Mobile Robots for the Simultaneous Exploration and 2D Determination of Radioactivity

MICHAEL MARSZALEK, MARTIN EDER
ANDREAS TROPSCHUG and ALOIS KNOLL
Technische Universität München
Department of Informatics -
Robotics and Embedded Systems
Boltzmannstr. 3, Garching bei München
GERMANY
{marszale, martin.eder, tropschu, knoll}@in.tum.de

HAGEN HÖFER
HÖFER & BECHTEL GmbH
Ostring 1, 63533 Mainhausen
GERMANY
info@hoefer-bechtel.de

Abstract: In the past years, various robotic inspection systems have been developed for industrial automation. The industry offers a wide range of applications for robotic systems. The inspection and cleaning of severs or an automation of a biotechnology laboratory are successful examples for robotic systems and automation [1]. In this paper, we present a novel system for the autonomous 2D determination of radioactivity and contaminated areas. The importance of autonomous determination of radioactivity, in the course of disassembly of nuclear power plants, will rise in the upcoming years. It enables remarkable possible savings, concerning work time and personnel expenses. Our mobile robot is able to scan incrementally the whole floor of a building, in order to assure clean environments in nuclear power plants or castors, and marks the contaminated areas in a generated map of the building for further evaluation. The paper discusses the main requirements of a navigation system for 2D detection of radioactivity, presents the hardware and software set-up and some real-world experiments with our mobile robot.

Key–Words: Industrial Application, Automation in Medicine, Automation, Robotics, SLAM, Path planning

1 Introduction

Generally, the challenging field of robotics has an enormous potential to support the industry in different fields. A robot is able to perceive and manipulate the physical world by means of different devices and can act in various working environments, for example planetary explorations, in the car industry or as a demonstrator in a theatre.
The market for service robots has an excellent chance for the future. An example for successful commercial systems are cleaning robots which have been studied and developed some years ago. Meanwhile, the market offers some cleaning robots for private purpose. The common enabling technology for a cleaning robot and our application is autonomous navigation in everyday indoor environments. In contrast, the main difference is the efficiency of our navigation system. While a hoover robot usually cleans the floor randomly, our system aims to document all radioactivity measurements for every reachable position in the environment. The handling with radioactivity in a nuclear power plant or radiology demanded some requirements, concerning the compliance of safety measures. One important safety task is the securing of a contamination-free working environment, typically needed in a nuclear power plant. Usually, the measurements are done by hand and thus a member of staff has to invest a lot of time for the safekeeping of a contamination-free building. It is necessary to prepare a documentation of every checked plane of the floor or walls. In the case of contamination, the exact position of the determined area has to be marked in a map of the building. The application of robots could be extremely timesaving, unforgeable and reduces physically demanding occupations. Our approach was driven by the idea for automation of the safekeeping of a contamination-free environment. We equipped a mobile robot with various devices which focuses on the detection of α, β and γ radiation in castors and nuclear power plants. The complexity of

Figure 1: Mobile Robot

our system is manageable because we focused on the floor of the areas. Thus, we did not need a robotic arm in order to reach walls and ceilings. The applied mobile robot, called *Robotino* (see Figure 1), is equipped with an omnidirectional drive, three wheels and multiple sensors. It is able to move in all directions, as well as to turn on the spot through its recent drive system. The control of the actuating elements and acquisition of sensor data results from the WLAN module. Alternatively, it is possible to use a network cable for the communication with Robotino. Festo Didactic offers a software named "Robotino View", which is an interactive graphic programming tool for Robotino. Furthermore, an open-source framework, called OpenRobotino[1], and the corresponding documentation are available for the development of applications. Robotino was additionally equipped with a laser rangefinder (SICK S300), two radioactivity detectors (CoMo 170) and an Intel Core Duo notebook for the developed navigation framework.

