

A CONCEPT FOR AN AUTOMATED APPROACH OF PUBLIC TRANSPORT VEHICLES TO A BUS STOP

C. Albrecht^{1,*}, S. Kraus², A. Zimmermann², U. Stilla¹

¹ Photogrammetry and Remote Sensing, Technical University of Munich (TUM), Germany - (christian.albrecht, stilla)@tum.de

² MAN Truck & Bus SE, Munich, Germany - (sven.kraus, andreas.zimmermann)@man.eu

KEY WORDS: Self-Localization, SLAM, Automated Driving, Graph Optimization, High Definition Map, Low-level Data Fusion

ABSTRACT:

This paper discusses the current methods for vehicle self-localization and compares previous findings to the use for urban public traffic vehicles. In specific, requirements for autonomous buses approaching a bus stop are defined. An autonomous system capable of reliable vehicle self-localization running in real-time in a city scenario shall be developed in a future work based on this paper. The comparison of filter-based estimation and graph-based optimization techniques shows that the latter suits the the automated approach to a bus stop in an urban environment the best. Based on these findings, a concept for self-localization of public transport vehicles equipped with a variety of imaging sensors with the help of a digital high definition map is presented. A current method is shown and a concept of improving the localization by inferring semantic information into landmark detection by low-level data fusion is provided. Validation and verification of the proposed fusion approach have to be carried out in the future, but a validation scenario is presented in this work.

1. INTRODUCTION

The ability to guide automated vehicles on well-structured streets and highways has been demonstrated and tested for several years now. (Geiger et al., 2013), (Cordts et al., 2016), (Fickenscher et al., 2018) In these situations, definite and characteristic features for localization are present and well-defined road markings ensure the lateral guidance of vehicles. In contrast to a highway scenario, the requirements for localization accuracy on the one hand and for resolution of high definition (HD) maps on the other are more critical in urban environments. Complex lane guidance as well as a multitude of ambiguous landmarks hinder an accurate estimation of the vehicle's pose. The recognition of traffic and particularly vulnerable road users are more difficult due to this ambiguity. Also the guidance of a bus in a city environment is a scenario only few work has focused on (Bouraoui et al., 2011) and mostly focuses on small transport vehicles instead of city buses.

To drive in urban environments, there are higher requirements for positioning accuracy, reliability and run-time. To be safe and stick to the defined road boundaries, a global positioning accuracy of 10 cm is required according to (Levinson and Thrun, 2010). As this accuracy was stated for cars, this paper compares existing approaches and gives an estimation for the needed accuracy for city buses. Due to direct interaction with pedestrians and other vulnerable road users, the autonomous system must run in real-time and should be fail-safe.

To cope with these conditions, modern cars are equipped with various sensors to perceive objects and measure the drivable area and its environment. A combination of different sensors specially designed for distance measurement, like LiDAR (Light Detection and Ranging), and ones for angular measurement, like cameras, ensure a complete environment recognition. In addition to mobile sensory installed on the vehicle, a pre-built HD map including e.g. pole-like objects,

road markings or curbs and their respective geometries as well as shapes of buildings shall be used for self-localization. Future work has to research, if special landmarks like bus bays or signalization of bus stops must be added and classified in a special way for the validation test case with a city bus or if they can easier be used for relative than for global localization.

Therefore, to address bespoke requirements for buses in urban environments, an autonomous system capable of reliable vehicle self-localization running in real-time in an urban area shall be developed. For recognition and association, the usage of machine learning methods will be evaluated in future work. The validation and verification of the presented method will be conducted by simulation and real-world scenarios in the field of automated city buses. The use case presented by this work is localization with a given HD map, so the focus of presented material is mainly on self-positioning in this map instead of a combined approach of localization and mapping.

A concept to achieve a reliable real-time localization in urban areas and a free space estimation based on the perceived environment data will be presented in this paper. By the evaluation of sensor fusion and usage of diverse sensor types, a positioning accuracy of 5 cm relative to the curb is expected to be reached. Based on this work, future investigations shall evaluate stages of data fusion at different perception levels to define the optimal fusion, either on raw data level or previously detected features or object. Global positioning error and accuracy of the localization system shall be compared to measurements of a differential global positioning system (DGPS) in combination with an inertial measurement unit, to validate the calculated accuracy against a reference system. Furthermore, the static environment perception shall be complemented with dynamic objects to analyze the current scene and possibly estimate hazards. Based on static and dynamic objects, the driving maneuver shall be adapted to the scenario. Object recognition is evaluated by placing objects or posing dynamical situations, where persons intruding the test vehicle's action space and real situations.

