Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Indoor SLAM using architectural plans

Bc. Jakub Havlíček

Supervisor: Prof. Dr.-Ing. Hartmut Bruhm
Field of study: Cybernetics and Robotics
Subfield: Cybernetics and Robotics
May 2019

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Havlíček Jakub**	Personal ID number:	**420400**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Control Engineering**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Cybernetics and Robotics**		

II. Master’s thesis details

Master’s thesis title in English:

Indoor SLAM using architectural plans

Master’s thesis title in Czech:

Vnitřní SLAM použitím architektonických plánů

Guidelines:

Problem:
In the master project proposed here, it shall be investigated and demonstrated how the quality of a SLAM can be improved if architectural plans are considered in the process.
Goals:
- Localization of the robot relative to inventory plans
- Fusion of inventory plan and occupancy grid from SLAM process
- Improving long-term stability of the SLAM proces

Solution approach:
- Mobile robot equipped with odometric sensors, LIDAR, IMU and Kinect-II
- Control architecture based on ROS and HECTOR SLAM
- Matching of selected regions of occupancy grid with architectural elements (walls, doors,…)
- Testing in Gazebo simulator and real world scenarios

Scientific challenges:
- Robustness w.r.t. deviations from inventory plan (additional obstacles: furniture, people, other robots) and drift in the localization process

Bibliography / sources:

[1] Thrun, S.; Burgard, W.; Fox, D.; Arkin, R. C. (2005): Probabilistic Robotics: MIT Press.
[2] Quigley, M.; Gerkey, B.; Smart, W. D. (2015): Programming Robots with ROS: A Practical Introduction to the Robot Operating System: O'Reilly Media.
[3] Förter-Grauel, David (2018): Kooperative Exploration mit autonom agierenden mobilen Robotern. Masterarbeit, Hochschule Aschaffenburg.
[4] Fellhauer, Daniel (2018): Kooperationsstrategien für autonome mobile Roboter. Projektphasenbericht I, Hochschule Aschaffenburg.

Name and workplace of master’s thesis supervisor:

Prof. Dr.-Ing. Hartmut Bruhm, University of applied sciences Aschaffenburg

Name and workplace of second master’s thesis supervisor or consultant:

doc. Ing. Tomáš Svoboda, Ph.D., Vision for Robotics and Autonomous Systems, FEE

Date of master’s thesis assignment: **24.01.2019** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until:
by the end of summer semester 2019/2020

Prof. Dr.-Ing. Hartmut Bruhm Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Acknowledgements

I would like to thank my supervisor Prof. Dr.-Ing. Hartmut Bruhm for the comprehensive support of my research. I would also like to thank my partner, family and friends for their psychological support during this work.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 24. May 2019

Abstract

Accurate localization and mapping help the robots to navigate in the environment. In this thesis, it is investigated how the quality of simultaneous localization and mapping can be improved by priors extracted from architectural plans. Particular attention is paid to the influence of integrating minor odometry errors and environmental variations from the architectural plans.

Keywords: SLAM, occupancy grid, ICP, architectural plans, ROS, mobile robot

Supervisor:
Prof. Dr.-Ing. Hartmut Bruhm
University of Applied Sciences,
Würzburger Straße 45,
63743 Aschaffenburg

Abstrakt

Přesná lokalizace a mapování pomáhají robotům orientovat se v prostředí. V této práci je zkoumáno, jak může být kvalita simultánní lokalizace a mapování zlepšena architektonickými plány. Zvláštní pozornost je věnována vlivu integrace drobných chyb odometrie a environmentálních variací z architektonických plánů.

Klíčová slova: SLAM, mřížka obsazenosti, ICP, architektonické plány, ROS, mobilní robot

Překlad názvu:
Vnitřní SLAM použitím architektonických plánů

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

Accurate localization and precise mapping is essential for a mobile robot in order to be able to successfully operate and to be an effective help for humans. Those are challenging tasks, that has not been completely solved in the real world conditions yet [1]. Mobile robots need maps to be able to navigate independently in the environment.

