A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera

Claudio Caraffi[1], Tomáš Vojíř[2], Jiří Trefný[2], Jan Šochman[2] and Jiří Matas[2]

***Abstract*— A novel system for detection and tracking of vehicles from a single car-mounted camera is presented. The core of the system are high-performance vision algorithms: the WaldBoost detector [1] and the TLD tracker [2] that are scheduled so that a real-time performance is achieved.**

The vehicle monitoring system is evaluated on a new dataset collected on Italian motorways which is provided with approximate ground truth (GT$'$) obtained from laser scans. For a wide range of distances, the recall and precision of detection for cars are excellent. Statistics for trucks are also reported. The dataset with the ground truth is made public.

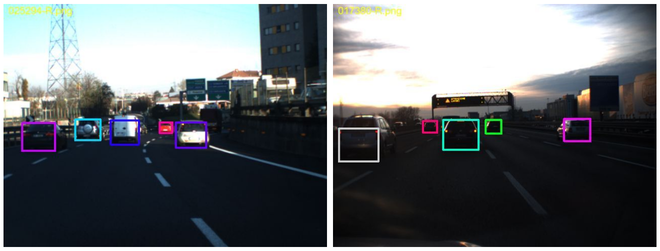
I. Introduction

We present a system for vehicle detection and tracking using a single camera mounted on a moving or stationary car. The system is running in real-time (10 Hz) on a single CPU core.

A wide range of sensors, e.g. lidar, radar, ultrasound and stereo based depth sensors, is available to driver assistance system designers. We opted for a single camera-based system since it is cheap, consumes minimum energy, is light and robust. It can easily by mounted on a motorbike or even, forward or rear facing, on a bicycle. In a car, multiple single-camera systems with different viewing directions, angles and distance ranges can be deployed. Visual information is complex to process, but it provides rich information about the environment. Vision as a sensing device has limitations (e.g. foggy conditions, driving against the sun) but these are well understood since they are similar to difficulties experienced by human drivers.

As the main contribution of the paper we present a novel system for detection and tracking of vehicles that integrates high-performance vision algorithms: the WaldBoost (WB) detector [1] and the TLD tracker [2]. We show how to control the WB detector and the TLD tracker to achieve real-time performance via process scheduling.

As a second contribution, a new dataset intended for evaluation of on-board systems for vehicle monitoring is presented. The dataset was collected on Italian motorways and includes a variety of lighting and traffic conditions, see Figures 1 and 6. For the dataset, an approximate ground truth was calculated from laser scans collected together with the visual data. The dataset and the approximate ground truth will be made public.



(a) Long range, variable light (b) Dusk conditions

Fig. 1. Examples of motorway conditions represented in the introduced dataset with vehicle detections of the presented system overlaid.

The motorway environment is constrained in comparison with a general road situation: no pedestrians, no incoming vehicles, a well-marked road with a uniform surface, no high-curvature bends and slowly changing slopes. On the other hand, the high percentage of trucks, occasional density of traffic and the high speed of some vehicles pose a challenge.

We evaluate the system on a selected subset of the dataset that includes varying conditions and we report performance in terms of detection and false positive rates as a function of vehicle distance and apparent vehicle width in the image (width in the image in pixels).

II. Related work

A. *Vehicle detection*

Object detection in static images is a well studied problem. In computer vision research, cars are common objects of interest due to their rigid structure, low appearance variations and common presence in everyday scenes [3], [4], [5], [6], [7], [8]. Early approaches were aiming mostly at high precision and recall rather than real-time performance and were based on statistical methods like SVM [4], [5], PCA [6], Neural Networks [7] or Bayesian decision-making [8].

A breakthrough in application of statistical learning techniques to real-time object detection was brought by the cascaded classifier of Viola and Jones [9] who proposed a method for training a sequential classifier working on simple-to-evaluate Haar-like features and demonstrated its real-time performance on the face detection problem. Hundreds of related papers have been published focusing on improving different aspects of the approach and applying it to various tasks, including car detection [10]. Of the follow up work on the Viola-Jones algorithm, of particular interest to this paper are methods focusing on automatic cascaded classifier training with respect to both classification precision *and* the average classification time [1], [11]. They allow training a time-precision optimized cascaded classifier without the

[1] C. Caraffi is with Advanced Technology division, Toyota Motor Europe, `<name>.<surname>@toyota-europe.com`

[2] The authors are with The Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic `trefny.jura@centrum.cz`, {`vojirtom`, `jan.sochman`, `matas`}`@cmp.felk.cvut.cz`

tedious manual intervention needed in the original Viola-Jones method.

An alternative approach inspired by the success of part-based detectors in Pascal VOC Challenge [3] is taken in [12]. However, despite the good detection performance, the complexity of the method allows the system to run at only 1-2 frames per second.

Driving assistance systems for urban environments require relatively complex algorithms and the problem is still considered to be very challenging [13]. In less demanding scenarios like motorway driving, various relatively simple heuristic-based vehicle detectors have been proposed in the literature. Some authors exploit the shadow cast on the road by cars which is typically darker than the rest of the road [14], some use the fact that car outer edges could be approximated by a U-shape curve [15]. Others rely on vehicle symmetry as the main cue [16], [17]. At the same time, methods that reduce the range of possible vehicle positions to be tested by constraining to feasible on-road locations are often applied [18]. The advantage of these systems is their real-time performance, but their assumptions about the the real-world scenes are simplistic and do not hold in general. Indeed, such papers often lack rigorous quantitative evaluation on some publicly available dataset and comparison to other methods. An exhaustive survey of this class of methods can be found in [19].

