On-Vehicle Video-Based Parking Lot Recognition with Fisheye Optics

Sebastian Houben[1], Matthias Komar[2], Andree Hohm[2], Stefan Lüke[2],
Marcel Neuhausen[1], and Marc Schlipsing[1]

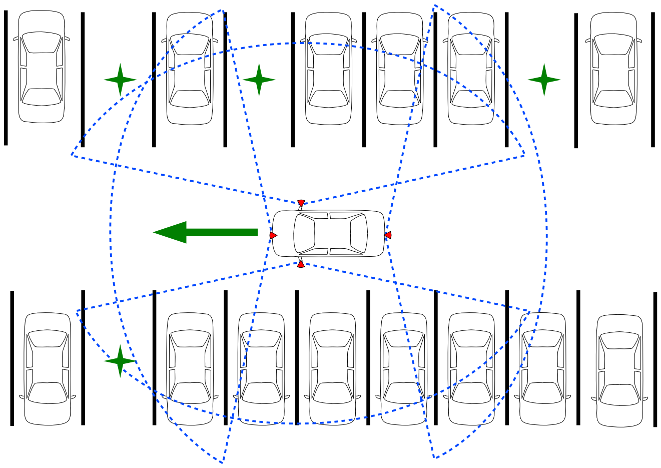


Fig. 1: The task at hand is to recognize parking lot rows while passing by at a moderate speed. Four fisheye cameras were installed to our prototype vehicle (front, rear, left, right) to provide a 360° surround view.

***Abstract*— The search for free parking space in a crowded car park is a time-consuming and tedious task. Today's park assistance systems provide the driver with acoustic or visual feedback when approaching an obstacle or semi-autonomously navigate the vehicle into the parking lot. However, finding a free parking lot is usually left to the driver. In this paper, we address this search problem via video sensors only. This can be used as a help to the driver to quickly pass a parking deck and, more important, can be regarded as a cornerstone to fully autonomously parking vehicles.**

A fisheye camera setup is presented that allows a 360° view around the car. Subsequently, we describe in detail the image processing steps for a) detecting b) locating and c) classifying nearby parking lots, *i.e.*, deciding whether it is free or occupied. We tested our algorithm on several video sequences and examined the detector's accuracy, precision and recall, and the classification performance.

Results show that the system is capable of performing this task with an accuracy of 0.21m for nearby parking lots. This allows to place the vehicle in an initial position for planning and performing the actual parking manoeuvre.

I. INTRODUCTION

Due to the lack of free parking space the search for a place to leave one's vehicle, especially in urban areas, is often time-consuming and tedious. The aim of the automotive industry and traffic management is therefore to automate parking as far as possible.

The task is twofold: Firstly, one has to find a free parking lot and, secondly, the parking manoeuvre must be performed. Modern driver assistance systems provide feedback from distance sensors or display of rear camera images. Also, for certain types of parking lots there are systems that plan a parking manoeuvre trajectory and execute it semi-automatically.

The localization task, however, is handled differently. Most solutions rely on permanently installed systems in the parking deck to manage availability of free parking lots.

Wolff et al. [1] describe a system of magnetic sensors installed at each parking lot that must additionally be equipped with transmission units to report their status. Likewise, radar, infra-red, and sound sensors are used (*cf.* Idris et al. [2] for a broad review). A recent paper by Ibisch et al. [3] describes the use of Lidar sensors to autonomously navigate a vehicle through a park deck.

On the other hand, since computation costs continue to drop, video-based systems become more and more popular as they can be implemented with already installed surveillance cameras (True [4], Ichihashi et al. [5], Seo and Urmson [6], Tschentscher and Neuhausen [7]). The authors report an average detection accuracy of 94% up to 99%.

Nevertheless, the systems are attached to a certain location and have to be connected to a parking management infrastructure. In this paper, we present an on-vehicle system that is able to locate parking lots while driving by at a moderate speed. While processing time is uncritical, motion blur should be avoided for the algorithm to perform optimally. The entire approach is based on video sensors only as we consider these to play a prominent role in future driver assistance hardware.

We describe our setup in section II, explain in detail the image processing steps in section III, analyse the detector's accuracy (Sec. IV), and close with possible extensions of our system in future development (Sec. V).