As shown by the last major nuclear incident in Fukushima power plant, for some tasks it is not only more appropriate to use robots, but it is the only viable option. If the level of radiation is life-threatening, the mobile robot is the only possible answer to the requirement of performing tasks in this environment.

A contaminated environment has a specific feature that although it may be life-threatening to humans, it does not have to be physically destroyed. The contaminated buildings may remain structurally intact. In such scenario, a mobile robot could benefit from prior knowledge about the environment extracted from available architectural plans. If the processing of the architectural plans could be further automated, the robot's deployment time could be reduced. Faster response to the incident could be beneficial.

1.2 Problem

This thesis shall investigate how the quality of simultaneous localization and mapping (SLAM) process can be improved with prior knowledge of the architectural plans of the building where the mobile robot is located. It should further demonstrate possible improvement in SLAM process if the prior information extracted from architectural plans are to be considered.

The robustness against the deviation between the environment and the global map created from architectural plans poses a specific scientific challenge. These deviations are caused by obstacles that the architectural plan does not contain or that are not extracted from it. For example, the deviations can be present due to additional furniture, people or other robots.

The drift in the localization process commonly associated with odometry measurements presents another specific challenge. It shall be investigated how the prior knowledge of the architectural plans can correct such drift.

1.3 Goals

Particular goals of this work are to localize the robot in the architectural plans, to fuse the architectural plans with the occupancy grid (OG) and to ensure stability of the SLAM process.

First, the robot should be able to find the position and the rotation of itself relative to the architectural plans. The estimate of the last position is assumed to be known. The goal is not to solve the task of global localization.

Second, the available architectural plans should be appropriately fused with the OG from SLAM process. OG is common form of map representation for mobile robots.

Third, the long-term stability of the SLAM process should be improved using the architectural plans. The SLAM process may be prone to the integration of small errors over time if it relies solely on inaccurate relative measurements.

1.4 Methodology

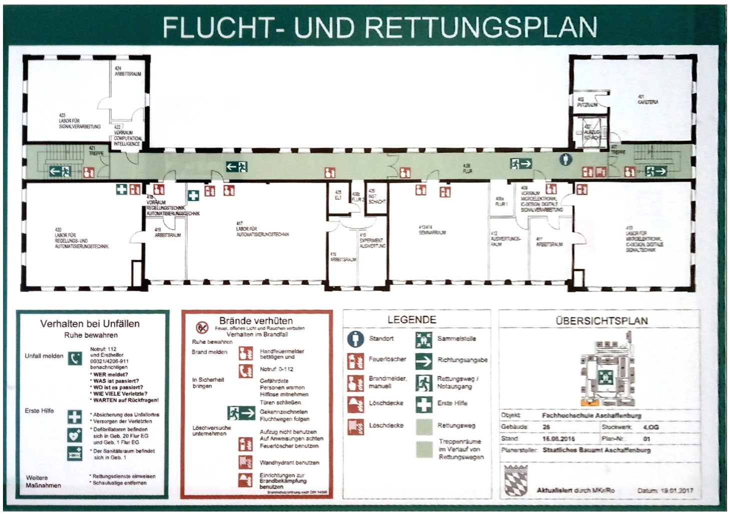

Figure 1.1: Escape plan[2]

The task should be solved for the EtaBot robot developed at the University of Applied Sciences in Aschaffenburg, Germany. It is mobile wheeled robot equipped with various sensors. The EtaBot robot is described in more detail in chapter 3.

The EtaBot is using control architecture based on Robot Operating System (ROS), therefore the solution should be implemented using the same architecture. The solution shall be experimentally verified in the Gazebo simulator under realistic condition. It should build upon previous work and it should embrace the existing platform. The solution is described in detail in chapter 5.

An escape plan placed on a wall on the 4th floor, by which the Gazebo model of the floor is created, in the building number 26 on the campus of the University of Applied Sciences in Aschaffenburg is shown in figure 1.1.

