Blurring the border between real and virtual parking environments

Daniel Becker[1], Andrew Munjere[2], Jens Einsiedler[3], Kay Massow[1], Fabian Thiele[1], Ilja Radusch[2]

***Abstract*— Modern multi-level indoor parking environments promise to alleviate the parking problems in modern cities but they are oftentimes stressful for human drivers. Increasing automation of the parking process has the potential for significant gains in efficiency, safety and comfort but requires highly accurate sensing and monitoring of the environment. Another challenge is the appropriate visualization of large amounts of sensor data from disparate sources, in an intuitively understandable way. We address these challenges with our platform VPIPE for realistic visualization of 3D parking environments, parking lots and sensor data of vehicles. As central building block for this platform, we propose a cost-effective camera-based parking lot monitoring system that uses a cascade of Random Forest and Artificial Neural Network classifiers. The achieved detection accuracy in our parking testbed is 94.98%.**

I. Introduction

Modern cities around the world suffer from problems related to parking: Overcrowded streets, lack of space and time-consuming searches for parking spots. One potential solution are multi-level car parks which represent an efficient way for temporary vehicle parking. However, human drivers are often overwhelmed, as driving in these confined, highly dynamic and sub-optimally illuminated spaces is challenging. Also, especially if the parking environment has a high rate of occupancy, finding a free parking lot can take a considerable amount of time.

To alleviate these problems, driver assistance systems can be employed: The first step is a navigation system, where the driver is guided towards an available parking lot. The second step is automatic driving, where the vehicle is finding an available parking lot entirely on its own. Both strategies significantly reduce the burden of the human driver. However, the difference is in the strictness of the requirements in the underlying sensor task, i.e. automatic driving requires a considerably higher level of accuracy, robustness, performance and error detection for each of the sensing tasks.

[1]Daniel Becker, Kay Massow and Fabian Thiele are with the Daimler Center for Automotive Information Technology Innovations (DCAITI), Ernst-Reuter-Platz 7, 10587 Berlin, Germany {daniel.becker,kay.massow,fabian.thiele}@dcaiti.com

[2]Andrew Munjere and Ilja Radusch are with the Fraunhofer Institute for Open Communication Technologies (FOKUS), Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany {andrew.munjere,ilja.radusch}@fokus.fraunhofer.de

[3]Jens Einsiedler is with the Fraunhofer Application Center Wireless Sensor Systems, Am Hofbrauhaus 1, 96450 Coburg, Germany jens.einsiedler@iis.fraunhofer.de

This work was also supported by the European Commission under TEAM, a large scale integrated project part of the Seventh Framework Programme for research, technological development and demonstration [Grant Agreement NO.318621]. The authors would like to thank all partners within TEAM for their cooperation and valuable contribution.

In this work, we present our platform for visualization of indoor parking environments (VPIPE), an extension of the *Physics Aware Behavior Modelling Advanced Car Simulator* (PHABMACS). Our vision is to blur the border between the virtual and real world by achieving a customizable visualization of the 3D environment and sensor data from both the infrastructure and vehicles. This platform will accelerate the development, validation and demonstration of localization, object detection, tracking and other algorithms. Also, synergies are created by using VPIPE for parking space monitoring, Human Machine Interface (HMI) for navigation, tracking of autonomous vehicles, validation of indoor maps, etc. Further improvements are yielded by applying Cooperative Driver Assistance systems (CoDAS) in parking environments, such as Cooperative Adaptive Cruise Control (CACC) [1] which are evaluated in the TEAM project [2].

To achieve this, VPIPE dynamically renders indoor environments based on maps, infrastructure and vehicle data. Firstly, OpenStreetMaps (OSM) [3] are rendered true to scale. Secondly, the environment is equipped with a variety of sensors. Thirdly, vehicle sensors are incorporated to further enhance the richness of environmental representation.

In our previous work, we have used ubiquitous surveillance cameras to detect and localize moving vehicles [4], [5], [6], as well as tracking and identification of unique endpoints [6], [7]. In order to quantitatively evaluate these systems, we proposed a highly-accurate Ground Truth [8]. In this work, we propose a parking lot monitoring (PLOM) approach based on infrastructure cameras that is a central building block, sas many ADAS in parking areas rely on an accurate knowledge of the lot occupation state. In addition, we deployed Velodyne Puck VLP-16 LiDAR scanners [9] in the environment which yield highly accurate point cloud data. An example of the flexible visualization capabilities of VPIPE is displayed in Fig. 1 which shows three representations of the same parking scene using the PLOM detection, simulation models and LiDAR scanner.

