Advanced Real-time Indoor Parking Localization based on Semi-Static Objects

Benjamin H. Groh*, Martin Friedl**, Andre G. Linarth*, Elli Angelopoulou*
*Pattern Recognition Lab,
University of Erlangen-Nürnberg, Germany
Contact Email: benjamin.groh@fau.de
**BMW Group Forschung und Technik,
Munich, Germany

***Abstract*—Indoor parking localization systems face two major challenges: the absence of GPS coverage and the dynamic character of the parking environment. The absence of GPS coverage can be compensated by data fusion of several sensors in order to localize an object in a map of the environment. However, this map may constantly change due to the dynamic nature of a parking garage.**
This paper describes an improvement to indoor localizations by a new map representation approach. The algorithm is developed for the application of an indoor parking system. The car park's map is based on the known permanent elements of the environment in combination with the knowledge of possibly changing elements. The permanent elements (e.g. walls) are considered static objects, while the known changing elements (e.g. parked cars) are modeled as semi-static objects. Such a representation avoids constantly updating the map with newly detected objects.
Our algorithm fulfills the real-time requirement of the localization task for a computer-controlled car through the use of a particle filter. The filter uses laser scanner measurements and odometry data to localize the car in a precalculated probability grid map, containing static and semi-static elements. The evaluation of the algorithm in a real car park scenario demonstrates the robustness and real-time capability of the proposed system with a RMS error in the absolute position of 0.33 m and in the heading angle of 1.03 ° at a computation time of 10 ms per cycle.

I. Introduction

A. Motivation

Due to urbanization, more and more people are going to live in metropolitan areas. One result of the increasing population is the raise in traffic in these centers. More parking space will be needed to deal with the increasing number of cars. A possible way of solving this problem could be a new structure of multi-story car parks that are available to autonomously driving cars. These vehicles could be left outside the car park and are expected to find their way to a prespecified parking position. The traffic inside the building could be minimized, accidents avoided and the available space maximally utilized. Further advantages would be improved comfort and time-saving benefits. Drivers would not have to find a parking spot but just leave their cars in a car drop-off zone.
The requirements of a reliable indoor localization involve real-time operation and robust positioning without the use of GPS. There are different approaches for solving the indoor localization task. In the particular case of localization inside a car park, the special dynamic nature of the environment has to be considered. The basic construction of the building will not change over time. In contrast, the utilization of the parking slots and the thereby created environment varies. In some slots there are parked cars and stay at the same position during the localization task, whereas other slots stay completely empty. The positions of the parking slots are considered known and can therefore be used in the localization.
This paper describes a new localization algorithm based on a particle filter, specifically developed for parking environments. Its uniqueness lies on the fact that it not only focuses on static objects, e.g. the walls of the building, but also uses parked cars as semi-static objects for accurate localization. Besides static and semi-static objects, there are also dynamic objects, e.g. moving cars, which should also be appropriately handled for safe and precise localization.

B. Related work

Prior localization projects have already addressed the combination of static and dynamic objects. Kümmerle et al. [1] proposed a SLAM (simultaneous localization and mapping) based indoor localization approach for multi-level car parks. It relied on LIDAR (light detection and ranging) data in the absence of GPS information. A similar approach was presented by Levinson et al. [2]. They proposed a graphSLAM algorithm that worked well in dense and dynamic environments. It used a particle filter and GPS data to create a road map. Ibisch et al. [3] introduced a tracking algorithm based on a network of LIDAR sensors. This tracking provided a distinction between static and active (or dynamic) objects and resulted in accurate localization. Bojja et al. [4] built their work on a 3D map representation of a multi-level parking garage. They proposed a localization method including a collision detection only based on gyroscope and odometry data. Suppe et al. [5] presented a further motivation for an autonomous parking system. By designing a semi-autonomous parking aid for garages, entering and exiting parked cars could be made easier, especially for elderly drivers. In their implementation they fused LIDAR and odometry data.
Most of the aforementioned work relied on a continuous SLAM algorithm to track changes in the environment. The

map of the environment was constantly updated and used for localization. SLAM algorithms are well developed and widely used in localization tasks. However, they can be expensive with respect to both computation time and memory requirements. None of the aforementioned work included the knowledge of parked cars or in general semi-static objects. Our contribution is to effectively utilize this additional information instead of relying on a SLAM algorithm. The structure of the building itself and the slot positions are assumed to be known beforehand and to remain constant.