A very different recent approach for vehicle detection is to use motion parallax [20] or more generally real-time multi-body visual SLAM [21]. Here the vehicles are detected as outliers to reconstructed (rigid) scene structure. This approach allows for both scene modeling and vehicle/pedestrian detection and tracking: however, the method still remains to be verified on more complex scene where multiple outliers clusters corresponding to multiple moving vehicles may not be easily separable into independent objects

In this paper we rely on the WaldBoost-based vehicle detector [1], [22]. WaldBoost has already demonstrated real-time performance ability on face detection problems, is easy to train and, given an adequate training set, it generalizes well to previously unseen vehicles.

B. *Vehicle tracking*

In the literature, the Particle Filter (PF) algorithm [23] is probably the most popular approach for vehicle tracking. It has been applied both to single object tracking [12], [21] as well as in an extended form which is able to track an unknown and variable number of objects [10], [20]. The advantage of the PF approach is that it can model complex object dynamics through non-parametric, particle-based, multi-modal motion distribution estimation. In [13] the Multi-Hypothesis Tracking (MHT) has been used instead of PF. Instead of modeling the distribution of possible states as in PF, MHT keeps only a small set of the most likely motion explanations.

The inherent problem of the above approaches is their sensitivity to drift from the true object position, especially in long sequences. They offer no correcting mechanisms and eventually fail when the object changes appearance significantly due to occlusion, change in lighting conditions or pose change.

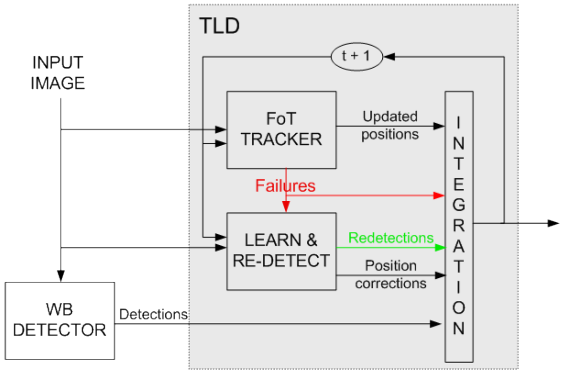


Fig. 2. The structure and flow of information in the proposed vehicle monitoring system.

Recently, discriminative methods have become popular in tracking literature posing the tracking as a foreground-background classification task [24], [25]. In these approaches, the problem of complex motion modeling is avoided by exhaustively searching the neighborhood of the predicted position. The methods also offer means for continuous appearance model updating through on-line learning algorithms.

One possible method for minimizing drift within this formalism is *co-training*. Two (or several) on-line classifiers are trained at the same time using either independent modalities [26] or comparing global (or generative) and local models of object appearance [27], [2], [28]. Very impressive results have been demonstrated using these approaches for objects undergoing appearance changes [2], [28] as well as for long-term tracking [2].

Driven by real-time requirements and the need for long-term tracking we adopt a modified TLD method [2] with extensions described in [29] as they represent probably the most robust and yet real-time approach for object tracking.

III. The System

The structure of the vehicle detection and tracking system is depicted in Fig. 2. The role of the WaldBoost (WB) detector, described in Sec. III-A is the discovery of new cars and trucks in the field of view. The new detections are tracked by a Flock of Trackers (FoT) which is detailed in Sec. III-B. The Learn and re-Detect module (Sec.III-C) uses on-line unsupervised learning to build a specific detector for all monitored vehicles; it allows long-term vehicle identity maintenance even in case of tracking failure. The information from the tracker, specific and generic vehicle detectors is integrated and passed on to the 3D pose estimation and surrounding vehicle maintenance module; these two modules are not described here due to lack of space.

A. *The Detector.*

The rear view vehicle detector is a WaldBoost [1] trained sequential classifier applied within a sliding window framework. WaldBoost is an AdaBoost-based algorithm which automatically builds a fine-grained detection cascade of the Viola and Jones type [9]. Wald's sequential probability ratio test (SPRT) performs early rejection of negative samples after evaluation of a single weak classifier. Fast rejection of negative samples is critical for detection speed, as a vast majority of tested windows do not contain a vehicle.

WaldBoost training is iterative, gradually building a more complex sequential classifier. In the first iteration, a standard AdaBoost learning search for the best weak classifier is performed. Then the rejection threshold for Wald's SPRT is estimated on a large pool of data. Finally, the pool is pruned and a bootstrap strategy is employed to collect additional non-object examples. To speed up the AdaBoost learning, the weak classifier selection relies only on a subset of the pool sub-sampled using the QWS+ strategy [30]. The weak classifiers are chosen from an extended set of multi-block local binary pattern features [31] and their contributions to the final decision are a function of the weighted error for each binary code as in the confidence-rated classification method [32]. The approach allows fast implementation using a look-up table.

The vehicle detector was trained on 5000 car samples from which about 80000 positive samples with random displacements and scale changes were synthesized. For the background class, about one billion negative windows were sampled from images without a car. The training samples were downscaled to the width of 26 pixels which corresponds to the minimum detection size of a car.

The detector is applied within a sliding window. The detector runs at 12 fps on sequences with 1024x768 pixel resolution and evaluates only 1.9 weak classifiers per scanning position on average. This speed was achieved for a shift between two evaluated positions equal to 1/13 of the window size (a two pixel shift for the smallest scanning window which is 26 pixels wide) and when window sizes at adjacent scales differ by a factor of 1.2. An additional speed increase is gained by excluding a fixed-height region corresponding to the sky from the search.

B. The Tracker

Tracking is performed by an adapted Flock of Trackers (FoT) [29]. The advantages of the FoT are its speed, about 5 milliseconds on a standard notebook for each tracked object, and its robustness to partial occlusion and imprecise initialization.