II. VEHICLE SETUP AND CALIBRATION

For our task we deploy a prototype vehicle with a four-camera setup that we will describe in the following. Fisheye cameras are installed at the front bumper, the rear trunk lid, and at each of the side mirrors. Since the field of view is larger than 180° we gain a complete surround view which strongly simplifies the marking detection between different cameras. The calibration of the system is divided into three

[1] University of Bochum, Institute for Neural Computation
`firstname.lastname at ini.rub.de`

[2] Continental Division Chassis & Safety
`firstname.lastname at continental-corporation.de`

steps: computing the intrinsic calibration parameters of all cameras, estimating each relative ground plane orientation, and connecting the ground plane images to a single bird's eye view.

For the sake of computation time the image transformation is implemented via a lookup-table for every considered ground plane pixel.

The intrinsic calibration,*i.e.*, the mapping of image point to view ray, is accomplished via the method of Scaramuzza et al.[8]. As a radial distortion function we choose a polynomial of degree 4.

Because of the wide-angle lenses in use the ground plane calibration is not straightforward. The well-established DLT algorithm [9] is infeasible. The estimation of the ground plane parameters is performed on rectangular calibration objects (several meters in size) with known dimensions that are marked in the recorded images. We found the following procedure to yield satisfying results: For an initial solution we intersect the view rays of the marked image points with the current candidate plane. We alternately minimize the sum of the scalar products of their connecting vectors (*i.e.*, find a plane where the intersection points form a rectangle) and the size error (*i.e.*, find a plane where the distances of the marked rectangle are correct). An iteration of those two substeps yields an initial ground plane estimation. To refine this solution, we subsequently minimize the back-projection error. That is, we predict each rectangle corner position from the respective three other corners and compare its back-projected image position with the actual one. Since this problem has several local minima the final optimization step relies on a good initial solution.

After this ground plane estimation has been performed for every camera image, the final surround view is created by computing a two-dimensional rigid transformation between the respective ground plane images. This is achieved by manually marking two or more corresponding ground points.

The fields of vision of the installed fisheye cameras overlap. Regarding the task at hand, it is reasonable to prioritize the front and rear camera over the side cameras in the overlapping area, since this allows a better and likely less occluded view into the parking lots that the vehicle passes. To avoid artificial edges at the transition from one camera to another the images are cross-faded. Please refer to Fig. 2 and compare the situation to the scheme in Fig. 1 for an overview.

III. IMAGE PROCESSING

Based on the bird's eye image, the parking lot detection algorithm that is presented here follows the steps below:

- Perform a coarse detection of the parking lot markings via a local symmetry measure
- Estimate the lateral distance from the vehicle to the parking lot row
- Estimate the parking lot row pattern and use it to predict possible distant parking lots
- Classify the occupation status of all parking lots
- Stabilize the results with the help of temporal smoothing

For an overview of the computed values please refer to Fig. 2.

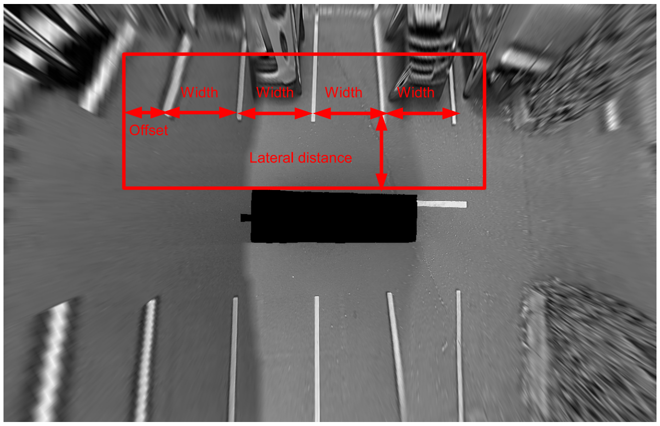


Fig. 2: Result of the calibration: the four camera inputs transformed to a bird's eye view of the vehicle surrounding