II. Methods

A. Sensor hardware and data collection

The available sensors for this project were a four-layer Ibeo Lux laser scanner (accuracy σ_{laser} = 0.1 m) located at the front of a test vehicle (BMW 5 series) and the odometry data of the car. The odometry measurements provided the current velocity and angular rate of the heading of the car (yaw rate). A quantitative analysis of the accuracy of the odometry data was done in [6]. The result of this analysis showed a standard deviation of σ_v = 0.02 $\frac{\text{m}}{\text{s}}$ and σ_ω = 0.0003 $\frac{\text{rad}}{\text{s}}$ for the velocity and yaw rate. The resulting uncertainty in position and heading was calculated in [6] and will be referred to as $\mathbf{\Sigma}_{odometry}$.
The evaluation of the localization system was based on outdoor test runs in a parking environment that was similar to an indoor car park. The map of the outdoor car park was modeled with a GPS-aided inertial navigation system with a position accuracy of up to 0.02 m. In addition, the car internal GPS system was used to localize the vehicle in the map as ground truth for the implemented position filter. The vehicle internal GPS provided an accuracy of up to 0.02 m in position and up to 0.1 ° in heading angle.
The test runs were executed in the described environment and included scenarios of clear views of walls as well as limited views caused by parked cars. The recorded data was stored for offline evaluation and analyzed using different filter configurations (three times each) of the presented localization system.

B. Particle filter based localization

The localization task was executed by a particle filter algorithm that is briefly explained in this chapter. It did not include a SLAM algorithm but relied on a precalculated probability grid map of the environment. The map itself is the novelty of this work: it explicitly models static and semi-static objects (see section II-C for further details).
Fig. 1 shows a schematic representation of the components of the implemented particle filter. The total number of particles is defined by M. The state of a particle $m \in \{1, 2, ..M\}$ is provided by $\boldsymbol{x}^{[m]} = (x, y, \theta)'$. It contains the 2D-position x, y in the map and the heading angle θ. The importance factor $\rho^{[m]}$ denotes each particle's likelihood of being in the true current position.
Initially, an approximate origin $\boldsymbol{x}_{origin}$ is expected to be available, e.g. by knowing the entrance coordinates of the car park or the last obtained GPS position. The state of each particle m is spread around the initial position $\boldsymbol{x}_{origin}$ according to the uncertainty $\mathbf{\Sigma}_{origin}$.

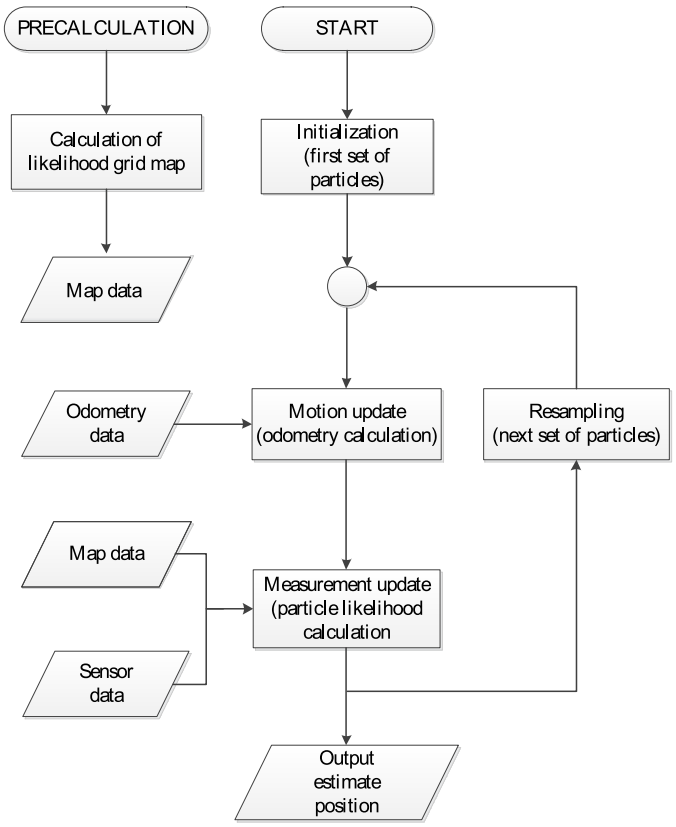


Fig. 1. Framework of the implemented particle filter

$$\boldsymbol{x}_{init}^{[m]} \sim \mathcal{N}(\boldsymbol{x}_{init}, \mathbf{\Sigma}_{origin}) \tag{1}$$