Chapter 2

Architectural Plans

In this work, the terms architectural plan, floor plan, inventory plan, architectural model and architectural drawing are used interchangeably. All of them describe a representation of a structure or an environment that the robot can traverse. This representation is usually created using computer-aided design (CAD) in advance for the purpose of construction of buildings. There are many different types of CAD software used to achieve this goal. Example of a computer model of a building is shown in figure 2.1.

Figure 2.1: Computer model of a building[3]

Architectural plans can contain different level of detail. Among the elements that the plan may include are walls, doors, windows, plumbing, power lines and other utilities. In some cases, architectural plans might be even supplemented with furniture or equipment. The German language has distinct word **Bestandsplan** for such architectural plan, which unfortunately does not have clear direct equivalent in neither the Czech or the English language. It could be translated as an architectural plan complete with planned building inventory of furniture and other equipment.

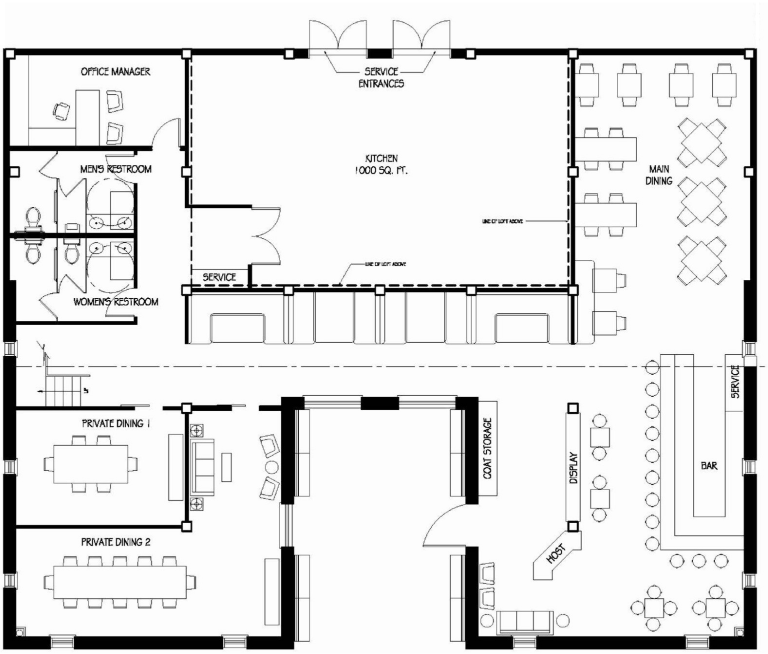

Figure 2.2: Human readable representation of a floor plan[4]

The advantage of using the architectural plans for aiding the SLAM process is that they do not have to be built specifically for this task, as they were already created for the construction of the building and therefore are available. Architectural plans can be processed directly, which theoretically allows to retrieve all of the details that they contain, but it also possesses a challenge of conversion from different file formats. Another option might be exploitation of exported human readable image. Example of such representation is shown in figure 2.2 The latter method also allows to use architectural plans of older structures, that were not created with the help of CAD. The downside being loss of some detail, that would have to be labeled by hand if needed.

Chapter 3

EtaBot Robot Hardware

Figure 3.1: EtaBot robot[5]

Figure 3.1 shows a picture of EtaBot robot, that is being developed at the University of Applied Sciences in Aschaffenburg. This robot is used as a development platform for this thesis. Figure 3.2 shows relation between different coordinate systems relevant for the task. Laser scanner measurement has to be transformed from laser scanner coordinate frame, in green color, to the robot coordinate frame, in blue color, before the relative position of robot coordinate frame in world coordinate frame, in red color, can be estimated.

The EtaBot is small mobile robot equipped with various sensors. Its sensory equipment include scanning laser rangefinder, Microsoft Kinect, ultrasonic sensors, incremental encoders in wheels and inertial measurement unit (IMU), according to [6].