This paper is organized as follows: In Section 2 related work is introduced, Section 3 explains the PHABMACS simulator, the basis of VPIPE, Section 4 presents the methodology of the PLOM system which is evaluated in Section 5, while the paper concludes in Section 6.

II. Related Work

Simulation tools relevant for ADAS research in the context of intelligent parking environments can be divided in three groups. The first group are microscopic traffic simulation tools like SUMO [10], which aim on modelling traffic flows and to support research with a focus on traffic influencing

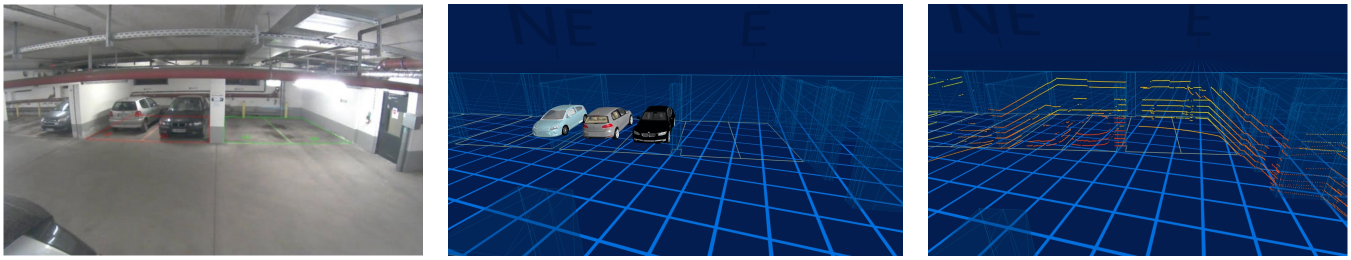

Fig. 1: Three vantage points on parking lots: PLOM detection (left), PHABMACS (center), LiDAR scanner (right).

applications rather than with physically realistic 3D representations. The second group, driving simulators, is dedicated to the involvement of human drivers into the simulation, like OpenDS [11]. These driving simulators are used for research on ADAS from a perspective of interaction with human drivers. For this reason, the simulation is designed to feel most realistic to the driver instead of mapping realistic vehicle physics. The third group of simulation tools, the most relevant group for our purpose, refers to the vehicle simulators such as TORCS [12] which aim on mapping realistic vehicle dynamics. TORCS, however is targeted towards racing scenarios and the visualization of race tracks. In contrast, the PHABMACS simulator is designed to map realistic 3D vehicle physics and to visualize a recognizable view of an existing real world environment with a minimum set of information, provided by OSM maps and sensor data.

With respect to indoor parking environments, parking lot detection systems (PLDS) often incur the disadvantage of high installation and deployment costs. In contrast, camera-based systems do not only provide an inexpensive and effective solution that is easy to deploy and covers a wide range of parking spaces, but also support other tasks such as surveillance. The substantial challenges for camera-based PLDS are the various environmental influences such as shadows, varying weather conditions and limited visibility as determined by the camera perspective. Most of the PLDS for parking lot occupancy detection in recent years are based on images and videos from cameras placed on the infrastructure side. For instance, template matching techniques based on reference images can be applied [13]. However, invariant descriptors such as SIFT [14] or SURF [15] in combination with the *bag of visual words* (BoW) [16] method have gained popularity due to their simplicity, performance and robustness [17], [18]. In this paper, we present a vision-based Parking Lot Monitoring (PLOM) system based on two classification methods (Random Forest [19] and Artificial Neural Networks [20]) that build multi-category image models based on invariant descriptors.

III. Simulation Environment

In the following, we provide a brief overview about the PHABMACS simulator, a distributed framework for testing ADAS [21] within a simulated 3D environment. PHABMACS, is the basis of VPIPE, even though only the visualization aspects are relevant to this paper. The ADAS under test in PHABMACS can utilize simulated sensor data as well as control simulated vehicles by using virtual actuators. In order to support such testing, the entities to be part of the simulation are driver, vehicles, and environment. The simulated driver is used to generate the input, which a driver normally generates by operating the vehicles actuators (throttle, brake, steering). In PHABMACS, this is realized by employing driver models, or using direct input from a human driver. The simulated vehicles map the vehicle dynamics on simulation objects, which generate sensor input used by the ADAS and reacts to the output on actuators of the vehicles. The environment simulates the influence, which objects have on the sensors and the dynamics of the vehicles, as well as road conditions, the street grid, and other environmental influences like slopes, air drag etc. PHABMACS is available as part of the VSimRTI suit[22].