In every localization cycle the initial importance factor $\rho^{[m]}$ is uniformly distributed over all particles before the motion update and the laser scanner measurements are considered.
In the motion update step the former particle state $\boldsymbol{x}_{t-1}^{[m]}$ (initially: $\boldsymbol{x}_{init}^{[m]}$) is updated. The state is propagated by applying the current odometry data Δv_{odom} and $\Delta\omega_{odom}$, together with the normally distributed odometry uncertainty $\mathbf{\Sigma}_{odometry}$, in order to obtain the estimate $\bar{\boldsymbol{x}}_t^{[m]}$:

$$\bar{\boldsymbol{x}}_t^{[m]} = f_m(\boldsymbol{x}_{t-1}^{[m]}, \Delta v_{odom}, \Delta\omega_{odom}) + \boldsymbol{w_t} \tag{2}$$

where $\boldsymbol{w_t} \sim \mathcal{N}(\bar{0}, \mathbf{\Sigma}_{odometry})$ and f_m is a possibly non-linear transition function that models the evolution of the underlying dynamic system.
The measurement update follows the likelihood fields for range finders method of [7]. It adapts the likelihood $\rho_t^{[m]}$ of each particle being in the true state. This calculation is based on the provided information of the probability grid map and the laser scanner measurements of K beams $z_t^{[k]}$, with $k \in \{1, 2, ..K\}$. Based on the assumption that each particle could represent the true position of the car, all laser scanner beams $z_t^{[k]}$ are compared to the map-based scenario which the laser scanner would detect if it had been in this very particle position $\bar{\boldsymbol{x}}_t^{[m]}$.

If, for example, a sensor beam returns a short distance to an obstacle, then all particles that represent a position facing a wall are rated as more likely than particles facing free space. The likelihood of a laser scanner beam - particle state combination is denoted by $p(z_t^{[k]}|\bar{\boldsymbol{x}}_t^{[m]})$. The product of all beam likelihoods represents the probability or weight $\rho_t^{[m]}$ of a particle.

$$\rho_t^{[m]} = \prod_{k=1}^{K} p(z_t^{[k]}|\bar{\boldsymbol{x}}_t^{[m]}) \tag{3}$$

Random objects (e.g. moving vehicles) that disturb one or more laser beams would result in an unlikely sensor measurement $z_t^{[k_{err}]}$ and $p(z_t^{[k_{err}]}|\bar{\boldsymbol{x}}_t^{[m]})$ would be set close to 0. Due to the product in (3) the weight of the corresponding particle would be set close to 0 and the possibly well positioned particle would probably be ignored in further cycles. In order to avoid this disturbance an error measurement value $p_{z,err}$ is applied in every likelihood calculation. Equation (3) is replaced by

$$\rho_t^{[m]} = \prod_{k=1}^{K} \left(\frac{p(z_t^{[k]}|\bar{\boldsymbol{x}}_t^{[m]})}{\alpha_t} + p_{z,err}\right) \tag{4}$$

with the normalization factor

$$\alpha_t = \frac{\max\left(p(z_t^{[k]}|\bar{\boldsymbol{x}}_t^{[m]})\right)}{1 - p_{z,err}} \tag{5}$$

to avoid sensor beams with a likelihood greater than 1. An appropriate value for $p_{z,err}$ was empirically determined to 0.3.
In the next step of the localization procedure, the resampling algorithm selects which particles will be used for the next iteration.
The multinomial resampling algorithm of [8] is used in this work. The basic idea of this approach is to draw M particles from the current set of particles. Particles can be drawn with replacement. The probability of one particle to be selected is given by its importance weight. The particle selection is thereby based on the probability of each particle, but still occurs on random conditions. Hence, particles with a low probability are less likely to be selected but are not excluded from the resampling task. The method is visualized in Fig. 2. M particles (in the example M = 6) of the former set are distributed along a circular line. Their relative size equals the normalized importance weight. Particles are drawn by selecting random positions at the circular line.
The last step of the particle filter algorithm is to estimate the current position. This is done by averaging all particles' positions and headings after the measurement update. The weighted average is calculated in respect to the important weights $\rho_t^{[m]}$. The iterative algorithm does not allow a position output before the standard deviation of all particles is smaller than a predefined threshold of 1 m in position and 35 ° in heading direction.
A detailed description of the basic function of particle filters is beyond the scope of this paper, but can be found in [9]. The details of the implementation of our filter algorithm are described in [6].

