2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

360° detection and tracking algorithm of both pedestrian and vehicle using fisheye images

Massimo Bertozzi, Luca Castangia, Stefano Cattani, Antonio Prioletti, Pietro Versari

Abstract— **All-around view is a mandatory element for autonomous vehicles. The European V-Charge project seeks to develop an autonomous vehicle using only low-cost sensors. This paper presents a detection and tracking algorithm that covers all the area around the vehicle using 4 fisheye cameras only. The algorithm is able to detect pedestrians and vehicles and track them, using cylindrical images. This paper presents the whole pipeline, from the image un-warping to the classification and the tracking algorithms, together with some results.**

I. INTRODUCTION

The EU-funded project Autonomous Valet Parking and Charging (V-Charge) aims to develop an autonomous electric vehicle that can move in parking lots or garages to park and reach re-charging spots. The peculiarity of this project is the use of a low-cost sensor setup, that is close to the one that it is possible to find in a consumer car. This guideline leads to the choice of the sensors suite, that is shown in Fig. 1: just two stereo cameras for long range narrow angle obstacles detection, plus 4 monocular fisheye cameras, aimed to detect moving obstacles all-round the vehicle and ultrasonic bumper sensors.

More details on the project aims can be found in [1]. Cameras are the main sensor of the V-Charge car platform: the presented algorithms employ four fisheye cameras (depicted in blue in Fig. 1) to cover the whole area close to the vehicle.

This paper presents a system to classify and track vehicles and pedestrians all around the autonomous vehicle. As this system is supposed to be used driving at low speeds in parking lots or garages in order to help in different maneuvers (moving forward and backward and making tight curves), the region of interest that must be considered covers the whole area around the vehicle, but it is limited to short distances. Stereo systems, which are not presented in this paper, are employed to detect front and back obstacles only.

The proposed algorithm is a development of the work presented in [2]. As an extension of the previous work, cylindrical images are investigated for classification and multi-camera tracking is developed.

A. Related works

In driving assistance systems, obstacle detection is a crucial component of collision avoidance, especially for dynamic object detection. Many sensors can be used for obstacle detection, such as LIDAR, RADAR and vision sensors. Vision sensors can be divided into pinhole cameras, with limited field of view, and fisheye cameras [3]. Fisheye cameras can be used for several applications, especially in backup aid and surround view systems [4], [5], [6] because these cameras can frame a wide area close to the vehicle. Feature based object detection algorithms are often used to detect specific kinds of objects such as pedestrians and vehicles [7]. Currently, the major interest relating to the objects classification in the automotive environment is dedicated to pedestrians. Luckily, all the technology developed for this category can be applied to simpler classes of objects like vehicles. The most exploited features in this field are the Haar Features [8], the Local Binary Pattern (LBP) [9], the Histogram of Oriented Gradient (HOG) [10], [11], the Integral Channel Features (ICF) and, recently, Aggregated Channel Feature (ACF) [12], [13], [14]. Features extracted in the image window are processed with an ensemble learning algorithm. Algorithms largely used to detect objects in automotive environments are based on SVM (linear, non-linear, latent) [10], AdaBoost (and variants) [11] or Random Forest [15]. In the state of the art, both Soft-Cascade techniques and search window pruning (using the hypothesis of flat-terrain) are widespread used [2], [10], [11], [14], to achieve real-time performance. The object detection in fisheye images can be done classifying directly in fisheye images [16], where different classifier are trained on different portion of fisheye image, or applying a classifiers to undistorted image [17].

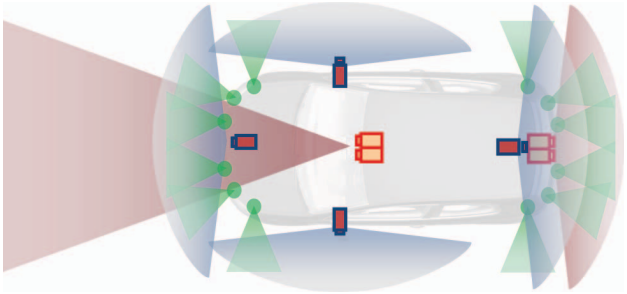


Fig. 1: V-Charge sensors layout: two stereo cameras (in red), on the front and on the back, for long range narrow angle detection; 4 fisheye monocular cameras (in blue) for all-round view obstacle detection; short range collision avoidance sonars (in green).

