An Autonomous Forklift with 3D Time-of-Flight Camera-Based Localization and Navigation

Ulrich Behrje, Marian Himstedt, and Erik Maehle
Institute of Computer Engineering, University of Lübeck
Lübeck, Germany
{behrje, himstedt, maehle}@iti.uni-luebeck.de

***Abstract*—In this paper we present an autonomous guided vehicle (AGV) based on a forklift. While centralized transport systems are widely used in the industry, these systems are expensive to set up and inflexible with regard to changes in the schedule. We in contrast investigate an approach that uses a decentralized autonomous forklift that uses a 3D Time-of-Flight (ToF) camera as its navigation sensor and is not dependent on artificial visual landmarks. The capability to process transport orders and maneuver in confined space with the required accuracy has been successfully tested in two warehouse environments.**

I. Introduction

Automated goods transportation with automated guided vehicles (AGV) is ubiquitous in intralogistics. The automotive industry, known for its high manufacturing efficiency, relies on the benefits of this automation technology to achieve a high degree of production automation, but it is also common in many other manufacturing industries and non-manufacturing sectors like clinic logistics or warehouse logistics. In contrast to other material handling systems like belt conveyors, roller conveyors or elevators, AGVs allow for the transportation of a huge variety of goods and are more flexible with regard to transport routes. Early systems consist of an inductive track guidance system that utilises wires embedded in the floor, a central control station and safety equipment on the vehicle. Transport routes are limited to the available tracks of the guidance system that had to be planned in advance. Initial installation work as well as floor modifications that become necessary to adapt the routes to modified manufacturing processes entail high costs and reduce the overall flexibility of the system.

Contemporary solutions are no longer dependent on wires embedded in the floor but rely on laser rangefinders as a means to localize the vehicle. Reflective beacons serve as optical landmarks and improve the localization accuracy and the robustness. The navigation system still relies on predefined paths in order to move the AGV to the desired destination. Safety equipment like light barriers or contact strips are present on AGVs and a centralized control system coordinates and monitors all transportation processes. While the reconstruction measures in order to customize the transport routes to new processes declined, the transport system still has to be stopped and reprogrammed by the vendor, resulting in expensive economical costs for the initial set-up and subsequent modifications.

This work presents the development of an autonomous forklift that may operate autonomously in a warehouse environment and does not suffer from the aforementioned high installation costs. To achieve this goal, the AGV uses a novel on-board sensing system constisting primarily of a 3D time-of-flight (ToF) camera instead of a laser rangefinder for mapping and localization and is not dependent on artificial landmarks. In order to pick up a pallet, comparable accuracy to laser rangefinder-based systems in localization and control is needed. The narrowed field of view and reduced range compared with laser rangefinder imposes a challenge. The environment is mapped in an initial teach-in process perfomed by a forklift operator. The transport routes for the autonomous operation are not planned in advance, but calculated during the operation and allow to transport pallets between arbitrary locations in the test environment, taking obstacles that were added after the set-up phase into account. Also, this new system is not dependent on a centralized control system. All localization and navigation algorithms are calculated locally on the vehicle. As a result, no reconstruction measures are needed and the AGV may be used in new or modified environments at high flexibility. Basis for the demonstrator is the commercially available forklift ETV 216 built by Jungheinrich AG. It is modified with additional sensors, computers, and safety equipment to avoid accidents. Several tasks are defined as use-cases that the demonstrator shall be able to fullfill: It shall be able to pick up and put down a pallet from a rack as well as from the floor.

II. Related Work

Visual navigation has drawn particular interest by commercial projects as, for example, Google Tango [1], NAV-VIS [2] and SLAM-Tec [3]. Recently an automous forklift has been announced by Seegrid which is based on stereo cameras [4]. However, the navigation of the majority of AGVs in warehouse environments still relies on 2D LIDAR sensors. For more than a decade, laser-based localization in such environments has been carried out using artificial markers. In recent years, we can observe an increased amount of applications omitting the use of these markers and instead solely utilizing the range data for position estimation. In the PAN-Robots Project [5] an onboard system with an omnidirectional stereo camera and 2D safety laser scanners is used for providing 3D surrounding perception including object detection, tracking, classification and environment representation. Furthermore, there is an infrastructure-based environment perception system at intersections. A centralized control system allows operating a fleet of AGVs in a warehouse. The position estimation is carried by matching 2D LIDAR data to a prior map without any artificial markers. Also the academic research on AGV applications mainly focuses on 2D LIDAR sensors. Röwekämper et al. propose to combine adaptive Monte-Carlo localization (AMCL) and scan-matching based on Iterative Closest Point (ICP) [6]. While AMCL enables an efficient approximation of the robot pose w.r.t. to a map with coarse resolution, ICP further accomplishes a fine-positioning w.r.t. reference scans at predefined target locations. The authors demonstrate a significant increase of the position accuracy in a number of experiments. Similarly, Vasiljevic et al. present a localization system for AGVs based on AMCL and ICP-based scan-matching [7]. It is shown that a successive fine-positioning based on Fourier transforms further improves the positioning accuracy down to a few centimeters [7]. It can be observed that the majority of state-of-the-art navigation systems for AGVs [7], [6] rely on the use of laser rangefinders in conjunction with the SLAM algorithm GMapping [8] for generating initial maps. Graph-based approaches to SLAM have been investigated for the last years and meanwhile have become the means of choice in robotics [9], [10]. While a significant number of SLAM approaches working with RGB-D cameras [11], [12], [13] were proposed in the literature, there exists only a small number that can be applied straightaway to solely range data, as for example, obtained from laser range finders and ToF cameras [14], [15], [16]. In addition to that, there have been published numerous vision-based approaches, either enabling a metric [17], [18], [19] or a topological localization [12], [20] using monocular [21], [18], [20], [12], stereo [19] and omnidirectional cameras [17]. However, the position accuracy of these hampers the use for automated warehousing applications unless additional sources are employed for fine-positioning. Gadd and Newman suggest the use of appearance-based place recognition and visual odometry for autonomous navigation in warehouses. The authors benefit from long-term experiences in visual navigation for outdoor environments which can also be transferred to intralogistics applications. Their paper includes teach-and-repeat experiments in a real environment, however, the navigation system is not employed for uses cases such as approaching and picking up pallets from the ground or rack. Today there exists numerous solutions for the localization w.r.t. a prior map using monocular cameras, stereo cameras and LIDAR data in academic research and commercial applications. The majority of AGV navigation systems still rely on 2D LIDAR sensors. To our knowledge, there has not been published any research or commercial product on the use of 3D ToF cameras as the primary sensor for navigation and localization of AGVs.