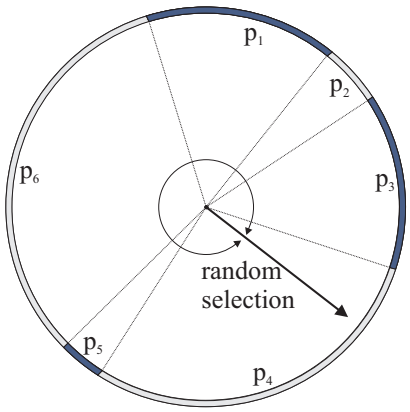


Fig. 2. Visualization of multinomial resampling: All particles are distributed relatively to their weight. Positions at the circular line are randomly chosen and thereby particles are selected.

C. Map representation

The main focus of this work was the integration of the parked cars as semi-static objects in the localization. This was modeled in the precalculated probability grid map of the parking environment. The environment was assumed to be known in advance. The probability grid map was tessellated into C square cells, each $0.05\,\mathrm{m}$ wide. The map represented the likelihood of each cell being occupied. A cell which was located exactly at a wall of the car park was assigned a high likelihood of being occupied. In other words, a cell close to a wall was most likely to be detected by a laser scanner beam. In contrast, a cell in the middle of the path was assumed less likely to be occupied and detected by the sensor. The map only had to be created once, as the environment was not assumed to change over time.
The floor plan of an example parking environment was constructed manually for this work and is provided in Fig. 3. It shows the original known structure before any computations. It contains the walls of the car park as static objects (black) as well as the semi-static objects (grey). These latter objects are depicted by lines which represent the boundaries of possibly parked cars. When cars are parked in front of the wall, the autonomous vehicle cannot detect the wall anymore. The collection of parked cars generates a new "wall", which has to be considered by the system. As a result, the structure representation contains two different classes: the wall elements (static) and the slot elements (semi-static).
Static objects were assigned an existence probability of $p_{exist,wall}$ = 1, whereas the semi-static objects were only assumed to have an existence probability of $p_{exist,slot} \in (0;1)$. Different uncertainties further influence the configuration of the probability grid map: the laser sensor uncertainty σ_{laser}, the parking position uncertainty σ_{pos} and a particle filter distribution, modeled by a particle filter dependent uncertainty $\sigma_{particle}$. The laser uncertainty was set to σ_{laser} = 0.1 m as provided by the manufacturer. There were two position uncertainties: $\sigma_{pos,wall}$ for the static and $\sigma_{pos,slot}$ for the

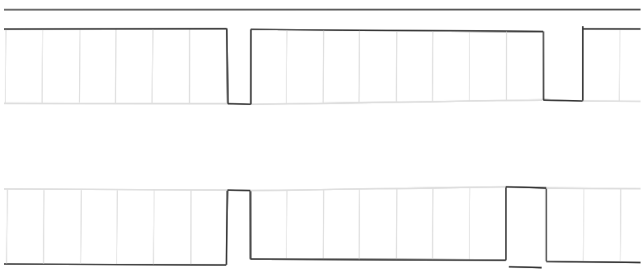


Fig. 3. The floor plan of the parking environment: wall elements (black), parking slots (grey)

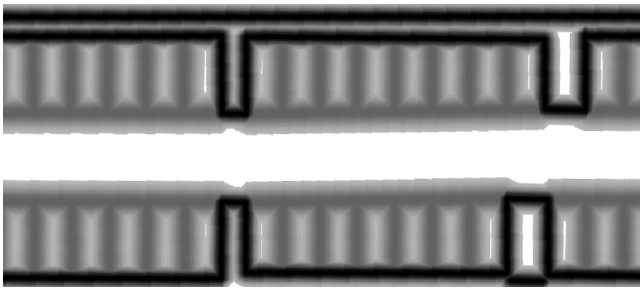


Fig. 4. Probability grid map of the parking environment: The black cells represent a high likelihood of being detected by a laser scanner beam. The lighter the shade of grey, the lower the likelihood value.

