 DOI: 10.3217/978-3-85125-524-9-08

An Autonomous Transportation Robot for Urban Environments

Konstantin Lassnig[1] and Clemens Mühlbacher[2] and Gerald Steinbauer[2] and Stefan Gspandl[3] and Michael Reip[3]

***Abstract*— The transportation of goods is a central task of today's economy. The cheap transportation of goods allows the wide spread of today's internet based sales. To perform such transportation tasks one currently relies on humans. This imposes constraints when the transportation can be performed and imposes constraints on the costs. To address this time and cost constraints an automatic transportation of goods is preferred.**

Such an automatic transportation can be performed by an autonomous robot, as the ones used in warehouse environments. Although such environments are diverse and undergo a certain amount of change they are still rather static environments. To allow robots to perform the transportation in outdoor environments several problems need to be tackled. One needs to deal with large operation areas, uneven ground, and dynamic objects. In this paper, we present a robot system which can cope with these problems and allows to perform transportation tasks in outdoor environments. The focus of this paper will be on the localization and navigation of the robotic in the outdoor environment allowing the robot to perform outdoor deliveries.

I. INTRODUCTION

The cheap transportation of goods is a central part of today's economy. Reasonable prices of goods which are sold over the internet, heavily depend on transportation costs. Today's supply chain requires a very dense distribution network and relies on the fact that sending a lot of packages on the same route is cheap. The larger number of goods for one route the cheaper it becomes. This is in contrast with the need for transporting goods to a single customer. Such a transportation is characterized by a few goods for one transportation route. To address this, robots offer a possible solution. Using a robot, the transportation can be performed in a flexible manner. Additionally, if multiple robots are used one can simply balance the load of transportation tasks on several robots.

The use of a robot fleet for transportation tasks is getting adopted for warehouse environments nowadays [1], [2], [3]. These robot systems allow transporting goods in the warehouse without the need of an adaption of the warehouse. This is achieved by using algorithms allowing a localization and navigation in an indoor environment [4]. These algorithms use a 2D map of the environment. Such a map can be stored easily in the robots memory for a warehouse but not for large outdoor environments such as a city. Furthermore, the 2D map can be easily used in a warehouse for navigation as one can assume a reasonable flat ground. Such an assumption cannot be made for an outdoor environment where the robot needs to ensure that it is not falling over road curbs.

To allow a robot system to be used for transportation tasks in a large scale outdoor environment, one needs to address the problems which are imposed by the scale of the environment as well as the uneven ground. In this paper, we show a robot system which addresses these problems. The size of the environment is addressed by splitting the environment into smaller areas allowing the robot to keep only a small map in its memory. To allow the robot to be globally localized one additionally stores how the small pieces are related to each other. To tackle the uneven ground only the area close to the robot needs to be considered. This space is represented as a 2.5D surface and interpreted to find possible holes.

The remainder of the paper is organized as follows. In the next section, we will discuss the software system used by the robot to perform transportation tasks in an outdoor environment. The proceeding section discusses how the robot localizes itself despite the size of the environment. In Section IV we discuss how the robot navigates in the environment. This section also comprises a description how the robot deals with the uneven ground. Afterward, we discuss some related research. Finally, we conclude the paper and point out some future work.

II. SYSTEM OVERVIEW

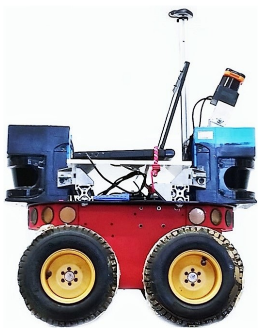

Fig. 1. The transport robot [5].

