DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

Integration and Evaluation of a Close Proximity Obstacle Detection for Mobile Robots in Public Space

Integration und Evaluation einer Hinderniserkennung für den Nahbereich mobiler Roboter im öffentlichen Verkehrsraum

Noel Blunder
Marko Thiel
Manuel Schrick
Johannes Hinckeldeyn
Jochen Kreutzfeldt

Institute for Technical Logistics
Hamburg University of Technology

To ensure safe operation of mobile robots in public spaces, the robots must be able to detect obstacles and identify non-traversable ground. However, there are relatively few suitable methods for mobile robots in urban environments available and equally few empirical data on such applications. This contribution identifies requirements for such systems, designs meaningful test scenarios, and evaluates an exemplary approach using a mobile robot platform with depth sensors. It is found that the method is suitable in principle, but that different roadway elements pose a challenge to detection. Corrupted sensor data due to tilting movements or overexposure also negatively affect the quality of obstacle detection.

[Keywords: mobile robot, obstacle detection, public space, traversability, urban test scenarios]

Um einen sicheren Betrieb von mobilen Robotern im öffentlichen Raum zu gewährleisten, müssen Roboter in der Lage sein, Hindernisse zu erkennen und unbefahrbare Bodenbereiche zu identifizieren. Es existieren jedoch nur vergleichsweise wenige geeignete Methoden für mobile Roboter in städtischen Umgebungen und ebenso wenig empirische Daten bezüglich solcher Anwendungen. In diesem Beitrag werden Anforderungen an solche Systeme identifiziert, aussagekräftige Testszenarien entworfen und ein beispielhafter Ansatz anhand eines mobilen Roboters mit Tiefensensoren evaluiert. Es wird festgestellt, dass diese Methode prinzipiell geeignet ist, jedoch verschiedene Fahrbahnelemente eine Herausforderung für die Erkennung darstellen. Fehlerhafte Sensordaten durch Kippbewegungen oder Überbelichtungen beeinträchtigen ebenfalls die Qualität der Hinderniserkennung.

[Schlüsselwörter: mobile Roboter, Hinderniserkennung, öffentlicher Raum, Befahrbarkeit, urbane Testszenarien]

1 Introduction

The detection of obstacles and thus impassable areas plays a key role in the development of perception systems for mobile robots. However, the specific requirements for such systems vary between different applications. Applications with predominantly flat floor surfaces, such as warehouses, allow relatively straightforward detection of obstacles and recognition of traversable areas. Obstacles are distinguished by the fact that they protrude from the flat surface of the floor. Mobile robots operating outdoors in public spaces often have to navigate uneven or sloped surfaces such as paved areas, gravel paths, and ramps. The uneven ground surface itself as well as the resulting relative motions between the robot and the ground plane render segmentation based on height thresholds difficult. In addition, certain parts of the ground itself might pose an obstacle if the surface is too uneven or too steeply inclined for a robot to travel on. Determining the traversability of ground surfaces in the robot's vicinity is essential to planning suitable driving trajectories. Urban environments present a particular challenge, as false detections can lead not only to damage or loss of the robot, but also to serious injury for affected individuals.

While obstacle detection and drivable area segmentation are common problems in the field of automated driving [1, 2], comparatively few solutions focus on compact mobile robots. Similarly, there are only few research studies and empirical data that address and evaluate the challenges of robotic applications in public spaces [3].

The objective of this work is to investigate which suitable approaches are available for the detection of obstacles in public spaces and which performance can be expected from these methods in practical real-world scenarios. For this purpose, a selected, open source available method is integrated into a delivery robot prototype from the current research project TaBuLa-LOG [4]. This robot is equipped

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

with four downward-facing stereo cameras on all sides to capture the ground surface in its immediate vicinity as 3D point clouds. Its control system is based on ROS (Robot Operating System), which enables straightforward integration of additional features. The entire system is then evaluated in various scenarios that replicate real-world challenges for such systems and address previously identified requirements. Those scenarios also include a combined logistics use case in which the mobile robot enters and exits an automated public transport (PT) shuttle via a ramp. This shuttle was also part of the TaBuLa-LOG project.

The main contributions are:

- Identification and definition of requirements for obstacle detection systems for mobile robots in public spaces or urban environments (Section 4).
- Derivation and development of meaningful, real-world test scenarios for such detection systems (Section 5).
- Trial and evaluation of a selected recognition/segmentation algorithm using empirical data from a mobile robot prototype with depth sensors in real-world test scenarios (Sections 6 and 7).

A short overview of relevant literature can be found in Section 2, with a focus on object detection approaches that are provided as open source software.

2 Related Work

Perception and interpretation of a vehicle's environment are some of the basic prerequisites for the safe operation of autonomous vehicles, especially in public traffic areas [5]. This creates a correspondingly broad field of research. Beyond ground-based applications, analogous systems are also being used in unmanned aerial and water vehicles [6, 7]. In this paper, the focus is on ground-based mobile robots.

2.1 Obstacle Detection for Mobile Robots

The existing literature on obstacle detection for mobile robots covers a wide variety of topics and approaches. This includes detecting obstacles in the sense of non-traversable areas. Published methods typically differ in the following characteristics:

- Task: object detection, segmentation
- Sensor type: LiDAR, RGB-D (e. g. time of flight; stereo camera), RGB (i. e. monocular camera), combination of sensors
- Operational environment: indoor, outdoor (urban, off-road)
- Employed method: geometric reasoning, machine learning/neural networks

The following publications were selected as representative examples.

Hua et al. [8] present a system for avoidance of small obstacles using RGB-D sensors and an encoder-decoder network based semantic segmentation. It is evaluated against indoor and outdoor data, including the *Citiscapes* dataset [9]. Using not a robot in the classical sense but an electric wheelchair Wang et al. [10] propose a self-supervised approach for deep learning-based drivable area segmentation. This also addresses the issue of limited datasets and training data. For recording data and evaluation, an RGB-D sensor is used. Pang et al. [11] present a method to detect obstacles in urban environments using a 2D LiDAR that is facing downwards. The method is evaluated using three scenes: a normal road, a school gate and a curved, sloped road segment with parked cars.

To the best of our knowledge there is no systematic evaluation of an obstacle detection method for mobile robots in a variety of urban scenarios. With the *SideGuide* dataset [12], however, there is a large dataset with RGB stereo images of sidewalks and various characteristic objects.

2.2 ROS-Compatible Open Source Implementations

In the context of this work, approaches that are available as open source implementations are of particular interest, as one of them will later be selected for practical evaluation on a robot prototype vehicle. Consideration is given to approaches that are already distributed as ROS packages as well as those that could be integrated into a ROS-based system.