*Corresponding author

2. SYSTEM REQUIREMENTS

A broad overview over the challenges and open questions in the field of simultaneous localization and mapping (SLAM) is given in (Cadena et al., 2016) as they evaluate different methods by the main aspects "robust performance, high-level understanding, resource awareness, and task-driven inference". In addition to that, (Bresson et al., 2017) show the current state of the art focusing on accuracy, scalability, availability, recoverability, updatability, dynamicity for SLAM Methods used in autonomous driving. These review papers give a good introduction to the topic and so only a few selected papers that are important for our future work will be further investigated here.

The measures introduced in these review papers are taken into account for the development of localization methods used in this work and define a reasonable initial set of requirements. These will be evaluated with the used sensor and controller setup. To validate the localization quality of the developed method, the localization system shall be included into a self-driving test vehicle in form of a city bus. As there are requirements in Germany for local passenger traffic in terms of barrier-free usage of busses and vehicle control, these will be taken into account for the scenario requirements. The maximum distance where the entry into a bus is specified barrier-free is 5 cm and manageable is 10 cm according to (DIN 18040-3, 2014). This constraint is valid for the relative position from bus to the curb and does not necessarily have an impact on the global localization accuracy needed in city driving. As the bus is wider than a passenger car, the global accuracy must be constrained to fulfill the requirement of driving according to lane markings. Future work will research how accurate the position control system of the automated bus can follow a given trajectory and evaluate the combined error of localization and guidance to not drive off the marked lanes.

As an initial estimation, (Levinson and Thrun, 2010) assume a global positioning error of 10 cm to be feasible for a passenger car, whereas (Schlichting, 2018) was able to achieve a localization quality of 5 cm without any sensor data fusion. With these insights on current research, future work must check if on the one hand the assumption of about 10 cm global positioning accuracy also holds for city buses and on the other hand if this localization quality can be achieved within the given scenario.

3. LOCALIZATION METHODS

Typically, the localization task for mobile robots or automated vehicles should be divided into two distinct systems according to (Grisetti et al., 2010):

- A frontend that evaluates data and odometry inputs while knowing the according sensors and their respective measurement quality. The main task of the frontend is to present sensor-agnostic description e.g. in the form of object lists with covariance matrices to later be used in the backend.
- A backend to estimate at least the current position of the robot based on the given frontend output. Most commonly used implementations of a backend are a filter based approach like e.g. (Thrun et al., 2001) or (Steß, 2017) or a graph based optimization as in (Wilbers et al., 2019c) or (Gao et al., 2018) which are later explained in more detail.

Having this division of tasks, frontend and backend should be modularly developed to be easily exchangeable. Furthermore the interface must be defined beforehand to ensure this exchangeability. If a landmark based approach is used for localization, parts of the same frontend can be used for a particle filter backend as in (Steß, 2017) and at the same time for a graph-based solution developed in (Wilbers et al., 2019b).

Within the localization system, there is a differentiation of methods. Either a full SLAM method is realized to localize a robot in a map built up at run-time or there is already a given map to localize in. As the use case in this work is a single localization task with a given HD map, the focus of the presented material is on the self-positioning in this map instead of on a combined approach of localization and mapping. Further material is shown and, if appropriate, discussed and evaluated for usage in this work's problem case.

To give a proof of concept, the PhD thesis of (Schlichting, 2018) is further investigated, as a quite similar topic is presented. Laser scanner data is evaluated in an urban environment and the following approaches for feature extraction are presented and later used for localization to estimate the position error:

1. Landmarks (pole-like objects and planes) are extracted from the laser raw data and later used for self-localization. The descriptor from (Brenner, 2009) is used to describe poles in this case.
2. Neural networks are used to extract features with an autoencoder pattern. A global localization is achieved by matching a sequence of feature patterns with reference patterns in a sliding-window search.
3. A scan image is created from multiple laser scans and correlated with a feature map consisting of intensity and height values. This correlation is later included into a method for change detection.

Different state-of-the-art laser scanners are evaluated within each of the different methods to compare the resulting localization accuracies. The mean standard deviation with method (1) for poles and planes detected is 6 cm for a Valeo Scala and 8-10 cm for a Velodyne VLP-16. The root mean square error (RMSE) for method (2) is greater than 1m and not as accurate as the landmark based approach. Artificially built up scan images of method (3) are compared to the previous measurement runs and lead to a RMSE of 5 cm and even better in inner-city scenarios (3 cm). With the knowledge of this non-fused approach, the requirement of a positioning accuracy of 5 cm is feasible and can be applied for this work.