In this paper, we discuss a robot which can perform a transportation task on a university campus autonomously. The robot can navigate indoor as well as outdoor. Furthermore, the robot considers the uneven ground outdoors to

[1]Konstantin Lassnig is with ARTI, Graz Austria. This author was with the Institute for Software Technology when contributing, Graz University of Technology, Graz, Austria. klassnig@arti-robots.com

[2]Clemens Mühlbacher and Gerald Steinbauer are with the Institute for Software Technology, Graz University of Technology, Graz, Austria. {cmuehlba,steinbauer}@ist.tugraz.at

[3]Stephan Gspandl and Michael Reip are with incubedIT, Hart bei Graz, Austria. {gspandl,reip}@incubedit.com

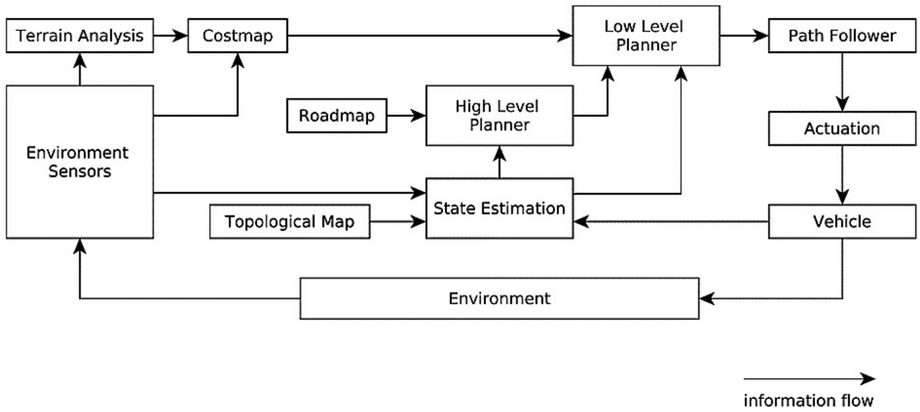

Fig. 2. System overview of the transport robot [5].

safely navigate between buildings. The robot is depicted in Figure 1 and is based on a pioneer 3-AT platform which allows the robot to navigate indoors as well as outdoor. Additionally, the robot has three laser scanners to detect obstacles. Two laser scanner are mounted horizontally to detect obstacles, like cars or people. Furthermore, these two lasers are used to localize the robot. The third laser is mounted tilted down to scan the ground in front of the robot. This laser is used to build a map of the local terrain. Besides the laser scanners, the robot has a GPS sensor for the localization and mapping. To improve the accuracy of the odometer of the robot an inertial measurement unit (IMU) is used which is mounted on the robot. To perform the transportation task, the robot uses the system architecture depicted in Figure 2.

The robot uses its sensors to estimate the current location. This is done using the robots odometer, the IMU, GPS and the horizontal laser scanners. Due to this redundancy, the estimation of the current location is stable in areas where one sensor may yield wrong results, e.g. the GPS sensor near tall buildings. To perform the estimation, the robot matches the sensor readings with the information of a topological map. The topological map consists of several small maps which are linked to each other to allow the robot to only keep small maps to be localized.

Using the estimation of the current location together with a road map, the robot plans a high-level path for navigation. The roadmap describes possible traversal routes in the environment on a higher level. Due to this abstraction, the planning can be done very efficiently even in the case of large environments. After generating a high-level plan, the plan is passed to the lower-level planner which tries to find a valid path in the environment for each path segment in the high-level plan. This is done by considering the current small local map of the environment as well as the sensor data which are used to build a cost map. If a valid path is found the robot tries to follow this path as accurate as possible.

To incorporate the information of the terrain the robot uses the tilted laser to perform a terrain analysis. Afterward, the results of this terrain analysis are used to update the cost map. Thus, holes, as well as small objects which are below the horizontal laser scans but bigger than the robots clearance, are added to the cost map as obstacles. This allows the robot to consider the terrain in the low-level planning.

In the following two sections, we will discuss the localization as well as the navigation in more detail.

III. LOCALIZATION

Starting from an initial known position the robot needs to know its location during the entire delivery. This is done through one part of the robot system which is used to localize the robot. This localization should ensure that the robot has an estimation of its global position. First, the robot corrects its odometer to get a good estimation of its 2D position using dead reckoning. Afterward, it uses the created topological map to localize itself.

