Scene context is more than a Bayesian prior: competitive vehicle detection with restricted detectors

Thomas Hecht[1], Mrinal Mohit[2], Egor Sattarov[3] and Alexander Gepperth[1]

***Abstract*— We present a new approach for making use of scene or situation context in object detection, aiming for state-of-the-art performance while dramatically reducing computational cost. While existing approaches are inspired by Bayes' rule, training context-independent detectors and combining them with context priors in hindsight, we propose to integrate these context priors already into detector design, through algorithmic choices and/or pre-selection of training examples. Although such *restricted detectors* will, as a consequence, be valid only in regions compatible with context priors, the corresponding simplification of the object-vs- background decision problem will lead to reduced computation time and/or increased detection performance. We verify this experimentally by analyzing vehicle detection performance in a realistically simulated inner-city environment where context priors are defined by a road surface mask obtained from the simulation tool. Comparing a restricted detector, based on horizontal gradient detection refined by neural network confirmation, to a generic HOG+SVM-based approach taking into account the road context prior, we show that the restricted detector shows superior vehicle detection performance at a vastly reduced computational cost. We show qualitative results that permit the conclusion that the restricted detector will perform well on real-world scenes if appropriate road context priors are available.**

I. Introduction

a) General context: This article is concerned with the real-time detection of relevant traffic objects in complex environments, e.g., inner-city road traffic. The nature of the encountered scenes poses grave difficulties to object recognition approaches based on the recognition of local patterns only, as the exact same patterns may have different semantics depending on, e.g., where in a scene they are encountered. To counter this, scene context is by now widely used to control and restrict the detection process[17], [13], [22], [9], [6].

b) Approach overview: We propose a new approach to incorporate scene context into object detection and validate it by evaluations in a simulated inner-city vehicle detection task with high visual realism, showing that an exemplary implementation of our approach achieves very high detection accuracy at dramatically increased execution speed. The approach relies on the appropriate *restriction* of search areas by scene context priors, and on the corresponding creation of *restricted detectors*, which will not work anywhere else.

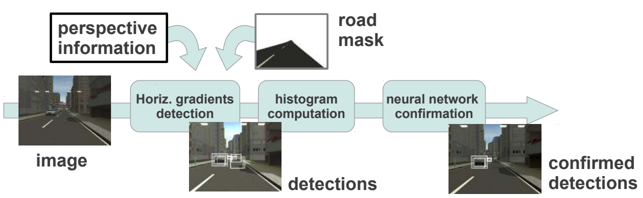

Fig. 1. General architecture of the restricted detector we propose. The input image is filtered for horizontal gradients which meet perspective and road constraints. Ensuing vehicle hypotheses are passed to a neural network confirmation stage which evaluates color and orientation histogram information within each hypothesis, leading to a confirmation or rejection of each hypothesis. This simple scheme works well because the restrictions imposed by road shape and perspective lead to an extreme simplification of the discrimination problem which the hand-crafted detection process takes into account. Our idea is that this could be a general principle in designing efficient object detectors.

The advantage of this restriction approach is the usually very strong simplification of the detection problem thus achieved: as a detector no longer needs to reject all possible kinds of background objects that may occur in places where no detections will take place anyway, it can direct the resources thus freed towards more productive uses. For a support vector machine, for example, a reduced problem complexity will usually be reflected in a lower number of support vectors, leading to a correspondingly increased execution speed.

For the purposes of this study, we chose to implement a vehicle detector along these principles. Bluntly, it poses the question ”If it's on the road, has the right size and has a horizontal lower edge, what else would it be but a car?”. The processing tool-chain of our restricted detector implementation is depicted in Fig. 1. We employ a mix of manual and adaptive model selection by adopting a two-step procedure where a very simple gradient-based heuristics selects vehicle hypotheses, and a trained neural network confirmation module filters these hypotheses based on simple-to-compute local image features. In line with our philosophy of restricted detectors, training examples for the neural network are selected exclusively from the road area.

c) Related work: The simple gradient-based heuristics we propose for our restricted detector was often adopted a decade ago[11] when vehicle detection was only performed on highways where there are few of the distractors encountered in inner-city scenes (shadows, irregular lane markings, adjoining buildings etc.). With the advent of more powerful computers, these approaches were mostly abandoned, and our approach contrasts with most recent work on the subject of vehicle, or more generally traffic object detection. In most