1) Models: When developing PHABMACS, the requirements for the simulation where oriented on cooperative ADAS [21]. To simulate these a higher physics accuracy is needed than any existing microscopic traffic simulator like e.g. SUMO [10] can offer. Simulation scenarios contain less vehicles than large scale traffic simulations but can still involve hundreds of vehicles. However, in order to achieve this tradeoff between simulating vehicle dynamics precisely and still having enough performance for simulating multiple vehicle instances, PHABMACS does not aim on simulation of accidents. Thus, the simulated vehicle behavior close to the limits of driving dynamics is realistic until accidents happen, not subsequently. A correctly simulated vehicle behavior in extreme situations like heavy over- or understeering is not strictly realistic, as the class of applications PHABMACS aims on, is meant to prevent these situations. For this reason, PHABMACS is based on rigid body dynamics [23] employing basic models for chassis components e.g. dampers and more precise models for the powertrain as this is relevant for the interaction of the ADAS with vehicle. These models are of minor importance for the research done in the context of this paper and we therefore do not go into further details at this point.

2) Architecture and Implementation: The architecture of PHABMACS is depicted in Fig. 2. This architecture takes separation of user code and the PHABMACS framework into account. The user code is a separate piece of software, which the user employs to control the simulator, setup scenarios, define custom sensors and connect the ADAS to

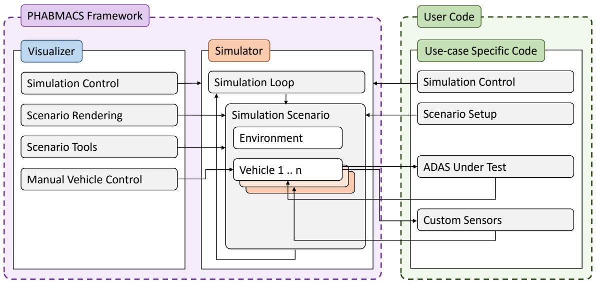

Fig. 2: PHABMACS architecture.

be simulated. The PHABMACS framework is separated into the visualizer and the actual simulator, which needs to be modified if changes on the models are needed. The framework also enables deployment of visualizer and simulator on different computers. In this way, multi-user scenarios with multiple human driver or observers can be set up, as well as high performance simulation scenarios with a runtime factor below real time, without visualization. Controlling the simulated vehicles in PHABMACS can either be done by the simulation, driving paths which are part of the scenario defined by the user, or by direct user input.

The implementation is realized in JAVA. For the basis of the physics engine of the simulator, JBullet has been chosen. The vehicle dynamics and the other models described earlier are custom JAVA implementations, which are used to calculate the forces to be applied to the objects in JBullet.

3) Visualizer: The Visualizer component displays the simulated world to the user utilizing advanced 3D graphics (cf. Fig. 1 and Fig. 2). This includes visualization of complex environments like parking garages, creating a recognizable visualization for the user from the limited set of 3D information of the environment, coping with low performance devices, and potentially a large number of objects to be rendered. In order to reconcile these requirements, we have designed the visualizer using the following features. In order to avoid additional effort on scenario setup, the visual models should be directly generated from the simulation models. The displayed environment is automatically generated from an arbitrary OSM file and includes streets, buildings, waterways, trees and terrain types. Moreover, an optional height map can be used to generate further terrain information. To get a high quality visual appearance the visualizer optionally uses cascaded shadow maps [24] and high-detail vehicle models. The employed visualization engine is a custom implementation, which uses LWJGL as JAVA based library to access OpenGL functions on the target system. There are options to connect ADAS which are no JAVA implementations, e.g. a Matlab/Simulink interface is available.

4) Parameterization of the Vehicle Model: As described, the precise simulation of vehicle dynamics is important for original purpose of PHABMACS. In order to reach such a precision, the vehicle models are parameterized to match the real world pendants of the vehicles to be simulated. In this way, from the perspective of the ADAS under test, the simulation gets close to the situation when being deployed in a real world vehicle.