semi-static objects. Walls always keep their location but cars are never parked in exactly the same position. Hence, the uncertainties were set to $\sigma_{pos,wall}$ = 0 and $\sigma_{pos,slot}$ = 0.8 m, according to the maximal possible variation in the parking position of an average car in a slot while still remaining inside the designated parking area. To ensure the proper functionality of a particle filter, a distribution of the particles was required. The laser sensor uncertainty was very low. Thus, only the very best particles, which almost perfectly match the measured data, would have been considered as true positions and all other particles would be set to a likelihood of zero. In order to avoid this problem, a particle distribution was employed, modeled as a zero-mean Gaussian distribution with a standard deviation of $\sigma_{particle}$. For the evaluated parking scenario a standard deviation of $\sigma_{particle}$ = 0.4 m was empirically determined.
Using all this information, the resulting uncertainty σ_{res} was calculated by

$$\sigma_{res} = \sqrt{\sigma_{laser}^2 + \sigma_{pos}^2 + \sigma_{particle}^2}\,. \tag{6}$$

σ_{res} was used to determine the total likelihood that a cell was occupied. In a loop over all cells C, the distances from the current cell $c \in C$ to the closest wall element $d_{c,wall}$ and to the closest slot element $d_{c,slot}$ were obtained from the map. The distance calculation was based on the Chamfer distance transformation algorithm as in [10]. The likelihoods that the corresponding cell was occupied by a wall or by a slot element ($l_{c,wall}$ and $l_{c,slot}$) were computed as a function of the distance, the resulting position uncertainty σ_{res} and the existence probabilities $p_{exist,wall/slot}$.

$$\begin{aligned} l_{c,wall} &= \frac{1}{\sqrt{2\cdot\pi\cdot\sigma_{res,wall}^2}} \cdot e^{-\frac{d_{c,wall}^2}{2\cdot\sigma_{res,wall}^2}} \cdot p_{exist,wall} \\ l_{c,slot} &= \frac{1}{\sqrt{2\cdot\pi\cdot\sigma_{res,slot}^2}} \cdot e^{-\frac{d_{c,slot}^2}{2\cdot\sigma_{res,slot}^2}} \cdot p_{exist,slot} \end{aligned} \tag{7}$$

The two likelihood values $l_{c,wall}$ and $l_{c,slot}$ of each cell were compared with each other and the higher one was assigned to the cell. This was necessary to avoid the failure of ignoring a higher likelihood value for the distance to a wall, by setting a low likelihood to a cell according to a closer, but less likely slot element.

$$l_c = \max(l_{c,wall}, l_{c,slot}) \tag{8}$$

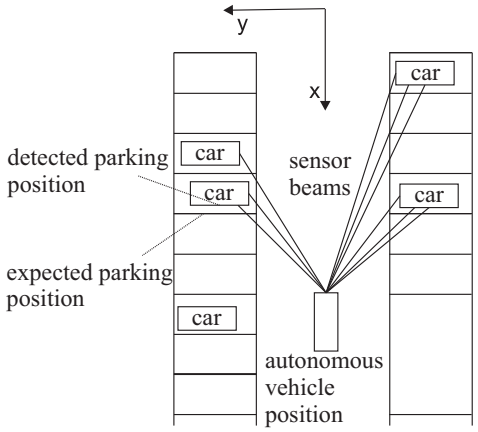
The resulting probability grid map is visualized in Fig. 4.

D. Correction of alignment errors

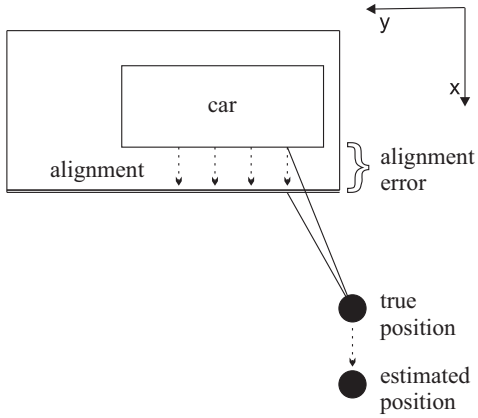
The slots of a car park are usually wider and longer than the parked cars. Thus, the cars were expected to be parked inside the slot but the exact position was not known in advance. This was represented in the map as parking uncertainty. By considering this uncertainty, no parked car could lead to completely failed localizations (failure which the algorithm cannot recover from), but it led to higher inaccuracies.
Problems occurred when mostly parked cars were detected and no walls were visible. Due to the distance between the parked cars and the borders of the slot element, there was almost always an offset in one direction, depending on the point of view of the autonomous vehicle (Fig. 5a). If the filter did not detect any wall element, the estimated position of the autonomous vehicle was adjusted in a way that the detected parked cars matched the expected slot element in the map (Fig. 5b).
When the cloud of particles was infinitely small, the filter could not recover from errors, even if the vehicle detected only wall elements. No particle would have been close to the true position to attract more particles because all the samples had already converged to a wrong estimated state. In order to correct these matching-based errors, a systematic particle spreading was necessary to allow the filter to recover from small localization errors. This spreading was implemented by $\mathbf{\Sigma}_{spread}$ as an additional motion component. It was incorporated by adding normally distributed deviations (with the corresponding variances) to each particle in every motion update step. Equation (2) was replaced by (9):

