2014 IEEE Intelligent Vehicles Symposium (IV)
June 8-11, 2014. Dearborn, Michigan, USA

Vehicle Detection for Autonomous Parking using a Soft-Cascade AdaBoost Classifier

Alberto Broggi[1], Elena Cardarelli[1], Stefano Cattani[1], Paolo Medici[1] and Mario Sabbatelli[1]

***Abstract*— This paper presents a monocular algorithm for front and rear vehicle detection, developed as part of the FP7 V-Charge project's perception system. The system is made of an AdaBoost classifier with Haar Features Decision Stump. It processes several virtual perspective images, obtained by unwarping 4 monocular fish-eye cameras mounted all-around an autonomous electric car. The target scenario is the automated valet parking, but the presented technique fits well in any general urban and highway environment. A great attention has been given to optimize the computational performance. The accuracy in the detection and a low computation costs are provided by combining a multiscale detection scheme with a Soft-Cascade classifier design. The algorithm runs in real time on the project's hardware platform.**

The system has been tested on a validation set, compared with several AdaBoost schemes, and the corresponding results and statistics are also reported.

Introduction

Autonomous ground vehicles (AGV) driving in dynamic environment, as well as top-notch advanced driver assistance systems (ADAS), require a reliable perception of the 360 ° environment around the vehicle.

Existing commercial ADAS systems (such as lane departure warning, adaptive cruise control) and collision warning depend on the perception of the environment in front of the car. More complex systems, like traffic jam and evasion assistance systems, control the vehicle in both longitudinal and lateral direction, but are yet to come on vehicles in series production.

On the research side, current state of the art in AGV driving [1] provides reliable all-around perception, at the cost of adopting a highly sophisticated suite of expensive sensors, such as sweeping laser range finders, RADAR systems, and color cameras.

The EU's 7th Frame Programme of Research V-Charge [2] (www.v-charge.eu) seeks to address these problems simultaneously: on one hand it is aimed to design and develop an autonomous electric car prototype, able to self parking while dealing with a highly dynamic environment; on the other hand, V-Charge guidelines were settled to enforce the adoption of close-to-market sensors, like *monocular wide angle field of views cameras*: they are cost effective, light and robust, easy to be installed and cleanly integrated on common cars; they also consume minimum energy. The resulting sensors layout is shown in Fig. 1: just one stereo camera for long range narrow angle obstacles detection, plus 4 monocular fish-eye cameras, aimed to detect moving obstacles all-round the vehicle.

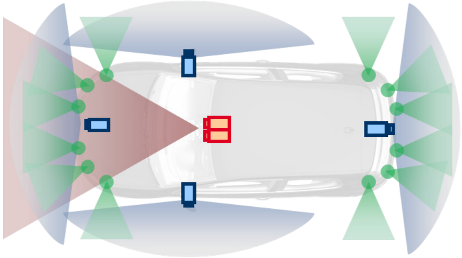


Fig. 1. V-Charge's sensors layout: one stereo camera on the front, for long range narrow angle detection; 4 fish-eye monocular cameras for all-round view obstacle detection; short range collision avoidance sonars (in green).

As the main contribution of this paper, we present a monocular algorithm for frontal and rear vehicle detection, developed as part of the V-Charge perception system; it is based on an AdaBoost classifier [3] which uses Decision Stump on Haar features as weak learner. The presented approach uses a multiple scales detection scheme coupled with a Soft-Cascade algorithm [4] to reduce the computational burden.

A. Related Work

Objects detection and classification, using monocular cameras installed on a moving vehicle, has been widely investigated by the Intelligent Vehicles community. An important milestone in application of machine learning techniques to real-time object detection is represented by the the cascaded classifier of Viola and Jones [5]: here a set of simple classifiers, working on simple-to-evaluate Haar-like features, are applied subsequently to a region of interest until an exit condition is reached (i.e. candidate is rejected or no more classifiers left). Several improvements to this techniques have been presented by many authors through the years, like in [6], [7]. Again, on the machine learning track it is possible to find several approaches that try to overcome some of the typical problems of passive learning, introducing the concept of active learning, where system exhibits some degree of control over the inputs on which it trains [8].