To correct the odometer of the robot we use an unscented Kalman filter (UKF) [6]. The Kalman filter uses the raw odometer of the robot to perform a prediction of the robot pose. This prediction is formed in a probabilistic manner with a position and a covariance matrix specifying the uncertainty. The covariance matrix is defined in such a way that the linear speed has a higher accuracy as the rotational speed, as the rotation is badly estimated through the raw odometry due to the slippage of the wheels during rotation. To correct the prediction the IMU data are used. The IMU data is used to provide an additional estimation of the robots velocity in all three axes as well as the global orientation the robot has in space. As in the case of the raw odometry, the IMU data update the estimate in a probabilistic manner with the help of a covariance matrix. The covariance matrix for the IMU data is formed in such a manner that the rotational speed is estimated more accurately than with the raw odometry. Due to the use of the Kalman filter, we have a better estimation of the robot pose instead of the very noisy raw odometer of the robot.

After the odometer is corrected the robot can perform its estimation on the topological map. The topological map is a graph with vertices which represent positions in the world and edges which represent connections between those posi-

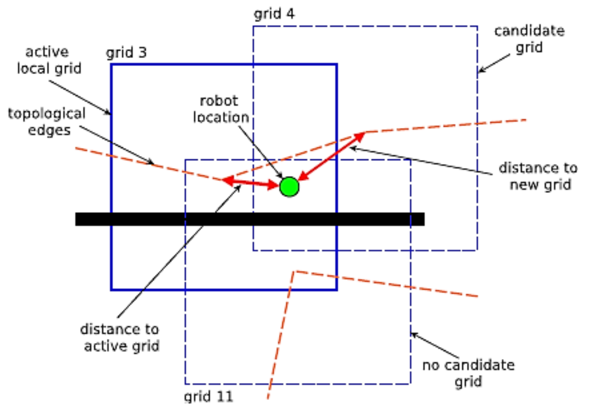

Fig. 3. Grid selection for the localization, together with the topological map [5].

tions. Each vertex is specified as a full 2D pose in the global reference frame allowing to specify the difference from the robots location to any frame in the graph. Furthermore, each vertex contains a grid map representing the local environment at this position. It is ensured that every position within the grid map can reach the center. To ensure proper connections of vertices a connection is only made if the combination of both grid maps allow the robot to reach one vertex from the other. Let's consider the simple example of a topological map as it is depicted in Figure 3. Grid 11 is close to grid 3, but due to the wall between these two grids, no connection between grid 11 and grid 3 is made. Thus, the robot knows which traversals are possible in the environment with the help of the grid map. We will exploit this knowledge to select the right grid for localization if the robot moves beyond the area of one grid.

Initially, the robot knows its starting location, this is done through the input of the user. After the robot has selected the initial position the vertex which is the closest to the current initial location is chosen. Additionally, the robot should check if it can move between the initial pose to the grid map vertex. Using this vertex, the robot can use the grid map of the vertex to localize itself. This is done with the help of a particle filter [7]. The particle filter uses the grid map to align the current laser measurements with the occupied cells in the grid map. Additionally, the robot uses the GPS signal to localize itself. This is done by anchoring each vertex with a GPS position. Thus, by using the current GPS signal the robot can estimate its position relative to the currently used vertex in the topological map. Using the grid map and the GPS the robot derives an estimation of its current location. If the robot is moving in the grid map the localization can be done with the current grid map. But as we assume a large space of the outdoor environment the robot will at some point reach the border of the grid map. In such a case the robot needs to decide which vertex in the topological map is the next one to localize itself. This is done by checking the distance to each vertex in the topological map which has a connection to the currently used vertex. The vertex with the smallest distance to the current robot pose is used for future localization. Thus, the robot will switch the vertex and the occupancy grid only if it is closer to that vertex than any other vertex which could be reached from the robot.

Let's consider the situation in Figure 3. If the robot is moving from grid 3 to grid 4. It checks the distance from the vertex of grid 3 and the distance of vertex of grid 4. But the robot does not check the distance to the vertex of grid 11 as the robot already knows that there is no possibility that it has traveled from grid 3 to grid 11. If the distance of grid 4 is larger than the distance to grid 3 the robot will use grid 3 for future localization.

Due to the use of the connections within the topological map one saves the effort to check all nearby vertexes if they should be used for localization. Furthermore, more important is that the robot will not select a vertex which cannot be reached. Let's consider the map in Figure 3. Grid 3 and 4 can be on the outside of the building whereas grid 11 is the inside of the building. Thus, if the robot is localized outside of the building it does not make sense that the robot jumps suddenly through the wall inside the building. As we don't consider grid 11 as an alternative such a jump is not possible. This also allows the robot to move close to the wall of the building without being incorrect localized.