The sensors can be generalized into two types, based on what kind of measurements they provide. The first type of sensors provides a scan of environment, which can be covnerted to pointcloud.

The first group includes laser scanner, Microsoft Kinect and to some extent the ultrasonic sensors. The second type of sensors provides relative position change of the robot, which can be considered as a measurement of odometry.

The second group includes the incremental encoders in wheels and the IMU. Sensors providing the same type of measurement can be theoretically interchanged, thus there is no need to solve the task for different sensors, but it is sufficient to consider the type of data that the sensors provides.

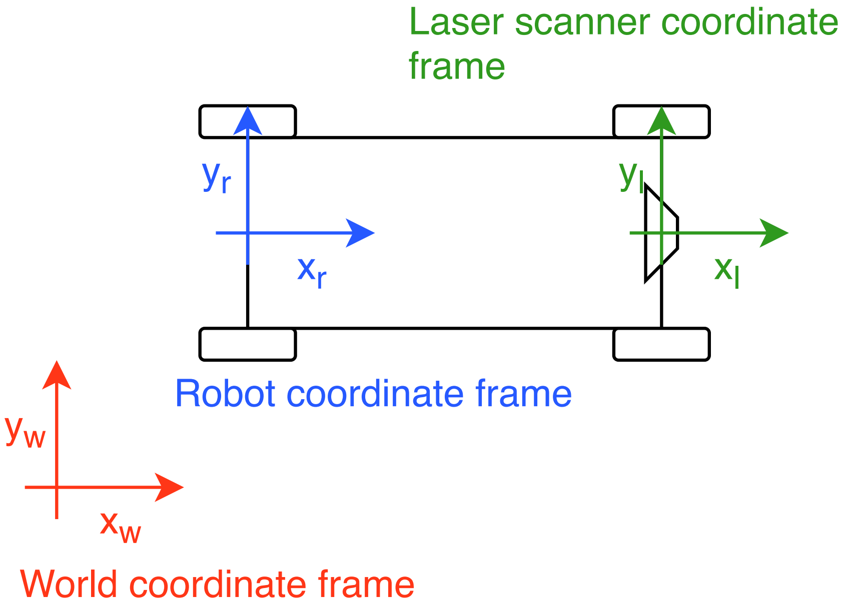

Figure 3.2: Coordinate frames

Chapter 4

Related Work

Boniardi et al. investigate robust robot localization in architectural plans in article [7] from 2018. They adapted general iterative closest point (ICP) algorithm for scan to architectural plan matching. Their system uses graph instead of occupancy grid to represent map.

They assume completely static environment during scanning for the purposes of defining and modeling the problem. Although they allow change of the environment between scans. They localize the robot in prearranged global map. The graph based representation, that they use, seems to be propitious alternative to occupancy grid based approach.

Advantage of method described in article [7] is, that it does not require part of the architectural plan to be continuously visible. It only assumes that the first scan contains recognizable part of the architectural plan. It leverages past observations and graph based map aligned to architectural plan.

Therefore it is able to correctly localize the robot even if it encounters a part of the environment that cannot be distinguished by a pointcloud measurement from the architectural plan. Such condition may be caused, according to the article, for example by a series of cabinets covering the full length of a wall. They further note, that they do not attempt to solve the global localization of the robot.

Boniardi et al. build upon their previous work [8] published a year earlier. They argue in this article, that human readable architectural plans are easier to understand for humans than specialized map representations mobile robots usually work with. In this article, they propose the same procedure based on general ICP algorithm with graph based map representation. They assume an architectural plan to be encoded as a binary image with known resolution. They conclude, that the proposed method works robustly and is comparatively good with state-of-the-art approaches.

Georgiou et al. address the problem of converting the architectural plans to a format appropriate for SLAM process in article [1]. Their method is capable of extracting walls and doors from an input architectural plan in human readable form. They suggest that additional information could be extracted from the input image as well. They conclude, that this process can be automated, but the possible benefits of extracted information in SLAM are yet to be investigated.