[1] Alexander Gepperth and Thomas Hecht are with ENSTA ParisTech, 828 Blvd des Marechaux, 91762 Palaiseau, France `firstname.lastname@ensta-paristech.fr`

[2] Mrinal Mohit is with the Indian Institute of Technology Kharagpur, Kharagpur, India - 721302 `mrinalmohitiit@gmail.com`

[3] Egor Sattarov is with Universit de Paris-Sud, Rue Noetzlin (Bat.660), 91190 Gif sur Yvette, France `egor.sattarov@upsud.fr`

current studies, context information is implicitly applied during detector training by selecting the most difficult negative examples in an automated[7] or semi-automated fashion by bootstrapping[19].

Additionally, many approaches apply context priors to the detected objects in hindsight[10], [8],excluding all detections not compatible with context priors. Bayesian fusion of detection likelihoods and context priors may also be performed, with subsequent decision about detections [9], [21], [15].

Common to all of these approaches is the training of *generic detectors* which are trained and evaluated independently and only combined with context after training which usually improves results considerably.

This independence of detector and context is normally reflected in a huge imbalance between positive (few) and negative training examples (very many) as defining the non-object class can, in general, be very difficult and takes up most of the resources of the detector, e.g., in the form of negative support vectors. If however context priors can restrict already the training problem, detectors may become much simpler, or use the freed resources for improved recognition performance.

II. Methods

The processing system we propose is organized as shown in Fig. 1. In this section, all relevant parts of the system are discussed individually, as well as the simulation tool used to generate training and test data.

A. The nisys Traffic Simulator

The nisys Traffic Simulator is a commercial product[1] targeting industrial research studies. It is capable of rendering realistic traffic scenes seen from a moving vehicle in real-time, making use of GPU acceleration if available. Scenes can either be randomly generated, in this case controlled by meta-parameters, or be completely user-defined. In the latter case, some of the degrees of freedom include:

- road geometry and appearance (own bitmaps can be supplied for almost all objects)
- placement of static objects: traffic signs, billboards, trees, houses, street lights, shrubberies, guardrails, trees
- placement of dynamic objects: pedestrians
- placement of other traffic participants (vehicles) with their own driver model, chosen out of a large number of 3D vehicle models
- complex parametrized driver models
- sun position, light type (diffuse, ambient,..) and shadow generation
- daylight/night and weather conditions (rain, snow, fog, ..)

Furthermore, the Traffic Simulation software can be remotely controlled via an XML-RPC network interface that is accessible from most programming languages, notably Matlab, Python and Lua. Via this interface, nearly all parameters of the simulation can be set and queried at run-time. In particular, the 2D positions and speeds of all currently visible vehicles can be obtained in this way which facilitated the creation of annotated training and test videos as explained in the following section.

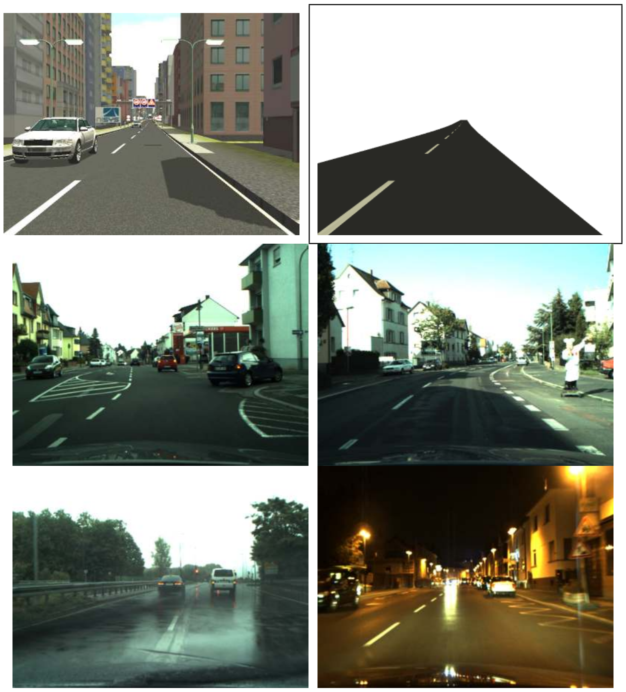

Fig. 2. Databases used for training and testing. Top row: simulated traffic image and corresponding road mask obtained from the simulation tool. Middle row, bottom row: images from substreams I,II,III,IV of the HRI RoadTraffic benchmark, corresponding to overcast/dry, low sun/dry, rain, and night conditions. Please note the realistic rendering of shadows on the road by the simulator as compared to similar shadows in stream II (middle row, right image).