A different approach is part-based detector, where classifiers are trained to recognize a specific and peculiar part of a vehicle [9], [10]. Other authors try to exploit more heuristic vehicles features, like the shadow cast on the road (assumed to be darker then the rest of the scene), the symmetry of the vehicle body, the presence of rear and front light, etc.

[1] A. Broggi, E. Cardarelli, S. Cattani, P. Medici and M. Sabbatelli are with VisLab - Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy {broggi, cardar, cattani, medici, smario}@vislab.it

978-1-4799-3637-3/14/$31.00 ©2014 IEEE

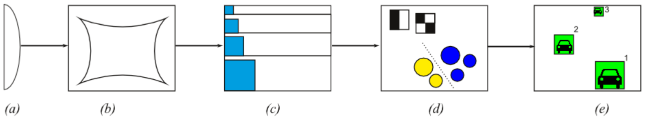


Fig. 2. Overall system design. *(a)* Image acquisition with fish-eye camera. *(b)* Calibration and lens distortion correction. *(c)* Definition of region of interest where investigate vehicle presence. *(d)* Execution of the AdaBoost classifier based on Haar features. *(e)* Tracking execution.

An exhaustive overview of these methods can be found in [11] and [12].

I. System Overview

The scheme in Figure 2 represents the principal steps of the proposed approach: perspective images, relative to the scene of the V-Charge vehicle surrounding area, are obtained from the corresponding fish-eyes then, an AdaBoost classifier, based on Haar features, is trained for vehicle recognition and executed in a set of region, each representing the theoretical area covered by potentials vehicles at different distances. Finally, to increase detection stability, a features based, odometry aided, 2D target tracking is performed, within the same camera/image and as well as among different adjacent cameras.

This paper focus only on detection, while tracking will be the subject of a further paper.

A. Perspective Virtual Views

Exploiting the extrinsic, intrinsic and distortion cameras' parameters [13], each fish-eye image (Figure 3) is reprojected onto several planes, obtaining a set of virtual unwarped pin-hole camera (i.e. perspective) images, as shown in Figure 4. These virtual planes are fully customizable in terms of their number, size, orientation and field of view.

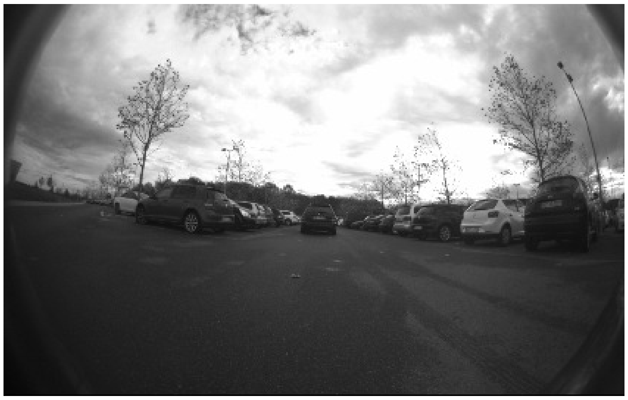


Fig. 3. Distorted image acquired with front fish-eye camera.

A procedure to correct fish-eye lens distortion is necessary to facilitate and speed up the training phase, as well as the determination of regions of interest (see Section II), since aspect ratio is generally preserved. However, radial distortion model, because of introducing noise and deformations on their sides, is not the most suitable to represent fish-eye images: in particular, the wider the resulting pin-hole field of view, the stronger these aberrations are. Virtual views approach partially overcomes these limitations, by splitting the single fish-eye image into several narrowed angle pin-hole images, keeping advantage of both wide angle fish-eye views and pin-hole unwarped model. Moreover, virtual views also allow special purpose images layout, specifically designed to deal with V-Charge scenarios. For example, when pulling out from a parking spot it is helpful to focus the vehicle detection along left and right axes: in this scenario, triggered by path planner, initially only *left and right virtual views* from *front camera* will be used; then, when the ego vehicle has pulled out enough, also left and right *cameras* will be activated, with their corresponding virtual *views*. On the other hand, when following a car in ACC mode, detection will be mainly focused on front camera's frontal view. Similarly, during normal driving, in no particular scenario, all the virtual views are activated.