II. PROPOSED SOLUTION

The scheme in Fig. 2 represents the main steps of the proposed approach:

All the authors are with Department of Information Engineering, Università degli Studi di Parma, Parma, Italy {bertozzi, castangia, cattani, prioletti, versari} @ce.unipr.it

978-1-4673-7266-4/15/$31.00 ©2015 IEEE

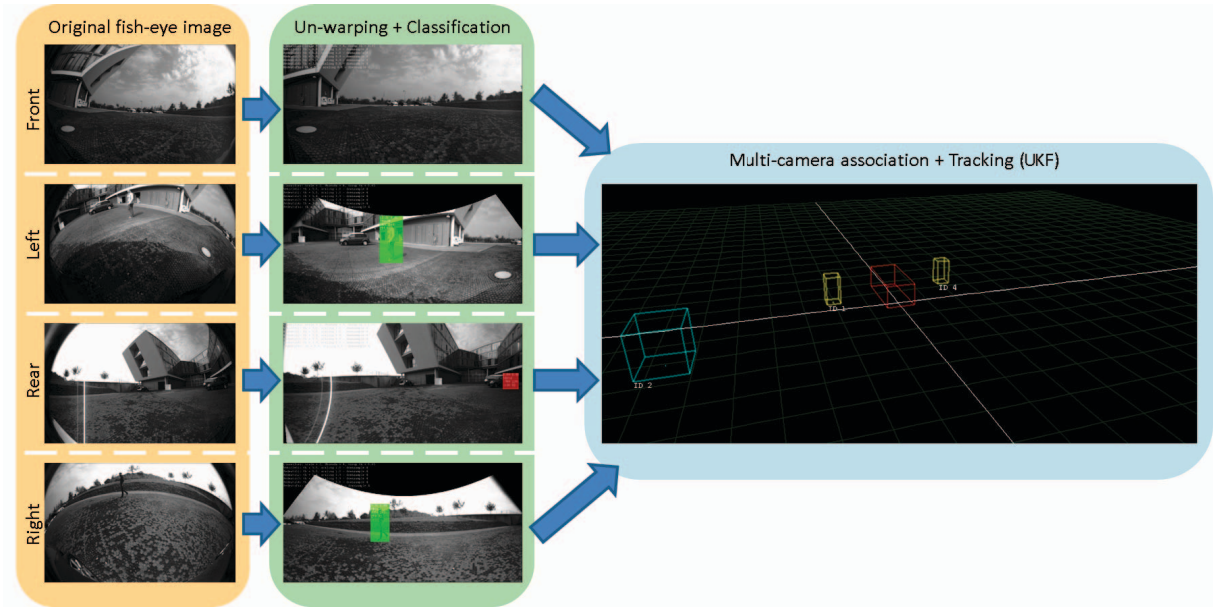


Fig. 2: Overall system design: from the left to the right it is shown the pipeline of the proposed approach. Firstly, each fisheye image is un-warped; the chosen model allows to have overlapping field of views between adjacent cameras. Secondly, the Soft-Cascade+ACF classifier is used to detect both vehicles (in red) and pedestrians (in green). The outputs from all views are associated each other and finally the Unscented Kalman Filter track all obstacles, allowing to follow an object moving around the vehicle through different field of views.

- Image un-warping:
  the fisheye images are un-warped in order to both correct the lens distortion and obtain a wide-angle view without strong aberrations.
- Classification:
  a Soft-Cascade+ACF classifier is run on each camera image to detect vehicles and pedestrians on all the surrounding space.
- Multi-camera tracking:
  in order to perform the 360° tracking, firstly an inter-camera association algorithm is performed then, an Unscented Kalman Filter is used to exploit those information to track and generate an all around description, as an unique high-level sensor with 360° field of view.