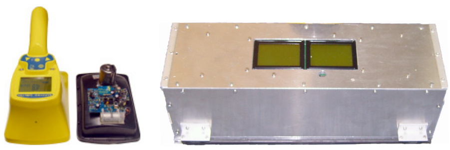

Figure 2: CoMo 170

The contamination monitor (see Figure 2) serves for the detection of radioactive contaminated surfaces. The hardware was built into a new box, which is mounted in front of Robotino. This box contains two detectors to attain a complete cover. The standard PC of Robotino is not capable of fulfilling the computation power requirement, why we run the navigation framework on a mounted notebook. A full, in-depth presentation of all features of our system is beyond the scope of this paper. Rather, we will focus on the core functions of the software respectively mobile robot. In section 2, we will briefly describe some recently developed robotic and navigation systems, followed by the overall architecture of the system in section 3. To validate our approach we perform multiple experiments in section 4 to confirm the accuracy of our system. For this purpose, we received a permission for an experiment in a laboratory of the Max-Planck institute in Mainz. The Alpha Particle X-Ray Spectrometer (APXS), which determines the consistency of stones and dust on the mars surface, was originally developed in this laboratory.

[1]http://www.openrobotino.org

2 State of the Art

In the past, several efforts have been made, regarding the creation of autonomous mobile robot systems. Since every creation had its specific requirements and involves the combination of several algorithms, we will try to give a review for our work below. A mobile robot for the automation of the complete sample management in a biotechnology laboratory is presented in [2]. The navigation of this mobile robot is based on a generated static map of the current working environment. An other example, presented in [3], is the service robot LISA (Life Science Assistant) which supports lab personnel in biological and pharmaceutical laboratories. It makes automated experiment cycles flexible and helps employees to prepare experiments, e.g. by collaboratively executing transportation tasks or filling microplates. A Hokuyo URG-04LX is used for the localization in an a-priori map and the collision detection. The map of the laboratory is handmade and is initially used for the localization. A generation of a static map of the buildings is time-consuming and presumes a new map for every domain. Our application needs a flexible adoption of the robot, which requires the ability to map and explore unknown environments right from the start. The main technology for this purpose is called Simultaneous Localization and Mapping (SLAM) and has been in the focus of many researchers for several years. SLAM deals with the problem of constructing an accurate map of an environment in real-time. Thereby, it has to handle imperfect information about the robot's trajectory through the environment. For example, a robot which was placed in an unknown environment, incrementally builds a consistent map of its surroundings, while simultaneously determining its location within this map. OpenSlam [2] offers some open-source SLAM implementations like GMapping or DP-SLAM. A Rao-Blackwellized particle filter based approach (GMapping) is presented in [4][5]. It is able to produce accurate maps of indoor and outdoor environments using occupancy grids. The open-source library requires some programming effort for the adoption and only runs under Linux systems. Another framework, called CARMEN [6], provides basic navigation primitives like path planning, mapping and localization. We used some generated log files for the evaluation of the mapping module. It works for small environments correctly, but it was not able to resolve the loop closing issue in large buildings. The commercial framework KARTO [3] is a very efficient solution for navigation tasks. It provides exploration strategies, mapping and trajectory planning.

[2]http://www.openslam.org
[3]http://www.kartorobotics.com

A free functional trial copy is available for download. The evaluation of the mapping, using our carmen log files, has been very promising.
[7][8] DP-SLAM needs odometry measurements and an accurate laser rangefinder, e.g. SICK S300, for localization and building a map. It uses a particle filter to maintain a joint probability distribution over maps and robot positions, as well as some efficient data structures, which allow an efficient mapping.
DP-SLAM does not need predetermined landmarks and is accurate enough to close loops without any special off-line techniques. The data association problem was also eliminated through the abandonment of landmarks. Moreover, it is not necessary to predetermine the environment.

In almost no exception, none of the open-source distributions can be used without additional programming effort. The Karto framework is an ideal solution for mobile robots, but we prefer an open-source software, due to the costs for the licence. GMapping is an efficient and accurate SLAM algorithm but it needs some additional adaptation for our application. In contrast, DP-SLAM was implemented and successfully tested during a past project of our faculty. It runs under Windows and Linux and can be easily configured for our mobile robot.