One potential approach is the segmentation of a point cloud into traversable ground planes and obstacles. A ROS node that accomplishes this task is the *obstacles_detection* node developed by Labbé and Michaud [13] as part of the *RTAB-Map* ROS package. The segmentation of the input point cloud is based on the orientation of their surface normal vectors, a method already addressed in some of the earliest approaches to obstacle detection [14]. The output of the system consists of two point clouds, one containing traversable points and the other containing points classified as obstacles.

The *RTAB-Map* node uses functions of the *Point Cloud Library* (*PCL*) [15], which provides a multitude of functions for point cloud processing. The available *PCL* ROS package allows the use of various point cloud processing functions, such as filtering measurements based on their surface normal vectors. Accordingly, the *PCL* would provide the means to develop an independent approach to

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

obstacle detection instead of relying on existing ready-to-use solutions.

A Python repository that could be integrated into a ROS-based system is *GndNet* [16] and implements point cloud ground-level segmentation. The approach is similar to *RTAB-Map* in terms of the overall goal, but the segmentation of traversable planes and obstacles is based on a neural network and is optimized for LiDAR data.

Besides segmenting point clouds into ground and obstacles, another approach employs the projection of the point cloud into an elevation map. Elevation maps display the environment in form of a grid map where each grid cell is assigned a corresponding height. Based on this representation, impassable areas can be determined by calculating the slopes between neighboring cells. As the integration of obstacle detection into the navigation of a robot requires a real-time application, dynamic robot-centric elevation maps are particularly suitable. These record the robot's environment in a predefined radius and are updated periodically. ROS packages for robot-centric and real-time elevation mapping approaches are *ANYbotics elevation_mapping* [17] and *hector_elevation_mapping* [18].

Another approach worth mentioning is the *Move Base Flex* navigation stack developed by [19] as an alternative to the ROS standard navigation stack. *Move Base Flex* enables navigation based on 3D data as opposed to a two-dimensional cost map structure used in the standard navigation stack. As part of this project, a package called *mesh_navigation* was developed that enables navigation using elevation maps [20].

3 METHOD

In this section, the methodological approach of this work is explained in detail. The integration and evaluation of an obstacle detection system for mobile robots is therefore divided in requirements analysis, the development and recording of test scenarios, the selection, parameterization and application of an obstacle detection approach as well as the evaluation of the system's performance.

3.1 REQUIREMENTS ANALYSIS

The requirements analysis describes the definition of requirements to be met by a system for obstacle detection in the close range of a mobile robot operating in public spaces. The focus here is on four different influencing factors: (1) Requirements of the existing system, (2) general characteristics of obstacles to be detected, (3) environmental conditions, and (4) the robot application. The system-specific requirements include the software and hardware specification of the robot, to allow the selection of a compatible approach. The general characteristics of obstacles to be detected define geometric dimensions of objects that would be considered obstacles. Requirements resulting from the environmental characteristics of the robot are partly oriented on the city of Lauenburg/Elbe, which represents the operating area of the mobile robot within the TaBuLa-LOG project. The application-specific requirements result from the use case of a mobile robot operating in public areas in combination with public transport vehicles.

3.2 DEVELOPING AND RECORDING TEST SCENARIOS

Based on the requirements derived in the first step, the development of test scenarios follows. To be able to evaluate a given obstacle detection approach, a set of real-world test scenarios are developed that cover all analyzed requirements. During this process, the focus remains on universal applicability to enable the evaluation of further approaches in subsequent work. To be able to apply an obstacle detection approach to the test scenarios offline, they are recorded as ROS bag files, further ensuring exact reproducibility of the experiments.

3.3 SELECTING, PARAMETERIZING AND APPLYING THE OBSTACLE DETECTION APPROACH

After recording the test scenarios, an algorithm is selected that most promisingly fulfills the identified requirements. It is particularly relevant whether the investigated approach fulfills the system-specific requirements in terms of hardware and software compatibility and whether general obstacle detection can be realized in the sense of the task. The selected obstacle detection algorithm is then parameterized and applied to the test scenarios offline. The parameterization is performed with the help of a separate scenario that is independent of the test cases. Based on the general obstacle properties, different geometric shapes are 3D-printed, which are to be classified by the algorithm as obstacles or as passable, and the parameters are adjusted accordingly. The application of the algorithm to the test scenarios takes place in an offline testing environment, which is set up on a computer that matches the one installed on the robot. This enables an offline application of the algorithm to the recorded test scenarios.

3.4 EVALUATING THE SYSTEM'S PERFORMANCE

To ensure universal applicability and comparability, universal metrics for the evaluation of the test scenarios through success and failure indicators are identified. The detection rate is used as a metric for evaluating detection performance. This describes the percentage of detected obstacles by dividing the number of detected obstacles $n_{obstacles\ detected}$ by the total number of obstacles present in each scenario $n_{obstacles\ total}$.

$$Detection\ Rate = \frac{n_{obstacles\ detected}}{n_{obstacles\ total}}[\%]$$

A metric to ensure evaluation of the system in terms of error susceptibility is the mean absolute error (MAE) of

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

the system. In the context of obstacle detection, errors occur when an obstacle is not detected as such or, on the contrary, an obstacle is detected although there is none. These cases are called false negative and false positive classifications. Accordingly, the MAE results from the sum of false negative classifications $n_{false\ negative}$ and false positive classifications $n_{false\ positive}$, which are normalized regarding the total number of evaluation steps $n_{frames\ total}$. The MAE thus reflects the average number of misclassifications per evaluation step, where each sequential evaluation step analyzes the images of all four cameras at once. Accordingly, the total number of evaluation steps results from the scenario duration and the frame rate of the cameras.

$$\mathrm{MAE} = \frac{n_{false\ positive} + n_{false\ negative}}{n_{frames\ total}}$$

For each evaluation step, the number of obstacles to be detected, the number of fully detected obstacles, the number of partially detected obstacles, the number of undetected obstacles (false negatives), and the number of falsely detected obstacles (false positives) are thus collected to calculate the metrics.

Finally, the system is evaluated based on the results from each test scenario. The key performance indicators (MAE and detection rate) are used to evaluate the system's performance and to identify potentials and challenges that need to be overcome.

4 Requirements for an Obstacle Detection System for Mobile Robots in Public Spaces

To guarantee a systematic evaluation, requirements for the detection of obstacles in the close range of a mobile robot in public space are derived regarding (1) system-specific requirements, (2) general obstacle properties, and (3) environmental conditions as well as (4) the specific application. These requirements serve as the basis for selecting an obstacle detection approach as well as developing scenarios for evaluating its performance.