IV. Navigation

With the help of the topological map, the robot can localize itself. Using this localization, the robot can plan its path to the destination. To do so, the robot uses a hierarchical planning approach. As we consider a large-scale environment the robot is not able to use a grid map of the complete environment. Thus, the robot uses a road map to plan the overall navigation. This allows the robot to plan for the large environment in a fast manner. After a plan is found using the road map the robot use the current grid to search for a mid-level plan within this map to move between different nodes in the roadmap. Finally, the robot uses a local planner to move along the mid-level plan and avoid obstacles which are not present in the grid map.

The road map which is used by the robot to generate a high-level plan consists of a graph of nodes which specify locations and edges which describe possible traversals between these nodes. A roadmap together with the high-level plan is depicted in Figure 4. The road map is constructed by considering the distance between the nodes and if the node is collision free. To check if a node is collision free the footprint of the robot and the local grid map of the position to check is used. As the robot, has not specified a complete description of the environment traversability, one uses the positions used during mapping as a seed for the road map calculation. This allows that the robot uses the positions and traversals which were created during mapping.

To plan a path within the road map the robot first determines the closest node of the road map to its current location. Afterward, the closest node to the destination is determined. The node close to the current position is the start node of the search and the node close to the destination is the goal node of the search. After determining these two nodes the robot performs a graph search for the shortest path through the A^* algorithm [8].

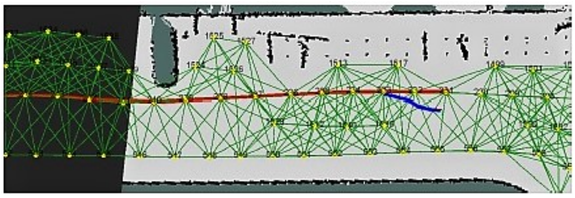

Fig. 4. Road map (green) together with the high-level plan (red) and the low-level plan (blue) [5].

After the robot has generated a high-level plan it generates a mid-level plan on the current grid map to plan to the next node in the road map which is part of the high-level plan. The node which is the next one to pass through is determined by the current location of the robot. The robot considers every node as reached which is in a certain range. To determine this node, the robot uses a queue of nodes within the high-level plan. After the head of the queue is in range the robot pops the head from the queue and uses the new head of the queue as the next goal to plan to. Additionally, if the node is the last node in the queue the robot plans to the destination as the high-level plan only ensure that the robot moves near the destination.

For the mid-level plan on the current grid map, the robot uses the information of the current grid map to determine if it can traverse a grid cell or not. Using this information, the robot uses its current location together with the next node to find a plan. This plan consists of a sequence of grid map cells to traverse. The sequence is found by using the A^* algorithm [8]. As the grid map only specifies a limited area of the world the algorithm can determine the path very fast. Additionally, the path which needs to be planned is most of the time short compared to the high-level plan.

Using the mid-level plan on the current grid map the robot has derived a path which should lead to the current node of the high-level plan considering the known static objects. As we consider a dynamic environment the robot needs to deal with these obstacles as well. This is done by creating a local plan with the help of the dynamic window approach [9]. The local plan is generated several times per second to allow to react to changes. To plan locally the robot uses a cost map which contains the static obstacles, the information from the horizontal laser scan, the information from the elevation map and information about the grass around the robot.

As we argued above one cannot assume that the robot moves on a flat surface. Thus, the robot needs to deal with the uneven ground. Through the construction of an elevation map in a local area, the robot can detect holes and barriers. The elevation map is constructed with the help of the sensor data of the tilted laser. Each of the laser measurements is transformed to specify a position in the world frame. Afterward, the measurement is projected on a grid which defines the height information of the environment. To in cooperate the sensor measurement into the grid a Bayes update per grid cell is used [10]. This allows the robot to deal with the noise of the sensor measurements. After generating the height information one detects holes and barriers by deriving the gradient for each grid cell. Using this gradient one can define a threshold which determines if this hole or barrier is traversable by the robot. If the gradient exceeds a certain limit the robot cannot traverse this grid cell and it is assumed to be a lethal obstacle for the local planner. An example of grid cells which are marked due to a large gradient is depicted in Figure 5. As the elevation map is projected through the gradient into the cost map one can use a standard 2D-planning algorithm to find a local plan.