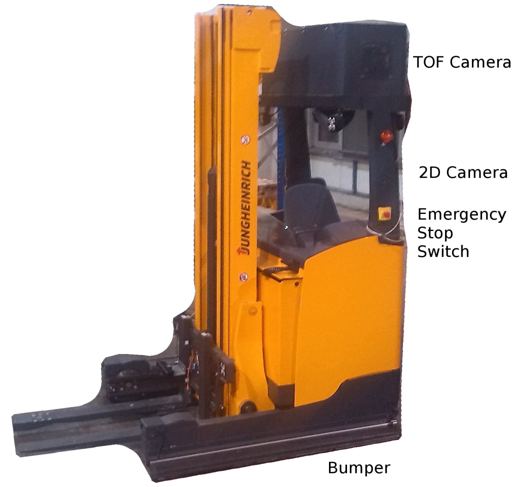


Fig. 1: The customized forklift based on an ETV 216.

III. System Architecture

An industrial 3D ToF camera (Basler tof640-20gm_850nm) is used as primary sensor for localization and navigation. It is installed at the top front of the forklift looking in the forward driving direction (Fig. 1). A secondary 3D ToF camera of this type is installed in the tip of a fork and is used solely to detect a pallet pose of a pallet in front of the forks with a template matching algorithm[1]. An Ethernet local network interconnects the cameras and several PC-based control computers located in a box on top of the forklift. One PC hosts the Automation interface with access to the vehicle CAN bus on one side and an Ethernet connection on the other side. The Drive controller is an industrial control computer that forwards drive commands to the automation interface. The HMI hosted by another PC uses speech and gesture recognition to generate user commands. One more PC is used for the localization and navigation module. Also paths are generated there and forwarded to the drive controller. In the demonstrator truck used in our experiments notebooks were used as PCs for their small size and ease of use. The communication between the several modules is done with a self-developed communication framework. Inside the localization and navigation module, the robot operating system (ROS) has been used.

A. Human machine interface

The human machine interface, which is described in [22], is realized by gesture and speech recognition utlizing the ToF camera and a wearable microphone. Is is developed in a companion project.

B. Automation interface

The Automation interface realizes the interface for the drive controller to the drive of the forklift. It is a newly developed module built on an industrial PC with CAN Bus controller which keeps intellectual property regarding the CAN messages shielded.

[1]This is based on a partner's project deliverable and not presented here.

C. Drive controller

The Drive controller is used to drive the AGV along a given path to the destination. It is based on a commercially available product and adapted to the needs of the AGV during the term of the project. It has originally been developed for the commercially available transport vehicle KATE [23] and had to be modified due to the different kinematics and dynamics of the much heavier forklift. The system relies on pre-defined paths that are defined in a global coordinate frame. It needs the current vehicle pose in the global frame as input as it has no integrated localization tools. A path consists of a series of waypoints and associated speed values. The drive controller leads the AGV along such a path, but especially in curves some deviations from the waypoints are to be expected. A path can be activated, when the current pose is identical to the starting point and orientation of the next path. As the AGV must be able to drive to arbitrary destinations, the possibility to program new paths during operation is heavily utilized.

D. Safety controller

A safety controller based on a PLC is used to avoid accidents during the development and testing phase. The list of safety sensors includes several emergency stop switches on the vehicle, a remote emergency stop switch, bumper switches and a SICK safety laser scanner. The laser scanner is not utilized by the localization and navigation module and has been covered during the final tests.

E. Localization and navigation module

The localization and navigation module (Fig. 2) is running on a PC with connection to Ethernet to get camera images and to interact with the other modules. It has several responsibilities:

1) Creation of an initial map during the set-up phase, including semantic information.
2) Localization in this map in normal operation.
3) Remembering of new obstacles in the map during operation.
4) Looking up the destination in semantic data and calculate a feasible path from the current pose to the destination.