The FoT estimates object motion from the displacement of local trackers which are spread uniformly within a region covering the object. Local trackers estimate displacements by the Lucas-Kanade (LK) method [33].

In the application considered, we assumed that object motion is sufficiently precisely modeled by translation and scaling. The motion is robustly estimated from a subset of reliable local trackers, the translation as the median of their displacements, the scale as the median of distance ratios of all pairs of corresponding local trackers. The reliable subset is selected on the basis of local tracker confidence estimates which are a function of past performance, of normalized cross-correlation of patches at the previous and current locations and of the consistency with adjacent displacement estimates. For details, see [29].

The median-based estimation method combined with local tracker confidence prediction makes the FoT robust to partial occlusions and to the failure of a fairly large proportion of local trackers. However, our current implementation of the FoT does not in general handle cases were most local trackers fail due to a global change in illumination, e.g. when passing under a bridge, entering a tunnel or in the presence of strong sharp car-size shadows on the motorway. This problem is caused by the underlying assumption of brightness constancy made by the Lucas-Kanade tracker. Such cases are handled by the re-detection described in the next section.

C. Learning and re-Detection (LrD)

Since tracking based on local optimization may fail, e.g. due to occlusion or rapid illumination change, the need to maintain a temporally consistent model of the environment requires the ability to re-detect a temporarily lost vehicle which in turn requires unsupervised on-line learning of detectors of specific vehicles. Such learning and re-detection capability is provided by a modified version of the TLD algorithm [2].

In TLD, the detector also uses the sliding window approach. The object is represented by on-line learnt Randomized Forest (RF) [34]. In comparison with the WaldBoost generic vehicle detector, the Randomized Forest is simple and typically more powerful since it solves a simpler problem: *specific detection* - to recognize a single particular vehicle in current conditions (illumination, background, etc) only.

The RF is a set of a restricted class of decision trees called ferns [35] with Haar-like features [9] associated with internal nodes. Observations at internal nodes define a single leaf node in every fern, where an estimate of object vs. background likelihood is stored. Initially, the estimates are based on a single example provided by the generic WaldBoost detector and its affine warps.

For each vehicle a new RF is learnt. The RFs consist of 10 ferns each with depth 7, which is a compromise between the speed of evaluation and the discriminative power of the model. Initially, we populate an RF with the positive examples generated by warping the validated object image patch and then a negative examples learnt incrementally, as in the TLD, by considering the positive responses of the sliding window detector which are far from object position as the negative examples. A detection is far from the object if the overlap[1] with the object position is less than 0.7. The current object position (provided by the FoT tracker) is learnt as a positive example. The learning takes place only if at the current position the similarity to the collection of object patches is higher than 0.75, where similarity is measured by the maximum (over patches) of the normalized cross-correlation.

[1] The overlap score O is defined in terms of areas of bounding boxes and of their intersection $\cap$: $O = \cap(\text{bb1}, \text{bb2})/(\text{bb1} + \text{bb2} - \cap(\text{bb1},\text{bb2}))$.

We omit other learning events from the TLD algorithm [2], because they are designed for situations where the object appearance changes (e.g. as a consequence of a rotation around it axis), which is not the case in the motorway scenarios.

The Learn and re-Detect (LrD) sub-system serves two purposes. First, the LrD is used for FoT position corrections which stabilize estimates of the vehicle trajectory and it prevents the FoT from drifting by accumulating error from imprecise object transformation estimation. Second, LrD contributes to the decision about object position in situations when illumination changes dramatically and the FoT fails. Since the features used in the RF are invariant to any locally monotonic illumination changes it is highly robust to illumination changes.

D. Tracker-Detector Interaction. Scheduling.

The vehicle monitoring system schedules two processes: (i) vehicle hypothesis initialization and validation and (ii) tracking, including positional correction and failure handling.

In the first process, the WaldBoost detects objects, and the detected objects with confidence above a threshold are passed on to be tracked by the FoT. After this initialization, WaldBoost detections that overlap already tracked objects are used to validate them; moving objects with multiple positive detection in consecutive frames are unlikely to be false positives. We set a threshold to object validation to three positive detections out of five consecutive detector runs.

The second process consists of tracking and positional correction of the objects with the LrD to minimize the localization error and avoid drifting. In the case of FoT failure, which is indicated by the FoT tracker from internal statistics adapted for the tracking car situations (i.e. number of consistent local trackers and scale change between two consecutive frames), the WaldBoost and LrD detector are run in the enlarged area predicted by a Kalman filter associated with the failing FoT to decide if there is a vehicle and where, and eventually reset the FoT.

Scheduling. To achieve real-time performance, we identified the most time-consuming components of the system (see tab. I showing the timing of individual components) and introduced the following scheduling for:

- The WaldBoost detector. It is *i*) run every 3rd frame on the relevant part of the image to detect new vehicles and *ii*) run in a small range of locations and scales to re-detect vehicle where tracking failed.
- Establishing a new object. The process is run one frame after detection and requires object patch warping and RF learning.
- The LrD. It is run for each established object in frames where neither the WaldBoost detector is run nor a new object is found. The LrD plays two roles. First, time permitting, the position of all existing trackers is checked. Second, negative examples for learning are collected for one object tracker at a time.

In summary, only one of the three time-consuming processes is performed in a single frame.