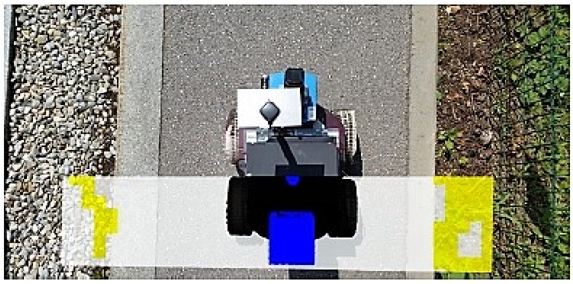

Fig. 5. Detection of edges with the help of the elevation map [5].

Besides the uneven ground, the robot needs also to consider the grass to proper navigate in an outdoor environment. During outdoor navigation, the robot should preferably stick to roads and sidewalks. Thus, the robot needs to detect the grass surrounding the robot. To perform this detection, the robot uses the tilted laser scanner. The tilted laser scanner does not only provide the information about the distance from obstacles but it also contains the information about the intensity of the reflection. Using the intensity and the distance one can identify grass in the environment. A simple linear separator is sufficient to detect grass properly. This relation between distance and reflection intensity with the linear separator is depicted in Figure 6. With the help of this classifier, the robot can detect grass in its vicinity. An example of this detection is depicted in Figure 7. Using this information, the robot adds increased costs in the cost map for the local planner on those positions which indicate grass. Thus, the robot avoids the grass if possible but will also traverse it if necessary.

By combining the grass, the information of the elevation maps, the laser scan measurements of the horizontal laser as well as the static objects in the map the robot can safely navigate locally. Thus, the robot neither hits an object nor falls down a step. We consider these data only for a local area around the robot. This has the benefit of a smaller memory footprint for each local cost map but also the drawback that this information cannot be used for localization or high-level navigation.

V. Related Research

Before we conclude the paper, we discuss some related research.

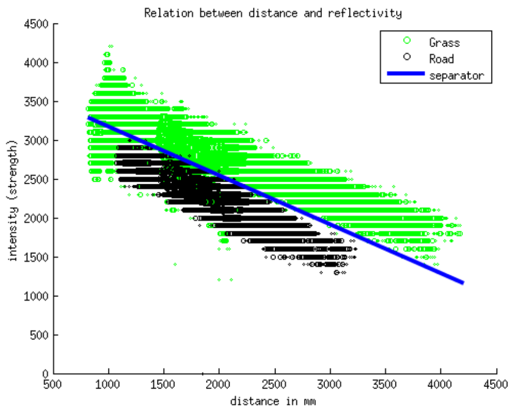

Fig. 6. Separator to detect grass with the help of the laser scanner [5].

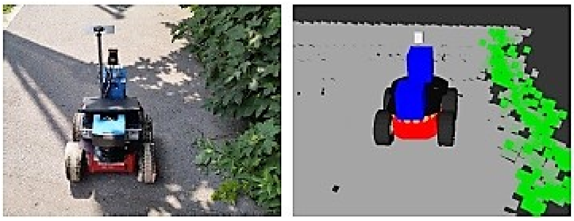

Fig. 7. Detection of the grass though the laser scanner [5].

We start our discussion of related research with the robot presented in [11]. The robot could take a long tour through Munich without a prior created map or GPS information. Instead, the robot was using its sensors to react locally in a safe manner and asked humans for information about the direction. This was done by approaching humans and the recognition of basic commands to derive the direction of the desired destination. In contrast, our robot has a prior created map which allows it to move autonomously without asking for directions. This is also desirable in the case of a transport robot which should transport goods to a customer.