3.1 Landmark Detection

For self-localization, the environment has to be analyzed, labeled and classified to extract landmarks, i.e. static objects that are detectable both at run-time and marked in a map. The landmark detection is usually executed in the localization frontend to be exchanged easily later. Given there is an either built-up or previously constructed map, landmarks have to be recognized as objects and then co-registered within the map. For recognition and association, the usage of machine learning methods shall be evaluated in future work. Here we will only focus on landmark based localization, whereas there are papers using direct features of a camera images (Gao et al., 2018) or of a laser point cloud (Hungar et al., 2019). As these methods

require to build up a feature based map instead of a landmark map as stated above, these methods are currently not taken into account for further development of this concept.

As dynamic objects are not part of a map, these have to be excluded from the map-matching. They would only lead to wrong correlations or false-positive measurements. Furthermore, changes in the static map are to be detected and evaluated in accordance with the local measurements. This can be done with change detection methods as in (Gehring et al., 2017). These changes can then be used for a map retraining step. For the localization itself, neither dynamic objects nor the static changes will be taken into account, as only static objects can be co-registered in (fused) sensor data and HD map.

Landmarks have to be found in sensor images at run-time. As a laser scanner provides both, very accurate distance and angular measurements, each point's position in the scan can easily be calculated in a local coordinate frame. Geometrically clustering is enabled by these properties, so LiDAR is used in (Wilbers et al., 2019b) for detection of pole-like and plane landmarks. Landmarks to be detected can be classified as follows:

- Pole-like objects like posts, traffic signs, trees, traffic lights etc.
- Planes like building walls
- Corners describing the intersection of two planes
- Lane markings with complex geometries

The geometric reference of all these given landmarks is specific to the used localization framework. The framework used as a basis is further described in Section 4.2.3.

3.2 Filter-based approaches

A localization approach utilizing a filtering method as backend estimates the current position from a priori knowledge and current measurements. Within this solution, only the last pose is estimated what makes the approach slim and runnable at real-time conditions. (Grisetti et al., 2010) call a system capable of running in these conditions "online SLAM".

To give an example, (Dellaert et al., 1999) show an algorithm for Monte-Carlo localization (MCL) and combine the advantages of grid-based Markov localization with the efficiency and accuracy of Kalman Filter techniques. (Thrun et al., 2001) improve the MCL approach by using a particle filter for improved accuracy.

In another approach, (Levinson et al., 2007) build up a 2D occupancy grid storing infrared remittance information. They used inputs of a global navigation satellite system (GNSS), an inertial measurement unit (IMU), wheel odometry measurements, LiDAR data and the GraphSLAM algorithm by (Thrun and Montemerlo, 2006). Furthermore, they are using a particle filter to localize in that map. They re-use their 2007 method in (Levinson and Thrun, 2010), but utilize a probabilistic grid instead of an infrared remittance grid also using GraphSLAM from (Thrun and Montemerlo, 2006).

In contrast to a full SLAM approach, (Steß, 2017) use a landmark-based method with a pre-built map to only self-localize in it. With a Particle Filter, they use prior knowledge from previous pose estimations and then randomly sample a new set of particles, of which the sample matching the current landmark measurements the best is declared as the current pose estimate.

3.3 Graph-based approaches

Instead of relying on a filter-based localization, a graph-based approach optimizes multiple poses built up in a directed factor-graph at a time. Hereby, old measurements can be used to infer new knowledge also on old poses and landmark positions, which also improves the current pose estimation. Because not only the current, but all poses and all measurements are optimized, (Grisetti et al., 2010) call the principle "full SLAM", although it has to compensate the higher cycle time when running on a robot or automated vehicle. (Bresson et al., 2017) define a graph-based optimization as the most suitable variant of SLAM for autonomous driving. Not only the accuracy is higher in this approach, but also a deterministic method is preferable for functional safety in autonomous driving. With this method's improved accuracy, a map retraining without worsening its quality is possible by acceptance or rejection of measurements by cars with adequate sensory.