B. Databases for training and testing

For training a *baseline detector* that will serve as a reference for the evaluation of the proposed restricted detector, as well as for evaluating its general ability to provide a performance baseline, we use the publicly available HRI RoadTraffic dataset[9], see also Fig. 2. For training the restricted detector, a sequence denoted S_1 consisting of 3000 color images generated at a rate of 10Hz, of resolution 640x480, is created using nisys TrafficSimulator, see II-A. For comparing the baseline and the restricted detectors, another sequence, denoted S_2, is generated using nisys TrafficSimulator. Care is taken to make S_1 and S_2 sufficiently dissimilar by using randomly generated sequences of buildings, trees, shrubberies, traffic signs, message boards and oncoming vehicles, as well as randomly generated curves and 3D road profiles. Due to the modeling of shadows, all of this has strong effects on the visual appearance of a scene so we are confident that sufficient dissimilarity is achieved. For running the restricted detector, road masks were obtained by supplying the simulator with own bitmaps for the road surface, containing a certain characteristic color, and later

[1] see www.nisys.de for further information

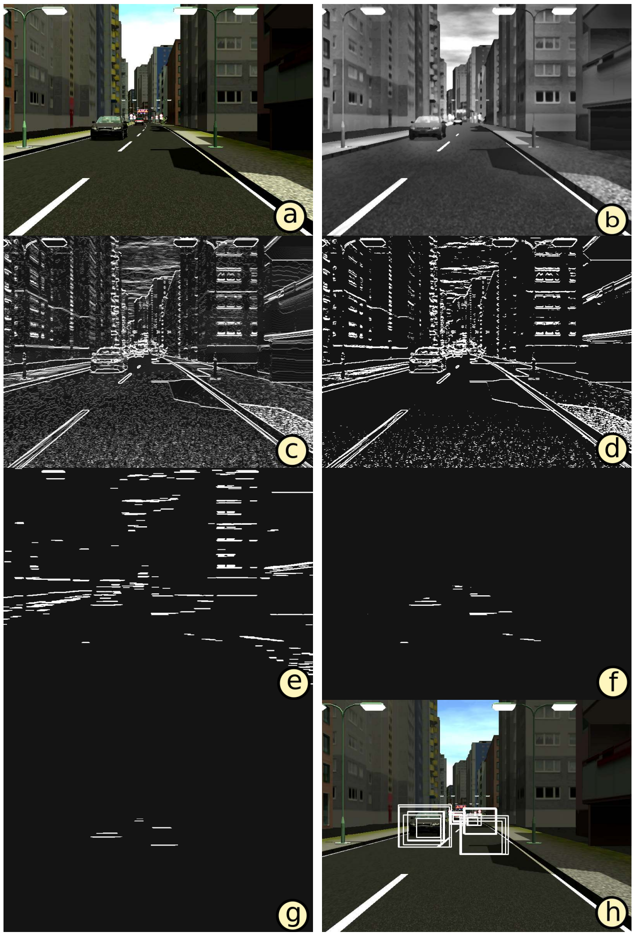

Fig. 3. Individual steps of the gradient-based detection process. (a) original color image with two visible cars. (b) gray-level image after adaptive histogram equalization and median filtering. (c) horizontal gradients obtained by Scharr filtering. (d) results of adaptive thresholds binarization. (e) result of morphological operations (erosion+dilation). (f) result of multiplication with the binary road mask. (g) lines detected by probabilistic Hough transform. (h) obtained detection boxes. Please note the extreme effect of applying the road mask by comparing images (e) and (f).

filtering the simulated images for this particular color[2].

C. Gradient-based hypotheses detection

The initial hypothesis generator is based on the observation that vehicles almost always exhibit strong horizontal gradients at or close to their lower border[11]. The detection of such oriented gradients is a very efficient operation, which makes this approach appealing, although evidently several pre-and post-processing steps are necessary to bring detection performance to competitive levels.