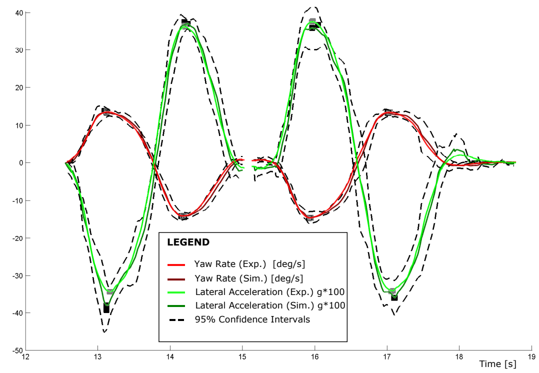

Fig. 3: Validation of simulation vehicle model.

This parameterization has been validated in two steps according to the graphical comparison technique [25]. First, static parameters of a real world vehicle like dimensions, mass, maximum steering angle, suspension spring constant, etc. are set to the simulation model. Second, experimental data captured from a real world vehicle is used to correct the vehicle models. In the context of this paper, we do not go into details. However, as one example of the result of validating the model used for the research in this paper, [ref] shows the graph for comparison of the lateral dynamics captured from real world experiment and simulation. According to the methodology for validation of vehicle lateral dynamics simulation described in [26], six selected test runs performing a double lane change maneuver where split, aligned and averaged. The 95% confidence interval was calculated for yaw rate and lateral acceleration using student's t-distribution. As depicted in the resulting Fig. 3, one of the major validation criteria is met, as the averaged simulation output are within the confidence interval.

IV. Parking Lot Detection

The proposed processing steps of our parking lot monitoring system are the following: In the first step (1.), we obtained an image from the camera which is monitoring multiple parking lots (cf. Fig.1 left). The second step (2.) is the segmentation of the entire image into individual regions of interests (ROIs) for each parking lot. Due to the fixed mounting point of the camera and fixed position of parking lots, the ROIs are set manually in an initial calibration step. In the next step (3.), a classifier evaluates each ROI to label it as either occupied or available. Finally (4.), the status for each parking lot is provided to PHABMACS in order to enable VPIPE to visualize the parking lot occupation state (cf. Fig.1 center).

Our proposed image classification step is based on BoW models which is a supervised learning technique using a codebook of visual words obtained by clustering the extracted local invariant image descriptors with the *K-Means*

		Prediction	
		Occupied	*Available*
Reality	*Occupied*	True Positive (TP)	False Negative (FN)
	Available	False Positive (FP)	True Negative (TN)

TABLE I: Confusion Matrix denoting the possible results of the PLOM classifier.

[27] algorithm. Put differently, an image is represented as a bag of visual words, which is a vector of K bins that counts the number of occurrences of particular image patterns or vocabulary. This histogram of visual words acts as input for multi-class classifiers.

The proposed PLOM is a binary classifier as it differentiates two possible classes for each parking lot input image, i.e. occupied and available. The confusion matrix in Tab.I illustrates the four possible states for the classifier results depending on its prediction and the actual reality. Moreover, the classifier is operated in two stages, training and testing resp. In both the training and testing stage, the first step is detection and representation of features (i.e. *keypoints*). The second step in the training stage is to use the features of step 1 and given class labels (i.e. supervised learning) to train a classifier. In the testing stage, this classifier determines the class label of an arbitrary new image.

We selected a dataset of about 22,000 images for the training process. For both classes (i.e. occupied and available), we use images from our parking test site, as well as images from publicly available datasets [28], [29]. For the evaluation of the detection accuracy, we used additional 757 images from our parking test site (see Fig.4) which are not in the training data set.

For the basic representation of image features, we investigated the following keypoint detectors and descriptors [30]:

- Difference of Gaussians (DoG) feature detector and the 128-elements SIFT descriptor.
- Fast Hessian feature detector and the 128-elements upright SURF descriptor.
- Oriented Brief (ORB) keypoint detector and descriptor.

Based on aforementioned detector and descriptor methods, a Random Forest (RF) [19] and an Artificial Neural Networks (ANN) [20] classifier was trained on the training data set. Furthermore, we have used a two-step empirical method selection process to identify the optimal combination of keypoint detector and classifier for a varying number of clusters in the BoW vocabulary. In the first step, we evaluated the detection accuracy of each combination of a single individual keypoint detector and classifier. In the second step, we have assessed the detection accuracy of cascades of individual classifiers.