4.1 System-Specific Requirements

The system-specific requirements serve to ensure the compatibility of the approach being evaluated with existing robots in terms of both software and hardware. For an obstacle detection approach to be selected, compatibility with ROS Melodic and a release under an open source license have to be fulfilled. The stereo cameras of the mobile robot produce 3D point clouds, which serve as input data for a selected algorithm. Additionally, the approach should be executable on a mobile computing unit with limited computing power.

4.2 General Obstacle Properties

The obstacle properties can also be derived from the TaBuLa-LOG mobile robot and may vary for different applications. Experiments in the context of approval procedures have shown that obstacles with a size of 5 cm as well as surfaces with a slope of significantly more than 30 % can lead to safety-critical problems and should therefore be detected. Since in this case a more detailed classification of detected objects is not necessary, the task of obstacle detection can be reduced to a binary decision between passable areas and obstacles respectively impassable areas.

4.3 Environment-Specific Requirements

Mobile robots in public areas must be able to operate in a variable and changing environment, which itself has its requirements for an obstacle detection system. The surface to be driven on varies from unpaved gravel paths to level and structured walkways to uneven cobblestones. Especially irregular and uneven surfaces cause the robot and its sensors to sway. This creates challenges regarding the detection of passable areas since parts of the ground are perceived as obstacles due to the changing orientation of the sensor system. Accordingly, the obstacle detection system should be able to robustly segment traversable ground from obstacles as independently as possible from changing ground structures. Operating on walkways demands to detect dynamic obstacles, such as pedestrians, in real-time as well as the boundary of the walkway which is usually the drop off of the curb to the street.

4.4 Application-Specific Requirements

The fourth and last influence on the requirements is represented by the given application of the robot. In the TaBuLa-LOG project the mobile robot is using a public transport shuttle to cover further distances. Thus, the robot must detect obstacles typical for public transport, such as luggage or the feet and legs of passengers. The robot enters and exits the shuttle via a wheelchair ramp. The ramp does not have a physical barrier to limit the traversable area, so the robot's obstacle detection should perceive the drop offs on each side of the ramp as obstacles. Figure 1 shows the robot entering the automated public transport shuttle in Lauenburg via a wheelchair ramp.

The environment- and application-specific requirements can be summarized into six points: (1) Robustness to uneven ground, (2) successful curb detection, (3) successful drop off detection, (4) successful pedestrian detection, (5) successful detection of typical static obstacles, and (6) successful detection of public transport-specific obstacles. Based on these six requirements, a set of test scenarios can be developed that allows the evaluation of obstacle detection approaches for mobile robots in public areas.

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

Figure 1. One of the TaBuLa-LOG robots entering the automated public transport shuttle in Lauenburg.

5 Experimental Setup

This section describes the experimental setup. It therefore includes the derivation of test scenarios from the previously presented requirements as well as the selection, presentation and parameterization of the obstacle detection approach.

5.1 Derivation and Recording of Test Scenarios

To evaluate the obstacle detection approach regarding the six main requirements presented in the previous section, a total of eight test scenarios were developed in a way that allows each requirement being tested in at least two scenarios.

In terms of robustness to ground irregularities, two scenarios (scenarios 1 & 2) were developed with different types of cobblestones as the ground surface to be driven on. The varying degree of unevenness allows for estimating whether the susceptibility to errors is proportional to the unevenness of the ground. Scenario 3 aims to verify the detection of curbs. It also includes various typical static obstacles, such as trees and flower beds. Scenario 4 involves the drop off between the walkway and the road in the form of a curb, which should also be recognized as an impassable area.

The focus of scenario 5 is the detection of a pedestrian walking around the standing robot. While the robot is stationary in scenario 5, scenario 6 envisions a dynamic and versatile sidewalk situation in which the robot drives past curbs, drop offs, pedestrians, and typical static obstacles such as parking meters, fences, or lampposts that should be identified as obstacles. The entry and exit of the robot into an autonomous public transport shuttle via a ramp are simulated in scenario 7 using a shuttle mock-up. In addition to the drop offs at the ramp's edges, the detection of passengers as well as luggage is tested in this scenario.

Finally, Scenario 8 represents a collection of situations that were initially assessed as challenging for the obstacle detection system. These include changing light conditions, the detection of delicate and detailed objects, as well as objects with grid-like or reflective features. Table 1 provides an overview of the coverage of the individual requirements by the developed test scenarios.

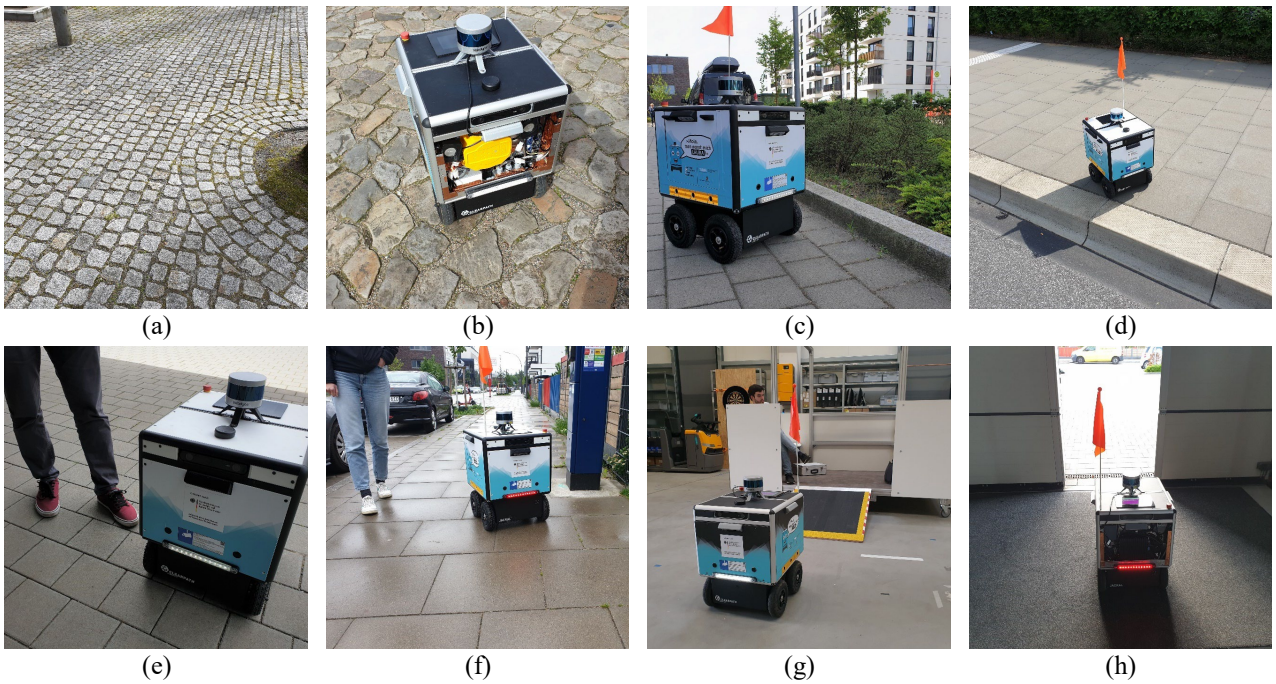

Figure 2. Overview over the eight test scenarios: (a) Scenario 1: Moderate cobblestone (b) Scenario 2: Rough cobblestone (c) Scenario 3: Curb detection (d) Scenario 4: Drop off detection (e) Scenario 5: Pedestrian detection (f) Scenario 6: Walkway (g) Scenario 7: Public transport shuttle (h) Scenario 8: Challenging situations (exemplary: changing light condition).