AVERAGE COMPUTATION TIME [ms]		
	Image resolution	
Process	640x480	1024x768
WaldBoost*	16.61	42.99
Warping + RF Learning	8.82	21.24
LrD position correction	5.06	3.99
LrD negative samples	2.65	2.74
FoT	3.12	6.29
WaldBoost verification	1.27	0.87
LrD verification	3.47	1.60

TABLE I

AVERAGE COMPUTATION TIME FOR PROCESSES WITH NON-NEGLIGIBLE DURATION, IN MILLISECONDS. *EXCEPT FOR WALDBOOST, THE TIME INDICATED IS PER OBJECT.

Currently, all these sub-systems run in one thread, therefore by introducing multiple threads the system can be easily parallelized, because of high processing independence of individual components.

IV. THE TME MOTORWAY DATASET

The dataset used to benchmark the system has been selected from the acquisition made in Northern Italy in December 2011 in cooperation with VisLab (University of Parma, Italy), using the BRAiVE test vehicle [36]. The “TME Motorway Dataset”, available for download at http://cmp.felk.cvut.cz/data/motorway, comprises:

- Image acquisition: stereo, 20 Hz frequency[2], 1024x768 grayscale losslessly compressed images, with bayer coded color information[3]; 32° horizontal field of view.
- Ego-motion estimate (confidential computing method).
- Laser-scanner generated vehicle annotation and classification (see Sect. IV-A).

The data provided is timestamped, and includes extrinsic calibration.

28 clips for a total of approximately 27 minutes of acquisition have been selected, to promote comparison on a dataset that could be downloaded in a reasonable amount of time. This selection includes variable traffic situations, number of lanes, road curvature, and lighting, covering most of the conditions present in the complete acquisition (see Fig. 1 and 6). The dataset has been divided in two sub-sets depending on lighting condition, named “daylight” (although with objects casting shadows on the road) and “sunset” (facing the sun or at dusk). For each clip, 5 seconds of preceding acquisition are provided, to allow the algorithm stabilizing before starting the actual performance measurement.

A. Approximate Ground Truth (GT')

To the best of our knowledge, the only publicly available vehicle-annotated dataset for motorway video sequences is the one introduced in [10][4]. This dataset includes 2 motorway sequences for a total duration of one minute, and annotation of vehicles in the image. An extension of publicly available

[2] In experiments, even frames (10 Hz) from right camera were used.

[3] OpenCV `CV_BayerGB2GRAY` and `CV_BayerGB2BGR` conversion codes are utilized to compute our results.

[4] Only very recently, public databases of recorded data from a moving vehicle have emerged [37].

data would be beneficial, but a tedious manual annotation job would be necessary, especially when it comes to heavier (and more interesting) traffic conditions. Furthermore, manual annotation cannot provide information about 3D world location of targets.

To overcome this problem, we propose to generate a comparison dataset using a different sensor, namely a 4-layer Ibeo laserscanner. We have developed an algorithm, currently of limited scientific interest, to detect vehicles from 3D point clouds. The results are then mapped into bounding boxes in the image using the static calibration information and a flat-ground assumption. The detections are tracked over consecutive frames, so that a consistent ID is provided. This also allows interpolating results at specific intermediate timestamps between 2 detections, dealing with the different acquisition frequencies of laserscans and images (12.5 Hz and 20 Hz respectively).

Thanks to the availability of 3D locations and ego-motion, it is possible to estimate if an object is moving, allowing discarding targets like signs misclassified because of their compatibility with vehicle size. However, also static objects are recorded, so that stopped vehicles can be easily tagged manually. Manual corrections allow also to quickly remove the few (and ID-consistent) false positives present. The final classification about “moving” or “not moving” object and vehicle width estimation are computed off-line, by considering the collected data until the moment the target is lost. The estimated width of the vehicle serves to classify vehicles into “car” or “truck”, using a threshold set at 2.1 meters: this classification is successful in the vast majority of the cases.

Limitations:

1) Due to the limited number of laser reflections available, in GT′ no motorbike is present.
2) The reliability of the generated data decays beyond 60-70 meters, when less than 3 laser reflections per vehicle become available.
3) The quantization error caused by the limited angular resolution of the laserscanner generates lateral jumps of the computed image box, so that a quantitative evaluation of the image tracker preciseness cannot be performed. Farther and darker objects show additional instability in their box side boundaries.
4) No vehicle length is currently provided, and vehicle height can only be set arbitrarily, partially depending on the estimated width.
5) Ego-vehicle pitching can cause a target to be temporarily lost and re-detected with a new ID. In particular, ego-vehicle oscillations and non-flat road pose challenges that will be addressed in the next section.

B. Matching between system results and GT′

As it can be reasonably expected, removing the human supervision of a manual annotation requires some complexity to be shifted on the successive parts of the process, in particular while designing the criteria used to match laserscanner detections with image detections. The source code of the program designed for this operation is made available with the dataset. We compute:

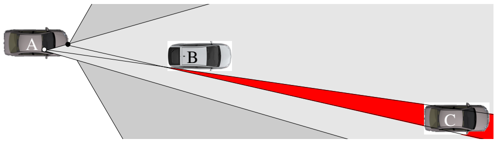


Fig. 3. Due to the presence of car B, car C is not visible from the point of view of the laserscanner, which is located at the center of the front bumper of car A (black dot). However, car C is visible from the camera located on the top right corner of the windshield of car A (white dot).

$$O = {O_w}^2 \cdot O_x \cdot \sqrt{O_y} \quad (\textit{Overlap score})$$

$$O_w = \frac{\min\left({w'}_G, w_S\right)}{\max({w'}_G, w_S)} \qquad O_x = \frac{\left\|\cap\left(\left[x_{0_G}, x_{1_G}\right], \left[x_{0_S}, x_{1_S}\right]\right)\right\|}{\min(w_G, w_S)}$$

$$O_y = \frac{\left\|\cap\left(\left[y_{0_G}, y_{1_G}\right], \left[y_{0_S}, y_{1_S}\right]\right)\right\|}{\min(h_G, h_S)}$$

where subscripts G and S denote an image box from GT′ and from our system output respectively, $[x_0, x_1]$ and $[y_0, y_1]$ are the intervals occupied by the box in the x and y coordinate respectively, and w and h are the width and the height of the box, with ${w'}_G$ as the width of the GT′ box computed using the estimated physical width and the instantaneous distance. After experimental observations, a decision threshold has been set at 0.35.