After arrival at the destination, the second planning phase is active: Plan locally, without using the global map, only using the pallet detection module in order to accurately position the AGV in front of the pallet. During this phase the localization data sent to the drive controller is not based on the output of the camera-based localization module which generates a global pose but based on odometry measurements and the local pose of the pallet. This module also includes the vehicle control system that does contingency handling. Expectable problems arise due to unmapped shelf locations or unreachable destinations. The vehicle control system is beyond the scope of this paper and hence not further detailed.

The collision avoidance module is active when the AGV is driving in forward direction and avoids collisions by reducing the vehicle speed in case an obstacle is detected near the planned path. This allows to handle temporary obstructions caused by moving objects like other AGVs gracefully. If an obstacle does not give way, the path planning module is triggered to find an alternative route.

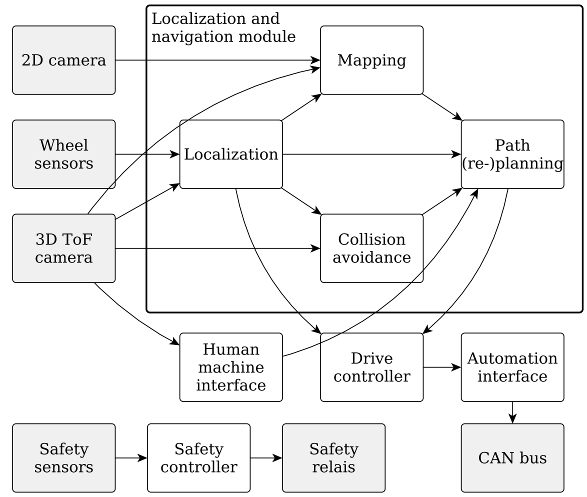


Fig. 2: Schematic overview of the system architecture.

IV. Preprocessing

A. Generating 2D Range Scans from Depth Images

Our localization and mapping algorithms require 2D range scans. This reduction allows a better runtime performance. Also it is not a restriction for the navigation since our approach addresses warehouse environments providing a planar ground. We therefore transform the input point cloud C into a 2D polar representation, with $p^{(k)} \in C$ as follows:

$$\begin{pmatrix} \theta \\ \rho \end{pmatrix}^{(k)} = \begin{pmatrix} atan2(p_y^{(k)}, p_x^{(k)}) \\ \sqrt{(p_x^{(k)})^2 + (p_y^{(k)})^2} \end{pmatrix} \tag{1}$$

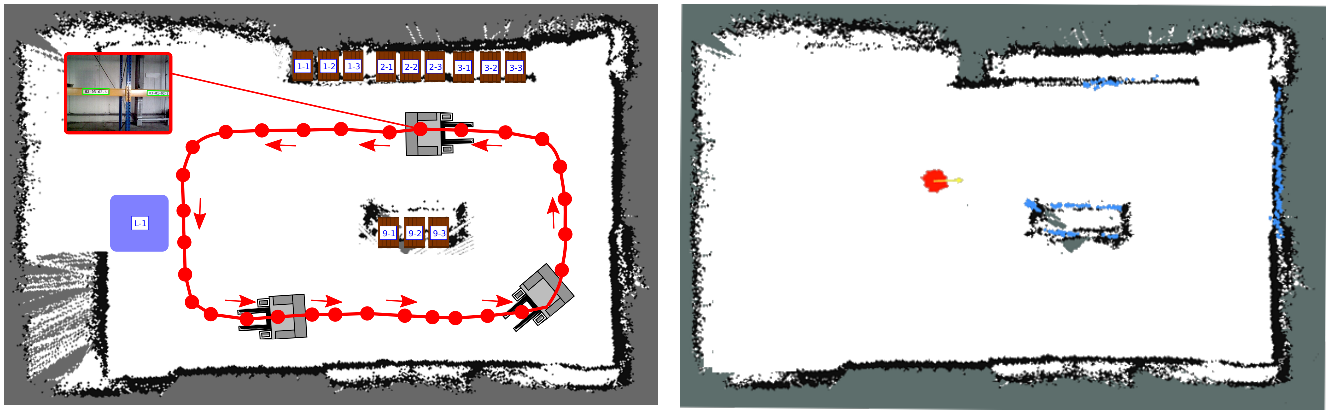
with $\theta^{(k)}$ and $\rho^{(k)}$ denoting the bearing and the range respectively for the k-th point. For each column of the input depth I_{dep} we select the minimal range value $\rho(k)$ being in the height range $z_{min} \leq z^{(k)} \leq z_{max}$. As a result of this step, we obtain a 2D range scan s with the number of measurements being equal to the number of columns of I_{dep}.

V. Mapping

The presence of a prior map of the environment is a fundamental precondition for the autonomous operation of our AGV. Therefore we manually steer the vehicle within the operation environment with the collected sensor data being used to generate an occupancy grid map (Fig. 3a) [24]. In addition to that we incorporate semantic information which is automatically recognized and can also be supplemented by the human operator. This initial mapping phase is further referred to as teach-in process.

A. Pose Graph SLAM

The availability of a map of the environment is a fundamental requirement of our system as it is used by most parts of our navigation stack. Since we focus on the application of



(a) SLAM path with map and semantic annotations.

(b) Map-based localization using AMCL and ToF camera data.