All of these steps are described in detail in the following paragraphs.

A. *Image un-warping*

Fisheye lenses provide very large wide-angle views (theoretically the entire frontal hemispheric view of 180°), but the produced images suffer from severe distortion since the hemispherical scene gets projected onto a flat surface. A procedure to correct the fisheye lens distortion is necessary unless you use an algorithm that takes into account this effect. A common approach involves the distortion correction by re-projecting the image on a virtual plane in order to obtain a pinhole image but, the wider the resulting pinhole field of view, the stronger aberration you get on image edges.

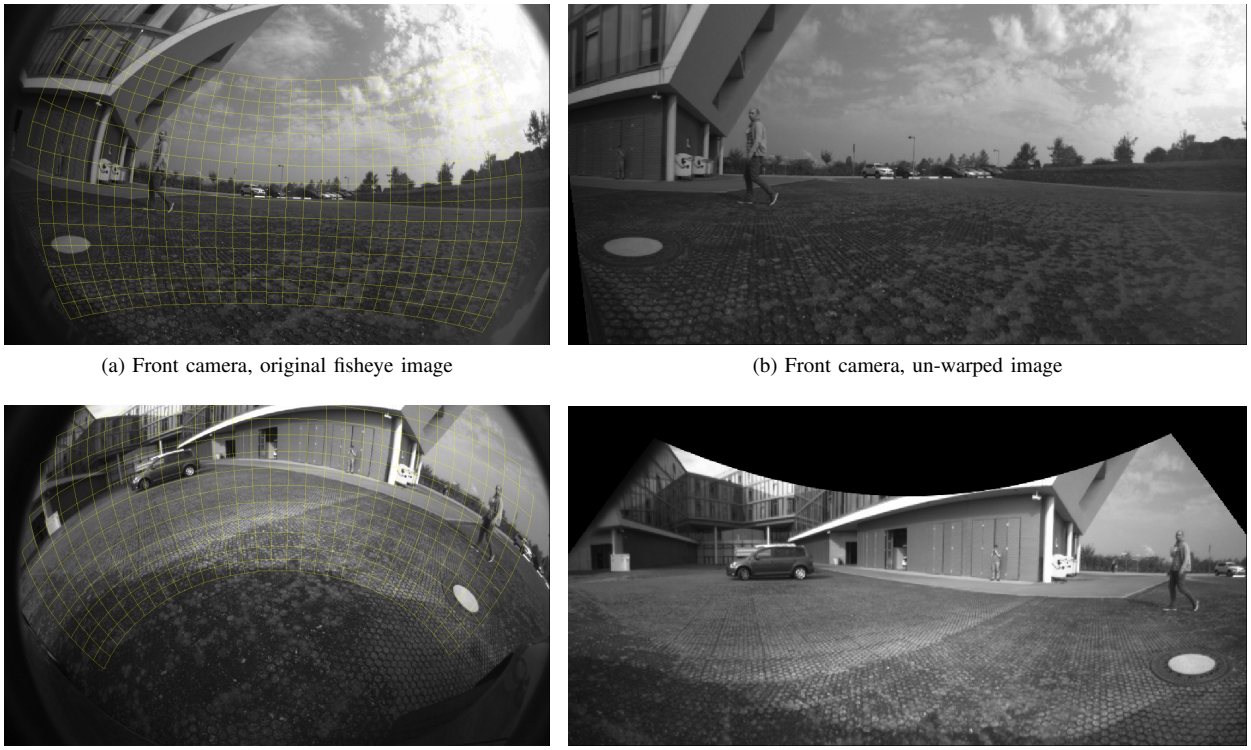
In [2] a virtual views layout has been presented in order to correct the lens distortion and exploit the large fisheye lenses field of view. However that approach has some drawbacks: to find all obstacles virtual views need to overlap; this means more pixels to process and conflicted areas where it is hard to determine what is the right detection. To overcome these shortcomings, it has been decided to increase the number of virtual views and merge them together in a single wide image: the final result is equivalent to re-project the original fisheye image onto a semicylindrical surface, as it is done in [18].