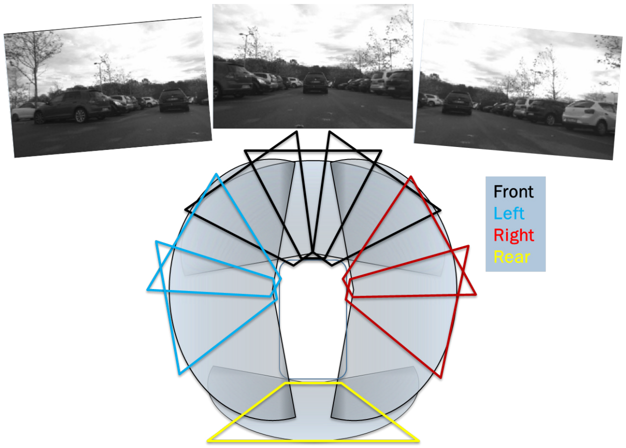


Fig. 4. Virtual views layout: 3 virtual views from front camera (black), 2 virtual views from left(blue) and right(red), 1 single virtual view from rear camera (yellow). On the upper row are shown the 3 virtual views obtained from the fish-eye image in Figure 3.

Virtual views are not the only way to deal with fish-eye distortion, but it is for sure the most straightforward and flexible; other models (e.g. Cylindrical [14]) will be considered for further developments.

II. Vehicles Detection

A Soft-Cascade AdaBoost classifier based on Haar features is trained for vehicle recognition and applied to the images to provide frontal and rear vehicle detection. This technique is certainly widespread in the literature and in recent years is being surpassed by other techniques such as SVM+HOG [15], [16] and, more recently, by SoftCascade+ICF [17]. For various reasons, however, it remains the fastest technique by a computational point of view and therefore remains worthy of investigation. Since the variants of this kind of techniques, proposed in the last years, are really large, section III describes all experiments performed and the results obtained in order to find the most suitable algorithm for the project.

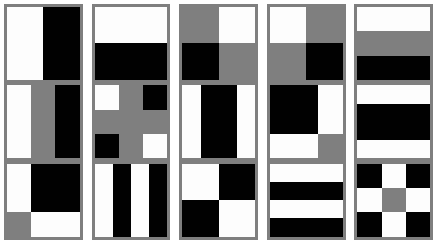


Fig. 5. Haar features involved in AdaBoost training.

One of the main contributions of this paper is the use of an expanded pool of Haar-like features. The list of available Haar features for the decision stump classifier is visible in Figure 5.

Particular attention is dedicated to the training of the AdaBoost classifier: the training patterns are extracted directly from the perspective images of the road environment to allow the vehicles recognition in real situation.

	Positive Samples		Negative Samples
	Front	*Rear*	
Training Set	10000	10000	$\sim$ 10000
Validation Set	5000	5000	100000

TABLE I

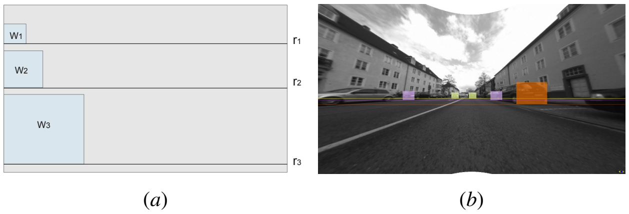
TRAINING AND VALIDATION SET DESIGN.

Therefore, as a result of this a set of 30000 samples, representing frontal and rear vehicle views, have been obtained and separated in training and validation set. Table I summarizes the design of the defined set: the training set consists of 20000 positives samples and approximately 10000 negative ones. 10 iterations of a bootstrap algorithm (like the one described, for example, in [18]) are used to select most representative negatives among billions of availables. In the first stage some negatives are randomly chosen and, in the subsequent stages, they are extracted from the worst false positives. The validation set, used to test the detector performance, consists of 10000 positives samples and 100000 negative ones.