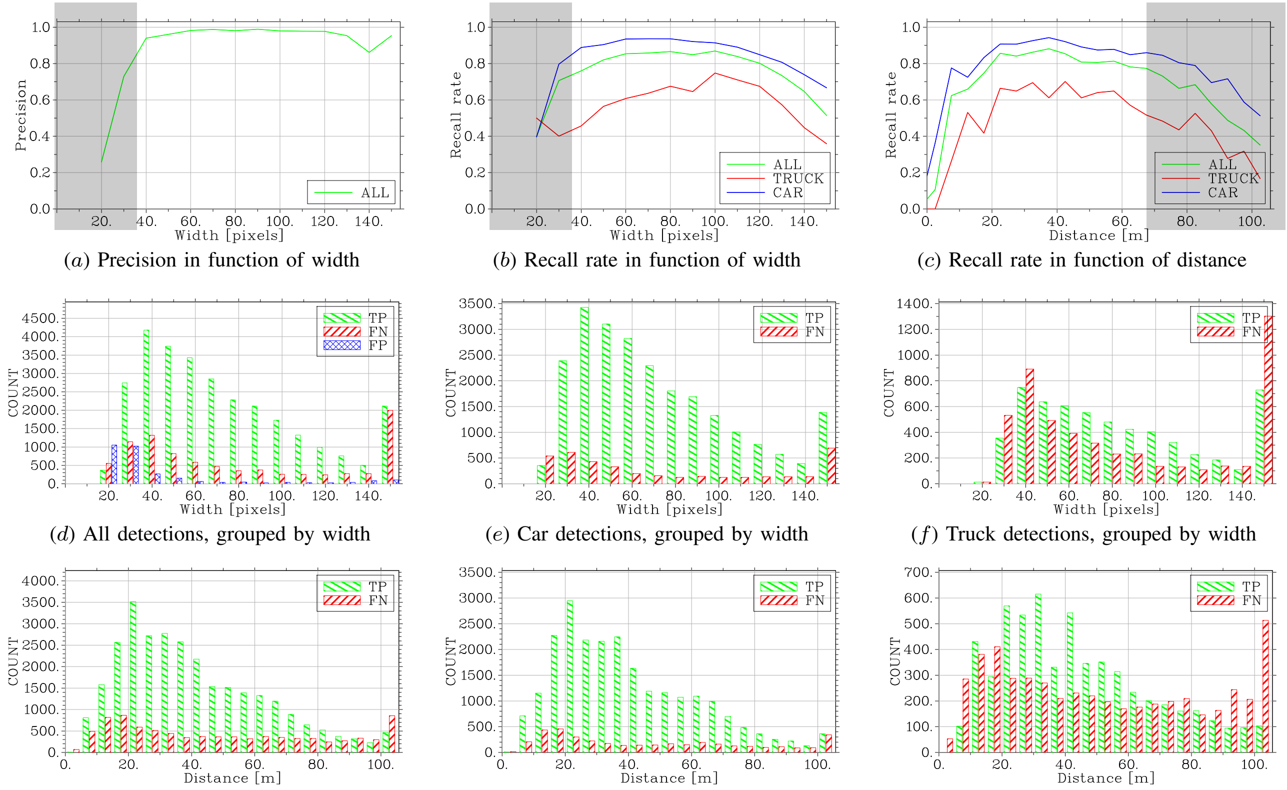
Conceptually, we separate the scale/area matching from the position matching. The term ${O_w}^2$ represents the area matching between GT′ and our system output: as the height of the target cannot be measured directly from the laser reflections, we consider the squared width as area-related value. O_w is the most reliable overlap measure, as ${w'}_G$ is computed using the physical width estimated over the whole tracking period of the object. O_x and O_y represent measures of the overlap on the x and y coordinate of the boxes. Given limitations 3, 4 and 5, we select a conservative denominator.

Ego-vehicle oscillations (pitching) cause misplacements of GT′ bounding boxes along the y coordinate, because reprojected in image coordinates through a static calibration. To compensate this error we reproject the GT′ detections in the image utilizing at each frame the calibration pitch angle that allows the best total matching score for all the detections (*Best Pitch Match*), implementing what can be considered a detector-based image stabilization. Nevertheless, the presence of non-flat roads makes O_y the less reliable measure. Therefore, we assign a reduced exponential weight (0.5) in the merging formula used to obtain O.

Given the displacement between the 2 sensors (see Fig. 3), there exist targets visible by only one of them, which should be removed from the statistics computation. This problem has been addressed considering vehicles as standing vertical surfaces, parallel to the image plane; although this covers most part of the cases in the dataset, also object length and orientation should be estimated and taken into consideration for a full understanding of the occlusions generated.