A tutorial on graph-based SLAM is given in (Grisetti et al., 2010) that explains the basic principles for graph-based localization. Implementing the proposed method, (Merfelds and Stachniss, 2017) infer a novel approach to build up a sliding-window pose-graph to estimate the pose history with multiple prior pose estimates in real-time (20Hz). These estimates originate amongst others from a particle filter based solution of the localization problem by (Steß, 2017). The introduced Landmark Based Localization (LBL) together with the PoseGraphFusion (PGF) will also be the base for our further work due to the following advantages:

- Various sensory as input handled by a localization frontend
- Optimization problem is solved as least-squares estimation with g2o presented in (Kümmerle et al., 2011)
- The state vector includes as well vehicle as landmark poses and vehicle odometry estimates in a directed pose-graph leading to a sparse information matrix for optimization
- Resource-adaptation and marginalization are shown in (Wilbers et al., 2019c)

(Wilbers et al., 2019a) state the differences of a Particle Filter based localization and a sliding-window graph based algorithm for localization for automated vehicles. In the paper, they are focused on online localization with the possibility to refine the map and also adding landmarks to the graph. Experiments were shown to estimate "(i) accuracy, (ii) adaptive behavior in terms of computational resources, and (iii) benefit of estimating old poses." An approach to use a PID Controller for limiting the size of graph is shown and validated. This ensures that the method fulfills run-time requirements. They propose to use a sliding-window optimization approach for localization because of the higher accuracy which is most beneficial for localization.

An algorithm for a pose-graph optimization for self-localization based on third-party HD maps is shown in (Wilbers et al., 2019b). Only the localization is focused instead of the mapping. The map will not be changed to avoid mapping errors due to localization errors. The method must run in (near) real-time, so multiple ways to constrain run-time are shown:

- sparsify landmarks and poses
- use sliding window (discard or marginalize)

The stages of the Graph Based Localization (GBL) are discussed in their paper and executed in the following order:

1. Landmark Detection
2. Local Association
3. Map Matching
4. Temporal Smoothing

First of all, the landmarks are detected in the local frame. This measurement is connected to the vehicle pose closest to the measurement time. Afterwards, multiple measurements of different timestamps are associated to one cluster with covariances to limit the numbers of optimization variables. With the use of the factor graph, landmarks that are not detected in every timestep from its first detection can be used. Unlike a standard ICP approach by (Besl and McKay, 1992), former estimates can improve the current estimation. These local clusters are then matched with the previously given high detailed map. Using temporal smoothing means re-adjusting previous map matches. I.e. associations of measurements and map landmarks are re-evaluated and changed or removed if the match was falsely set up. This not only improves the positioning quality but also makes the system aware of previous errors. In Section 4 we use this method and integrate a low-level sensor data fusion to further improve the localization accuracy to enable map retraining without worsening the map's quality.

4. PROPOSED METHOD

Facing the questions for autonomous driving in urban areas, in this section a method is presented to give an estimation of the needed and possibly reachable accuracy. First of all the usage of different sensor types is shown and the principle of low-level sensor data fusion is explained. The first milestone will be to evaluate if a low-level sensor data fusion improves the localization system's accuracy and which combination of sensors improves accuracy the most. To state an example, the combination of a angle-measuring sensors and a distance-measuring sensor is shown. Figure 1b and 1b illustrate this example with a centered camera and a 360° laser scanner on the side.

After this short introduction and evaluation of requirements for the sensor setup, the method to be used will be shown in more detail. As a basis, the sliding-window pose-graph optimization shown by (Wilbers et al., 2019b) is used for localization, transferring their knowledge on localization in passenger cars to the environment of autonomous bus driving. The current method will be described and the planned improvement inferred by our future work will be discussed.

4.1 Sensor requirements

The method by (Wilbers et al., 2019b) uses a graph-based optimization to estimate the current pose based on odometry and sensor measurements. To perceive the environment and detect landmarks, mainly the laser scanner is used for poles, surfaces and corners. The sensor setup used for our validation setup contains cameras for object detection with a field of view (FOV) of 120° and a top-view system with 190° FOV cameras. Additionally a 360° laser scanner and long-range radars will be used to perceive the environment in a redundant way. For

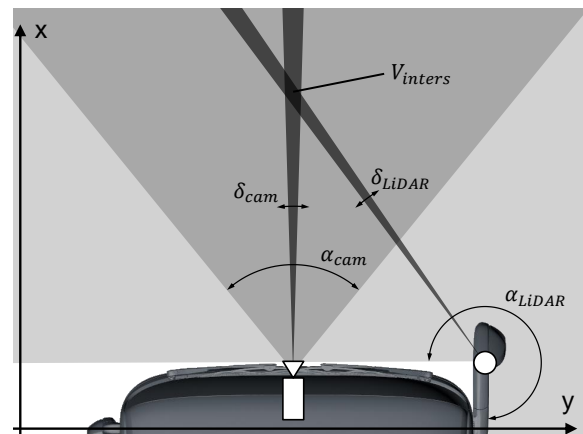


Figure 1a. Top-down view of an exemplary sensor setup at the test vehicle's front for low-level data fusion of camera (center) and LiDAR (right, at the mirror). The measuring area of the respective sensor is highlighted and their FOV is marked by an arrow.