In [12] the method to deal with large maps was described. The authors use a topological map to allow an efficient representation of large areas. The vertices in the topological map are spots of interests such as a square or a crossing. The edges represent paths between these places. For each edge, a traversal behavior is defined. Thus, one can use different behaviors to perform the traversal. With the help of this method, the robot could drive autonomously in a park. Our robot uses, in contrast, a topological map which contains enough information to allow the robot to be always localized not only in interesting places. Furthermore, the robot uses a denser road map allowing it to plan its route more accurate.

A very close related work to ours was presented in [13]. The robot navigated more than 3km in the city Freiburg in an autonomous fashion. To localize itself, the robot used a topological map where each vertex in the graph contains a map of one part of the environment. In contrast, our approach additionally used the GPS signal for estimating the robot pose within the particle filter. To navigate the robot, the method presented in [13] created a high-level plan using the graph of the topological map. Each vertex is connected to those vertices in the graph which allow moving between these two locations. Thus, using this graph the robot can derive a simple high-level plan for the navigation. Whereas the robot uses a planner on grid map basis to navigate between different vertices of the topological map. This contrasts with our approach as we use a finer grained road map for the high-level planning which allows us to choose the path more precisely.

VI. Conclusion and Future work

The transportation of goods is an essential part of our today's economy. The transportation often takes place in outdoor environments by delivering goods to costumers. To provide cost-efficient and flexible deliveries, robots are a promising solution.

In this paper, we presented an autonomous transport robot which is capable of navigating in large scale outdoor environments. To perform this transportation, the robot addresses the problem of a large-scale environment, uneven ground, and grass which should only be traversed if necessary. To deal with the large scale of the environment the robot uses a topological map. This map stores areas of the environment which are loaded on demand. This allows that the robot only needs to keep a small part of the environment in its memory and perform the localization on it. We furthermore showed how the robot can exploit the topological map to switch between the different parts to allow the robot to be localized during the complete delivery. To deal with the uneven ground, the robot builds an elevation map for its local environment. Afterward, the robot determines within the elevation map dangerous terrain and avoids it. To deal with the grass we have shown a simple solution with a linear classification for laser scan measurements. This detection allows the robot to detect grass precisely enough to avoid the grass if possible.

The robot presented in this paper mainly used several laser scanners to localize itself and it is left for future work to add more sensors to perform localization as well as navigation. Especially cameras would be of interest as they allow a detailed localization in many areas which don't offer features for a laser scanner. The additional use of a camera would increase the quality of the terrain classification.

References

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," in *Proceedings of the 19th national conference on Innovative applications of artificial intelligence - Volume 2*, ser. IAAI'07. AAAI Press, 2007, pp. 1752–1759. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620113.1620125

[2] E. Guizzo, "Three Engineers, Hundreds of Robots, One Warehouse," *Spectrum, IEEE*, vol. 45, no. 7, pp. 26–34, 2008.

[3] C. Mühlbacher, S. Gspandl, M. Reip, and G. Steinbauer, "Improving dependability of industrial transport robots using model-based techniques," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3133–3140.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[5] K. Lassnig, “An autonomous robot for campus-wide transport tasks,” Master’s thesis, Faculty of Computer Science and Biomedical Engineering, Graz University of Technology, 2016.
[6] S. J. Julier and J. K. Uhlmann, “New extension of the kalman filter to nonlinear systems,” in *AeroSense’97*. International Society for Optics and Photonics, 1997, pp. 182–193.
[7] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” *AAAI/IAAI*, vol. 1999, no. 343-349, pp. 2–2, 1999.
[8] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[9] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
[10] A. Kleiner and C. Dornhege, “Real-time localization and elevation mapping within urban search and rescue scenarios,” *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 723–745, 2007.
[11] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss, “The autonomous city explorer: Towards natural human-robot interaction in urban environments,” *International Journal of Social Robotics*, vol. 1, no. 2, pp. 127–140, 2009.
[12] K. Košnar, T. Krajnık, V. Vonásek, and L. Preucil, “Lama-large maps framework,” in *Proceedings of Workshop on Field Robotics, Civilian-European Robot Trial*, 2009, pp. 9–16.
[13] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, “Autonomous robot navigation in highly populated pedestrian zones,” *Journal of Field Robotics*, vol. 32, no. 4, pp. 565–589, 2015.