international conferences, and (co-)authored 10 books.