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

Table 1. Derivation of test scenarios from requirements.

Requirement	Scenario 1	2	3	4	5	6	7	8
Robustness	✓	✓						
Curb detection			✓			✓		✓
Drop off detection				✓		✓	✓	
Pedestrian detection					✓	✓		
Typical static obstacles			✓			✓	✓	✓
PT-specific obstacles					✓		✓	

Scenario 1: Moderate cobblestone; **Scenario 2:** Rough cobblestone; **Scenario 3:** Curb detection; **Scenario 4:** Drop off detection; **Scenario 5:** Pedestrian detection; **Scenario 6:** Walkway; **Scenario 7:** Public transport (PT) shuttle **Scenario 8:** Challenging situations

To record the test scenarios, environments were sought that best matched the developed test scenarios. In Figure 2 these scenario locations are shown. Sensor data from four downward-facing stereo cameras mounted around the robot was recorded in the form of ROS bag files. Figure 3 shows the robot's sensor fields of view in a dimensioned representation as well as the actual sensor data from scenario 1 in the form of 3D point clouds overlaid with camera images for visualization purposes. The right image

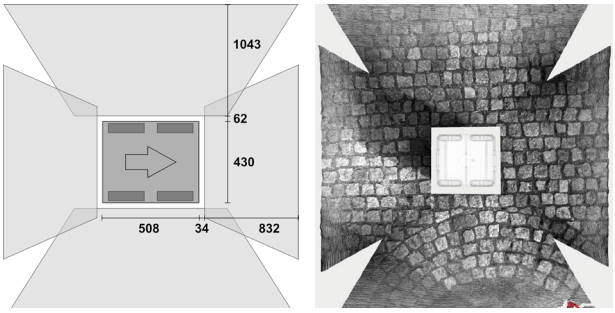

Figure 3. Dimensioned (in mm) representation of the sensor fields of view (left) and visualization of the real sensor data (right).

of Figure 3 further illustrates how all four camera images are viewed simultaneously within one evaluation step. This is done with the help of the visualization software *RViz*.

5.2 The Obstacle Detection Approach

As shown in Section 2, six alternative open source approaches were identified for further consideration. These are the *RTAB-Map obstacles_detection* ROS node, the *GndNet* Python application, the *elevation_mapping* ROS package, the *hector_elevation_mapping* ROS package, the *Move Base Flex (MBF)* navigation stack, and the development of a custom solution based on *Point Cloud Library (PCL)* functions.

To select a suitable approach, particular focus was placed on hardware and software compatibility of the approaches with the existing system as well as on the ability to distinguish drivable surfaces from obstacles. After comparing the possible alternative approaches (see Table 2), the *RTAB-Map obstacles_detection* ROS node and the custom solution based on the *PCL* functions emerge as the most suitable solutions regarding the system-specific requirements.

Table 2. Selecting an obstacle detection approach.

Approach	Compatibility		Traversability detection
	Hardware	Software	
RTAB-Map	✓	✓	✓
GndNet	N/A	✓	✓
ANYbotics	✓	N/A	✓
hector	✓	N/A	✓
MBF	✓	×	✓
PCL	✓	✓	✓

A closer look at the *RTAB-Map* node reveals that it is also based on those *PCL* functions that are relevant for segmenting traversable surfaces and obstacles and gives access to all relevant parameters of the individual functions. Accordingly, the additional effort of a custom development exceeds the benefit, since with the *RTAB-Map obstacles_detection* ROS node a ready-to-use, compatible, and manually adaptable implementation is already available, which is thus selected for further examination. To segment sensor data into traversable areas and obstacles, the *RTAB-Map obstacles_detection* node uses the surface normal vectors of the individual measurement points as a decision criterion. Thus, the measurement points are classified based on their surface orientation in three-dimensional space. A schematic overview of the functional principle is shown in Figure 4.

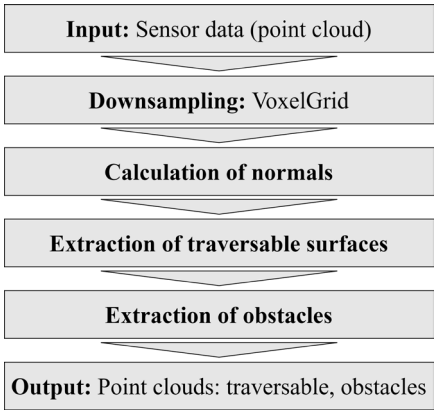

Figure 4. Functional principle of the RTAB-Map obstacles_detection node.

The system input is sensor data in the form of 3D point clouds, which in this case are generated by stereo cameras. To increase data processing efficiency, the input point cloud is first downsampled using a voxel grid. The

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

calculation of the normals is based on a defined number of neighboring points or their spatial location. With the help of a threshold value, it can then be determined by how many degrees a normal vector may deviate from a perpendicular orientation to be still classified as traversable. If this threshold value is exceeded, the measuring point is classified as an obstacle.

To reduce the effect of outlying measurement values, clustering is then performed, in which a minimum size can be used to specify how many measurement points must be contained in a cluster to be considered a contiguous obstacle. Finally, the original measurement points are divided into two separate point clouds, one each for traversable areas and obstacles. They serve as the output of the node and can then be further processed or directly integrated into the navigation of the robot.

5.3 Parameterizing the Obstacle Detection

Before the obstacle detection system can be applied to the test scenarios, it first needs to be calibrated independently from the test scenarios. This is done using 3D printed objects that are constructed according to the initial requirements. To consider the minimum size of obstacles, a cuboid and a cylinder with edge lengths of 5 cm are used to be detected as obstacles by the system. To also represent the maximum slope of traversable surfaces, two wedges are constructed, which are characterized by surface slopes of 30 % and 45 %. While a slope of 30 % does not represent an obstacle, a slope of 45 % should be recognized as one. Based on these objects, the parameters of the *RTAB-Map obstacles_detection* node are adjusted until the results matched the expected values. Figure 5 illustrates the described calibration process.