In the first step we tested the suitability of each method for our classification task. To this end, the classifiers were trained with different $\boldsymbol{K}$ number of clusters. Thus followed the validation and analysis of the results obtained after each training round. These results were then recorded and evaluated in a

Fig. 4: Examples of evaluation input images (top row occupied, bottom row available).

confusion matrix (see Tab.I), thus the training parameters of the classifiers and the size of the BoW vocabulary $\boldsymbol{K}$ are quantitatively optimized.

The second step addresses the consequences of a combination of methods, to compensate the disadvantages of each method and to benefit from the respective advantages. With this in mind, the classifiers were individually and in combination also trained and verified. In order to simultaneously achieve a high detection rate at the same time the lowest possible false alarm rate and low computational time, the parking lot occupancy detection is carried out in cascaded classifiers. The idea of cascaded classifiers is that, it executes consecutively multiple basis classifiers in order to improve the classification result iteratively and reducing the misclassifications by a further classifier.

The results of the empirical method selection presented in Section V reveal that the optimal detection accuracy for the evaluation data set is achieved for a RF-ANN-ANN cascade based on FAST and SIFT descriptors.

V. Evaluation

Our underground carpark test site (cf. Fig. 5) is equipped with 16 *AXIS M3114VE 2MM (FW: 5.40.9.2)* network cameras (LOT0 to LOT15) providing images at a resolution of $1280 \times 800\,\text{px}$ via Gigabit Ethernet encoded as JPEG. Overall, 65 parking lots are monitored as one camera covers between three to six parking lots. The cameras mounting heights are approx. 2 m; the distance to the lots is between 4 m to 7 m. The detection software is implemented in C++ using the library OpenCV 3.1 on a PC with an Intel(R) Core(TM) i7-4700MQ and 16 GB RAM on Ubuntu 14.04 LTS (64 bit). In the following, we first present the results of the aforementioned two step empirical method selection process, to determine the optimal classifier cascade. Subsequently, we evaluate this classifier cascade approach with a manually labeled evaluation data set of our parking testbed.

1) Empirical method selection process: We assessed the detection accuracy of each method in order to choose the best combination of feature detector, descriptor and classifier. Also, we used a confusion matrix (see Tab.I) with the 4 classes True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) to evaluate the prediction quality of our approach on the evaluation data set.

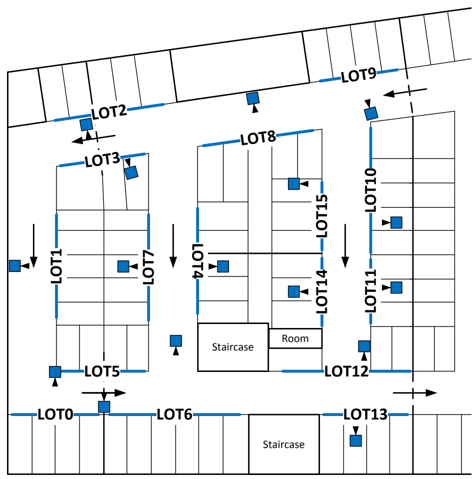

Fig. 5: Site plan with parking lot monitoring cameras.

	TM	PLOM
True Positives	344	346
True Negatives	328	373
False Positives	53	6
False Negatives	32	32
Accuracy	88.77%	94.98%
MCC	0.78	0.90
True Positive Rate	91.49%	91.53%
True Negative Rate	86.09%	98.42%

TABLE II: TM and PLOM detection results.

Phase 1: Descriptor and classifier selection

Considering the accuracy of the different descriptors over number of training image, the best result is achieved for a BoW cluster size of ***K***=1500. For this value, SIFT and SURF descriptors based classifiers achieved the highest accuracy, for the given data sets.

Regardless of the number of training images, the ORB-based classifiers could not be improved beyond a detection accuracy of about 83%. As a result, the ORB classifiers were no longer taken into account in the subsequent second phase. Furthermore, to take an advantage of the efficiency of the ORB detector and the performance of the SIFT and SURF descriptors, the FAST detector was used in connection with the SIFT or SURF descriptors.