$$\bar{\boldsymbol{x}}_t^{[m]} = f_m(\boldsymbol{x}_{t-1}^{[m]}, \Delta v_{odom}, \Delta\omega_{odom}) + \boldsymbol{w}_t + \boldsymbol{s}_t \tag{9}$$

where $\boldsymbol{s}_t \sim \mathcal{N}(\bar{0}, \mathbf{\Sigma}_{spread})$. Appropriate values were empirically determined as 0.0005 m for the additional spreading in x and y direction and 0.0025 rad for the additional deviation of the heading angle.



(a) True positions of autonomous vehicle and parked cars



(b) Alignment error: The algorithm matches the position of the parked car to the slot boarder line. The resulting position estimate of the autonomous car is misaligned respectively.

Fig. 5. Parking position evaluation: example of parking situation that leads to misalignment

E. Reduction of computation time

Besides the implementation of the semi-static object based localization, the real-time capability of the position filter was also an important criterion of this work. Project internal specifications required a cycle time of less than 50 ms.

The most straightforward solution for reducing computation time was to reduce the number of particles. Pretests showed that the localization accuracy during run-time did not considerably improve for configurations of more than 75 particles. Fewer than 75 particles resulted in failed localizations. Thus, the number of particles was initially set to 300 and was not allowed to drop below 75 during operation.

Another approach to reduce the computation time was based on the reduction of the number of laser scanner layers. The preprocessing and computation of sensor data can be accelerated by limiting the selected layers and thereby, limiting the sensor data points. Hence, two methods for decreasing the computation time were implemented: a KLD (Kullback-Leibler distance) sampling controlled adaptation of the number of particles and the reduction of a number of laser scanner layers.

1) Adaptation of the number of particles via KLD-sampling: A high number of particles was only necessary for the initialization task. Unless the car drove into an unknown area and had to be relocated again, a much lower particle number was sufficient to maintain the correct position. Therefore, KLD-sampling (10) was used to adapt the number of particles M in each cycle [11], [7].

The KLD algorithm discretizes the map of the environment into a three-dimensional state space. Initially, every grid cell is supposed to be empty. In a loop through all the particles, each particle is set to its corresponding cell. Every time a particle hits a so-far empty cell, a parameter k is increased by one. Thus, k represents the number of grid cells covered by at least one particle. We also employ two error bounds ϵ and δ. $1-\delta$ is the probability that the maximum error, which exists between the true posterior and the sample-based approximation, is less than ϵ. As a result, $z_{1-\delta}$ can be taken from standard statistical tables and represents the upper $1-\delta$ quantile of the standard normal distribution. The number of particles M needed to fulfill the predefined error bounds (ϵ = 0.2, δ = 0.9) is calculated in each iteration by (10). The resampling process continues until M particles have been created.

$$M = \frac{k-1}{2\epsilon} \cdot \left(1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} \cdot z_{1-\delta}\right)^3 \quad (10)$$

2) Reduction of the number of layers: Initially, four layers were used for the localization. Our algorithm allowed the selective use of either only a single layer or a combination of layers. Using fewer than four layers resulted in fewer sensor points. This could result in worse localization accuracy, but at an improved computation time. In order to overcome possible errors induced by the reduction of layers, an alternating layer approach was implemented [7]. Thereby, in every cycle only the sensor points of one layer were obtained. The four layers were alternately selected in every iteration, so that after four cycles all layers have been obtained once, but the computation time remained low.

III. Evaluation

A. Necessity of semi-static object modeling

One goal of the evaluation was to show the necessity of explicitly defining semi-static objects in the indoor localization task. Therefore, the modeled map was modified by removing all semi-static objects (e.g. Fig. 3 without the grey shaded parking slots). Our experiments showed that localization only based on static objects is not possible. The laser detection of parked cars was drawn to walls in situations when only parked cars were detected but no parked cars were modeled in the map. The resulting alignment error could not be corrected by the filter and the localization task failed.