Fisheye cameras manufactured to follow the equidistance mapping function, are designed such that the distance between a projected point and the image optical center is proportional to the projected ray incident angle, scaled only by the equidistance parameter. Exploiting this relationship, the resulted cylindrical model is conveniently described by the following equations:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c_u + \operatorname{atan2}(x, z) \cdot f_\theta \\ c_v + k_v \cdot \frac{y}{\sqrt{x^2+z^2}} \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin \frac{u-c_u}{f_\theta} \\ \frac{v-c_v}{k_v} \\ \cos \frac{u-c_u}{f_\theta} \end{bmatrix} \tag{2}$$

where (c_u, c_v) is the un-warped image virtual optical center, k_v is the vertical focal length, f_θ is the equidistance parameter, (x, y, z) is the 3D point being projected to the image and (u, v) is the related point in image domain. The Equation 1 is the projection of a (x, y, z) 3D point from camera reference frame to cylindrical image reference



(a) Front camera, original fisheye image

(b) Front camera, un-warped image

(c) Left camera, original fisheye image

(d) Left camera, un-warped image

Fig. 3: Examples of fisheye images and the related un-warped images; into the fisheye images is depicted the corresponding semicylindrical surface. The extrinsic parameters have been also used in the un-warping phase and this is considerable especially on side cameras. Note also the overlapping field of view between the two un-warped images.

frame, while the Equation 2 is the relation from a (u, v) 2D point from cylindrical image reference frame onto a semicylindrical surface in the camera reference frame. To preserve the original aspect ratio it is convenient to impose $k_v \equiv f_\theta$.

Exploiting the extrinsic, intrinsic and distortion cameras parameters [19] a proper semicylindrical surface has been defined for each fisheye camera; to facilitate the subsequent algorithms, such the classification phase, those semicylinders have been placed just in front of each camera, keeping their axis perpendicular to the vehicle reference z plane. The Fig. 3 shows some fisheye images and the resulted un-warped images.

These choices imply that each image row correspond to a circumference arc in vehicle reference z plane; this means that objects on the same row in the same image are at the same distance from the related camera: this property is useful for many optimizations in the classification phase. Noteworthy is also that the whole un-warped image does not respect the pinhole camera model but, locally, the difference between a virtual pinhole camera and the un-warped image is hardly visible; a detailed analysis is covered in section III-A.

B. Classification

Pedestrians and vehicles detection is based on a Soft-Cascade+ACF classifier. The first phase, regarding the computation of the integral channel, is shared between pedestrians and vehicles detector in order to speed-up the system.

Several classifiers with different pattern sizes have been trained reducing the processing time required in the scaled images computation. Five classifiers are used for pedestrians, with dimensions 32×64, 48×96, 58×116, 68×136 and 81×162; two, instead, for vehicles, with dimension 38×38 and 45×45. Two scales for each classifier are computed. The pedestrian classifier with pattern size 32×64 is used to detect only far pedestrian: differently from the other classifier, it is run just at the smallest octave for each scale.

Moreover, the detection is not performed in regions that do not respect the typical range of vehicle/pedestrian size: for each candidate window, through the world-to-image coordinates transformation, its theoretical area is calculated and is checked if it respects the suitable dimensions for a vehicle or pedestrian. Candidates at the same scale-octave are filtered with a 3×3 non maxima-suppression.

The final training set is composed by 48000 cars images from a private dataset, 4000 pedestrians images from KITTI+INRIA datasets and 5000 negatives examples, randomly extracted from the initial negatives images. Positive samples are enlarged by 8% to include also the information contained in the border. 5 bootstrapping cycle are performed to improve the detection performance and reduce the false positive rate. At each cycle, the detector is trained on an

augmented set composed by the initial negative samples and the hard negative ones obtained by the previous cycle.