Phase 2: Combination of methods

Due to the method combination, we were able to use a BoW cluster size larger than 10,000 without resulting into overfitting the classifiers. We found that a RF-ANN-ANN classifier cascade achieved the best detection results with ***K***=20,000. For this value, FAST and SIFT descriptors lead to the lowest overall error rate (7.40%).

2) Quantitative evaluation: We use a basic template matching approach similar to [13] (referred to as TM) as baseline for the evaluation of the detection accuracy, in comparison with our proposed RF-ANN-ANN cascade (referred to as PLOM). The TM approach uses a reference image for each parking lot ROI representing the *available* state. Each test image is compared against the reference image and the proportion of matching area by the TM approach determines the parking lot occupation state.

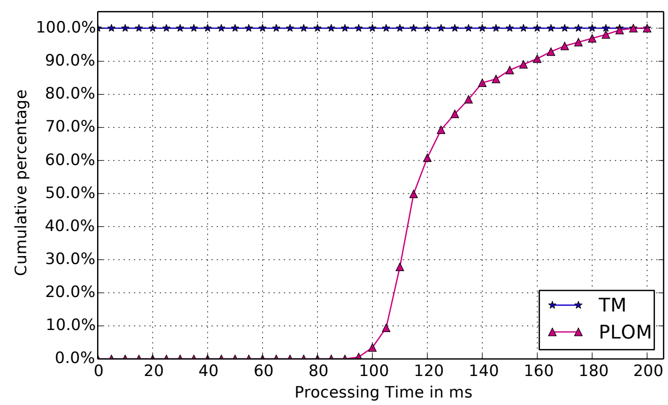

Fig. 6: Cumulative Density Function (CDF) of processing time for TM and PLOM algorithms, in ms.

The evaluation image data set contains 757 manually labelled images for both occupied and available parking lots captured randomly over a period of months. Thus, these images represent a realistic environment as they contain different levels of illumination, dirt and moisture. Also, some cars are not perfectly aligned to the parking lot boundaries. Moreover, some images contain arbitrary objects in the parking lots, such as pedestrians or building materials. For the subsequent evaluation, we use the following evaluation metrics (see Tab. I):

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2}$$

$$TPR = \frac{TP}{TP + FN} \tag{3}$$

$$TNR = \frac{TN}{TN + FP} \tag{4}$$

The $accuracy$ represents the overall rate of correct detections in relation to the total number of image detections. The MCC (Matthew's Correlation Coefficient) score has been selected due to its interesting properties: In a range of -1 to $+1$, where 0, -1 and $+1$ can be interpreted as chance results (i.e. *coin flip*), all detections incorrect and all detection correct resp. The TPR (True Positive Rate) and TNR (True Negative Rate) indicate only the correct detection rate within their class, i.e. occupied and available respectively.

Tab. II shows the evaluation results for TM and PLOM. Generally, TM and PLOM exhibit an $accuracy$ of 88.77% and 94.98% or a MCC of 0.78 and 0.90 resp. Also, TM and PLOM show a TPR of 91.49% and 91.53%, as well as a TNR of 86.09% and 98.42% resp. TM shows a relatively high number of false positives, i.e. empty parking lots declared as occupied. In this case, some parking lots will not be utilized which incurs economic consequences for the operator. Also, both TM and PLOM show a relatively high number of false negatives, i.e. occupied parking lots being declared as available. This can be problematic as vehicles might be guided towards these occupied lots, leading to potential overutilization and congestion. Thus, in the process of tuning the classifier's parameters a trade-off between a higher TPR and a lower TNR should be chosen if possible.

Fig. 6 illustrates the processing time of TM and PLOM for each image frame (containing in average 4 ROIs). TM and PLOM have a median processing time of $4ms$ and $107ms$ resp. The PLOM processing time standard deviation is $14ms$ and the 95 percentile processing time is $174ms$.

VI. Conclusion and Outlook

In this work, we have presented a highly accurate real-time parking lot detection system which is a central building block for our simulation and visualization platform VPIPE, an extension of the PHABMACS simulator. This system achieves a detection accuracy of 94.98% in a realistic parking environment under challenging environmental and illumination conditions. Further synergies can be achieved by combining results of our previous research into PHABMACS, i.e. detection and localization of vehicles and pedestrians, identification and tracking as well as collision warnings.