Fig. 3: This figures illustrates results obtained using mapping and localization algorithms. We utilized SLAM for building an initial map which is supplemented by semantic information of the operation environment (a). The resulting SLAM path (red), the detected rack signs (blue text on pallets) and a taught reloading area (L-1, blue box) are plotted on top of an occupancy grid map which is generated using the ToF camera data. An example of the localization with respect to a prior map based on AMCL using the ToF camera data is shown in (b). Particles approximating the vehicle's pose (red), the estimated pose (yellow) and the 2D range scan (blue) projected from the ToF camera data. Note the large field of view of the camera which is not yet common for ToF cameras.

AGVs in warehouses it is inevitable that mapping algorithms are able to cope with large-scale environments. Therefore our system makes use of graph-based simultaneous localization and mapping (SLAM). In particular, the graph's vertices refer to robot poses x_i while edges describe the spatial displacements of consecutive poses x_i and x_{i+1}. A robot's movement from the discrete state x_i to x_{i+1} is modeled as action u_i:

$$x_{i+1} \sim \mathcal{N}(f(x_i, u_i), \Sigma_i) \tag{2}$$

with Σ_i expressing the covariance matrix for the transition of state x_i to x_{i+1}. In order to account for the pose drift accumulated over time by odometric measurements, loop closures are incorporated as well. Again, we model the spatial displacement of a loop closure from x_i to x_j by an action u_{ij}. We can postulate the following about the state x_j:

$$x_j \sim \mathcal{N}(f(x_i, u_{ij}), \Lambda_{ij}) \tag{3}$$

with Λ_{ij} denoting the covariance matrix for loop closure from state x_i to x_j. The maximum a posteriori is estimated given all states X and actions U which results in the optimized poses X^*. This, in turn, is utilized to derivate a least-squares expression of our SLAM problem being solved using Gauss-Newton:

$$X^* = \operatorname*{argmin}_X \sum_i \left\| e_i^{odo} \right\|^2_{\Sigma_i} + \sum_{ij} \left\| e_{ij}^{lc} \right\|^2_{\Lambda_{ij}} \tag{4}$$

Here, $e_i^{odo} = f(x_i, u_i) - x_{i+1}$ refers to the optimization term of consecutive poses while $e_{ij}^{lc} = f(x_i, u_{ij}) - x_j$ expresses the loop closure term. It is important to keep the sparse structure of graph in order to enable high performance for the optimization at the runtime. This entails that loop closure candidates have to be carefully selected. Due to the limited field of view and operating range of the utilized ToF camera, it is important to steadily detect and incorporate new loop closures, but avoid overwhelming the optimization. Our SLAM algorithm relies on a feature-based loop closure detection given a 2D range scan being constructed from a depth image. This is accompanied by a robust outlier detection based on switchable constraints [25]. For a more comprehensive derivation the reader is referred to our prior work presented in [16], [26].

B. *Semantic Annotation*

Our map is annotated by semantic labels which bridges the gap from map positions with metric coordinates and application-specific goals of our operation environment. This is necessary in order to enable our AGV to reapproach specific goals such as racks, charging stations or reloading areas. Each semantic annotation is associated with a triplet consisting of $< class, description, pose >$, with the pose being defined as a 2D position and associated orientation. The following classes are considered by our system:

1) Rack identification signs
2) Reloading areas
3) Charging stations

The classes (2) - (3) are provided by a speech recognition system of a human-machine interface [22]. The poses of these annotations are set to the current vehicle pose once the HMI speech command is received during a prior teach-in process. This entails that charging stations and reloading areas will consistently be approached with the same orientation as being initially learned. Rack signs of class (1), in contrast, are automatically detected by a custom rack label recognition system working on RGB images captured with the side camera (see Fig. 2). For the sake of completeness, the key components of this system are summarized in the following:

1) Region of interests are detected
2) Conversion from RGB to monochrome image
3) Local feature extraction using MSER
4) Hough transformation for line detection
5) Filtering lines in regards of plausibility for sign
6) Extraction of putative sign corner points
7) Estimation of a homography corner points
8) Transformation of the image content inside a sign
9) OCR for reading sign content
10) 6D pose estimation of the rack sign

At this stage of the project, we assume a predefined sign format in order to simplify the pose estimation for a monocular RGB camera. The utilized algorithms can be applied to any sign format. This also enables the use of structure from motion (SfM) methods to estimate the sign poses with the absolute scale being obtained from the forward-facing ToF camera and the odometry. However, we have not yet investigated this. The poses for rack sign annotations of the map are obtained by transforming the observed rack signs (known real-world dimensions) from side camera's to vehicle coordinate system. Based on our SLAM algorithm we transform the pose from the vehicle's to the map coordinate frame which results in the final pose of the associated rack sign. Thanks to our semantic annotation, our AGV is able to obtain poses with respect to the map of specific goals such as racks, reloading areas or charging stations in the operation environment which, in turn, enables the AGV to reapproach them. The combination of automatic sign recognition and speech recognition for custom targets allows a substantial speed-up of the entire teach-in procedure since it avoids post processing the generated map.