C. Multicamera tracking

Tracking is performed jointly through the four triggered cameras allowing to follow an object moving around the vehicle.

1) Association: The "all-around tracking" requires that the association must be performed not only between objects of the same class in subsequent frames, but also between objects of the same class but coming from different cameras. The matching in subsequent frames is performed using the bounding boxes overlap. Several comparison metrics, instead, have been analyzed for the multi-cameras one:

- Euclidean distance: objects coming from all the cameras are projected in the world coordinates space and the euclidean distance represents the matching cost.
- Polar distance: objects are still projected in the world coordinates but, in this case, the polar distance represents the matching cost.
- Image distance: objects on each camera are projected on other cameras (if the projection is possible) and the matching cost is represented by the bounding boxes overlap.

The "Image distance" metric has been selected for the multi-camera matching process because achieves better results and its metric is coherent with the subsequent tracking algorithm.

An association based on features matching between different cameras has been also tested but discarded due to the high computational cost for the features extraction and matching.

Once defined a comparison metric, the Hungarian algorithm is used to find the best association and maximize the matching cost. After this step, each tracked object may have been associated with one or two elements, depending if it has been seen in one or more cameras.

2) Tracking: An Unscented Kalman Filter (UKF) is used for the tracking process; the filter state is composed by the tracking point, the object width and its relative speed. The predicted position is obtained combining the host vehicle movement information (from the inertial sensor) and the object position and speed. While the filter state is in the vehicle reference frame, the filter observations are in the images reference frame.

The bounding box extraction process requires the conversion of the top-left and bottom-right points of the object in the world coordinates and, then, the projection of these points in the image coordinates. In case of an object in the front camera, these points can be calculated subtracting and adding on the x axis the width value to the tracking point. This procedure, instead, would not be valid for objects in different cameras. Then, a different method has been used for the bounding box extraction: firstly, the tracking point and the top center point of the object are projected in the image coordinates; from these two points it is possible to extract the bounding box height in pixels and use it to find the width and, then, the top-left and bottom-right points in image coordinates starting from the projected tracking point.

As described in the II-C.1 section, one or two elements can be associated to a tracked object after the matching process. In case of a "single" association, the observation is represented by the tracking point and object width in image coordinates: through the Equations 1, 2 is possible to switch between image and world coordinates. In case of a "double" association, instead, the observation is represented by a couple of tracking points and widths in pixels from the objects in the two different cameras. In this case, the predicted observation are the projections of the object state in the two image coordinates.

Extra care is required to manage the two different obstacles types, pedestrians and vehicles. This involves different filter parameters, regarding the object movement and, the different extraction of the object bounding box from its state: while vehicles have the same ratio between height and width, the pedestrians height is twice their width.

III. RESULTS

A. Cylindrical images

The basic assumption of our approach was the classificator compatibility with cylindrical images; since no object detection evaluation dataset with fisheye images was available, we chose to evaluate "differences" between un-warping fisheye images using pinhole model and cylindrical model. For this purpose we defined "difference" d as follow:

$$d(u,v) = \left\| (u,v) - (u',v') \right\| \tag{3}$$

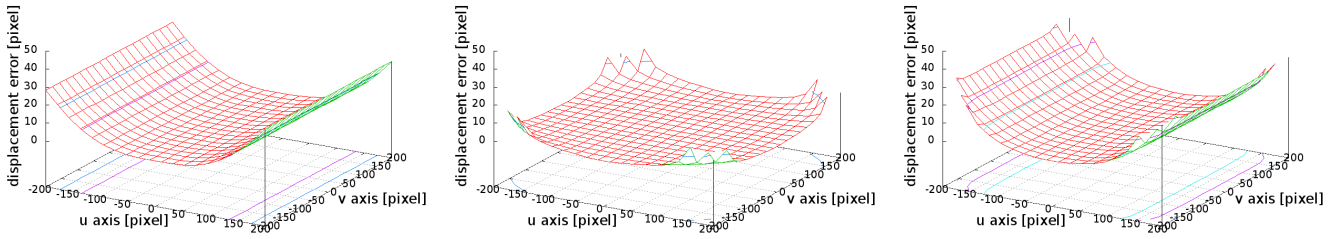
where (u,v) is a point in image domain and (u',v') is the point obtained by re-projecting (u,v) from pinhole model to cylindrical model.