V. Experiments

In the charts of Fig. 4 and 5 we report the statistics collected on the two datasets, “sunset” (ex. Fig. 1b, 6a and 6e) and “daylight”, while Fig. 6 and the complementary



(*a*) Precision in function of width

(*b*) Recall rate in function of width

(*c*) Recall rate in function of distance

(*d*) All detections, grouped by width

(*e*) Car detections, grouped by width

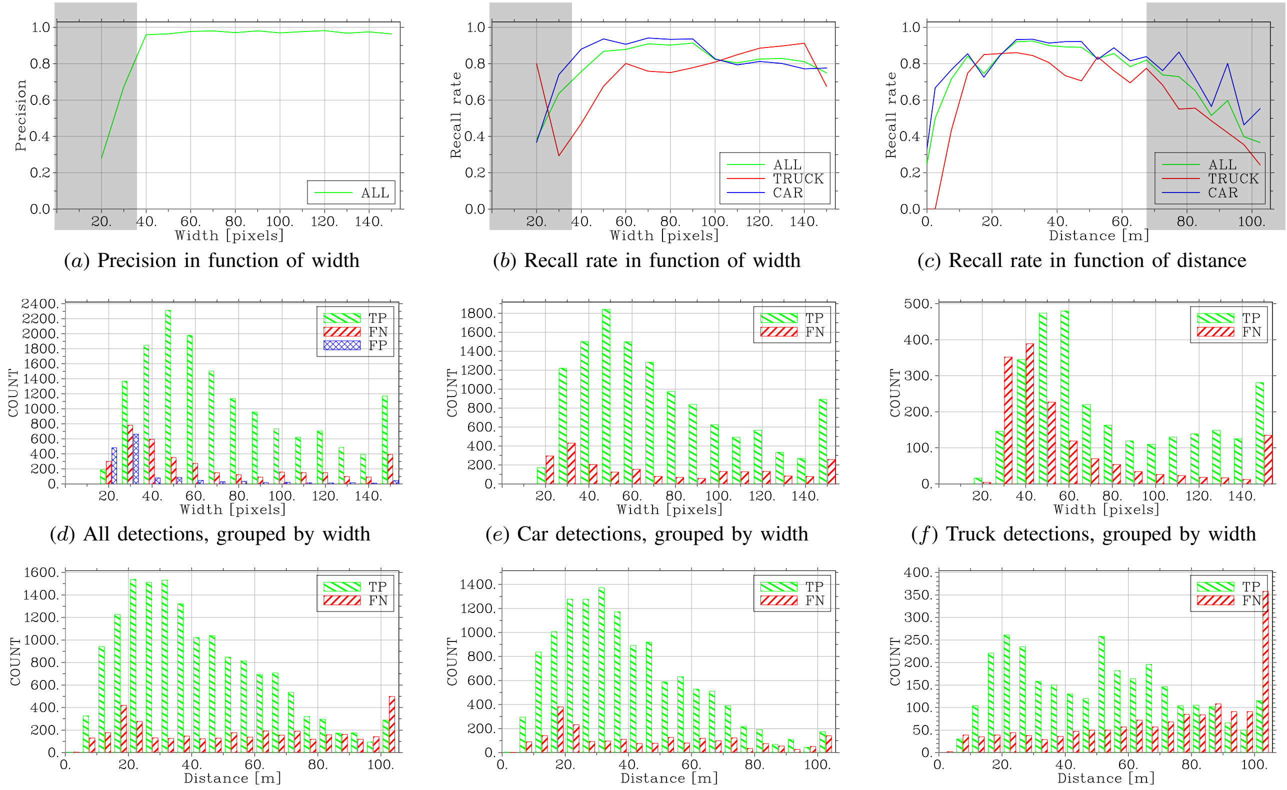
(*f*) Truck detections, grouped by width

(*g*) All detections, grouped by distance

(*h*) Car detections, grouped by distance

(*i*) Truck detections, grouped by distance

Fig. 4. Statistics collected on the “Daylight” subset. A grey box is placed over the chart region where the GT′ is not considered reliable (see limitation 2).