3 Software Architecture

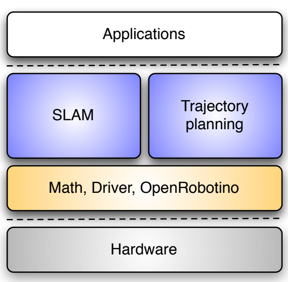

Figure 3: Software Architecture

Our navigation framework (see Figure 3) consists of a SLAM library, which interacts with a trajectory planning module. Thus, the robot is able to plan a path within known parts of the map. The radioactivity detectors log the results of the measurements, while driving through the building. The hardware is accessed via an API for Robotino and some additional drivers for the laser rangefinder and the radioactivity detector. Based on the software architecture, it is possible to develop various applications for mobile robots. The topic of the paper is focused on radioactivity detection, and therefore we will address its main components (SLAM and Trajectory planing) below.
A **SLAM** library for our application needs to meet the following requirements:

- Accurate determination of the pose
- Correct map of the building
- Handling of dynamic objects
- Real-time ability
- Easy configuration for various scanners
- Usability for Windows and Linux

Our SLAM library is based on an algorithm, called DP-SLAM [7][9] and fulfils all listed requirements, except handling of dynamic objects. Generally, this requirement addresses the interaction with humans. We avoid collisions with them by using the integrated bumpers and laser rangefinder measurements. Our group has long experience working with this SLAM library and the results were always promising for indoor buildings. The library works with various laser scanners and allows an easy configuration, for example particle amount or scanner resolution. It produces an occupancy grid map as output which is the basis for the trajectory planing. This is why we preferred this algorithm instead of familiarise with a new framework. We briefly review the basics of the algorithm and refer the reader to the detailed description of the algorithm in [7]. This slam algorithm is based on a particle filter and an ancestry tree. A particle filter, also called Monte Carlo method (SMC), is one possible approach for localization. Like in a Hidden Markow Model, it is necessary to define a state transition, a hidden state and the observation. In this case, the robot's position is the hidden state that should be tracked. The state transitions represent the movements of the robot and the corresponding observations are extracted from the sensor readings. All sensor readings are noisy or ambiguous, because no scanner is adequate enough to resolve ambiguities. The state transitions are represented by a motion model, which acts as the proposal distribution in the sampling process of the particle filter.
The ancestry tree is the core data structure for an efficient maintenance of particles. It represents the elapsed time during the mapping process. The leaves represent the current particles and a corresponding occupancy grid.
The **Trajectory Planning** is based on occupancy grid data, provided by the SLAM library. We differentiate between disturbances, dynamic stationary obstacles, free space and static obstacles by using probability values of the occupancy grid. For example, a

high probability in a cell stands for a static obstacle. The trajectory planning is continuously updated with the current pose of the robot and the occupancy grid, which represents the environment. These information are used by the algorithm in order to place waypoints in free space. Each *waypoint* describes a possible position of the robot. The distance between the waypoints and the distance to walls can be set optional, but it has to be chosen carefully to avoid collisions with static objects. A bounding box (geometry of the robot) is placed on every possible position next to the robot and a validation outputs the usability of this position. If the box is located inside the explored area and no obstacle overlaps the box, the algorithm accepts that position as a possible waypoint. A road represents the way between two waypoints, and it also enables a collisions-free driving.
Road mapping is implemented by a graph data structure. It reduces the problem of negotiating a path through a complex shaped 2D representation of the world to finding a shortest path through a connected graph from node A to node B, for example.
The trajectory planning computes a path through the known part of the environment, starting from the robot's current position. The robot drives to every generated waypoint, as long as a waypoint exists in the map. The map of the environment is regularly updated with changes in the surrounding and registrations of unseen parts of the building. The trajectory module reacts accordingly to these changes and extends the path by new waypoints in these uncovered areas. Thus, we abandon an **Exploration phase** which would prolong the whole radioactivity search process. The only one constraint for this approach is the securing of a closed area. This is necessary to stop the trajectory planning from exploring new parts of the building.
The **Documentation** of the radioactivity measurements is done by logging all measurements for every reachable position in the map. For this purpose, the geometry of the box with its detectors, as well as the deviation from the centre of the robot has to be defined. Additionally, all results are marked in the corresponding map. For example, the detection of radioactivity is marked as a radioactivity sign and clean areas are repainted in the map in green. The position of detected radiation has a deviation of +/- 3 cm, due to uncertainties in the localization process.