Finally, to achieve a trade-off between detection accuracy and speed, a Soft-Cascade design [4] is adopted, allowing to introduce a significant computational saving. AdaBoost consists of a classification by majority system: all the weak classifiers have to be evaluated before the classification can made the final decision. Cascades techniques instead permit to evaluate weak classifiers in a certain order and to provide an early classification under certain circumstances. The Haar features are ranked and evaluated in sequence. For each feature a rejection threshold is computed in order to classify the sample as negative without proceeding further in the chain when the response exceed this value.

A. Multiscale Approach

Without additional information about vehicle position and distance (e.g. a laserscanner or a stereovision system) a multiscale approach is used to detect vehicles. It is important to note that the scales space is not continuous but need to be quantized: to balance between accuracy and performance 4 scales per octave were chosen. However, to improve the performance, the image is not processed entirely for each scales, but only in some specific areas. Particularly, the set of regions where to investigate, through the classifier execution, the presence of vehicles are determined considering three main constraints:



(*a*) (*b*)

Fig. 6. Determination of the region of interest through the world to image coordinates transform. *(a)* In the image, the rows r_1, r_2 and r_3 represent the detection distances of hypothetical vehicles, while w_1, $w2$ and $w3$ are the corresponding investigated windows. *(b)* The dimension of the searching window is defined according to the area occupied by a vehicle, with its typical dimension, at the specific distance represented by an image row. In this case different colors are used to enhance the strong relation between the dimension of each area processed and the relative image row.

- the typical range of vehicle size;
- the intrinsic and extrinsic calibration parameters, taking account of variation of pitch angle and height during normal vehicle motion;
- the world to image coordinates transform, calculated according to the pin-hole camera model.

Thus, each row in the input image is processed reconstructing, through the exploiting of the world to image coordinates transform, a set of windows. Each window corresponds to the theoretical area that a vehicle, with a predefined size, may occupy at the particular distance represented by the specific image row, with the assumption that the vehicle is strictly on the ground (Figure 6). For each scale and octave a different set of areas is generated.

To improve performances only 4 scales per octave are computed and, exploiting the peculiarity of Haar features, only scale images need to be computed. The octaves, in fact, are obtained not shrinking the original image by a factor 2, but enlarging the Haar features of the trained classifier by the same factor. This procedure is executed at algorithm run-time and not in the training phase. In this way, only 4 downsampled images are required to look for all possible size of vehicles, .

One of the big advantages of using Haar features is that their evaluation can be performed very efficiently exploiting the integral image. So, for each downsampled image an integral image should be computed. However it is possible to obtain directly the 4 integral images associated with the four scales without downsampling the source image, simply resampling with a bilinear filter the integral image associated with the original image. Exploiting this approach, the computational time for the downsampling and the regeneration of the integral images is totally saved.

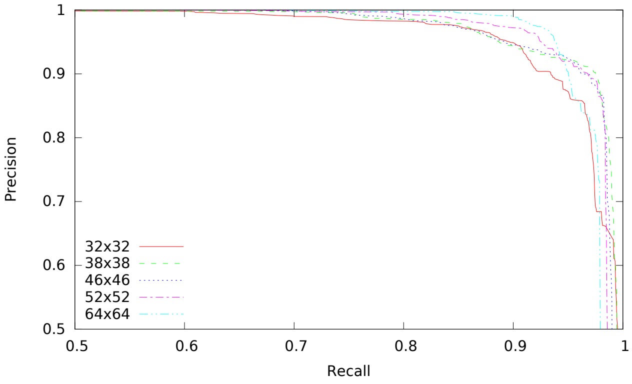


Fig. 7. Classification performance with different sizes of training samples using 300 Haar features and Discrete AdaBoost algorithm.