Our future vision is to blur the lines between the real and virtual environments by gaining an all-encompassing representation of objects measured by disparate sensors, both in the environment and vehicle. Different aspects of the virtual representation can be used for different purposes, such as navigation, tracking of autonomous vehicles, collision detection, parking space surveillance, etc. A myriad of new possible visualization and interaction concepts arises when connecting the PHABMACS simulator with virtual reality hardware components.

References

[1] Clemens Dannheim, Markus Mader, Christian Icking, Jan Loewenau, and Kay Massow. Team–co2 reduction through online weather assistant for cooperative acc driving. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, pages 369–373. IEEE, 2013.

[2] Angelos Amditis, Panagiotis Lytrivis, Ioannis Karaseitanidis, Matthias Prandtstetter, and Ilja Radusch. TomorrowâĂŹs transport infrastructure: from static to elastic mobility. In *20th ITS World Congress*, 2013.

[3] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.

[4] J. Einsiedler, O. Sawade, B. Schäufele, M. Witzke, and I. Radusch. Indoor micro navigation utilizing local infrastructure-based positioning. In *Intelligent Vehicles Symposium (IV), 2012*, pages 993–998. IEEE, 2012.

[5] J. Einsiedler, D. Becker, and I. Radusch. External visual positioning system for enclosed carparks. In *Positioning, Navigation and Communication (WPNC), 2014 11th Workshop on*, pages 1–6. IEEE, 2014.

[6] D. Becker, J. Einsiedler, B. Schäufele, A. Binder, and I. Radusch. Identification of vehicle tracks and association to wireless endpoints by multiple sensor modalities. In *Proceedings of the 4th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013.

[7] D. Becker, B. Schäufele, J. Einsiedler, O. Sawade, and I. Radusch. Vehicle and pedestrian collision prevention system based on smart video surveillance and c2i communication. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 3088–3093. IEEE, 2014.

[8] Daniel Becker, Fabian Thiele, Oliver Sawade, and Ilja Radusch. Cost-effective camera based ground truth for indoor localization. In *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on*, pages 885–890. IEEE, 2015.

[9] Velodyne LiDAR PUCK. PuckâĎć. Velodyne, 2015.

[10] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility. In *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.

[11] Rafael Math, Angela Mahr, Mohammad M Moniri, and Christian Müller. Opends: A new open-source driving simulator for research. *GMM-Fachbericht-AmE 2013*, 2013.

[12] Enrique Onieva, David A Pelta, Javier Alonso, Vicente Milanés, and Joshué Pérez. A modular parametric architecture for the torcs racing engine. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 256–262. IEEE, 2009.

[13] Andre Teguh Guntoro Thiang and Resmana Lim. Type of vehicle recognition using template matching method. In *Proceedings of the International Conference on Electrical, Electronics, Communication and Information*, pages 7–8, 2001.

[14] D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[16] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *In ECCV International Workshop on Statistical Learning in Computer Vision*, 1(1-22):9–84, 2004.

[17] N. True. Vacant parking space detection in static images. Technical report, University of California San Diego, 2007.

[18] C. Wah. Parking space vacancy monitoring. Technical report, University of California San Diego, 2009.

[19] L. Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[20] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[21] Meng Wang. *Generic Model Predictive Control Framework for Advanced Driver Assistance Systems*. TU Delft, Delft University of Technology, 2014.

[22] Björn Schünemann. V2x simulation runtime infrastructure vsimrti: An assessment tool to design smart traffic management systems. *Computer Networks*, 55(14):3189–3198, 2011.

[23] H Goldstein. Classical mechanics,(1980), ed.

[24] Andrew Lauritzen, Marco Salvi, and Aaron Lefohn. Sample distribution shadow maps. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, pages 97–102, New York, NY, USA, 2011. ACM.

[25] Emir Kutluay and Hermann Winner. Validation of vehicle dynamics simulation models–a review. *Vehicle System Dynamics*, 52(2):186–200, 2014.

[26] Emir Kutluay and Hermann Winner. Assessment methodology for validation of vehicle dynamics simulations using double lane change maneuver. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, pages 1–12. IEEE, 2012.

[27] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

[28] Alex Berg, Jia Deng, and Fei-Fei Li. Imagenet large scale visual recognition challenge 2010, 2010.

[29] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2012 (voc2012) results, 2012.

[30] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *Computer Vision–ECCV 2012*, pages 759–773. Springer, 2012.