Fig. 3 outlines the various processing steps of the detector, whereas the following paragraphs given an in-depth overview of individual processing steps.

d) Preprocessing: Adaptive histogram equalization[20] is applied to 11x11 blocks of the gray-level version of the original color image. As this technique tends to amplify noise, we apply a 5x5 median filter beforehand, which is preferable to a Gaussian filter as it does not eliminate local structure while still removing noise[1], [2].

e) Oriented line detection: Horizontal gradients are computed using a Scharr filter which reputedly gives better edge detection precision than Sobel filtering[14]. Subsequently, an automated thresholding procedure is applied to identify the strongest gradients[18]. We subsequently apply a morphological erosion operation with a linear kernel and dilate the result. The result of this operation, which is still a binary image, is AND-combined with the binary image of the road mask (coming from the simulator). A probabilistic Hough line detection algorithm[16] is subsequently applied, with parameters $\Theta_\rho = 1$, $\Theta_\theta = \frac{\pi}{2}$, $\Theta_\nu = 6$, $\Theta_\Lambda = 6$ and $\Theta_\gamma = 6$, which respectively denote the thresholds for the perpendicular distance to origin, the angle with the horizontal axis, the number of votes, the minimal size in pixels, and the distance of pixels to another line. Each more or less horizontal line found in this manner will trigger a rectangular detection, where the image coordinates of the detection rectangle are computed as indicated in Fig. 4.

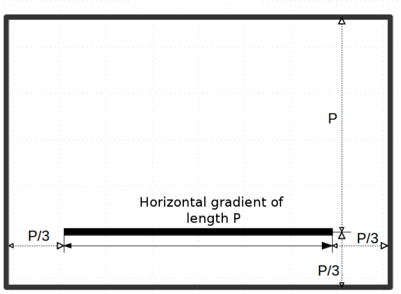

Fig. 4. Creation of a detection rectangle from a detected horizontal line. The rectangle is chosen larger than the horizontal line that triggers it because the shadow under a car is usually more narrow than the car itself. Some tuning was required to achieve satisfactory results.

f) Post-processing: We apply a perspective-based post-processing step to the set of detection rectangles. As we suppose that detections are placed on the ground plane which we assume to be flat, we can calculate the width L of detections in 3D according to the formula $L = \kappa_s * \frac{P}{y}$, where κ_s indicates the height of the camera in meters (known from simulator), P the width of the segment in pixels and y the distance in pixels to the horizon line which is always in the middle of the image. Here we use the fact that the simulated camera is mounted in exact parallel alignment to the ground plane; more complex camera positions would require slightly more complex formulas which would however achieve exactly the same thing. We eliminate a detection if its width L is not between 1.5 and 2.5 meters.

D. Visual features extraction

The set of detection rectangles obtained from the gradient-based detector is analyzed for its visual content with the goal of providing a concise yet discriminative description for subsequent neural network analysis. Essentially we are looking for a transformation $D_i \rightarrow f_i$ which maps an image sub-rectangle D_i to a vector of visual characteristics f_i

[2] These two sequences, along with road masks and annotations, may be downloaded from www.gepperth.net/downloads.html

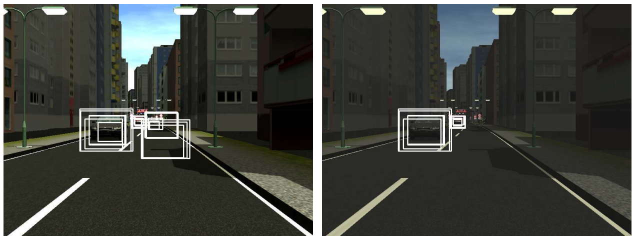

Fig. 5. Typical effects when filtering gradient-based detection results (left) by the neural network confirmation module (right). In particular, we observe that the detection of shadows on the road is prevented by the confirmation step.

which will facilitate subsequent object category decisions. To this end, we analyze each D_i in terms of color and form. More precisely, we compute feature histograms along the modalities of color and local orientation which have the pleasant property of constant dimensionality, independently of the pixel size of a detection. For color, we transform the color image to the HSV color space and compute normalized histograms from the Hue, Saturation and Value channels, each histogram performing a quantization into C_H, C_S and C_V bins.

We compute local orientation by convolving D_i with the horizontal and vertical Scharr kernels S_h, S_V:

$$S_h(x,y) = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix} S_v(x,y) \quad = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix} \tag{1}$$

and subsequently computing $\phi(x,y)$ as

$$\phi(x,y) = arctan\left(\frac{(S_v * D_i)(x,y)}{(S_h * D_i)(x,y)}\right) \tag{2}$$

A histogram with C_O bins is computed over all orientations thus obtained from D_i. The final feature vector f_i is formed by simple concatenation of all color and orientation histograms and therefore has dimension $\Upsilon = C_H + C_S + C_V + C_O$.