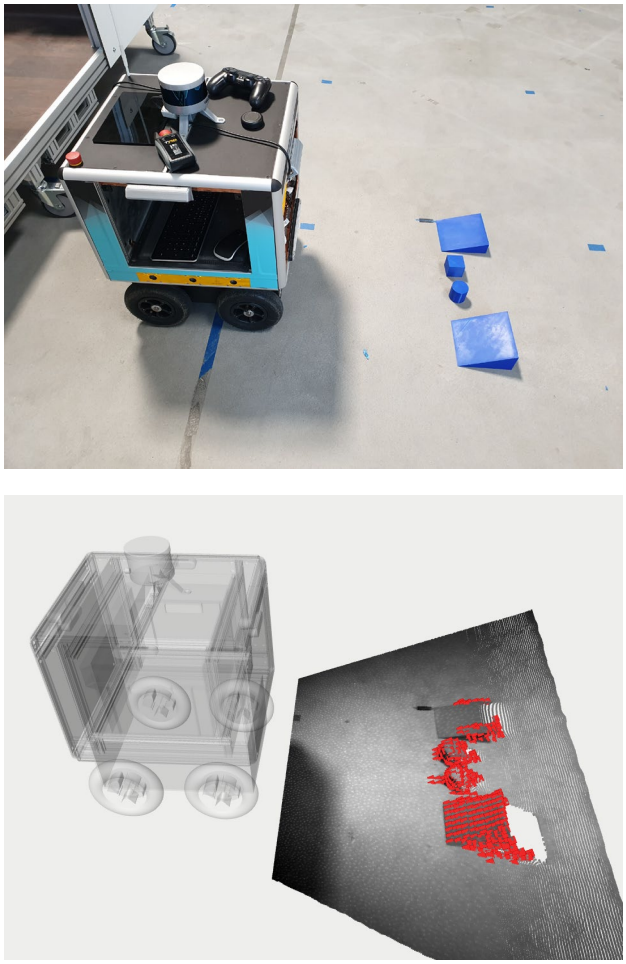

Figure 5. Calibration of the obstacle detection system using 3D printed objects (above) and the front sensor image with the output of the obstacle detection visualized in RViz (below).

6 Experimental Results

In this section, the results of the conducted experiments will be briefly summarized.

The eight test scenarios comprise a total of 1,334 evaluation steps, which together include 2,638 obstacles. Of these obstacles, 2,208 were fully detected, 178 were partially detected, and 252 were not detected (false negatives). In addition, a total of 1,657 false positive classifications were recorded. This results in an overall detection rate of 83.7% and a mean absolute error (MAE) of 1.43 misclassifications per evaluation step. Table 3 provides an overview of the summarized results of all eight test scenarios. Scenarios one and two do not focus on the detection of explicit obstacles, but rather on the effect of differently uneven surfaces on the error development. Thus, there is no value for the detection rate for these scenarios.

7 Discussing the Experimental Results

The discussion of the experimental results is the main focus of this section. Beyond the evaluation of the results throughout the different test scenarios, this section also addresses the limitations associated with this work.

The test scenarios in which the obstacle detection system performs best in terms of both detection rate and MAE are (3) curb detection, (5) pedestrian detection, (6) walkway, (8) and challenging situations. These four scenarios form a distinctive cluster with high detection rates between 88.37 % and 100 % and simultaneously relatively low mean absolute errors between 0.12 and 0.65. The detection of curbs (3) and pedestrians (5) stand out in particular with detection rates of 100 % and 99.61 %. Regarding the MAE, the fifth scenario performs minimally worse as well, but with 0.32 misclassifications per evaluation step, it is still in a very good range in the overall comparison. Thus, the detection of a curb on flat ground (Figure 6), and the detection of a moving person (Figure 7) pose no challenge.

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

Table 3. Experimental results and key performance figures for all eight test scenarios.

	Scenario								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**total**
Frames	141	144	140	147	136	156	202	268	1,334
Obstacles	0	0	269	139	255	404	1,206	365	2,638
Fully detected	0	0	269	0	254	357	973	355	2,208
Partially detected	0	0	0	13	1	20	144	0	178
Not detected	0	0	0	126	0	27	89	10	252
False positive	81	284	17	59	44	74	945	153	1,657
Detection rate	N/A	N/A	100%	0.00%	99.61%	88.37%	80.68%	97.26%	83.70%
Ø FN/Frame	0	0	0	0.86	0	0.17	0.44	0.04	0.19
Ø FP/Frame	0.58	1.97	0.12	0.40	0.32	0.47	4.68	0.57	1.24
MAE	0.58	1.97	0.12	1.26	0.32	0.65	5.12	0.61	1.43

Scenario 1: Moderate cobblestone; **Scenario 2:** Rough cobblestone; **Scenario 3:** Curb detection; **Scenario 4:** Drop off detection; **Scenario 5:** Pedestrian detection; **Scenario 6:** Walkway; **Scenario 7:** PT shuttle **Scenario 8:** Challenging situations

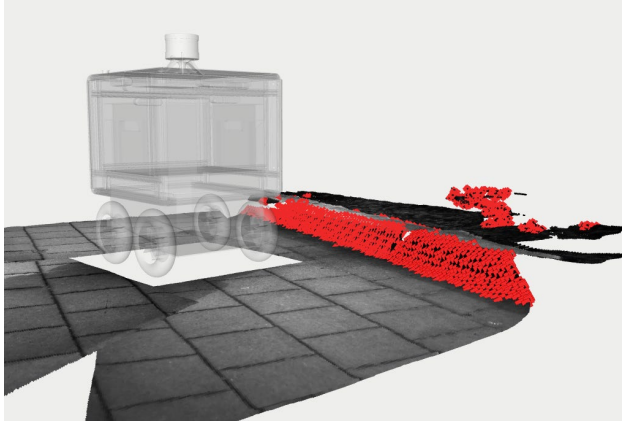

Figure 6. Successful detection of a curb on a flat ground surface.