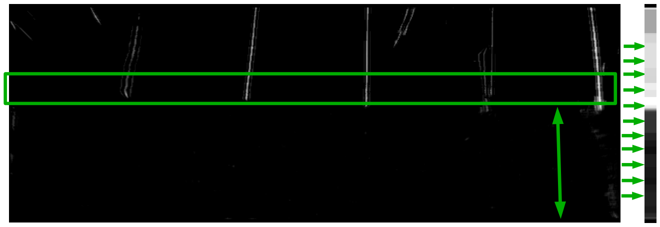


Fig. 3: The symmetry image provides a likelihood for the presence of bright vertically oriented structures of a given size (here: ≈ 10cm). The lateral distance to the parking lot (green arrow) can be determined by means of the function p that transforms each scan-line into a likelihood of a parking lot pattern being present (*cf.* Sec.III-B). The green rectangle shows the area that is subsequently regarded for computing the offset and parking lot width (*cf.* Sec.III-C).

A. *Local symmetry*

We deploy a coarse but fast detector for a lateral position likelihood of the parking lot markings. As these are bright lines of a fixed width w, we can efficiently apply the following symmetry measure for each image line. For this, we compute the gradient in horizontal direction and consider the k gradient pixels with highest magnitude for each image line. All pairs of those pixels are traversed and vote for their respective centers if their distance is near w and their sign differs. The weight of the votes is determined by the corresponding gradient magnitude of the bird's eye image at the very same position. In this way the detector encompasses both the width and the brightness as features of the markings. All vote centers form the symmetry image (*cf.* Fig. 3).

B. *Lateral distance to vehicle*

In the following steps, the aforementioned symmetry image is used to pinpoint the parking lot row. Firstly, the lateral distance of the vehicle to the parking lot markings is estimated. For this purpose, we define a function p that maps the lateral offset to the likelihood of a parking lot row being present at that position.

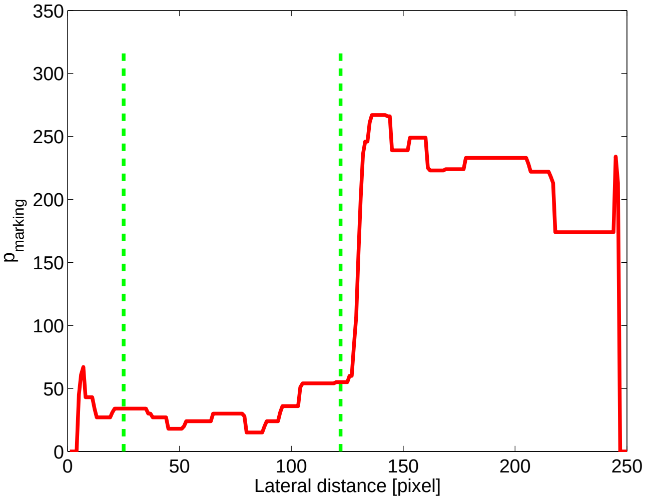


Fig. 4: The function p computed on the image in Fig. 3. The two markings indicate the minimal distance of the parking lots and the position the signal significantly changes, respectively. The first is needed to estimate the noise of the signal, the latter provides the estimated parking lot distance.

Since estimating a regular pattern, as described in section III-C, is too costly we deploy the following heuristic: We expect a parking lot row to create strong symmetry answers and those to be equally distributed over the symmetry image line. Especially the second criterion is important to increase robustness against false signals from other symmetric structures, *e.g.*, irregularities within the pavement. Thus, a sliding average of each line is divided into a number of equidistant bins for which a quantile q of the resulting symmetry values is determined. Plotting those quantiles against the lateral distance yields the desired function p. Figure 4 shows an example computed from the symmetry image in Fig. 3.

The rising edge of p identifies the start of the parking lot row. It is determined by finding the distance at which the function growth exceeds the sliding standard deviation by a given factor. By normalizing the growth by the standard deviation a adaptation to pavement structure as well as changing lighting conditions is introduced. The detailed choice of this parameter is uncritical and can be performed manually since the growing edge usually is very distinct.

C. Estimating the parking lot row

The lateral distance now enables us to examine the distribution of the symmetry votes parallel to the driving direction. Since these are mainly induced by the parking lot markings the resulting profile should resemble a shifted version of the Dirac comb (or Shah function). Thus, it can be parametrized by the period, *i.e.*, parking lot width, and an offset (*cf.* Fig. 5).