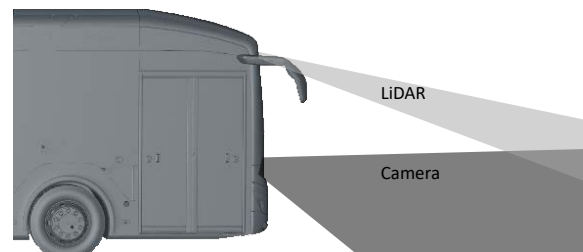


Figure 1b. Sensor setup at the test vehicle's front for low-level data fusion of camera (center) and LiDAR (top) as seen from the side

safety features, an ultrasonic sensor system will be used to avoid collisions with obstacles. Each of the sensors is at least able to provide an object candidate list.

To improve the amount of found landmarks and the robustness of the detection, a low-level sensor data fusion should provide a point cloud where each point is signed with a class label provided by a semantic segmented camera image. To get the information from the camera image to the laser point cloud there are some pitfalls:

- Camera and laser scanner are not in the same place and axis, so their relative position has to be accurately calibrated beforehand
- Different measurement principles of passive sensors like the camera and active sensors like LiDAR lead to the challenge that not every data point in the one sensor image has a corresponding data point in the other sensor's data.
- Some positions lay in the FOV of one sensor but not of the other because of geometrical restrictions given by the position as in Figure 1b.
- There might be occlusions of visible space because of the parallax effect although the position is in both FOVs.

To show the basic principle of low-level sensor data fusion, an exemplary sensor configuration between one camera and one laser scanner is provided. Figure 1a shows the setup as seen from the top and Figure 1b as seen from the right of our test vehicle. Figure 1a shows that a forward-looking camera should be combined with a laser scanner at the right outer mirror covering a 360° field that is obstructed by the bus itself in the back-left. In Figure 1a the laser scanner's remaining, unobstructed FOV is highlighted in gray. The camera image is facing forward and not obstructed. Points that can benefit from the sensor fusion need to be contained in the camera's FOV indicated by α_{cam} and the laser scanner's FOV indicated by α_{LiDAR} in Figure 1a.

An exemplary superposition is shown by two cones originating from each sensor with their respective angular resolution (e.g. δ_{cam}) and forming an intersection volume V_{inters} . With this info we can already see, that the scanner's whole point cloud cannot be classified by the camera at once. Because the LiDAR system will be attached to the outer mirror of the bus, it will be mounted higher than the camera and has a bigger blind area to the front than the camera. In this case, the camera's FOV is enabling feature detection in areas, that the laser cannot even see. Future work should evaluate, if the labeling in this field is feasible or can be masked to improve run-time.

4.2 Localization routine

To evaluate the presented research questions, we propose the concept shown in Figure 2. This method mostly uses the work presented by (Wilbers et al., 2019b) improved by a low-level sensor data fusion approach as red in section 4.1. The proposed method shall be developed for and tested in an automated test vehicle. The results shall be used to answer our proposed questions for accuracy and robustness. The developed system is further divided into localization frontend and backend as stated in section 3 and should be discussed briefly. Except the map retraining part, the original work is also presented in (Wilbers et al., 2019b).

The process is divided into the four parts

1. Data Capture
2. Low-Level Data Fusion
3. Localization Frontend
4. Localization Backend

where the data capture step is handling all the pre-conditions and the supply of respective sensor data and their pre-processing. Boxes represent outputs, while rounded boxes represent functions or processes. Stacked boxes signify multiple outputs over time, e.g. multiple images from the same camera, while single boxes show a accumulated state using multiple measurements from prior time steps. From that point, the automated system can use the given inputs for localization. Following, the second part is the earlier discussed low-level sensor data fusion using only raw inputs from the respective sensors. In this section we will focus on the fusion of a camera image and a raw laser scan, although other sensors can be used in a later approach. In addition to the previously provided sensor raw data, the fused output can then be used in the sensor-aware localization to find landmarks and build up a pose-graph as described in (Wilbers et al., 2019b). This graph