E. Neural network based hypothesis confirmation

a) Training data generation: Given a video sequence, in our case sequence S_1 obtained from the simulator described in Sec. II-B, the gradient-based detector is run on each image, and each detection D_i is automatically grouped into the classes ”vehicle” or ”non-vehicle” based on whether or not it overlaps with an annotation A_j. Annotations are 2D bounding boxes obtained from the simulator which tightly enclose a single vehicle in the image. Overlap is determined using a standard overlap criterion:

$$o(D_i, D_j) = \frac{D_i \cap A_j}{D_i \cup A_j} \tag{3}$$

for which we chose a threshold of $\theta_O = 0.5$. This way of collecting training samples for classifier training ensures that the training and eventual test data come from exactly the same probability distribution (as required by statistical learning theory) since they are generated by the exact same process.

b) Network parameters: A neural network classifier, more precisely a multilayer perceptron[12] is trained to sort detections D_i into the categories ”vehicle” and ”non-vehicle”, relying on the descriptor f_i whose computation from a detection D_i found by the gradient-based detector is outlined in Sec. II-D. The network has Υ input neurons, $\frac{\Upsilon+2}{2}$ hidden neurons and a single output neuron, with a sigmoid activation function $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$ applied to all layers. Each layer contains a bias unit. The training algorithm is momentum-based back-propagation using a maximal number of epochs N, learning rate ϵ, gradient step γ and momentum $\mu = 0.1$. A target value of 0.98 for the output neuron is supplied for the ”vehicle” class, whereas the ”non-vehicle” class requires an output of 0.02.

c) Decision making and ROC generation: Due to the sigmoid transfer function, the single output neuron will respond to input feature vectors f_i with values in the interval $o(f_i) \in]0,1[$, expressing a confidence about the presence of either the ”vehicle” or the ”non-vehicle” class. For ROC computation, we threshold this confidence using a variable threshold between 0 and 1, and plot the resulting values for recall and false detections per image. Prior to ROC computation, a non-maxima suppression step as described in Sec. II-F is carried out on the set of all detections augmented by neural network scores. For deciding whether a detection D_i is a false, missed or correct detection, we use the overlap measure $o(D_i, D_j)$ of Eqn. (3), demanding that a detection D_i and and at least one annotation D_j have an overlap $o(D_i, D_j) \geq 0.35$.

F. Non-maxima suppression

Both for evaluating baseline and restricted detectors, non-maxima suppression is performed to reduce false detections. This standard post-processing step in object detection expects a set of rectangular detections with associated real-values scores $\{D_i, s_i\}$, and produces a thinned out list of detections/scores where only the locally most confident detections survive. In detail, the algorithm runs as follows, relying on the overlap measure $o(D_i, D_j)$ defined in Eqn. (3):

```
**Algorithm** *Simple NMS*($\{D_i, s_i\}$)
Sort $\{D_i, s_i\}$ in descending order of score $s_i$
for $i \leftarrow 1$ to $N$
    for $j \leftarrow i+1$ to $N$
        if $D_i$ not marked for deletion
          then
            if $o(D_i, D_j) \geq \theta_{\text{NMS}}$
              then mark $D_j$ for deletion
  Erase marked detections and return list
```

In all evaluations, a threshold value of $\theta_{\text{NMS}} = 0.1$ is used.

G. Baseline detector for comparison

For providing a performance baseline, we train a support vector machine (SVM) with a linear kernel and $C = 1$ and apply it in a sliding-window fashion at 5 spatial scales to each image of the test videos described in Sec. II-B. In