In order to facilitate the fit, the average over several scanlines, *e.g.*, the green area in Fig. 3, is inspected and smoothed by a median filter. The resulting signal is then thinned out by considering only those values that are maximum among their direct neighbours. This leaves very little relevant function values.

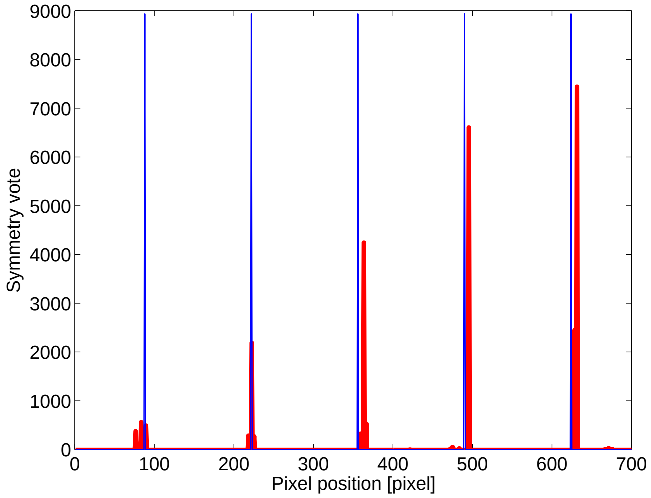


Fig. 5: The symmetry signal and the fitted Dirac comb. The objective function for the fit accumulates the largest symmetry values in a vicinity to each peak. These summands are weighted by a Gaussian of the distance. The period and offset allow to extrapolate the row of parking lots to distances with less distinct markings.

As an objective function we define the sum of the respective largest symmetry values in a vicinity of each Dirac peak. The summands are weighted by a Gaussian of the distance to that largest symmetry value. At this point we do also gain a confidence for the presence of a parking lot row via the quality of the fit. The ratio of the unweighted sum over the total signal energy yields a robust indicator.

With the help of the Dirac comb it is now straightforward to predict the parking lot markings at a further distance. This helps to find lots with a symmetry answer too low to be detected individually.

D. Occupation status

After having pinpointed the parking lot positions, image sections with the estimated width and fixed depth are passed to a classifier described in the following: Since we need to tell free from occupied space, a binary classification problem needs to be solved. Given a large set of labelled training images, we applied *Linear Discriminant Analysis* (LDA) to train a linear classifier. Linear discrimination using LDA gives surprisingly good results in practice despite its simplicity. We apply regularization to ensure proper conditioning of the covariance matrix [10, Sec. 4.3.1].

In order to robustly classify any kind of vehicle as *occupied* we decided not to train an explicit car detector, but to provide the classifier low-level structural features. A *Difference-of-Gaussian* (DoG) filter is applied to each trial image and a histogram over the area of the whole parking lot is generated therefrom. Depending on the filter size the feature vector is rather small (20–50 features) and, thus, supports fast classification from few examples.

The involved feature parameters (filter size, histogram resolution, etc.) as well as the regularization parameter

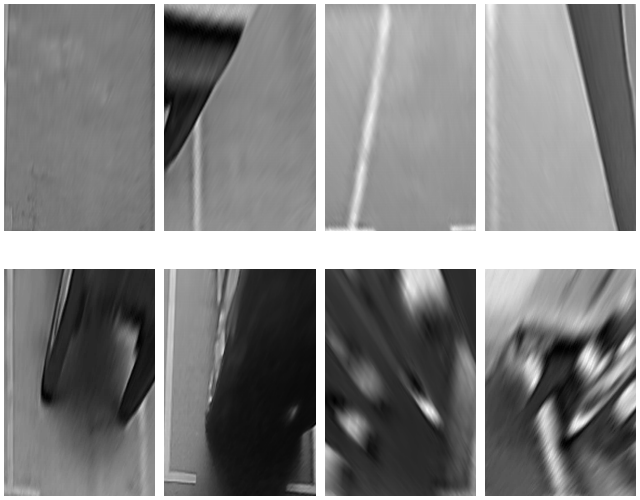


Fig. 6: The upper row show examples of free parking lots, the lower row examples of occupied ones.

were optimized through grid search. Cross-validation on two independent training datasets was performed therefor.