TABLE I
EVALUATION OF COMPUTATION TIME OF DIFFERENT APPROACHES

approach	computation time [s]
basic	36
particle adaptation	15
layer reduction	22
full system	10

TABLE II
EVALUATION OF ACCURACY OF DIFFERENT APPROACHES

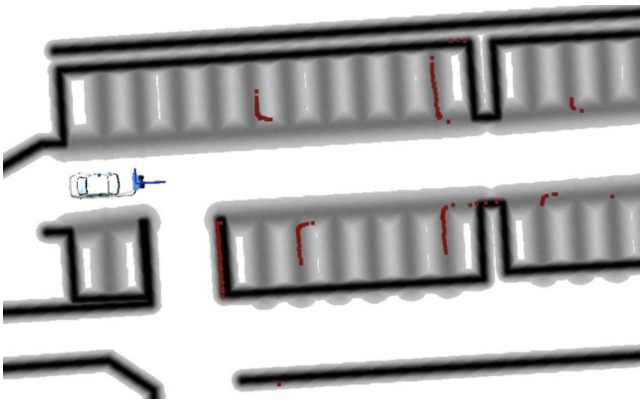
approach	abs. position error [m]	angular error [°]
basic	$\mu = 0.31 \mid \sigma = 0.14$	$\mu = 0.86 \mid \sigma = 0.83$
particle adaptation	$\mu = 0.33 \mid \sigma = 0.16$	$\mu = 0.91 \mid \sigma = 0.82$
layer reduction	$\mu = 0.30 \mid \sigma = 0.14$	$\mu = 0.95 \mid \sigma = 0.84$
full system	$\mu = 0.33 \mid \sigma = 0.17$	$\mu = 1.03 \mid \sigma = 0.98$

B. Accuracy and computation time of proposed methods

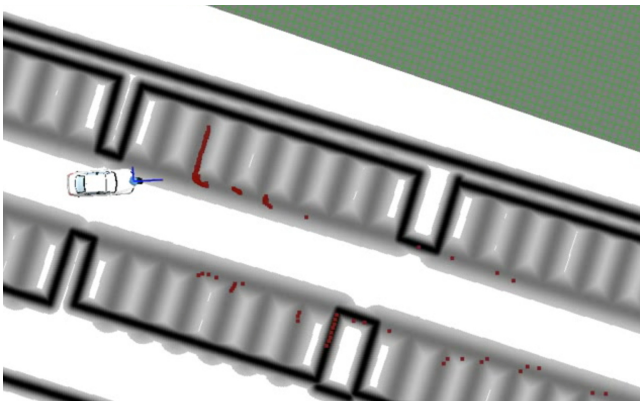
A quantitative evaluation of the algorithm was based on a test run that contained different scenarios of indoor parking maneuvers, including one of mainly detecting parked cars. We used two criteria for evaluating the quality of the filter: the computation time and the localization accuracy. We explicitly analyzed the impact of each design decision of our methods: basic approach, particle adaptation, layer reduction, full system. The basic approach is the described algorithm including the semi-static objects with a constant number of 300 particles. The particle adaptation version included the KLD approach with a possible reduction to a minimum of 75 particles. The layer reduction variant included the alternating layer approach with constant 300 particles. The final state analyzed the combination of all suggested methods.
The resulting computation time and the accuracy of each version of the localization algorithm are presented in tables I and II. The full system, including all filter enhancements, resulted in a absolute position root mean square error of μ_{pos} = 0.33 m and a heading angle root mean square error of $\mu_{heading}$ = 1.03 °. The standard deviation of the absolute position error was σ_{pos} = 0.14 m and $\sigma_{heading}$ = 0.98 ° for the mean heading angle error in a cycle time of 10 ms.

C. Video analysis

An example localization video can be found online at [12]. It shows the performance of the filter in a qualitative way. Example snapshots are shown in Fig. 6. The particles are visualized in blue while the heading direction is shown in black. The red dots and lines represent the obtained laser scanner data. First, the initial localization based on static and semi-static objects is demonstrated. We then perform a parking task. The next scene shows the robustness against non-mapped moving objects. The system only detects one passing car for a short period of time without losing its current position (playback in slow motion). The initial situation and a scenario of almost explicitly detecting parked cars are shown in Fig. 6a and 6b.