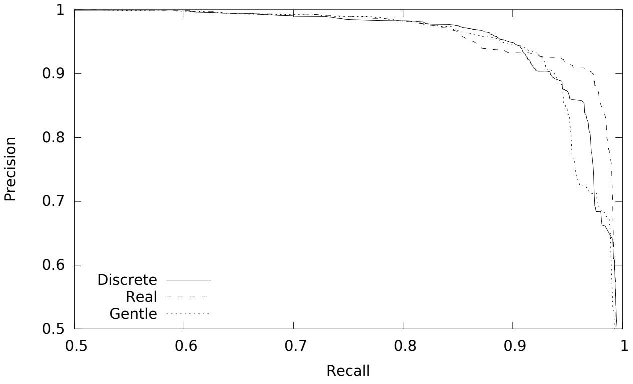


Fig. 9. Comparison among Gentle, Rear, Discrete AdaBoost for the recognition of the validation set using 300 Haar features.

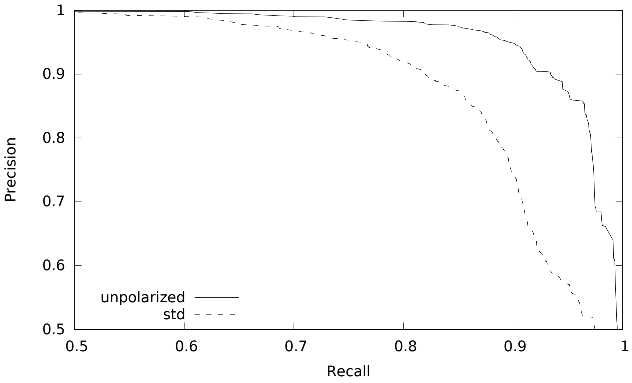


Fig. 8. Classification performance using classical 5 Haar features and proposed 15 unpolarized features (32 × 32 training size using 300 Haar features and Discrete AdaBoost scheme) .

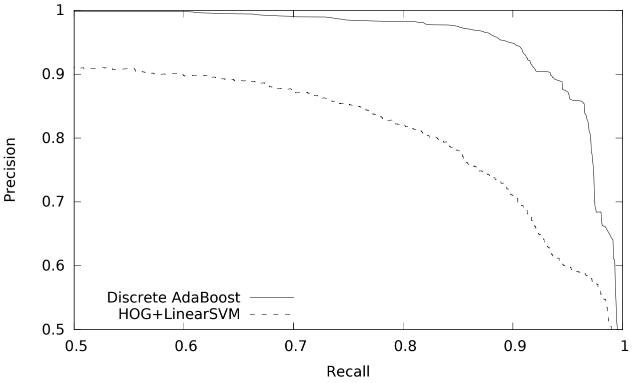


Fig. 10. Classification performance using AdaBoost with 300 Haar features and a 288 HOG descriptor using linear SVM on 32 × 32 samples.

III. Results

Firstly, different pattern dimensions are considered for the training of the classifier, from 32 × 32 pixels to 64 × 64 pixels. To take into account the information contained in the border of positive samples, all patterns are enlarged by 8% before being cropped and resampled.

The Precision-Recall curves in Figure 7 demonstrate that the bigger is the training samples, slightly better classification performance can be reached. However, since oversampling is not performed, the 32 × 32 pixels training samples allow to achieve greater detection distances than others. Therefore, the training pattern dimension used for the final classifier is 32 × 32 giving more priority to the capability of covering great detection distances, with respect to providing high classification performances.

Another subject of investigation is the class of Haar features that best discriminate vehicles. In Figure 8 is shown a benchmark between a classifier trained using classical 5 Haar features and one trained using all 15 features shown in Figure 5. This extended version has been choosen since, with the 15 features set, slightly better results are obtained.

Three different AdaBoost designs are involved in testing: Gentle, Rear and Discrete AdaBoost [19]. The obtained results are shown in Figure 9: the PRC curves demonstrate that all AdaBoost variants provide quite similar results so Discrete-AdaBoost is selected for simplicity.