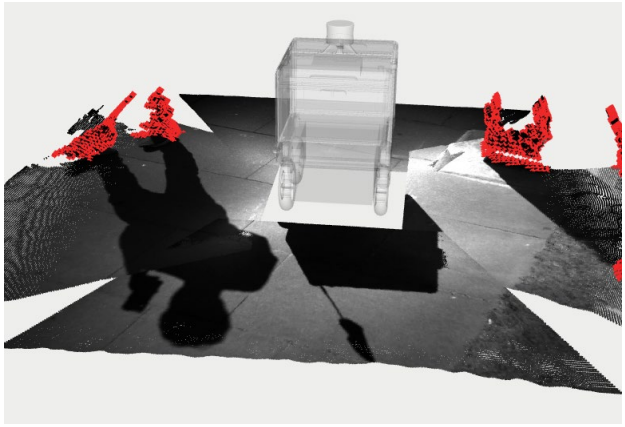

Figure 7. Successful detection of a pedestrian and other obstacles with clearly visible vertical surfaces.

Scenario 8, consisting of several individual challenging situations, shows decreasing performance regarding both key metrics. In terms of detection rate, this is due to the partial failure to detect a gully cover that was selected as particularly challenging because of its grid-like structure. However, as soon as the robot is closer to the object, it can be successfully detected as an obstacle. Furthermore, concerning the MAE, the situation with changing lighting conditions can be identified as the cause of the decrease in performance. Overexposure of the stereo camera image leads to partially critical false positive misclassifications directly next to and in front of the robot. Despite the demanding situations, the remaining individual scenarios do not pose a critical challenge for obstacle detection.

The obstacle detection performance in the walkway scenario (6) is lower than in the previous scenarios in terms of both indicator values. Considering the detection rate, the failed identification of the drop off between the walkway and the road can be identified as the cause of the error. The failed detection of these safety-critical obstacles also decides that the overall performance of this scenario cannot be rated as satisfactory despite otherwise satisfactory values of the key metrics overall. False negative classifications correspond to a present but unrecognized obstacle and thus tend to have a significantly larger impact on safety. Failure to detect pedestrians would result in a high risk of injury. Undetected drop offs could additionally cause damage to the robot itself. False positives on the other hand result in sudden and unexpected stops of the robot. This hinders operation, but also poses a risk for rear-end collisions.

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

Regarding the MAE, the performance within scenario (6) is mainly inhibited by misclassifications due to overexposed sensor images as well as reflective surfaces. Thus, the exposure conditions again represent an influencing factor on the system performance. A case of false positive classifications due to overexposure is shown in Figure 8.

Considering the MAE only, the obstacle detection performance within the moderate cobblestone scenario (1) is also on the same level as the before mentioned scenarios: Apart from misclassifications resulting from strong tilting movements when turning on the spot, almost no other critical misclassifications occur while driving over uniform and regularly laid cobblestones. Accordingly, the *obstacles_detection* node shows robustness against driving on moderately uneven ground surfaces.

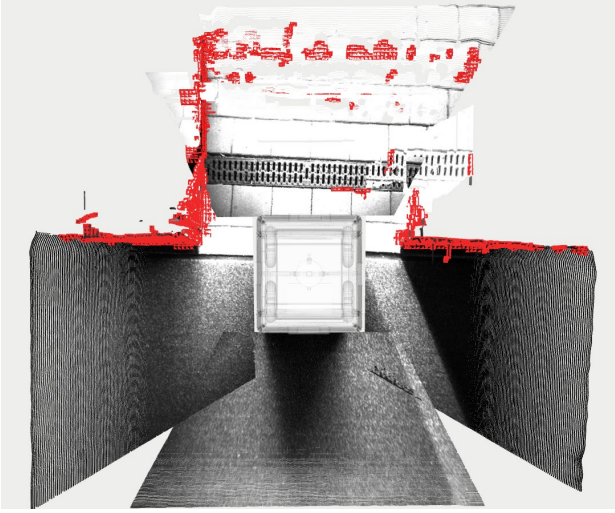

Figure 8. Critical false positive classifications immediately in front of the robot due to a large overexposed area of the stereo camera image (leaving a building through a door).

The remaining scenarios (2) rough cobblestone, (4) drop off detection, and (7) public transport shuttle cause significantly greater difficulties for the obstacle detection system. In the rough cobblestone scenario (2), critical misclassifications already occur during regular travel at increased speed due to stronger tilting movements resulting from increased unevenness of the ground surface compared to scenario (1) with moderate cobblestone. These tilting movements result in a characteristic error pattern that is shown in Figure 9. On the side to which the robot tilts, the measurement points of the sensor point cloud appear compressed, while the measurement points on the other side have gaps. Both lead to a geometric distortion, which entails a deflection of the normals and thus a classification as obstacles. Overall, the MAE of 1.97 is significantly higher than in scenario (1). This suggests that the number of defects occurring on average increases the more uneven and irregular the ground surface.

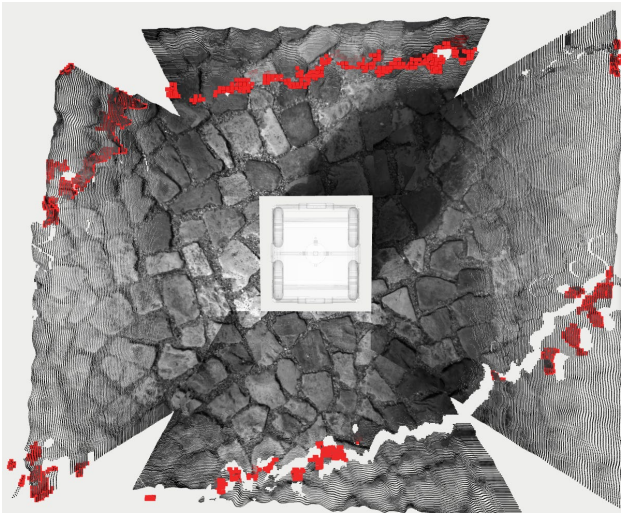

Figure 9. Characteristic error pattern of tilting movements of the robot resulting in false positive classifications while driving on rough cobblestone.

While the performance in the shuttle scenario (7) is comparable to the scenarios of the top group considering only the detection rate, the MAE of 5.12 misclassifications per evaluation step represents an enormously high value. The cause of this high error rate lies again in tilting movements and the inclined position of the robot when driving onto the shuttle ramp. This leads to a virtual inclination of all sensor data as well as the characteristic error pattern of tilting movements, which affects the sensor data around the robot (see Figure 9 for reference) and can lead to critical errors.

In addition, this scenario results in a failed detection of the drop offs on the sides of the shuttle ramp. Failing to detect these critical obstacles already precludes an overall evaluation of the performance as satisfactory for the same reason as in the walkway scenario (6). The detection rate of 80.68 % within this scenario also illustrates that, apart from the ramp drop offs, the detection of the public transport-specific obstacles does not pose a challenge for the *obstacles_detection* node. The enormously high MAE value of 5.12 misclassifications per evaluation step within scenario (7) also has a significant impact on the overall metric. Excluding scenario (7) from calculating the overall MAE results in a reduction from 1.43 to only 0.79 average misclassifications per evaluation step, and therefore nearly halving the original value.