VI. Localization

In order to enable our AGV to reapproach specific points within our prior map, as for instance racks or charging stations, a map-based localization is indispensable. Therefore a particle filter is used to estimate the vehicle's pose given a prior map of the operation environment. In particular we use adaptive Monte-Carlo localization (AMCL) which enables an efficient and robust tracking of multiple vehicle pose hypotheses (Fig. 3b) [24].

a) Motion Model: For our localization we utilize an odometry motion model as exhaustively described in [24]. In order to increase the accuracy of odometric motion estimation we utilize wheel encoders built into the non-driven loading wheels and the speed and angle of the driving wheel in a sensor fusion module based on a kinematic model of the forklift. Integration is done with the fourth-order Runge-Kutta method.

b) Observation Model: We make use of a likelihood field sensor model in order to obtain smooth observation likelihoods [24]. This model is also common for laser-based AMCL.

Similarly to our mapping, a 2D range scan as a top-down projection of the input depth image (see Section IV-A) is passed to AMCL. This allows an efficient localization in 2D using common methods. We observed that assuming a planar ground, a 2D localization is sufficient for the warehouse environment we are addressing in this paper. This assumption is also common for state-of-the-art laser-based positioning systems (e.g. [16], [27]. However, in contrast to these we are given a significantly reduced FOV ($85°$ vs. $180°$ and more) and limited operating range ($13\,\mathrm{m}$ vs. $30\,\mathrm{m}$ and more). First, this is being accounted for by the fusion of multiple wheel sensors. Second, we accept a reduced absolute localization accuracy by the use of an additional relative positioning system estimating the vehicle's pose with respect to a goal point in a local coordinate frame. In particular, the ToF camera data is used to estimate poses of palleted goods[2]. Thanks to this, we are able to fulfill drive requests in the presence of localization errors of about $0.5\,\mathrm{m}$ which will be shown in the experimental section. This a notable achievement as it enables to reduce the requirements of map-based localization systems and a transition from expensive laser-based systems to more affordable ToF cameras. In the context of this research platform we have also investigated localization algorithms with high robustness in a presence of significant changes in the environment by incorporating object recognition and exploiting prior knowledge [28], [29].

VII. Path Planning

Path planning is vital to the operation of an autonomous forklift. The task of path planning can be defined as follows: Given a robot in an environment with obstacles, find a feasible path from the current robot pose to the goal pose that avoids all abstacles [30]. In the case of an autonomous forklift, the environment can be considered a flat two-dimensional environment. A robot pose hence consists of the position defined by x and y coordinates and the orientation defined by the angle θ. A warehouse is a structured environment that typically consists of shelves, narrow corridors and open areas. The path planning takes place in the global two-dimensional map of the warehouse. In order to take a pallet from a shelf the forklift has to be positioned at the center in front of the pallet and perpendicular to its front. The warehouse map therefore must be annotated with the storage locations.

The path planning algorithm must take various constraints into account. First of all, the dimensions of the truck must be known and used by the path planning algorithm to prevent collisions with obstacles. Because the robot's footprint is not circular, the overrun surface increases in curves. The distance of the center of the footprint to obstacles must therefore be larger in curves and can be reduced when planning a straight line. The path planning must also handle the kinematic limitations of the nonholomonic drive, e.g. a path with sideways movement is not feasible. The forklift has two fixed driven wheels and one manoeuvrable castor wheel. Its kinematics can be modelled as tricycle kinematics. In case there is a new obstacle that is not present in the current map of the environment, the obstacle must be detected

[2]Note that this is based on another partner's project deliverable and hence not further presented here.

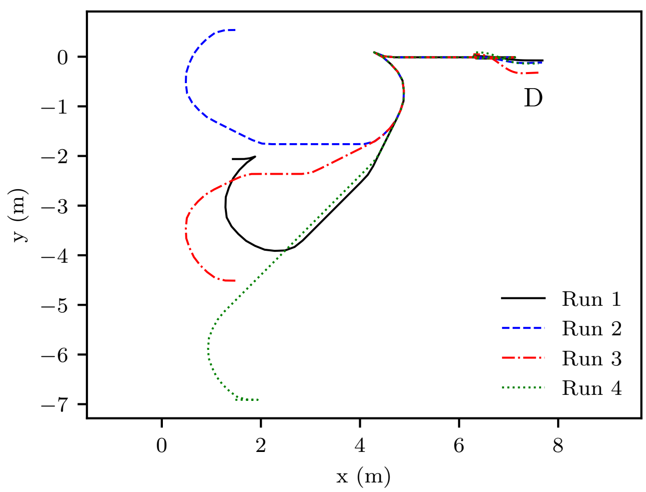


Fig. 4: This figure shows paths from four different starting locations to the same rack destination D that are generated and driven in order to take a pallet out of the rack. The segment part of each path accounts for inaccuracies in the global localization as well as deviations of the current pallet pose from the pose during the map creation. The pallet could be picked up in every run.