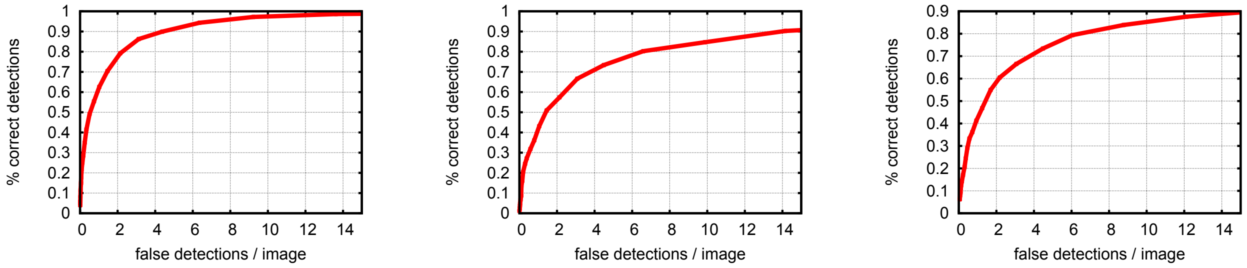

Fig. 6. Vehicle detection performance of the baseline detector on various videos from the HRI RoadTraffic dataset, demonstrating competitive performance of the baseline detector which justifies its use a reference system. Left: stream I (daylight, overcast), middle: stream II (daylight, low sun), right: stream III (daylight:rainy).

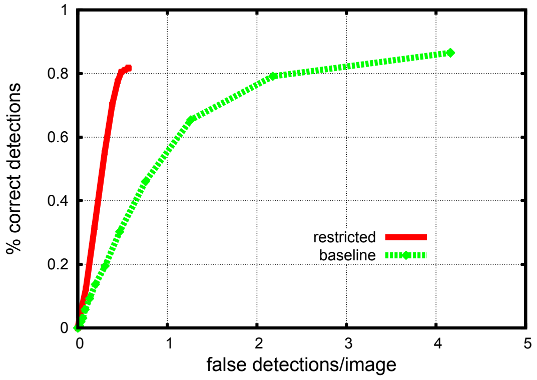

Fig. 7. Comparison of baseline and restricted detectors (see text for comparison details) using ROC analysis on the simulated video stream S_2, see Sec. II-B. A clear superiority of the restricted detector may be observed. The performance of the restricted detector does not increase beyond 0.9 on the y axis, which is due to the gradient-based detection step that is not affected by varying the threshold of the neural network confirmation module. Detections that are therefore missed by this step can not be detected whatever threshold is applied to the confirmation module.

this, we follow the method outlined in [5]. As we wish to compare our results to a detector for which we can verify decent real-world performance, SVM training is performed on the publicly available HRI RoadTraffic[9] dataset. More precisely, we train an SVM on 50% of the vehicle objects in substreams I,II and III which correspond to overcast, low sun and rain conditions, where each substream is split into alternating train/test intervals of 30s. HOG parameters are (following the terms of [5] and using standard parameters whenever not mentioned specifically): block size 8x8 pixels, cell size 4x4 pixels, window size 48x48 pixels, window stride 4x4 pixels.

Initially a set of ”non-vehicle” examples is selected randomly and is refined in successive bootstrapping[19] iterations. For ROC generation, we let the SVM compute the distance of an evaluation example to the optimal separating hyperplane acquired during learning. A variable threshold is applied to this ”confidence” in order to take a binary decision about the presence of a vehicle, which allows the plotting of the resulting detection and false detection rates in a ROC. Prior to ROC computation, a non-maxima suppression (NMS) step is performed as described in Sec. II-F. For deciding whether a detection D_i is a false, missed or correct detection, we use the overlap measure $o(D_i, D_j)$ of Eqn. (3), demanding that a detection D_i and and at least one annotation D_j have an overlap $o(D_i, D_j) \geq 0.35$.

H. *Implementation issues*

All described algorithms are implemented using the free OpenCV computer vision library[3] in its version 2.4.6, either through the Python interface (gradient-based detector, neural network) or the native C++ API (HOG+SVM baseline). For SVM training we use the libsvm toolbox[4].

III. Experiments

We conduct two quantitative and one qualitative experiments with the aim of comparing the potential of baseline detector (standard HOG+SVM model, see Sec. II-G) and the restricted detector (horizontal gradient detection and subsequent neural network classification, see Secs. II-C, II-D, II-E):

- Real videos: training and testing of the baseline detector
- Simulated videos: training and test of the restricted detector and subsequent comparison to baseline
- Real videos: Qualitative evaluation of gradient-based vehicle detection

A. *Training and evaluation of baseline detector*

We train and evaluate the baseline detector using the HRI RoadTraffic benchmark dataset as described in Sec. II-G. After 5 iterations, a ROC evaluation on the test objects in each substream clearly shows that a performance comparable to that reported in [9] is achieved, see Fig. 6. This tells us that the trained baseline detector can provide a valid point of reference for assessing the performance of the restricted detector described previously. At the last bootstrapping iteration, SVM training includes 2972 vehicle and 60.000 non-vehicle examples.