With a detection rate of 0 %, the obstacle detection in the drop off detection scenario (4) performs by far the worst regarding this key figure. The main purpose of the scenario is the detection of an again safety-critical drop offs between the walkway and the road, which cannot be achieved successfully similar to the walkway (6) and shuttle (7) scenarios, resulting in the extremely low detection rate. The extremely low detection rate of this scenario has an enormous impact on the aggregated detection rate, similar to the

DOI: 10.2195/lj_proc_blunder_en_202211_01
URN: urn:nbn:de:0009-14-55802

effect of the shuttle scenario (7) on the MAE. Excluding the drop off detection scenario (4) when calculating the overall detection rate, it increases from 83.7 % to 93.18 %.

In summary, the *RTAB-Map obstacles_detection* node is robust to a certain degree against unevenness of the ground, provides stable and detailed results for clearly visible obstacle surfaces, and is operational on mobile computing units. Especially the detection of drop offs as well as the susceptibility to tilt and changing light conditions turned out to be challenging for the given approach. The use of the system in its current form in mobile robots operating in public space is therefore not advisable due to the discussed shortcomings. In particular, the failure to detect drop offs poses too great a safety risk.

It is noticeable, that the error causes lie less in algorithm-internal errors than in corrupt sensor data. In the case of the unrecognized drop offs it can be stated that these do not exist in the sensor point clouds at all. This is because in the case of the curb, the vertical face of the drop off is hidden from the sensors behind the top edge. In the case of the shuttle ramp, on the other hand, no vertical surfaces exist that could be detected as obstacles. Figure 10 and 11 illustrate how hidden drop offs like for example curbs or of shuttle ramps cannot be detected as obstacles as they are not existent in the sensor data that serves as input for the *obstacles_detection* node

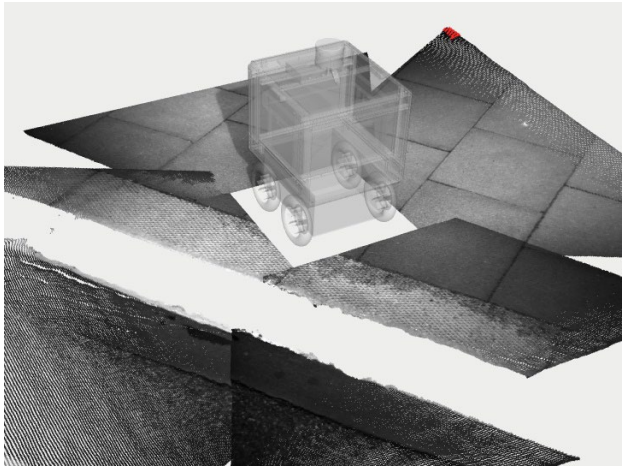

Figure 10. The drop off of a curb with its vertical surface hidden from the sensors is not represented in the sensor point clouds and poses a significant and safety-critical challenge to the obstacle detection.

Accordingly, in order to use the investigated approach in a real environment and especially in combination with a public transport shuttle, it would have to be ensured that any drop offs are represented as surfaces with steep gradients in the data used as input for the obstacle detection. Only then can they be successfully detected as obstacles by the *obstacles_detection* node.

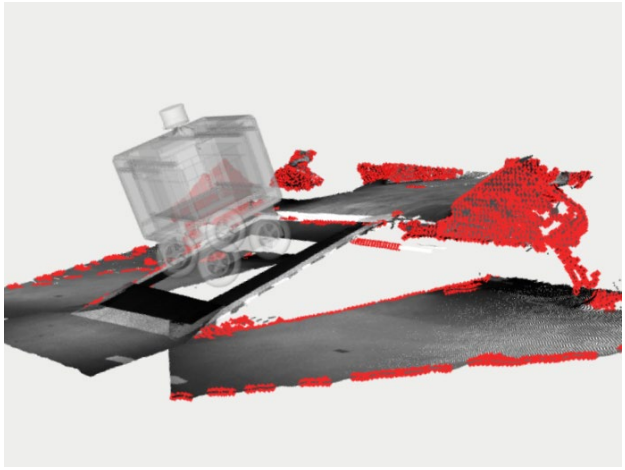

Figure 11. Occluded regions that are not directly captured in the sensor data cannot be classified as non-traversable.

7.1 Limitations

Although the present work was able to identify fundamental strengths and weaknesses of the approach studied, specific questions could not be answered or could only be answered to some extent. While the mainly quantitative experiment evaluation allows a general comparison of different approaches, the purely objective consideration of the error metrics does not allow a final conclusion on the actual severity of occurring errors. The assessment of an error's severity is currently based on a subjective estimation drawn from previous experience. To ensure universal transferability to different use cases, this should be the starting point for identifying a more objective evaluation option.

Another limitation regarding general transferability is the application-specific parameterization of the detection algorithm used. The parameterization was performed with specifically selected test objects (see Section 5.3). Their geometric properties such as size and shape reflect constraints of the research project in which this study was conducted, e.g. sensor properties and driving dynamics of the robot used.

8 Conclusion and Outlook

In this paper, the integration and evaluation of an 3D obstacle detection system for the close range of mobile robots operating in public areas was carried out. Four domains could be identified which impose requirements on object detection systems for such robots: System specific requirements, general obstacle properties, environment-specific requirements, and application-specific requirements.

Based on these requirements, eight test scenarios, consisting of real scenes of the operation of a mobile robot, were developed to allow performance evaluation. These

include, for example, operation on sidewalks as well as in combination with public transport vehicles, driving over different ground surfaces, and detection of pedestrians. Furthermore, also based on the requirements, the *obstacles_detection* node of the *RTAB-Map* ROS package was selected for the implementation of obstacle detection and evaluated using the test scenarios and performance and error metrics.

Overall, the *obstacles_detection* node provides very good results for scenarios with clearly visible vertical obstacle surfaces while being robust to a certain degree against unevenness of the ground surface. On flat surfaces, against which obstacles stand out, obstacle detection performance is best, as expected. The detection of obstacles with non-enclosed, detailed surfaces, such as fine branches, also performed well.

Three particular challenges can be identified for the *RTAB-Map* obstacle detection system, based on the experimental results across all scenarios:

- The detection of drop offs that do not appear as explicit obstacles in the sensor data
- Faulty classifications as a result of tilting movements of the robot and its sensors
- Faulty classifications as a result of changing lighting conditions

All three challenges occur in multiple scenarios. However, the source of these errors lies in the point clouds generated by the sensors rather than in the algorithm itself. The point clouds contain gaps where drop offs are located, distortions and missing points due to tilting movements of the robot, as well as increased noise of the measurements due to overexposure of stereo camera images. Since this flawed point cloud data serves as input for the *obstacles_detection* node, even correct classifications lead to incorrect overall results. One possible solution for dealing with gaps in point cloud data could be the adoption of elevation maps. These create a closed surface of the terrain and are therefore also suitable for detecting drop offs, which are represented as steep sloping surfaces. Implementations of elevation mapping approaches are also available as ROS packages (see chapter 2).