E. Temporal smoothing

All measured values, *i.e.*, the lateral distance, the Dirac comb's parameters, its quality of fitting, and the classification result on each parking lot image, are temporally integrated in order to yield more stable results. This is implemented by means of a Kalman filter, containing a constant-velocity model, for the Dirac comb's shift and an exponential smoothing of all other variables.

F. Implementation and real-time capability

The image processing pipeline was implemented in C/C++ and tested on a Intel Core i7 CPU (2 GHz). Although the algorithm is eligible for parallelization the computation was performed on a single core. Performance measurements yielded that the whole system worked at 15 fps, the computation of the bird's eye image (without further processing) took 15 ms per frame.

IV. EXPERIMENTS

In order to evaluate our approach we labelled 3 sequences with 150 to 495 frames. The scenarios comprise manoeuvres in roofed parking decks and open-air under stable weather conditions. The annotations were carried out manually by marking all parking lots in the bird's eye image and indicating the occupation status. For practical reasons the annotations were only performed every 5th frame. Hence, in total 2073 parking lots were labelled.

The accuracy of the parking lot detection was examined by five means:

- the error in correctly predicting the presence of a parking lot row,
- the errors in correctly detecting parking lots (false positives / false negatives)
- the deviation in the lateral distance estimation

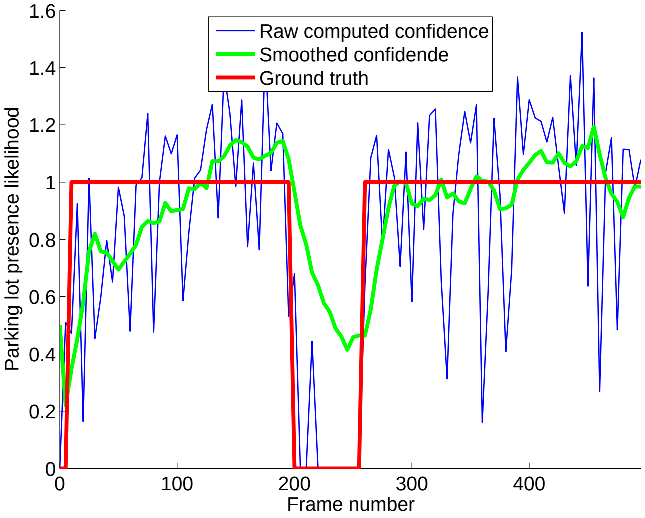


Fig. 7: This plot shows the presence of parking lot rows in one annotated sequence over time. The raw and smoothed fitting confidence donates a measure that reliably follows the ground truth.

- the deviation in parking lot width estimation,
- and the classification error for the occupation status.

In the following sections we inspect these errors on the complete image processing pipeline. This also includes the temporal smoothing of the single-frame results by adequate filter techniques. Since we expect the performance to decrease with greater distance to the vehicle, the deviation from ground truth will as well be presented with respect to their relative position.

A. Presence of parking lot row

As presented in Sec. III-B the vehicle's distance to a parking lot row is estimated by detection of a rising edge in the lateral overall symmetry in the bird's eye image. A temporal average of the goodness of fit value (*cf.* Sec. III-C) is a useful criterion for the presence of a parking lot row. If the lateral distance cannot be determined the goodness of fit is set to 0 for that frame.