4 Results

To validate the SLAM library we performed an experiment at the corridor of the computer science department in Munich (see Figure 4) to confirm the accuracy

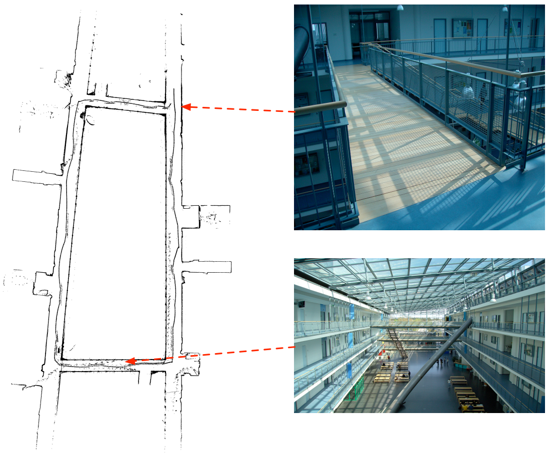

Figure 4: Third floor of the Computer Science Department

of the mapping. The evaluations were done online and off-line, in order to experiment with different parameters. The offline results are more accurate, due to the higher amount of particles, but also need a lot of computation time. The online tests were done with optimized parameters in order to enable a real-time mapping while exploring a building.
The closing loop problem is one of the difficulties in the mapping process. The map has to be accurate after completing a loop. The robot traveled approximately 130 m with a speed of 10 m per minute for this experiment. The size of the environment is roughly 39 m x 16 m. It is a challenging domain because of its asymmetric characteristics. Figure 4 illustrates, that is was possible to complete the loop, applying 100 particles for the localization and a grid resolution of 35 grids per meter. This figure was generated in an online modus while driving through the corridor. So far, it was not possible to map a bigger area. It leads to bad performance and the probability of losing its own position dramatically arises. Our approach is designed for laboratories, which usually have a smaller geometry. Thus, we focused on the overall performance of our system in the next experiment. The second scenario (see Figure 5) illustrates the operation in a laboratory of the Max-Planck Institute in Mainz, while interpreting the measurements of the radioactivity detectors. The robot is randomly placed and starts its exploration. Immediately after the first mapping of the subarea, the trajectory planning starts setting the waypoints in the map. We used the same configuration for the slam algorithm as in the last described experiment. The maximal accepted distance of the S300 is 8 Meter with $0.5°$ angle resolution over a $270°$ area. The waypoints keep a distance of 0.2 m to the wall and to each other. We decided for a small dis-