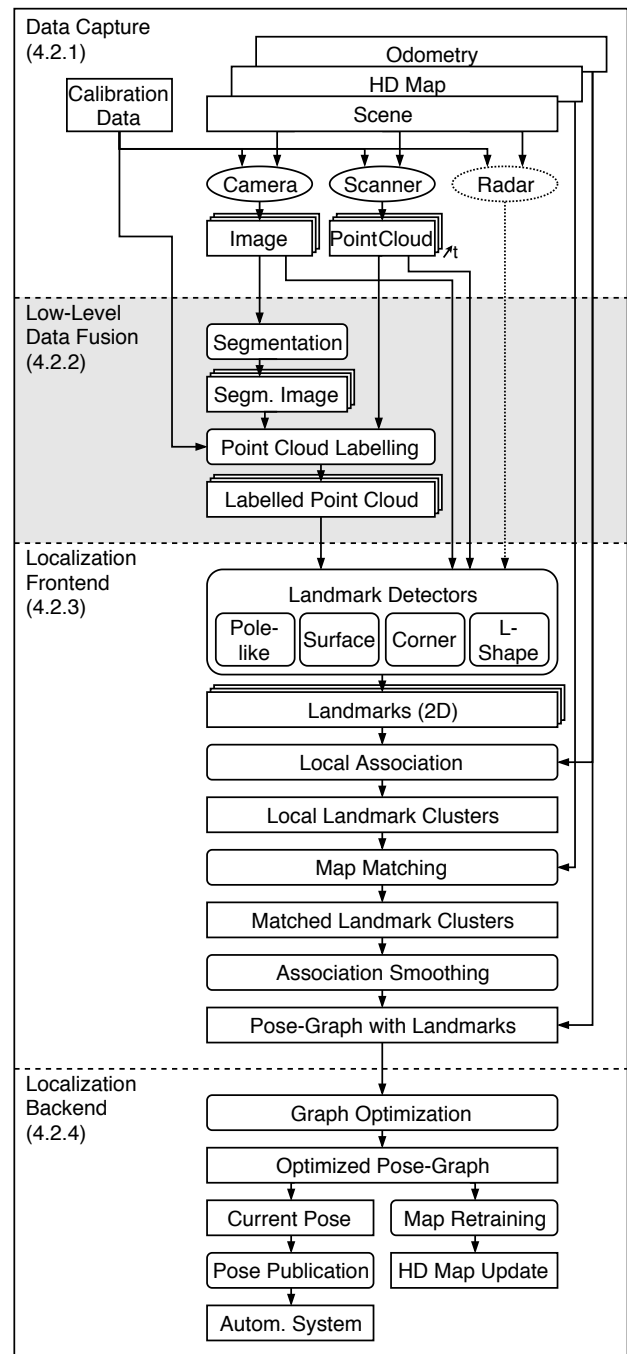


Figure 2. Abstract concept on vehicle self-localization of a city bus in an urban environment using a high definition map, diverse sensor types, low-level sensor data fusion and optimization of a pose-graph including landmarks.

This concept is based on the work by (Wilbers et al., 2019b), our impact is highlighted in gray.

representation is then transferred to the localization backend, where it is optimized with the g2o framework by (Kümmerle et al., 2011). The final optimized pose-graph is used to provide the latest pose to a module planning the trajectory for the next autonomous maneuver and to optimize the map and detect changes in the static environment.

4.2.1 Data Capture

The first part in the presented concept takes care of all needed conditions. Not only there must be a previously recorded and labeled high definition map, but also all sensors must be extrinsically and, if appropriate, also intrinsically calibrated and the calibration data must be available at run-time. In our use case, the used map will be a pre-built third party HD map where only a localization has to be conducted. Given all these pre-conditions, the scene is perceived by the sensors, mainly camera and laser scanners. Other sensors like radar or ultrasonic sensors will be added to the perception system if appropriate. As the error of standard GNSS systems is very high in urban areas, e.g. due to occlusions by high buildings, this sensor input is ignored. (Wilbers et al., 2019b) show that a fusion can also corrupt the estimation as a integration of very inaccurate GPS position worsens the final positioning error. Each module provides their raw data using the known calibration data in form of an image or a raw scan to the next module. In particular for the laser scanner output, it has to be defined, that the output is not yet transferred into a 3D or geodetic space, but still has the form of [vertical scan line, horizontal scan line, distance] to be used by the next functions. This comes in handy to use convolutional neural network (CNN) on laser raw data as on camera images.