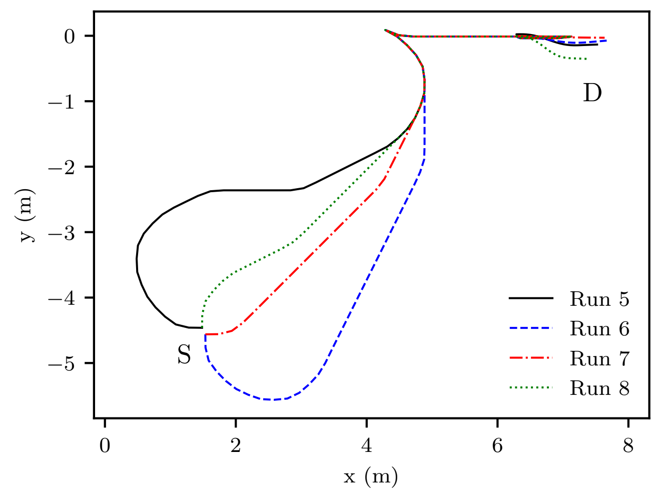


Fig. 5: This figure illustrates paths from the same starting position S with different initial orientations to the same destination D utilized to pick up a pallet. The pallet is taken from the rack in three out of four runs as the pallet detection did not properly identify the local pose of of the pallet in run 8.

when approached and an alternative path has to be found. The collision avoidance module checks the camera data in the front of the AGV and calculates the distance of obstacles to the path in the range of $2\,\mathrm{m}$. If there are obstacles present, the AGV is slowed down by resending the current path with a modified velocity profile until the vehicle comes to a full stop at a distance of $0.3\,\mathrm{m}$. As the interface to the driving controller is path-based it does not allow avoiding actions. If the vehicle stops, the path-planning is started again to find a feasible path around the new obstacle after a delay of $10\,\mathrm{s}$. If the obstacle moves away within that period the initial path is continued. This behavior allows for short stops when a person crosses the driveway while avoiding a costly replanning which often takes longer than the delay.

The localization works best when driving along a straight line and deteriorates when driving a curve. As the pose in front of a pallet has to be reached with high precision, the curvature of the last section of a path should be minimal. The fork can be moved sideways by $5\,\mathrm{cm}$, allowing for a maximal position error of $5\,\mathrm{cm}$. The requirements in regard to the maximal orientation error has been derived experimentally: if the final orientation has a deviation of more than $2°$ the pallet cannot be picked up.

To fulfill these requirements, the path planning task is subdivided into two sub-tasks. First, a path to a pose in front of the shelf is planned in the global map. The last section of this path is planned as a straight line, in order to achieve maximal accuracy. When this pose has been reached, the shelf or pallet in the shelf can be detected with the camera and its pose can be measured. The second path is planned as a bezier curve, which starts with a high curvature and merges asymptotically into a straight line. This ensures that the goal pose is reached on a path with minimal curvature and allows to compensate for unavoidable uncertainty of the pallet pose in the map.

An often employed approach in path planning is based on discretization of the search space into a grid and a grid-based search under the assumption that every adjacent cell can be reached from the current position at all times. This approach cannot be used in our case as the vehicle cannot drive around sharp corners. Another approach that has been employed in this project is the discretization of the state space into a subset of states and connections that represent feasible motions. This state lattice is a graph representaion of the problem and graph search methods can be used to find suboptimal or optimal paths [31]. The path planning software is based on the open source software Search-Based Planning Library [32]. If the state lattice is constructed using motion primitives that can be driven by the forklift, any solution found in the graph is guaranteed to be feasible. We utilized a set of 112 motion primitives with 16 different angles. Some primitives are directed in reverse direction, allowing for manoeuvering in narrow curves. Further, as the footprint is not round, the distance to obstacles is different on straights and in curves. Without the utilization of the exact footprint, a path through a narrow corridor could not be found.

VIII. Experiments

The presented system was evaluated in a number of experiments that took place in two test environments that resemble a warehouse. Both test environments are equipped with high shelves that carry pallets and lattice boxes. The size of the first test hall is $12\,\mathrm{m} \times 18\,\mathrm{m}$ and a single rack is positioned at one side. The tests conducted in the first hall shall show if the path planning module is able to calculate feasible paths

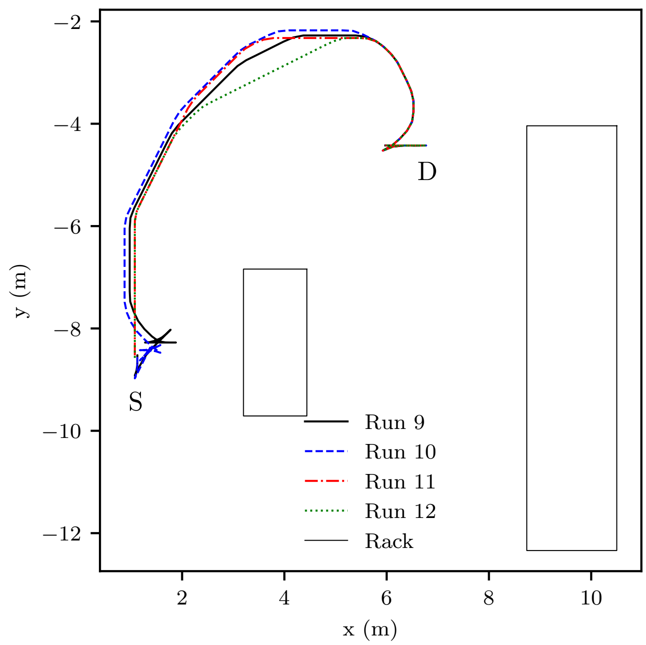


Fig. 6: In this figure the paths of the trial runs 9 to 12 are shown. The AGV has been located in a narrow corridor and is commanded to drive to a destination D outside of the corridor. It is reached without collisions in every run. In runs 9 and 10, the driving direction is changed multiple times during the necessary turn.