This analysis helps to understand the expected object distortion given its size in the equivalent pinhole model, and consequently it is possible to estimate the maximum acceptable size. In Fig. 4 the results[1] of this analysis are shown: firstly we evaluated the per-axis re-projection displacement errors, then we computed Euclidean norm as described by Equation 3.

Predictably, the main displacement error occurs along u axis: points further than 150 pixels have displacement error u component of approximately 20 pixels. With a similar analysis of "Euclidean norm of displacement errors" graph (Fig. 4 c) we can overestimate the error of objects as wide as 200 pixels in less than 10 pixels, and less than 20 pixels for object as wide as 300 pixels; along the v axis we have less constraints.

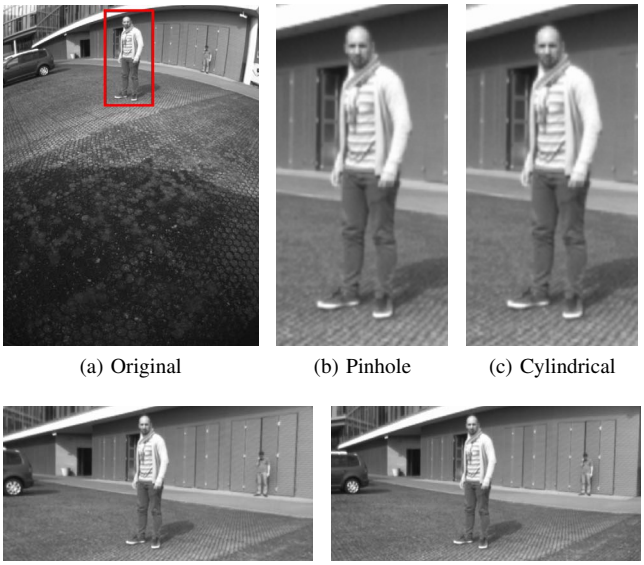
It is important to notice that the further the points are from the pinhole image center, the stronger aberration you get: then on the one hand the "difference" as defined by Equation 3 is increasing, but on the other hand the pinhole image we are comparing to begins to be affected by aberration. In Fig. 5 is shown a qualitative comparison between the original fisheye image and pinhole model and cylindrical model.

[1]These results are computed using the real intrinsic camera parameter utilized in our system: $f_\theta = k_v \simeq 266.67$ [pixel].



(a) Displacement errors along u axis (b) Displacement errors along v axis (c) Euclidean norm of displacement errors

Fig. 4: Displacement errors between pinhole model and cylindrical model. In (a) and (b) are shown the displacement errors along u and v axes, in (c) is depicted the Euclidean norm of displacement errors as described by Equation 3. The greater the error is, the higher the displacement between pinhole model and cylindrical model is at that coordinate.



(a) Original (b) Pinhole (c) Cylindrical

(d) Pinhole (e) Cylindrical

Fig. 5: Comparison between un-warping the original image (a) by re-project it on a plane (b) or on a semicylindrical surface (c); those images are focused only on a small area in order to highlight the small differences between the two approaches. In (d) and (e) the differences are more evident but still hard to see: with the same number of pixels in (d) there are some aberrations on sides, meanwhile in (e) the field of view is slightly larger.

B. Classification

The classifier, used in the previous version of the system, has been improved on several aspects:

- The pre-processing parts of vehicle and pedestrian classification are merged in a common stage.
- The scales number has been reduced using more classifiers with different pattern size.
- The shrinking factor has been introduced.

These changes lead to reduce the computational time and to achieve real-time performance processing the images coming from all the four cameras.