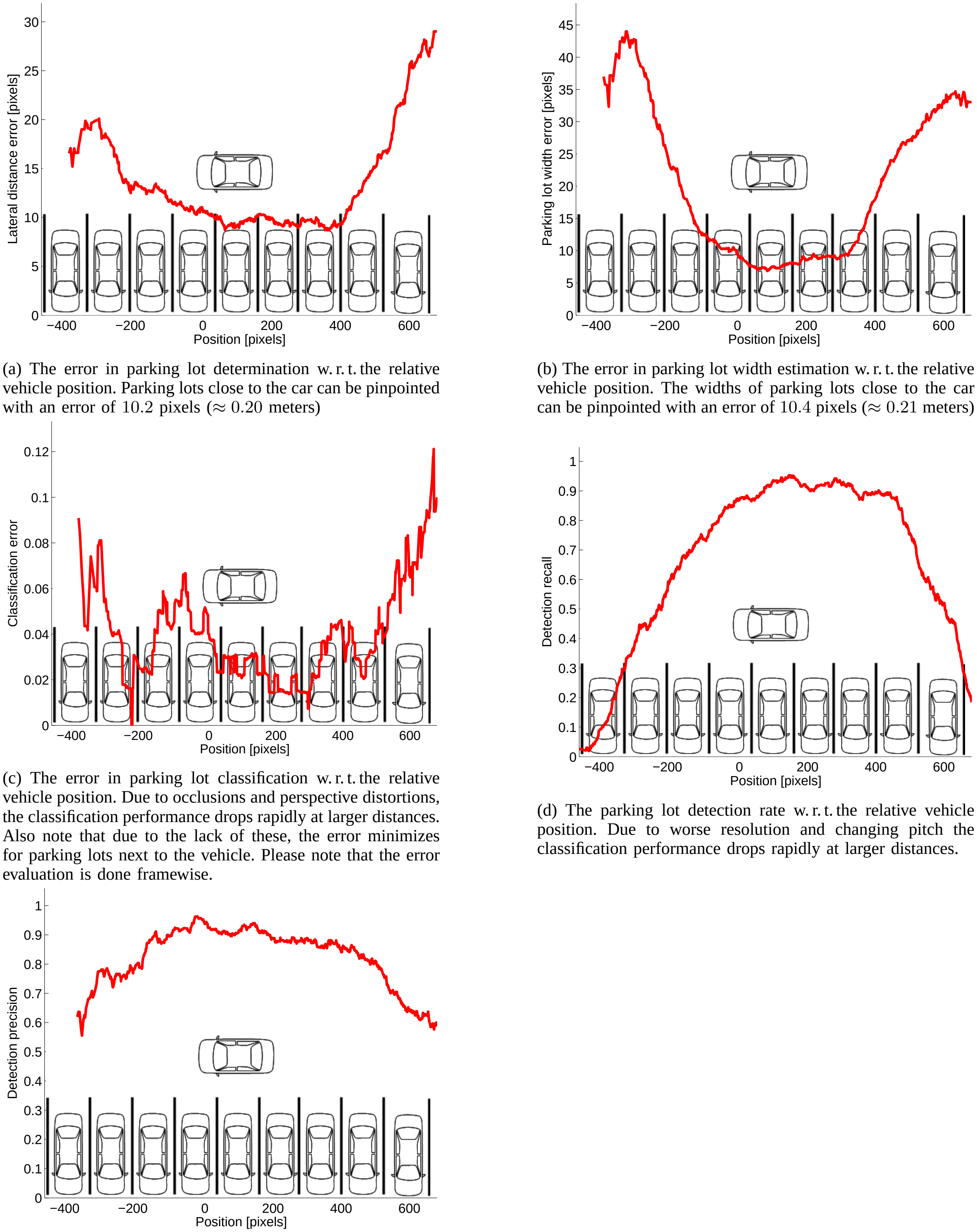
Figure 7 shows the course of the goodness of fit for a sequence. Ground truth, raw and temporally smoothed values are plotted.

B. Precision and recall of the parking lot detection

If the presence of a parking lot row has been verified the detector described in Sec. III-B and III-C is initiated to recognize the single parking lots. Figure 8d and 8e show its reliability by the averaged precision and recall relative to the vehicle position. The quick drop in performance for parking lots w. r. t. vehicle distance can be accounted to the aberration in the bird's eye image due to little resolution for distant regions as well as the changing pitch of the car.

C. Deviation of lateral distance

The accuracy of the lateral distance estimation is depicted in Fig. 8a, the mean error accounts to 12.4 pixels (≈ 0.25



(a) The error in parking lot determination w. r. t. the relative vehicle position. Parking lots close to the car can be pinpointed with an error of 10.2 pixels (≈ 0.20 meters)

(b) The error in parking lot width estimation w. r. t. the relative vehicle position. The widths of parking lots close to the car can be pinpointed with an error of 10.4 pixels (≈ 0.21 meters)

(c) The error in parking lot classification w. r. t. the relative vehicle position. Due to occlusions and perspective distortions, the classification performance drops rapidly at larger distances. Also note that due to the lack of these, the error minimizes for parking lots next to the vehicle. Please note that the error evaluation is done framewise.