(*a*) Precision in function of width

(*b*) Recall rate in function of width

(*c*) Recall rate in function of distance

(*d*) All detections, grouped by width

(*e*) Car detections, grouped by width

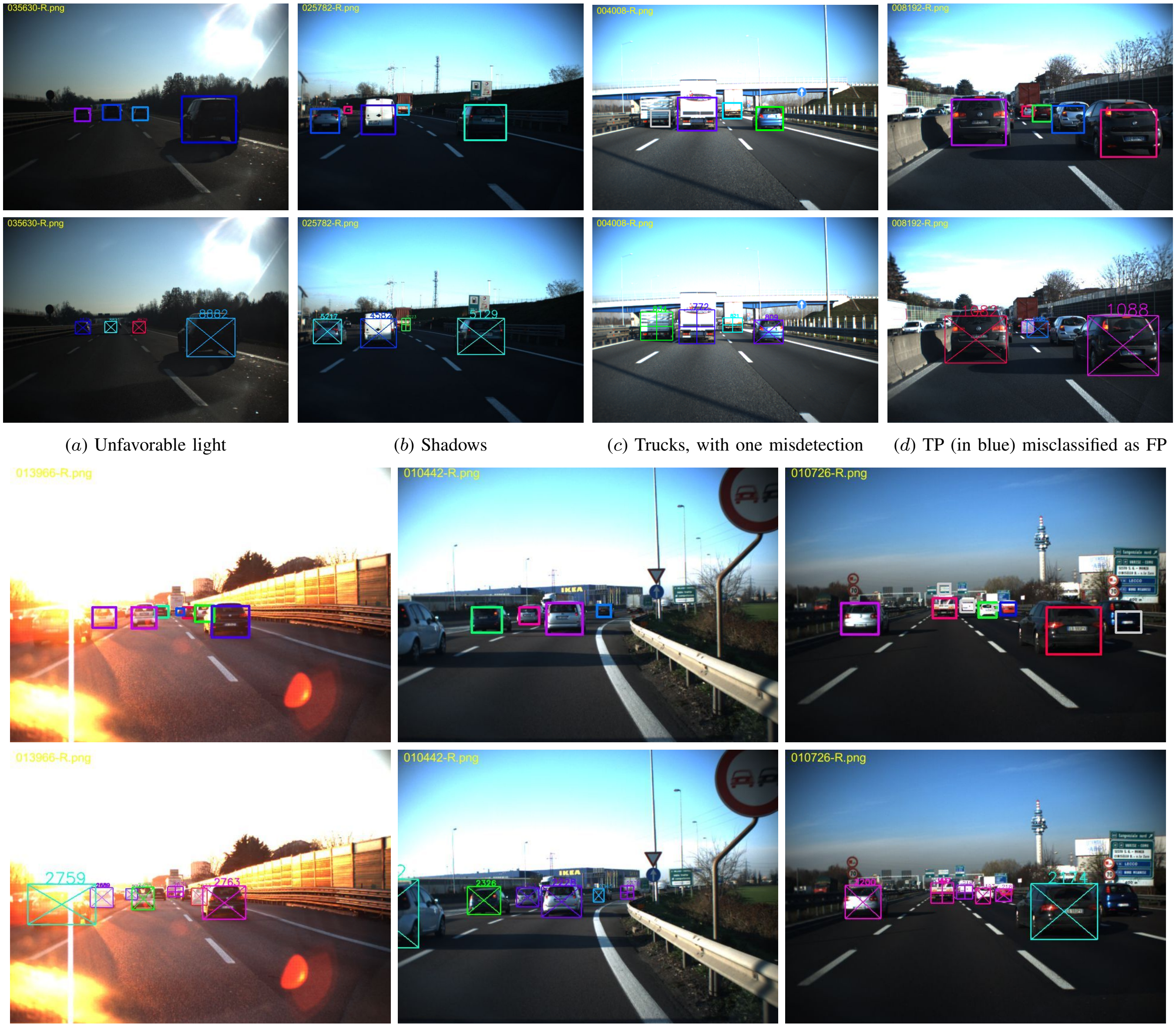
(*f*) Truck detections, grouped by width

(*g*) All detections, grouped by distance

(*h*) Car detections, grouped by distance

(*i*) Truck detections, grouped by distance

Fig. 5. Statistics collected on the “Sunset” subset.



(a) Unfavorable light (b) Shadows (c) Trucks, with one misdetection (d) TP (in blue) misclassified as FP

(e) Sunset (f) Curve (g) 4-lanes motorway

Fig. 6. Representative results on the TME Motorway Dataset. Odd rows show results from our system, even rows the laser scanner generated ground truth. In odd rows, a white bounding box marks a target that has not been validated yet. In even rows, a diagonal cross (*saltire*) marks cars, a vertical (*Greek*) cross marks trucks. A unique color is associated to each ID, as can be appreciated in the complementary video.

video show the result of our system and the GT′ for some significative cases. We compute *recall rate* as $\frac{TP}{TP+FN}$ and *precision* as $\frac{TP}{TP+FP}$, where TP, FN and FP are respectively the number of true positives (match between system output and GT′), of false negatives and of false positives.

The availability of information like width, distance and vehicle category allows us breaking down the statistics in the intent of highlighting strong points and limits of our system. For example it is possible to notice:

- The low number of false positives/high precision for target whose width is beyond 60 pixels.
- The apparently surprising low recall rate (0.8) for closer targets. This can be explained by the fact that overtaking vehicles remain in the proximity of the ego-vehicle for a shorter time than that required by the system to validate the object. This suggests that some work should be done to shorten the validation period for close targets.
- The relatively low performance of the system on trucks (see also fig. 6c). This problem will be addressed by redefining the training set of our algorithm (which currently includes only a limited portion of trucks) or by running in parallel a specific detector for trucks.

The low quality of the GT′ starting from 60-70 meters does not allow measuring quantitatively the performance of the system beyond that distance. In the submitted videos it is possible to qualitatively appreciate that our system outperforms GT′ for distant targets, with stable tracking/no ID loss even across illumination changes.

Qualitatively speaking, the majority of false positives is generated on sides of vehicles, which are not part of the negative training set for the detector. This choice increases significantly the recall rate of the detector, but it should be balanced by geometry considerations to filter unrealistic object hypotheses.

VI. Conclusions

A system able to consistently detect and track vehicle rears in images from a single camera was presented. The system showed good performance in terms of recall, precision and false positive rates even in bad lighting conditions. The evaluation was carried out on a new dataset that will be released to the scientific community.

The system is real-time without being resource-greedy, requiring a single core of a single CPU, which leaves space to integration with other algorithms or extension to parallel multi-class or multi-view object detection, including, e.g., motorcycles. We believe the system has a high potential for adoption since the required hardware is cheap and compact.

Finally, a new semi-automatic method for performance measurement was presented. We showed its limits, but noted the importance of extended public datasets and of extra information like the 3D position of targets, which allows e.g. benchmarking trajectory reconstruction algorithms.

Acknowledgements

The authors were partially supported by the Grant Agency of the CTU Prague under Project SGS11/125/OHK3/2T/13.

References

[1] J. Sochman and J. Matas, "WaldBoost - Learning for Time Constrained Sequential Detection," in *CVPR*, vol. 2, 2005, pp. 150–157.

[2] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, Jun. 2010.

[4] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *Int. J. Comp. Vision*, vol. 38, no. 1, pp. 15–33, Jun. 2000.