Our classificator is competitive in terms of the detection quality with respect to other state-of-the-art methods. The pedestrian classifier miss rate on Daimler Pedestrian dataset is 0.35 at 10^{-1} FPPI, 0.59 at 10^{-2} FPPI and 0.75 at 10^{-3}; the vehicle classifier miss rate on our private dataset is 0.11 at 10^{-1} FPPI, 0.27 at 10^{-2} FPPI and 0.50 at 10^{-3}.

C. 360° tracking

The inter-camera tracking algorithm considerably improves the system performance with respect to perform it individually on each camera. Overlapping fields of view guarantee a higher probability to detect an object than in one camera, both in terms of different points of view from which the object is observed then in terms of possible occlusions in a camera but not in the other one. Moreover, it is possible to exploit the past history of an object passing through the cameras, increasing the tracking accuracy.

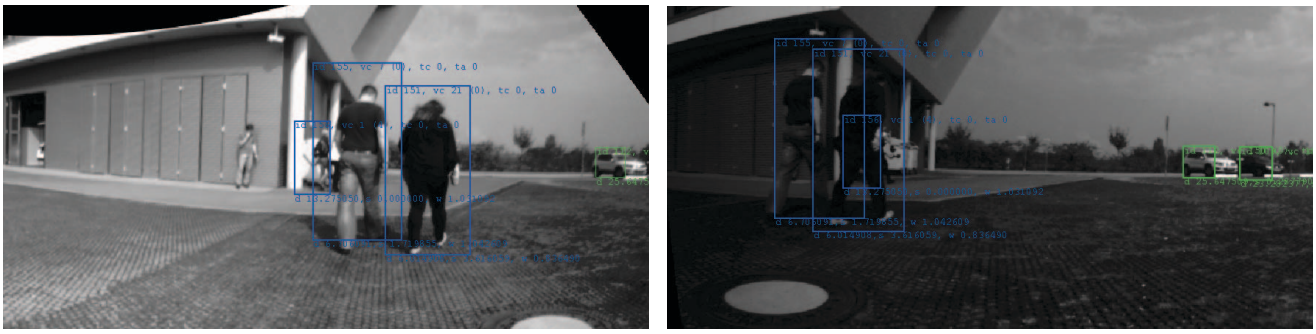
In Fig. 6 is shown an example where we exploited the two points of view to overcame one occlusion and to improve the detection accuracy.

D. Conclusions and future works

In this paper, a method to detect and track pedestrians and vehicles all around the V-Charge prototype is presented. The sensor suite is limited to four fisheye cameras, installed on the front, back, right and left parts of the vehicle. Main contributions of this method are the classification using cylindrical images and the 360° tracking. The obtained results are remarkable in parking slots and garage environments. A great attention has been given to optimize the computational performance[2], to reach a processing time that allows the managing of different manoeuvres at low speeds, processing four cameras images. Even if the system provides promising results in most situations, when the obstacles are as close to the vehicle to be only partially framed, the classification can provide very noisy results, especially in distance estimation.

The tracking stage still needs further developments: vehicle orientation should be detected in order to improve the

[2]The whole pipeline represented in Fig. 2 works at $12.5Hz$, running together with other systems, on a Intel® Core™ i7-2600 (4 cores, 3.4 GHz) computer.



(a) Left cylindrical image

(b) Front cylindrical image

Fig. 6: Example of 360° tracking approach benefits: the multiple point of views are exploited to overcome occlusion (the far pedestrian is visible only in (a)) and to increase the detection accuracy (the two walking pedestrians were detected on both images).

association by different points of view; moreover, tracking of partially visible obstacles should be included. A huge test stage is also mandatory to validate the method in the specific environment considered for the V-Charge project.

IV. ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013, Challenge 2, Cognitive Systems, Interaction, Robotics, under grant agreement No 269916, V-Charge.