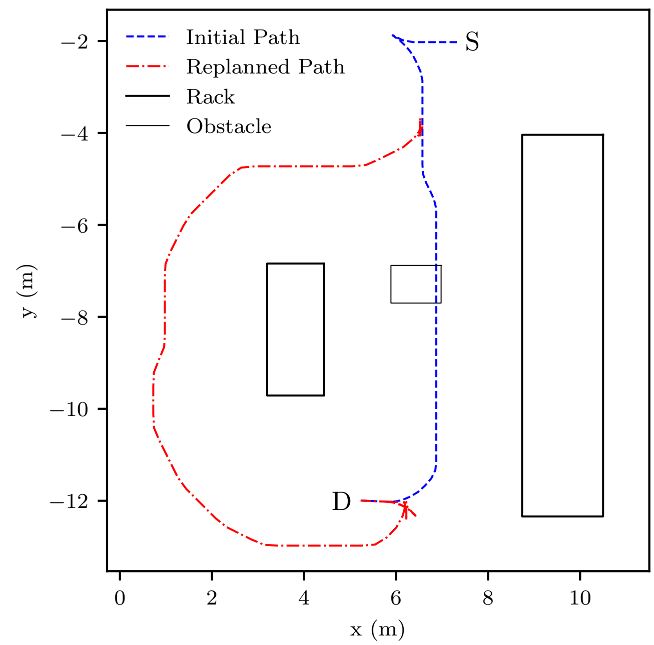


Fig. 7: This illustration shows the replanning of a path. The obstacle between the racks is not mapped and therefore not considered in the initial planning. An alternative route is found when the obstacle is detected after the initial turn.

and if the localization is accurate enough to allow for driving along these paths. As there was no external tracking system as ground truth available, multiple stock removal orders are carried out so the success rate of these tasks can serve as a useful reference. This method focuses on the area near the shelf where the highest precision is needed. A deviation from the planned path in large distance to the rack is acceptable in these tests, provided that no collision happens.

In the first experiment, a pallet is picked up out of the same storage location for four times. The starting position is changed in every run while the orientation was kept unchanged. The resulting paths are visualized in Fig. 4. The AGV reaches the same location in front of the shelf in each run, indicating that the paths are feasible and the localization works with sufficient accuracy. After run 2, the pallet has been put into the shelf with a slighty wrong orientation and a sideways offset. This unintentional deviation is being compensated by the final part of the path in run 3 and the pallet can be picked up in all four runs.

The second experiment is similar to the first one, except that the orientation of the AGV is changed while the start position is kept at the same location (Fig. 5). The AGV can successfully drive to the correct storage location at the shelf and pick up the pallet in three out of four runs in this experiment. In run 8, the AGV located itself not centrically in front of the pallet, but with a sideways mismatch of $40\,\mathrm{cm}$. Measurement records show that the pallet detection module had incorrectly located the pallet pose with an offset which is equal to half of the pallet width, which is the root cause for this failure. These two experiments were also carried out in the second test environment with similar results: The navigation module was able generate a feasible paths each time and the pallet could be successfully picked up in all tries.

The size of the second hall is $11.5\,\mathrm{m} \times 17\,\mathrm{m}$ and two racks are placed on the side and in the middle respectively to create two narrow corridors between the racks and between the center rack and the wall. In the third experiment (Fig. 6), the AGV is positioned in the corridor between the wall and the center rack to test if the generated paths leading to a goal outside of the corridor can be followed without collisions. As this experiment is conducted to test the maneuverability, the runs are stopped without picking up a pallet from the rack. In the runs 9 and 10, the vehicle must turn around in the corridor, in the runs 11 and 12, the AGV is oriented in direction of the goal. The goal is reached without collision in every run. During the turn the vehicle reverses 2 times in run 9 and 3 times in run 10. The minimal distance to the rack was $15\,\mathrm{cm}$.

The fourth experiment, shown in Fig. 7, is intended to test the capability of handling unmapped obstacles. A pallet is placed in the corridor between the racks while the AGV is oriented in direction of a wall and cannot detect the new obstacle. The initially planned path to a goal behind the obstacle leads as expected through the obstacle. This path is followed until the vehicle approaches the pallet when it stops and finds an alternative path through the other corridor. The goal is reached without colliding with the obstacle.

IX. Conclusion

In this paper we have presented an autonomous forklift which uses a 3D ToF camera as its primary sensor for localization and navigation. It can be used off the shelf for transport tasks in warehouses with no need for any artificial or visual landmarks. The localization and navigation software is based on ROS and implemented on several onboard PCs interconnected by Ethernet. For the generation of the initial map with semantic annotations a pose graph-based SLAM algorithm with teach-in is used. The localization within the map employs adaptive Monte-Carlo localization (AMCL) with particle filters. The path planning is tailored to the constraints of the forklift kinematics and the warehouse environment. Practical experiments with picking up and delivering pallets in a realistic test warehouse have shown that the required accuracy is reached for successfully fulfilling such tasks. Further work will consist of a better avoidance strategy for dynamic obstacles which is currently still based on stopping the forklift and replanning.