[5] Z. Sun, G. Bebis, and R. Miller, "On-road Vehicle Detection using Gabor Filters and Support Vector Machines," in *14th Intern. Conf. on Digital Signal Processing (DSP)*, vol. 2, 2002, pp. 1019–1022 vol.2.

[6] J. Wu and X. Zhang, "A PCA Classifier and its Application in Vehicle Detection," in *Neural Networks, Proceedings. IJCNN '01. International Joint Conference on*, vol. 1, 2001, pp. 600–604 vol.1.

[7] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle Detection and Recognition in Grayscale Imagery," *Control Engineering Practice*, vol. 4, no. 4, pp. 473–479, 1996.

[8] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection applied to Faces and Cars," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 746–751 vol.1.

[9] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[10] S. Sivaraman and M. Trivedi, "A General Active-Learning Framework for On-road Vehicle Recognition and Tracking," *IEEE Trans. on ITS*, vol. 11, no. 2, pp. 267–276, june 2010.

[11] L. Bourdev and J. Brandt, "Robust Object Detection via Soft Cascade," in *CVPR*, 2005, pp. 236–243.

[12] A. Takeuchi, S. Mita, and D. McAllester, "On-road Vehicle Tracking using Deformable Object Model and Particle Filter with Integrated Likelihoods," in *Intelligent Vehicles Symp.*, june 2010, pp. 1014–1021.

[13] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object Detection and Tracking for Autonomous Navigation in Dynamic Environments," *The International Journal of Robotics Research*, vol. 29, 2010.

[14] C. Tzomakas and W. von Seelen, "Vehicle Detection in Traffic Scenes Using Shadows," IR-INI, INSTITUT FUR NEUROINFORMATIK, RUHR-UNIVERSITAT, Tech. Rep., 1998.

[15] E. Richter, R. Schubert, and G. Wanielik, "Radar and Vision-based Data Fusion - Advanced Filtering Techniques for a Multi-object Vehicle Tracking System," in *Intelligent Vehicles Symposium, 2008 IEEE*, june 2008, pp. 120–125.

[16] A. Broggi, P. Cerri, and P. Antonello, "Multi-resolution Vehicle Detection using Artificial Vision," in *Intelligent Vehicles Symposium, 2004 IEEE*, june 2004, pp. 310–314.

[17] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle Detection Combining Gradient Analysis and AdaBoost Classification," in *Intelligent Transportation Systems Conf.*, sept. 2005, pp. 66–71.

[18] T. Kowsari, S. Beauchemin, and J. Cho, "Real-time Vehicle Detection and Tracking using Stereo Vision and Multi-view AdaBoost," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, oct. 2011, pp. 1255–1260.

[19] Z. Sun, G. Bebis, and R. Miller, "On-road Vehicle Detection: a Review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 5, pp. 694–711, may 2006.

[20] J. Arróspide, L. Salgado, and M. Nieto, "Vehicle Detection and Tracking using Homography-based Plane Rectification and Particle Filtering," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, june 2010, pp. 150–155.

[21] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime Multibody Visual SLAM with a Smoothly Moving Monocular Camera," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[22] J. Sochman, "Learning for Sequential Classification," Ph.D. dissertation, CTU in Prague, 2009.

[23] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[24] R. Collins and Y. Liu, "On-line Selection of Discriminative Tracking Features," in *IEEE International Conference on Computer Vision (ICCV)*, oct. 2003, pp. 346–352 vol.1.

[25] H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via On-line Boosting," in *Proc. BMVC*, 2006, pp. 6.1–6.10, doi:10.5244/C.20.6.

[26] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-Tracking Using Semi-supervised Support Vector Machines," in *IEEE 11th International Conference on Computer Vision (ICCV)*, oct. 2007, pp. 1–8.

[27] Q. Yu, T. B. Dinh, and G. Medioni, "Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ser. ECCV '08, 2008, pp. 678–691.

[28] L. Cehovin, M. Kristan, and A. Leonardis, "An Adaptive Coupled-layer Visual Model for Robust Visual Tracking," in *IEEE International Conference on Computer Vision (ICCV)*, nov. 2011, pp. 1363–1370.

[29] T. Vojíř and J. Matas, "Robustifying the Flock of Trackers," in *Computer Vision Winter Workshop*, 2011.

[30] Z. Kalal, J. Matas, and K. Mikolajczyk, "Weighted Sampling for Large-scale Boosting," *Proc. Brit. Machine Vision Conf*, 2008.

[31] J. Trefný and J. Matas, "Extended Set of Local Binary Patterns for Rapid Object Detection," in *CVWW 10: Proceedings of the Computer Vision Winter Workshop 2010*, L. Špaček and V. Franc, Eds. Prague: Czech Society for Cybernetics and Informatics, 2 2010, pp. 37–43.

[32] R. Schapire and Y. Singer, "Improved Boosting Algorithms using Confidence-rated Predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[33] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA)," in *Proc. of the DARPA Image Understanding Workshop*, April 1981, pp. 121–130.

[34] L. B. Statistics and L. Breiman, "Random Forests," in *Machine Learning*, 2001, pp. 5–32.

[35] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast Keypoint Recognition Using Random Ferns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 448–461, march 2010.

[36] A. Broggi, S. Debattisti, M. Panciroli, P. Grisleri, E. Cardarelli, M. Buzzoni, and P. Versari, "High Performance Multi-track Recording System for Automotive Applications," *Intl. Journal of Automotive Technology*, vol. 13, no. 1, Jan. 2011.

[37] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Computer Vision and Pattern Recognition (CVPR)*, Providence, USA, June 2012.