REFERENCES

[1] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Mühlfellner, S. Wonneberger, J. T. S. Rottmann, B. Li, B. Schmidt, T. N. Nguyen, E. Cardarelli, S. Cattani, S. Brüning, S. Horstmann, M. Stellmacher, H. Mielenz, K. Köser, M. Beermann, C. Häne, L. Heng, G. H. 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 *IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, 23–26 June 2013, pp. 809–816.

[2] A. Broggi, E. Cardarelli, S. Cattani, P. Medici, and M. Sabbatelli, "Vehicle detection for autonomous parking using a Soft-Cascade AdaBoost classifier," in *Procs. IEEE Intelligent Vehicles Symposium 2014*, Dearbon, MI, USA, June 2014, pp. 912–917.

[3] B. Zhang, V. Appia, I. Pekkucuksen, A. Batur, P. Shastry, S. Liu, S. Sivasankaran, K. Chitnis, and Y. Liu, "A surround view camera solution for embedded systems," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, June 2014, pp. 676–681.

[4] F. Nielsen, "Surround video: a multihead camera approach." *The Visual Computer*, vol. 21, no. 1-2, pp. 92–103, 2005. [Online]. Available: http://dblp.uni-trier.de/db/journals/vc/vc21.html#Nielsen05

[5] M. Gressmann, G. Palm, and O. Lohlein, "Surround view pedestrian detection using heterogeneous classifier cascades," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1317–1324.

[6] C.-C. Lin and M.-S. Wang, "A vision based top-view transformation model for a vehicle parking assistant," *Sensors*, vol. 12, no. 4, pp. 4431–4446, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/4/4431

[7] L. Castangia, P. Grisleri, P. Medici, A. Prioletti, and A. Signifredi, "A coarse-to-fine vehicle detector running in real-time," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, Oct 2014, pp. 691–696.

[8] J. Choi, "Realtime on-road vehicle detection with optical flows and haar-like feature detectors," University of Illinois at Urbana-Champaign," Computer Science Research and Tech Reports, 2012.

[9] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 975–982.

[10] M. Gabb, O. Löhlein, R. Wagner, A. Westenberger, M. Fritzsche, and K. Dietmayer, "High-performance on-road vehicle detection in monocular images," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 336–341.

[11] J. D. Ortega, M. Nieto, A. Corts, and J. Flrez, "Perspective multi-scale detection of vehicles for real-time forward collision avoidance systems." in *ACIVS*, ser. Lecture Notes in Computer Science, J. Blanc-Talon, A. J. Kasinski, W. Philips, D. C. Popescu, and P. Scheunders, Eds., vol. 8192. Springer, 2013, pp. 645–656.

[12] R. Appel, P. Perona, and S. Belongie, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, no. PrePrints, p. 1, 2014.

[13] R. Benenson, M. Mathias, R. Timofte, and L. J. V. Gool, "Pedestrian detection at 100 frames per second." in *CVPR*. IEEE, 2012, pp. 2903–2910. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#BenensonMTG12

[14] R. Brehar and S. Nedevschi, "Scan window based pedestrian recognition methods improvement by search space and scale reduction," in *Procs. IEEE Intelligent Vehicles Symposium 2014*, Dearborn, MI, USA, June 2014.

[15] J. Marin, D. Vazquez, A. Lopez, J. Amores, and B. Leibe, "Random forests of local experts for pedestrian detection," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 2592–2599.

[16] K.-Y. Byun, B.-S. Kim, H.-K. Kim, J.-E. Shin, and S.-J. Ko, "An effective pedestrian detection method for driver assistance system," in *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, Jan 2012, pp. 229–230.

[17] G. Cheng and X. Chen, "A vehicle detection approach based on multi-features fusion in the fisheye images," in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, vol. 4, March 2011, pp. 1–5.

[18] W. Schulz, M. Enzweiler, and T. Ehlgen, "Pedestrian recognition from a moving catadioptric camera," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, F. Hamprecht, C. Schnörr, and B. Jähne, Eds. Springer Berlin Heidelberg, 2007, vol. 4713, pp. 456–465. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74936-3_46

[19] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.