4.2.2 Low-Level Data Fusion

The principle of low-level sensor data fusion is shown exemplary by the combination of camera and laser data. The approach of combining information of these sensors can be performed in multiple ways. The first option is to detect object with the camera and confirm or reject them by fusion with laser objects. This approach would be an object fusion instead of a low-level data fusion. On the other hand, each pixel of the camera image could be augmented with an according distance measurement from the laser scanner. In this case, there would be the possibility to overcome the problem of the camera not having depth information on single frame measurements.

As the LiDAR system is currently defined as the main sensor by (Wilbers et al., 2019b), we want to present a method projecting the camera image on the laser scan. With that we can use either pre-trained deep CNNs like for example DeepLabv3 (Chen et al., 2018) or a9 self-trained neural network better tailored to our use case to semantically segment the given camera image and then assign the found class labels to the laser scan points. To use the information from the camera in the laser point cloud, correspondences have to be found. For the labeling process, the calibration data of both sensors has to be used to know their rotation and translation in the vehicle coordinate frame. Due to the sensors not being located in the same position, situations may occur where in one sensor's scan, given points are visible that are obstructed by an object in the other sensor's data. In this case, a robust labeling cannot be performed and should be either prevented or solved by a model based approach. These approaches have to be defined and then developed by future work.

4.2.3 Localization Frontend

Being able to use all raw sensor inputs and preprocessed or fused sensor data, the localization frontend's main task is to find landmarks and link them to the current pose estimate. The procedure used is shown in (Wilbers et al., 2019b), but adapted to the use case of automated bus driving in urban areas. Landmarks are defined as representations of objects, that can be possibly detected by multiple sensors and are defined and

Sensor type	Function
Top-View Camera	Detection of lane markings and curbs
Front Camera	Detection of dynamic objects and lane markings
LiDAR	Pedestrian recognition
	Object detection
Radar	Free space estimation
	Detection of dynamic objects
Ultrasonic	Odometry input
	Near field observation and safety

Table 1. Different sensor types integrated to an autonomous test vehicle for this works validation scenario

recognizable in a HD map, in our case a 2D-representation with additional information, e.g. object type, height and skew of the respective road element. Landmarks that can currently be evaluated are pole-like objects as a single 2D-point, a surface represented as a line consisting of two 2D-point and more complex structures with multiple points. Each point is also attached with covariances representing their uncertainty. Found landmarks from different time steps are then locally associated in a vehicle-relative coordinate frame. Known odometry is used to propagate old measurements to the current time step and enable the association process. The found association clusters are then matched with the given HD map. The connections from local cluster to map landmark are afflicted with errors, that are minimized later. The matches are evaluated at each time step and can be re-organized if there is a major assumption error in the current graph. After that step, the pose-graph is build up with all pose-, odometry-, measurement- and matching-assumptions as described in (Wilbers et al., 2019b). This graph is then provided to the localization backend.

4.2.4 Localization Backend

The sensor-agnostic localization backend solves the given weighted least squares problem by minimizing all errors for the assumptions declared in the graph as in (Wilbers et al., 2019b). From this optimized pose-graph the last node is the estimate for the current pose and the backend provides this information to the trajectory planner module of the automated system. On the other hand, the graph also holds the positions of all detected landmarks over time with all corresponding map landmarks. With the localization system becoming more accurate, it might be beneficial to re-estimate the position of landmarks in the HD map. An approach for validation of landmark matches and map-retraining shall be developed in future work.

5. VALIDATION SCENARIO

The localization quality of the proposed method will be evaluated using a city bus equipped with diverse sensors as shown in Table 1, computing units and actuators for autonomous maneuverability. In the given use case, the test vehicle should approach a bus stop autonomously, swerve into the bay and come to a full stop safely. As there is no bus already set up with all the sensors and actuators needed for carrying out experiments, no validation of the states procedures can be shown here.

While in manual mode, the system has to recognize the bus stop and estimate the free space for the maneuver. After identification, the driving task is provided to the autonomous system, which localizes itself with the help of landmarks and calculates a collision-free trajectory into the bay. While executing the driving task, collisions with static and dynamic objects have to be prevented.

Furthermore, there are requirements for local passenger traffic in terms of barrier-free usage of buses and vehicle control. The maximum distance for which the entry into a bus is specified manageable for wheelchair users is 10 cm (DIN 18040-3, 2014). Amongst others, these requirements will be taken into account for the whole system in addition to the previously defined system requirements.