Finally, only for comparison purpose, a benchmark between Haar-AdaBoost and HOG-LinearSVM has been made and shown in Figure 10. On this benchmark, the nonlinear decision stump classifier used in AdaBoost provides better results compared to the Linear SVM algorithm. The discussion will be definitely different in the case of nonlinear SVM.

The positive training set is composed by front and rear images of cars. For the recognition step a single classifier, trained either with frontal and rear vehicle views, is designed to optimize the computational overhead: with the acquired data it has been experimentally demonstrated that to obtain the correct and complete recognition of the training set defined, two distinct classifiers need 400 features (106 for the frontal detection and 264 for the rear one) while, with the use of a single classifier, the 100% of the training set correct recognition is reached in less time, with 356 features. To provide a compromise between the accuracy of the multiscale detection scheme and the computational time required, the final designed classifier has been trained with 350 features.

The algorithm performance has been tested in the acquired images, evaluating with the defined validation set either the Discrete AdaBoost scheme and its Soft-Cascade version. By comparing the correct detection rates obtained through the application of the two classifiers, it has been demonstrated that, from a detection performance point of view, they provide equivalent results: measuring the classification capabilities for the validation set designed, either the traditional AdaBoost and its Soft-Cascade version provides the same correct detection rate on the validation set.

Experimentally, it has been also demonstrated that the Soft-Cascade design allows to reduce 5 times the computational cost: comparing, runtime, a Discrete AdaBoost classifier trained for 350 stages and its SoftCascade version with a rejection rate of 0.04, the traditional AdaBoost classifier needs all the stages (350) to correctly discard a negative sample, while with the SoftCascade scheme the same negative pattern is correctly classified in average after only 30 evaluations.

The computational saving introduced by the Soft-Cascade scheme is further demonstrated measuring the individual times required by the two classifiers to perform vehicle detection in the 1280×800 input images. The timing results, obtained varying the number of scales for octaves defined according to the multiscale detection scheme, are shown in table II: by increasing the number of the scales, the image bounding box occupied by a vehicle is reconstructed more precisely, at the expense of a greater computational cost.

Number of scales	AdaBoost	Soft-Cascade
1	53 ms	14 ms
2	98 ms	26 ms
4	184 ms	52 ms

TABLE II
COMPUTATIONAL ANALYSIS.

To ensure a compromise between the results reliability and the processing time requested, the multiscale detection scheme adopted is a Soft-Cascade with 4 scales for octave: the union of this classification design and the dimension defined for training patterns (32×32) provides vehicle detection in a range between 2 and 25 meters.

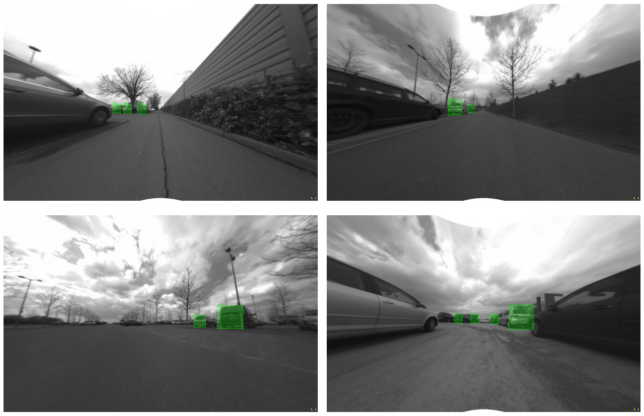
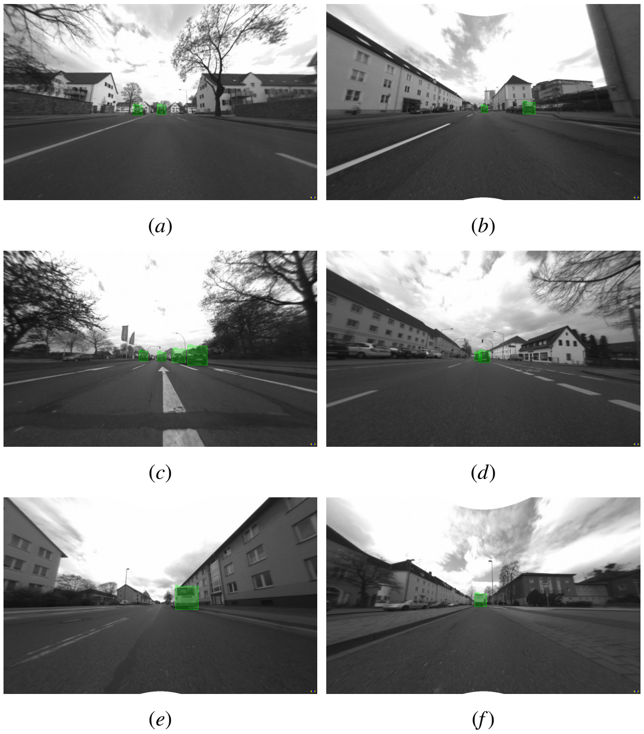