(a) The autonomous vehicle is shown at the estimated position in the known map. The laser scanner points are visualized according to this position and match some wall and some semi-static parking slot elements.



(b) The screenshot shows the scenario of almost explicitly detecting parked cars. The position of the vehicle remains robust in the estimated state and the laser scanner points accurately match the expected slot positions.

Fig. 6. Screenshots of the video analysis

IV. DISCUSSION AND FUTURE WORK

The analysis of the implemented system showed a rather high average offset of μ = 0.33 m. This was, amongst others, a result of the map creation uncertainty. The reference GPS uncertainty has to be considered, as well as the fact that the map was created manually during this work. However, a more important cause for the absolute offset was the unavailability of the exact parking position. The scenario of this evaluation was based on the visibility of many cars and few walls. The error in real indoor localization environments (which can usually rely on a setup with more walls than the outdoor evaluation testbed) is expected to be smaller but could not be more precisely evaluated due to the lack of reference data since there is no GPS reception inside the car park.
The evaluation showed failed localizations if only static objects were used and semi-static ones were ignored. This result could be expected in a challenging surrounding with many parked cars.
The results of comparing different approaches showed a considerable decrease in computation time when activating the KLD-sampling, the layer reduction and finally when both

methods were combined. Furthermore, the layer reduction (mean offset $\mu = 0.30\,\text{m}$) led to slightly better results than the basic approach (mean offset $\mu = 0.31\,\text{m}$). The difference of 1 cm can only be explained by one layer obtaining a wrong value in the basic case, which coincidentally did not happen in the alternating layer version. In general, a comparable accuracy can be assumed. More interesting is the accuracy result when using the KLD-sampling. In the KLD-sampling approach the error bounds ϵ and δ lead to a slightly higher uncertainty. This can be seen in the KLD-sampling approach itself, as well as in the final approach.

The proposed algorithm showed stable localization in inaccurately known environments by including semi-static objects in the map model. Thereby, a SLAM algorithm was not necessary. However, one could incorporate the concept of semi-static objects into a SLAM methodology. The static elements of the environment would not have to be considered in the map building process while the semi-static objects could be mapped by the SLAM algorithm in accordance to the known existence probability calculation of our approach.

V. Summary

In this work, a real-time localization algorithm for the application of an indoor parking system was implemented and evaluated. As major innovation, a map representation based on static and semi-static objects was introduced. Furthermore, different approaches for decreasing the computation time of the algorithm were compared. The evaluation showed that our suggested system was robust to localization failures even in scenarios of only detecting parked or moving cars. The position accuracy of the final evaluation results in a mean offset to the GPS reference system of 0.33 m and a standard deviation of 0.14 m at a computation time of 10 ms.

References

[1] R. Kümmerle et al., "Autonomous driving in a multi-level parking structure," in *Int. Conf. on IEEE Robotics and Automation (ICRA)*, 2009, pp. 3395–3400.

[2] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Int. Conf. on IEEE Robotics and Automation (ICRA)*, 2010, pp. 4372–4378.

[3] A. Ibisch et al., "Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors," in *4th Symp. on IEEE Intelligent Vehicles*, 2013, pp. 829–834.

[4] J. Bojja et al., "Indoor 3d navigation and positioning of vehicles in multi-storey parking garages," in *38th Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 2548–2552.

[5] A. Suppé et al., "Semi-autonomous virtual valet parking," in *2nd Int. Conf. on Automotive User Interfaces and Interactive Vehicular Applications (AutmotiveUI)*, 2010, pp. 139–145.

[6] B. Groh, "Localization in parking infrastructure," Diploma thesis, Pattern Recognition Lab, University of Erlangen-Nürnberg, Germany, 2012.

[7] S. Thrun et al., *Probabilistic Robotics*. MIT Press, 2005, vol. 1.

[8] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *4th Int. Symp. on Image and Signal Processing and Analysis (ISPA)*, 2005, pp. 64–69.

[9] F. Gustafsson, "Particle filter theory and practice with positioning applications," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 25, no. 7, pp. 53–82, 2010.

[10] G. J. Grevera, "Distance transform algorithms and their implementation and evaluation," in *Deformable Models*. Springer, 2007, ch. 2, pp. 33–60.

[11] D. Fox, "Adapting the sample size in particle filters through KLD-Sampling, *Int. Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003.

[12] "Localization example video," BMW framework, 2012. [Online]. Available (2014-03-01): `www5.cs.fau.de/files/parking-localization.mpeg`