6. OUTLOOK TO NEXT STEPS

In this paper we presented differences of particle filter and graph-based localization methods and proposed a concept for a localization pipeline usable for public passenger traffic in urban areas. Future work wants to focus on different stages of sensor fusion (raw data fusion and/or object fusion) and evaluate when it is feasible to use a fusion approach or not. Another focus of this work should be the evaluation of map-retraining methods for a full-SLAM approach with continuous updates on the global map. For that the reliability of change detection has to be ensured and an update principle has to be presented. Not only the data structure for updates but also the impact on the map and other vehicles will be shown by this work.

REFERENCES

- Besl, P.J., McKay, N.D., 1992. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Bourauoi, L., Boussard, C., Charlot, F., Holguin, C., Nashashibi, F., Parent, M., Resende, P., 2011. An on-demand personal automated transport system: The citymobil demonstration in la rochelle. *In: IEEE Intell. Vehicles Symp.*, 1086–1091.
- Brenner, C., 2009. Global localization of vehicles using local pole patterns. *In: Lecture Notes in Computer Science*, Vol. 5748, Berlin: Springer.
- Bresson, G., Alsayed, Z., Yu, L., Glaser, S., 2017. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Trans. Intell. Vehicles*.
- Cadena, C., Carlone, L., Carillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.*, 32(6), 1309–1332.
- Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. *In: The European Conference on Computer Vision (ECCV)*.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. *In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dellaert, F., Fox, D., Burgard, W., Thrun, S., 1999. Monte carlo localization for mobile robots. *In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.
- DIN 18040-3, 2014. *Barrierefreies Bauen - Planungsgrundlagen - Teil 3: Öffentlicher Verkehrs- und Freiraum*.
- Fickenscher, J., Schmidt, S., Hannig, F., Bouzouraa, M.E., Teich, J., 2018. Path planning for highly automated driving on embedded gpus. *Journal of Low Power Electronics and Applications*.
- Gao, X., Wang, R., Demmel, N., Cremers, D., 2018. LDSO: Direct sparse odometry with loop closure. *In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*.
- Gehring, J., Hebel, M., Arens, M., Stilla, U., 2017. An approach to extract moving objects from MLS data using a volumetric background representation. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-1/W1, 107–114.
- Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The kitti dataset. *Int. J. Robot. Research*.
- Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W., 2010. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.*, 2(4), 31–43.
- Hungar, C., Köster, F., Jürgens, S., 2019. Ein Beitrag zur kartenbasierten Positionierung von Fahrzeugen mittels Mustererkennung in LiDAR-Daten. Technical report, *AAET 2019 Automatisiertes und vernetztes Fahren*.
- Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W., 2011. g2o: A general framework for graph optimization. *In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 3607–3613.
- Levinson, J., Montemerlo, M., Thrun, S., 2007. Map-based precision vehicle localization in urban environments. *In: Proc. Robotics: Science and Syst. Conf. (RSS)*.
- Levinson, J., Thrun, S., 2010. Robust vehicle localization in urban environments using probabilistic maps. *In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 4372–4378.
- Merfelds, C., Stachniss, C., 2017. Sensor fusion for self-localization of automated vehicles. *J. of Photogrammetry, Remote Sensing and Geoinformation Sci.*, 85(2), 113–126.
- Schlichting, A., 2018. Fahrzeuglokalisierung durch Automotive Laserscanner unter Verwendung statischer Merkmale. Reihe C, Verlag der Bayerischen Akademie der Wissenschaften, München.
- Steß, M., 2017. Ein Verfahren zur Kartierung und präzisen Lokalisierung mit klassifizierten Umgebungscharakteristiken der Straßeninfrastruktur für selbstfahrende Kraftfahrzeuge. Ph.D. dissertation, Gottfried Wilhelm Leibniz Universität Hannover.
- Thrun, S., Fox, D., Burgard, W., Dellaert, F., 2001. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1), 99–141.
- Thrun, S., Montemerlo, M., 2006. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *Int. J. Robot. Research*, 25(5-6), 403–429.

Wilbers, D., Merfels, C., Stachniss, C., 2019a. A comparison of particle filter and graph-based optimization for localization with landmarks in automated vehicles. *In: Proc. IEEE Int. Conf. Robotic Computing.*

Wilbers, D., Merfels, C., Stachniss, C., 2019b. Localization with Sliding Window Factor Graphs on Third-Party Maps for Automated Driving. *In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA).*

Wilbers, D., Rumberg, L., Stachniss, C., 2019c. Approximating marginalization with sparse global priors for sliding window SLAM-graphs. *In: Proc. IEEE Int. Conf. Robotic Computing.*