Fig. 11. Vehicle detection in parking lots.



(*a*) (*b*) (*c*) (*d*) (*e*) (*f*)

Fig. 12. (*a*) Frontal and rear vehicles detection at different distances. (*b*) Static and dynamic vehicles detection at different distances. (*c*) Multiple vehicle detection, with a partially occluded car. (*d*) Detection with a strongly occluded vehicle.

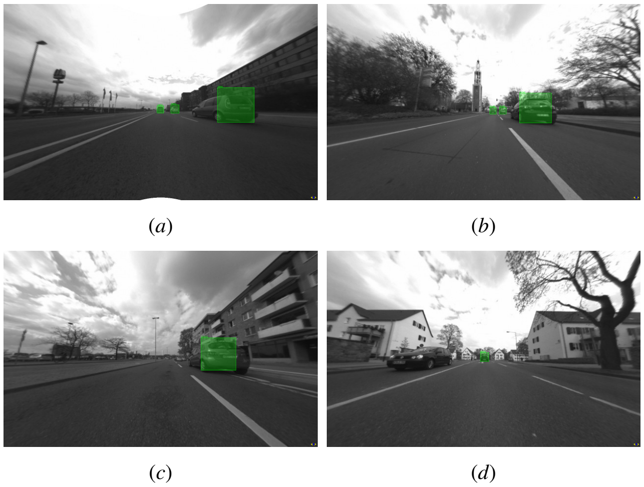
Concerning the qualitative evaluation of the detection results, despite the typical working scenario for the V-Charge vehicle, is a parking lot (Figure 11), the developed system demonstrates promising results also in urban and extra-urban environments achieving static and dynamic vehicles detection (Figure 12 *a-b*) and coping also with challenging situation as vehicle occlusions (Figure 12 *c-d*). The general purpose design, based on multiscale processing, allows to detect different types of vehicles as trucks and buses (Figure 12 *e-f*).

IV. CONCLUSIONS

An algorithm has been presented for frontal and rear vehicles detection in monocular fish-eye images. It is based on the execution of a AdaBoost classifier with a Soft-Cascade scheme on Haar features. The multiscale detection system allows to recognize vehicles at a distance between 2 and 25 meters, providing an high detection rate also before typical improvements introduced by a tracking technique. Computational analysis, performed on IntelCore i7@3.40GHz, demonstrates that the time requested to process 1280×800 images is less than 52 ms per frames.

V. FUTURE WORKS

The main limit of the proposed approach concerns the method used for the perspective images reconstruction: the fish-eye images reprojection onto a single plane involves the generation of noisy contributions for specific parts of the



(*a*)

(*b*)

(*c*)

(*d*)

Fig. 13. Vehicle detection in noisy region of the perspective images. The noisy contributions in the side part of the images involve a random classifier behavior. The bounding box may appears bigger than the area occupied by a vehicle and bad centered on it (*a*)-(*b*), it may cover partially the area (*c*) or it could not be detected (*d*).

images (see Figure 13). While the optimal solution appears to be the cylindrical unwarping approach, the first attempt was the adoption of a multi-view approach, as explained in Section I. This method is effective in reducing the distortion deformations, but, on the other hand, it increase the computational power required (more images to be processed) and needs some non trivial cross-views obstacle fusion.