(d) The parking lot detection rate w. r. t. the relative vehicle position. Due to worse resolution and changing pitch the classification performance drops rapidly at larger distances.

(e) The parking lot detection precision w. r. t. the relative vehicle position.

	Avg. Value	Avg. Value (near distance)
Lateral distance deviation	12.4px	10.2px
Width estimation error	16.7px	10.4px
Classification error	3.6%	3.2%
Detection rate	0.59	0.86
Detection precision	0.84	0.90

TABLE I: Several performance quantities of the presented image processing pipeline are shown for the whole surrounding and parking lots in a close environment (≈ 4.0m in front and behind the vehicle).

m), but is below 10.2 pixels (≈ 0.20 m) for currently passed parking lots.

D. Deviation of parking lot width

The spatial distribution of the error within the width estimation (*cf.* Fig. 8b) shows a similar course. On the average, we get an error of 16.7 (≈ 0.33 m), and accordingly 10.4 (≈ 0.21 m) at a nearer distance

E. Classification performance

The distribution of the parking lot classification performance is depicted in Fig. 8c. Again, the loss in performance for distant parking lots can mostly be attributed to occlusion of parking lots by other parked vehicles. The average classification error accounts to 3.6% and 3.2% for near vehicles. One can observe as well that the classification of parking lots recorded with the side cameras is most reliable because the relevant image region is only moderately distorted.

V. FUTURE WORK

We have presented a system for video-based parking lot recognition that is based on a stitched view around the vehicle. The detection algorithm searches for regular patterns of markings to stabilize single-detections and to predict the positions of possible parking lots at a farer distance. The task of classifying the occupancy is performed by a linear classifier on Difference-of-Gaussian features.

In the future we want to test the accuracy of our system with respect to world dimensions. Since the current surround view still suffers from moderate distortion, we will conduct experiments to refine the initial calibration of the camera system. Furthermore, for a versatile parking pilot it is imperative to deal with arbitrary oriented and less well-marked parking lots. The hardware in use allows for a higher resolution of the bird's eye image but at the cost of computation time. It is to be expected that this will further reduce the errors in the current detection process.

Our results show that the current detection accuracy of 0.33m average is sufficient to allow for placing the vehicle nearby a free parking lot and provide the necessary initialization for computing and performing the parking manoeuvre.

ACKNOWLEDGMENT

This publication is a result of a Continental funded research cooperation.

REFERENCES

[1] J. Wolff, T. Heuer, H. Gao, M. Weinmann, S. Voit, and U. Hartmann, "Parking monitor system based on magnetic field sensors," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1275–1279.

[2] M. Idris, Y. Leng, E. Tamil, N. Noor, and Z. Razak, "Car park system: A review of smart parking system and its technology," *Information Technology Journal*, vol. 8, no. 2, pp. 101–113, 2009.

[3] A. Ibisch, S. Stümper, H.Altinger, M. Neuhausen, M. Tschentscher, M. Schlipsing, J. Salmen, and A. Knolls, "Autonomous driving in a parking garage: Vehicle-localization and tracking using environment-embedded lidar sensors," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2013, submitted.

[4] N. True, "Vacant parking space detection in static images," University of California, San Diego, Tech. Rep., 2007.

[5] H. Ichihashi, T. Katada, M. Fujiyoshi, A. Notsu, , and K. Honda, "Improvement in the performance of camera based vehicle detector for parking lot," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, 2010, pp. 1950–1956.

[6] Y.-W. Seo and C. Urmson, "Utilizing prior information to enhance self-supervised aerial image analysis for extracting parking lot structures," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 339–344.

[7] M. Tschentscher and M. Neuhausen, "Video-based parking-space detection," in *Proceedings of the Forum Bauinformatik*, 2012, pp. 159–166.

[8] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easy calibrating omnidirectional cameras," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 7–15.

[9] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.