B. Comparison of baseline and restricted detector

We evaluate the baseline detector parametrized as described in Sec. II-G on the simulated video stream S_2. To ensure a more or less fair comparison, we filter baseline detection results by the same road mask used for the restricted detector. The confirmation module of the restricted detector is first trained on training examples from video S_1 as laid down in Sec. II-E and subsequently evaluated on video S_2 as well. Both for training and evaluating the restricted detector, the restricted detector is parametrized as follows, using the notation and procedures of Sec. II-C: we choose 40 histogram bins for each channel of the HSV histograms and 180 bins for orientation histograms, which gives a total feature vector size of $\Upsilon = 180 + 3 \times 40 = 300$. This is at the same time the size of the input layer of the neural network, whose hidden layer has $\frac{\Upsilon+2}{2} = 151$ elements. Using the method described in Sec. II-E, we collect in total 9970 vehicle and 8500 non-vehicle examples for network training. Back-propagation training parameters are: $N = 500, \epsilon = 0.01, \gamma = 0.1, \mu = 0.1$.

We want to stress that video S_2 which is used for performance evaluation and comparison, was not used in any way for the training of either the baseline or the restricted detectors. As the results of Fig. 7 plainly show, the restricted detector outperforms the baseline detector my a significant margin even though we apply the road mask to the baseline detector as well. As far as execution speed is concerned, the baseline detector, which is implemented in pure C++, runs at about 1.5 Hz on our Pentium i5, 2.8 GHz desktop computer with 4GB RAM running Ubuntu Linux (without GPU acceleration or similar funny business). The restricted detector, which is implemented in Python, operates at about 25 Hz under the exact same conditions. These times are calculated for images of size 640x480 pixels. The restricted detector thus outperforms the baseline detector by a large margin in this respect, as well.

C. Feasibility of the restricted detector on real videos

In this qualitative experiment we wish to verify whether the restricted detector may be applicable to real traffic videos as well instead of simulated ones. To this end, we use again the HRI RoadTraffic benchmark, see [9] and Sec. II-B. Although this benchmark data set does contain road information, we wish to ensure its quality before basing an evaluation on it. We thus simply let the gradient-based detector run on stream I of the benchmark and collect particularly representative images to underline our conclusions. As the training of the neural network confirmation module would require a road mask, we omit the confirmation step as well. To restrict detections, we therefore just rely on perspective filtering as outlined in Sec.II-C, which on its own already removes any detections that lie above the horizon. The results are shown in Fig. 8 and suggest that the gradient-based detector, suitably restricted by a road mask and followed by an adequate confirmation module, has very good potential on real videos. We can conclude this because of the following observations: first of all, cars are detected in virtually all cases even when they are small, and secondly because the number of false detections remains reasonable despite the lack of a confirmation step.

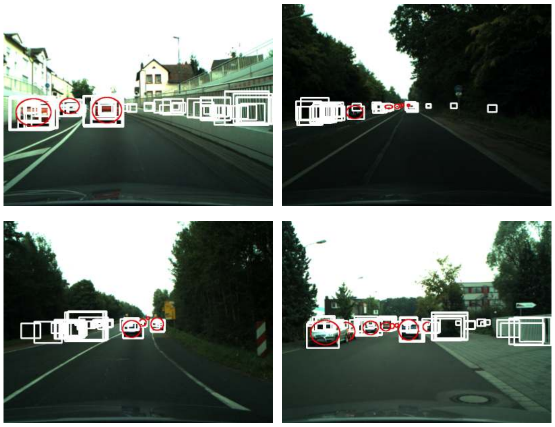

Fig. 8. Qualitative impressions when running the gradient-based detector on real video streams. White boxes: detections, red circles: actually present cars. It can be observed that vehicles are found very reliably (few missed detections), and that most false detections could already be eliminated by using knowledge about the road.