Actually the challenge of cross-image tracking and fusion exist in general, regardless the unwarping method selected, since the four fish-eye cameras are overlapping. In particular, some of the V-Charge scenarios include also overtaking and oncoming vehicles, where obstacle tracking across cameras will be essential: it is clear from Figure 1 how only mono detection is able to perform such a task, given the actual sensors suite. In a further paper it will discussed how cross-images (as well as cross-views) tracking has been implemented, both in 2D (image) and 3D (world coordinates).

ACKNOWLEDGMENT

The work described in this paper has been developed in the framework of the V-CHARGE Collaborative Project (FP7-269916), an European research activity funded by the European Commission to develop and demonstrate autonomous valet parking for electric vehicles. The authors gratefully acknowledge V-Charge partners for the support to this research.

REFERENCES

[1] Alberto Broggi, Andrea Cappalunga, Claudio Caraffi, Stefano Cattani, Stefano Ghidoni, Paolo Grisleri, Pier Paolo Porta, Matteo Posterli, Paolo Zani, and John Beck, "The Passive Sensing Suite of the TerraMax Autonomous Vehicle," in *Procs. IEEE Intelligent Vehicles Symposium 2008*, Eindhoven, Netherlands, June 2008, pp. 769–774.

[2] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Muhlfellner, S. Wonneberger, J. Timpner, S. Rottmann, Bo Li, B. Schmidt, T.N. Nguyen, E. Cardarelli, S. Cattani, S. Bruning, S. Horstmann, M. Stellmacher, H. Mielenz, K. Koser, M. Beermann, C. Hane, L. Heng, Gim Hee Lee, F. Fraundorfer, R. Iser, R. Triebel, I. Posner, P. Newman, L. Wolf, M. Pollefeys, S. Brosig, J. Effertz, C. Pradalier, and R. Siegwart, "Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 809–816.

[3] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, London, UK, UK, 1995, EuroCOLT '95, pp. 23–37, Springer-Verlag.

[4] Lubomir Bourdev and Jonathan Brandt, "Robust object detection via soft cascade," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, Washington, DC, USA, 2005, CVPR '05, pp. 236–243, IEEE Computer Society.

[5] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Intl. Conf. on Computer Vision & Pattern Recognition*, Dec. 2001, vol. 1, pp. 511–518.

[6] A. Haselhoff and A. Kummert, "A vehicle detection system based on haar and triangle features," in *Intelligent Vehicles Symposium, 2009 IEEE*, 2009, pp. 261–266.

[7] B. Alefs, "Embedded vehicle detection by boosting," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, Sept 2006, pp. 536–541.

[8] S. Sivaraman and M.M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 267–276, 2010.

[9] A. Chavez-Aragon, R. Laganiere, and P. Payeur, "Vision-based detection and labelling of multiple vehicle parts," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 1273–1278.

[10] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[11] Zehang 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, 2006.

[12] S. Sivaraman and M.M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1773–1795, 2013.

[13] Lionel Heng, Bo Li, and Marc Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1793–1800.

[14] Zsolt Kira, Raia Hadsell, Garbis Salgian, and Supun Samarasekera, "Long-range pedestrian detection using stereo and a cascade of convolutional network classifiers," in *IROS*, 2012, pp. 2396–2403.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, June 2005, vol. 1, pp. 886–893 vol. 1.

[16] D.F. Llorca, R. Arroyo, and M.A. Sotelo, "Vehicle logo recognition in traffic images using hog features and svm," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 2229–2234.

[17] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc J. Van Gool, "Pedestrian detection at 100 frames per second.," in *CVPR*. 2012, pp. 2903–2910, IEEE.

[18] Christoph Gustav Keller, Markus Enzweiler, and Dariu M. Gavrila, "A new benchmark for stereo-based pedestrian detection.," in *Intelligent Vehicles Symposium*. 2011, pp. 691–696, IEEE.

[19] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, pp. 2000, 1998.