Across the various scenarios, there are several objects which, from a purely geometric point of view, represent obstacles, but could theoretically be traversed by the robot due to their physical properties (e.g. tufts of grass or leaves). While the *RTAB-Map* node correctly recognizes them as geometric obstacles, an additional semantic classification would allow the robot to drive over these types of obstacles.

Financial Disclosure

This article is based on research that was undertaken in the project "TaBuLa-LOG - Kombinierter Personen- und Warentransport in automatisierten Shuttles" sponsored by the German Federal Ministry of Transport and Digital Infrastructure between 2020 and 2021.

References

[1] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D Object Detection for Autonomous Driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2147–2156, doi: 10.1109/CVPR.2016.236.

[2] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore, 2017, pp. 5067–5073, doi: 10.1109/ICRA.2017.7989591.

[3] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion," in *2009 IEEE Intelligent Vehicles Symposium*, Xi'an, China, 2009, pp. 215–220, doi: 10.1109/IVS.2009.5164280.

[4] C. Gertz *et al.,* "Endbericht des Projektes TaBuLa-LOG: Kombinierter Personen- und Warentransport in automatisierten Shuttles," Technische Universität Hamburg (TUHH), Hamburg, 2022.

[5] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, "Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014, pp. 873–878, doi: 10.1109/ITSC.2014.6957799.

[6] J. Hu, Y. Niu, and Z. Wang, "Obstacle avoidance methods for rotor UAVs using RealSense camera," in *2017 Chinese Automation Congress (CAC)*, 2017, pp. 7151–7155, doi: 10.1109/CAC.2017.8244068.

[7] L. Steccanella, D. D. Bloisi, A. Castellini, and A. Farinelli, "Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring," *Robotics and Autonomous Systems*, vol. 124, p. 103346, 2020, doi: 10.1016/j.robot.2019.103346.

[8] M. Hua, Y. Nan, and S. Lian, "Small Obstacle Avoidance Based on RGB-D Semantic Segmentation," in *2019 International Conference on Computer Vision workshops: ICCV 2019 : proceedings : 27 October-2 November 2019, Seoul, Korea*, Seoul, Korea (South), 2019, pp. 886–894, doi: 10.1109/ICCVW.2019.00117.

[9] M. Cordts *et al.,* "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.

[10] H. Wang, Y. Sun, and M. Liu, "Self-Supervised Drivable Area and Road Anomaly Segmentation Using RGB-D Data For Robotic Wheelchairs," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4386–4393, 2019. doi: 10.1109/LRA.2019.2932874.

[11] C. Pang, X. Zhong, H. Hu, J. Tian, X. Peng, and J. Zeng, "Adaptive Obstacle Detection for Mobile Robots in Urban Environments Using Downward-Looking 2D LiDAR," *Sensors 18 (6)*, pp. 1749, 2018. doi: 10.3390/s18061749.

[12] K. Park *et al.,* "SideGuide:A Large-scale Sidewalk Dataset for Guiding Impaired People," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 10022–10029, doi: 10.1109/IROS45743.2020.9340734.

[13] M. Labbé and F. Michaud, "RTAB-Map as an Open-Source Lidar and Visual Simultaneous Localization And Mapping Library for Large-Scale and Long-Term Online Operation," *J Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019, doi: 10.1002/rob.21831.

[14] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 362–373, 1988, doi: 10.1109/34.3900.

[15] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 1–4, doi: 10.1109/ICRA.2011.5980567.

[16] A. Paigwar, O. Erkent, D. Sierra-Gonzalez, and C. Laugier, "GndNet: Fast Ground Plane Estimation and Point Cloud Segmentation for Autonomous Vehicles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 2150–2156, doi: 10.1109/IROS45743.2020.9340979.

[17] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," 2014, doi: 10.3929/ethz-a-010173654.

[18] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. von Stryk, "Hector Open Source Modules for Autonomous Mapping and Navigation with Rescue Robots," in *RoboCup 2013: Robot World Cup XVII* (Lecture Notes in Computer Science), D. Hutchison et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 624–631.

[19] S. Putz, J. Santos Simon, and J. Hertzberg, "Move Base Flex A Highly Flexible Navigation Framework for Mobile Robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, 2018, pp. 3416–3421, doi: 10.1109/IROS.2018.8593829.

[20] S. Putz, T. Wiemann, M. K. Piening, and J. Hertzberg, "Continuous Shortest Path Vector Field Navigation on 3D Triangular Meshes for Mobile Robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 2256–2263, doi: 10.1109/ICRA48506.2021.9560981.

Noel Blunder, M.Sc., Research Assistant at the Institute for Technical Logistics, Hamburg University of Technology. Noel Blunder studied Logistics & Mobility and International Management & Engineering with a focus on information technology at the Hamburg University of Technology between 2015 and 2022.

Marko Thiel, M.Sc., Research Assistant at the Institute for Technical Logistics, Hamburg University of Technology. Marko Thiel studied Mechanical Engineering and Theoretical Mechanical Engineering at Hamburg University of Technology.

Manuel Schrick, M.Sc., Research Assistant at the Institute for Technical Logistics, Hamburg University of Technology. Manuel Schrick studied Computer Science at University of Lübeck and RWTH Aachen University.

Dr. Johannes Hinckeldeyn, Senior engineer at the Institute for Technical Logistics, Hamburg University of Technology. After completing his doctorate in Great Britain, Johannes Hinckeldeyn worked as Chief Operating Officer for a manufacturer of measurement and laboratory technology for battery research. Johannes Hinckeldeyn studied industrial engineering, production technology, and management in Hamburg and Münster.

Prof. Dr.-Ing. Jochen Kreutzfeldt, Professor and Head of the Institute for Technical Logistics, Hamburg University of Technology. After studying mechanical engineering, Jochen Kreutzfeldt held various managerial positions at a company group specializing in automotive safety technology. Jochen Kreutzfeldt then took on a professorship for logistics at the Hamburg University of Applied Sciences and became head of the Institute for Product and Production Management

Address: Institute for Technical Logistics, Hamburg University of Technology, Theodor-Yorck-Straße 8, 21079 Hamburg, Germany; Phone: +49 40 42878-3586, Email: noel.blunder@tuhh.de