Acknowledgment

This work has been supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) under grant number 01MA13005D.

References

[1] E. Marder-Eppstein, "Project Tango," in *ACM SIGGRAPH 2016 Real-Time Live!*, ser. SIGGRAPH '16. New York, NY, USA: ACM, 2016, pp. 40:25–40:25. [Online]. Available: http://doi.acm.org/10.1145/2933540.2933550

[2] G. Schroth and F. Reinshagen. (2018) NAV-VIS. (Date last accessed 20-Sep-2018). [Online]. Available: https://www.navvis.com/

[3] (2018) SLAMTEC. (Date last accessed 20-Sep-2018). [Online]. Available: https://www.slamtec.com/

[4] L. Dormehl, "Self-driving forklift takes the human factor out of the warehouse work," (Date last accessed 06-July-2018). [Online]. Available: https://www.digitaltrends.com/cool-tech/seegrid-gp8-series-6-forklift

[5] L. Sabattini, M. Aikio, P. Beinschob, M. Boehning, E. Cardarelli, V. Digani, A. Krengel, M. Magnani, S. Mandici, F. Oleari *et al.*, "The PAN-Robots Project: Advanced Automated Guided Vehicle Systems for Industrial Logistics," *IEEE Robotics & Automation Magazine*, vol. 25, no. 1, pp. 55–64, 2018.

[6] J. Roewekaemper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard, "On the Position Accuracy of Mobile Robot Localization based on Particle Filters Combined with Scan Matching," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, Oct. 2012, pp. 3158–3164.

[7] G. Vasiljević, D. Miklić, I. Draganjac, Z. Kovačić, and P. Lista, "High-accuracy Vehicle Localization for Autonomous Warehousing," *Robotics and Computer-Integrated Manufacturing*, vol. 42, no. C, pp. 1–16, Dec. 2016. [Online]. Available: https://doi.org/10.1016/j.rcim.2016.05.001

[8] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Robotics and Automation, 2005. Proc. of the 2005 IEEE International Conference on*, Apr 2005.

[9] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A General Framework for Graph Optimization," in *ICRA*, Shanghai, 2011.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 3281–3288.

[11] J. Hartmann, J. H. Klüssendorff, and E. Maehle, "A unified visual graph-based approach to navigation for wheeled mobile robots," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1915–1922.

[12] M. Labbe and F. Michaud, "Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, June 2013.

[13] D. Galvez-Lopez and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[14] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient Sparse Pose Adjustment for 2D Mapping," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[15] E. Einhorn and H. M. Gross, "Generic 2D/3D SLAM with NDT maps for lifelong application," in *2013 European Conference on Mobile Robots*, Sept 2013, pp. 240–247.

[16] M. Himstedt, S. Keil, S. Hellbach, and H.-J. Böhme, "A robust graph-based framework for building precise maps from laser range scans," in *Proceedings of the Workshop on Robust and Multimodal Inference in Factor Graphs. IEEE ICRA*, May 2013.

[17] H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell, "Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT," *Robotics and Autonomous Systems*, vol. 54, pp. 758–765, 2006.

[18] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke, "Metric Localization with Scale-Invariant Visual Features Using a Single Perspective Camera," in *European Robotics Symposium 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 195–209.

[19] P. Elinas and J. J. Little, "σMCL: Monte-Carlo Localization for Mobile Robots with Stereo Vision," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[20] M. Gadd and P. Newman, "A Framework for Infrastructure-Free Warehouse Navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015, pp. 3271–3278.

[21] J. Lee, C.-H. Hyun, and M. Park, "A Vision-Based Automated Guided Vehicle System with Marker Recognition for Indoor Use," *Sensors*, vol. 13, no. 8, pp. 10 052–10 073, Aug 2013. [Online]. Available: http://dx.doi.org/10.3390/s130810052

[22] L. Overmeyer, F. Podszus, and L. Dohrmann, "Multimodal speech and gesture control of AGVs, including EEG-based measurements of cognitive workload," *CIRP Annals - Manufacturing Technology*, vol. 65, no. 1, pp. 425–428, 2016.

[23] Götting KG, "KATE Kleine automatische Transporteinheit," (Date last accessed 01-July-2018). [Online]. Available: https://www.goetting.de/dateien/downloads/2018-03%20KATE%20Flyer%20de%206seitig.pdf

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, 2005.

[25] N. Sunderhauf and P. Protzel, "Towards a robust back-end for pose graph SLAM," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 1254 –1261.

[26] M. Himstedt and E. Maehle, "Geometry matters: Place Recognition in 2D range scans using Geometrical Surface Relations," in *Mobile Robots (ECMR), 2015 European Conference on*, Sept 2015.

[27] C. Reinke and P. Beinschob, "Strategies for contour-based self-localization in large-scale modern warehouses," in *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 223–227.

[28] M. Himstedt and E. Maehle, "Online semantic mapping of logistic environments using RGB-D cameras," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, 2017.

[29] ——, "Semantic Monte-Carlo localization in changing environments using RGB-D cameras," in *Mobile Robots (ECMR), 2017 European Conference on*. IEEE, 2017, pp. 1–8.

[30] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[31] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, 2005, pp. 1–7.

[32] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.