IV. Discussion

a) Summary: We have proposed the concept of restricted object detectors, where restrictions arise from scene or situation context priors that can be computed on-the-fly. To validate this concept, we implemented an exemplary restricted detector for vehicles and showed that its vehicle detection performance is superior to a state-of-the-art visual vehicle detector based on the well-known HOG+SVM technique [5]. We showed beforehand that this baseline detector is a valid reference system by evaluating its performance on a public benchmark database and finding performance comparable to that reported in previous studies. What is more, the increased performance goes along with a 10-fold *increase* in execution speed. Lastly, qualitative experiments at least make it plausible that the restricted detector may perform just as well in real scenes when road information is available.

b) Critical review of experiments: The comparisons made in Sec. III are not completely fair. In particular, the baseline detector is not trained on synthetic data but on real ones, thus it may be argued its performance is not what it could be. While this is certainly true to an extent, one may object that the gradient detection process is not specifically trained on synthetic data either, and will work conveniently on real videos as shown in Sec. III-C. Furthermore, the baseline detector profits from the same road mask operation as the restricted one, thus arguably improving detection accuracy by a significant margin, which is illustrated by comparing its results on synthetic and real scenes (Figs.6, 7). To conclude, we may state that the comparison is not completely ideal but very far from being significantly biased in one direction or the other. Another criticism concerns the comparison of execution times, since the time for computing the road mask should arguably be added to the execution time of the restricted detector. However, as the road mask may be used for many different purposes, including obstacle avoidance and navigation, we believe that it should at the very least be attributed equally to all processes making use of road information, which would make the execution of the restricted detector still competitive.

c) Next steps and outlook: The logical next step is to apply a refined version of the restricted detector presented here to real vehicle detection benchmarks, either computing the road on-the-fly in the manner of [10], or obtaining it from annotations. The potential features for use by restricted detectors are huge: stereo information, information about hypothesis movement coming from a tracker, and many more could be used for pre-selecting hypotheses and reduce the decision problem even more than could the road mask alone.

Subsequently, an application to pedestrian detection is targeted, in combination with a detection of sidewalks. Generally, we expect that any detection problem where context can simplify the discrimination task sufficiently will profit from the systematic design of restricted detectors.

References

[1] E. Arias-Castro and D. L. Donoho. Does median filtering truly preserve edges better than linear filtering? *The Annals of Statistics*, page 11721206, 2009.

[2] A. C. Bovik, T. S. Huang, and D. C. Munson. The effect of median filtering on edge estimation and detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):181194, 1987.

[3] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated, 2008.

[4] C. Chang and C. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[6] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *International Conference on Computer Vision (ICCV)*, 2009.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, Sept. 2010.

[8] A. Gepperth, B. Dittes, and M. Garcia Ortiz. The contribution of context information: a case study of object recognition in an intelligent car. *Neurocomputing*, 2012.

[9] A. Gepperth, S. Rebhan, S. Hasler, and J. Fritsch. Biased competition in visual processing hierarchies: A learning approach using multiple cues. *Cognitive Computation*, 3(1):146–166, 2011.

[10] C. Guo, J. Meguro, M. Kojima, and T. Naito. Detection of pedestrians in road context for intelligent vehicles and advanced driver assistance systems. In *ITSC*, 2013.

[11] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen. An image processing system for driver assistance. *Image and Vision Computing*, 2000.

[12] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 1999.

[13] D. Hoiem, A. Efros, and M. Hebert. Putting objects into perspective. *International Journal of Computer Vision*, 80(1), 2008.

[14] B. Jhne, H. Scharr, and S. Krkel. Principles of filter design. *Handbook of computer vision and applications*, 2:125151, 1999.

[15] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[16] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119137, 2000.

[17] K. Murphy, A. Torralba, D. Eaton, and W. Freeman. Object detection and localization using global and local features. In J. Ponce, editor, *Toward Category-Level Object Recognition*, Lecture Notes in Computer Science. Springer, 2005.

[18] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):2327, 1975.

[19] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Computer Vision, 1998. Sixth International Conference on*, pages 555–562, 1998.

[20] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355368, 1987.

[21] J. Schmuedderich, N. Einecke, S. Hasler, A. Gepperth, B. Bolder, R. Kastner, M. Franzius, S. Rebhan, B. Dittes, H. Wersing, J. Eggert, J. Fritsch, and C. Goerick. System approach for multi-purpose representations of traffic scene elements. In *International IEEE Annual Conference on Intelligent Transportation Systems*, 2010.

[22] J. Vogel and O. D. Freitas. Target-directed attention: Sequential decision-making for gaze planning. In *International Conference on Robotics and Automation (ICRA)*, 2007.