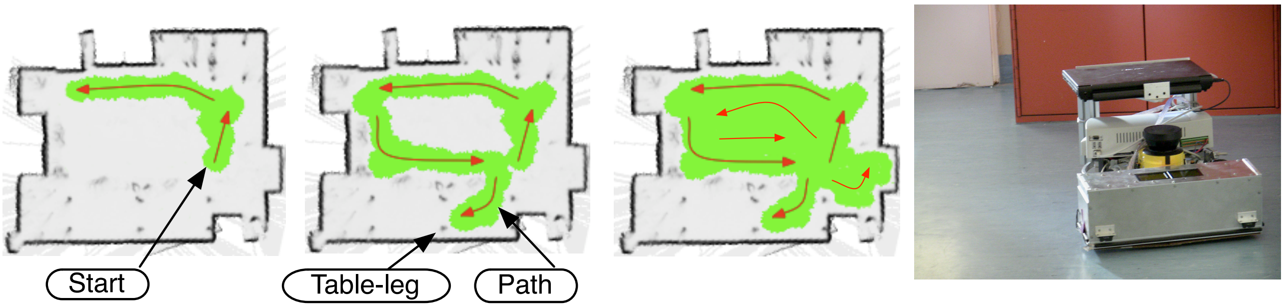

Figure 5: Search for Radioactivity

tance between the waypoints because it enables an approximated continuous path around obstacles and we wanted to assure to cover the whole floor for the radioactivity measurements. After reaching a waypoint, using its omnidirectional drive, it marks this position as completed and takes the next waypoint as target position. Finally, it traverses the seen section of the environment by itself until every possible location in the map was reached. The red arrows shows the direction of the robot's movement. It starts to drive with a speed of 40mm per second along the wall and continues its search in the middle of the room until every reachable position was completed. The uncovered areas next to the walls resulted from a safe distance to the walls. Figure 5 also shows that the robot did not cover the areas under the tables. The geometry of the robot did not allow to drive in such narrow passage. The search for radioactivity was accomplished without any collisions. Before this assignment, the functionality of the radioactivity detectors were tested using a gas mantle from a gas-powered lamp. The green area stands for the clean bottom. In the case of detected radiation a radioactivity sign would mark the corresponding position in the map. The robot drove with a speed of 40 mm/s and needed around 1 hour for the laboratory. The radioactivity detectors have a minor sampling rate why we need to assure a slow driving. Approximately 70% of the laboratory can be covered at the moment. We work on some extensions, which should solve that problem and guarantee a 100% coverage of the bottom.

5 Conclusion

In this paper, we have presented a completely new application for securing of a contamination-free environment. So far, a comparable approach has never been deployed for this purpose. We extended a SLAM approach with the ability to plan a path and drive autonomously through an indoor environment. The DP-SLAM re-implementation with its extensions enables an adoption for various robots or laser rangefinders. There are some disadvantages, which have to be solved in future works. The SLAM algorithm is not able to map outdoor environments in real-time. We suggest to implement a Rao-Blackwellized particle filter which is able to compute an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation [5]. This approach reduces the uncertainty about the robot's position and allows to track its own position in large-scale and outdoor domains. Another improvement is the implementation of an efficient occupancy grid through a Quadtree, for example. Both proposals could drastically speed up the performance of the slam algorithm.
The strategy for an optimal coverage of the floor needs to be adapted in order to guarantee a 100 % coverage for all possible environments. The experiments within this framework have been promising. That is why we want to extend the progress in near future. We aim for using our approach for a real industrial application.

Acknowledgements: The authors would like to acknowledge the support of the Höfer & Bechtel GmbH and Robotic Equipment Corporation GmbH for this research. We also thank Gerd Bauer for many helpful suggestions and comments.

References:

[1] H.-B. Kuntze, H. Haffner, Experiences with the development of a robot for smart multisensoric pipe inspection. In Proc. 1998 IEEE International Conference on Robotics and Automation (ICRA 98), Leuven, Belgium, May 1998

[2] Martin Wojtczyk and Michael Marszalek and Alois Knoll and Rdiger Heidemann and Klaus Joeris and Chun Zhang and Mark Burnett and Tom Monica: Automation of the Complete Sam-

ple Management in a Biotech Laboratory. In: BCS Int. Acad. Conf., 2009-09-21

[3] E. Schulenburg, N. Elkmann, M. Fritzsche, A. Girstl, S. Stiene and C. Teutsch. LiSA: A Robot Assistant for Life Sciences. In Proc. of the 30th Annual German Conference on AI (KI 2007), Osnabrck, Germany, September 2007

[4] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard: Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters

[5] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard: Improving Grid-based SLAM with Rao-Balckwellized Particle Filtlers by Adaptive Proposals and Selective Resampling

[6] Michael Montemerlo, Nicholas Roy, Sebastian Thrun: Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2003

[7] A. Elizar and R. Parr: "DP-SLAM 2.0". In: Proceedings of the IEEE International Conference on Robotics and Automation, May 2004

[8] A. Elizar and R. Parr: DP-SLAM:Fast,Robust Simultaneous Localization and Mapping without Predetermined Landmarks. In: IJCAI-03, 2003

[9] A. Elizar and R. Parr: Hierarchical Linear/Constant Time SLAM Using Particle Filters for Dense Maps. In: NIPS-05, Dec 2005

[10] Teddy N. Yap, Jr. and Christian R.Shelton: Simultaneous Learning of Motion and Sensor Model Parameters for Mobile Robots. In: IEEE International Conference on Robotics and Automation, 2008

[11] Michael Marszalek.: Simultaneous Localization and Mapping in Combination with Machine Learning Methods. Diploma Thesis 2008

[12] Sebastian Thrun, Wolfram Burgard, Dieter Fox: Probabilistic ROBOTICS 2005