Chapter 5

Solution

5.1 System Overview

The EtaBot robot is equipped with various sensors that allows it to sense the surrounding world. Those sensors can be divided into two main categories. First category sensors provide data in a form of set of geometrical points in sensor coordinate frame, that are representing the presence of real world objects. Those sets shall be called scans. The second category sensors provide data as an estimation of position consisting of translation and rotation of the robot.

It was decided to restrict the robot movement to a plane, as solving the problem in two dimensions might significantly reduce its complexity. This decision can be justified by the fact, that there are usually flat floors in buildings under normal circumstances. Figure 5.1 shows, how individual components of the solution interact with each other.

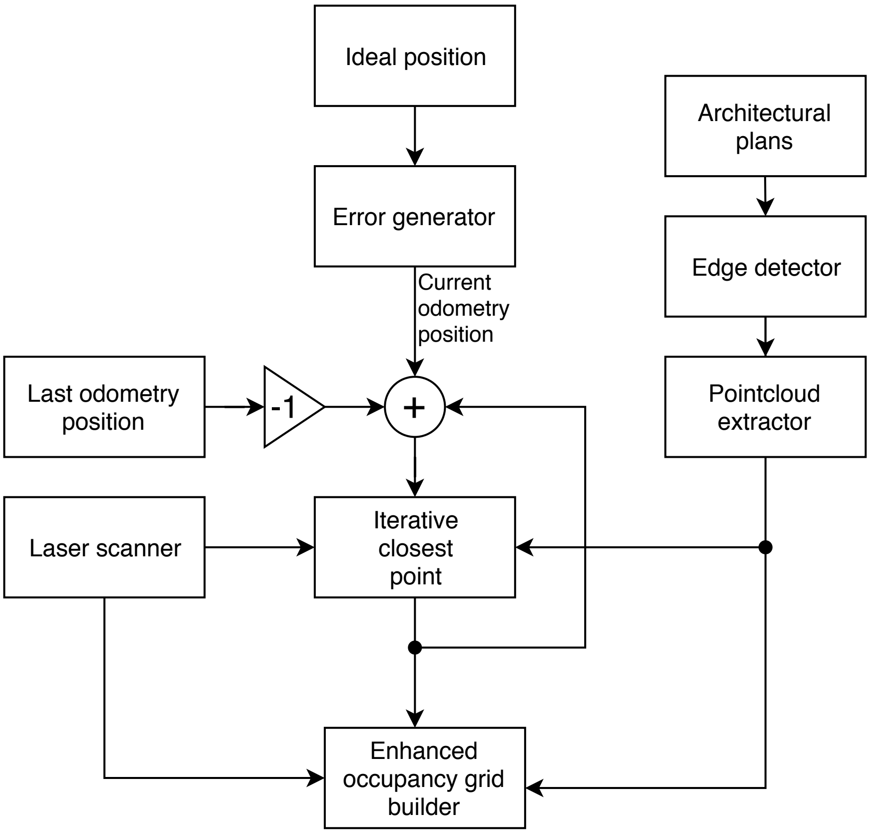

Figure 5.1: System scheme

It is assumed, that a sufficient part of the architectural part is continuously visible. The ICP algorithm is used to estimate the robot's position in the architectural plan, this algorithm further refines the position estimation given by sensor measurements. Namely by the odometry sensors. Odometry sensors provide prediction of the relative robot movement. According to article [7], ICP algorithm with laser scanner measurements has been proved highly robust.

However, odometry constantly collects small mistake. So although the initial position of the robot in the world is assumed to be known, the odometry itself cannot be effectively used for robot localization on its own. It needs suitable corrective measurement of the absolute position of the robot. In this case the corrective element is provided by the ICP algorithm.

ROS node for ICP position estimation was created. It remembers the old position estimate, adds change in position estimate from odometry sensors and computes new position estimate by matching current scan to world map. World map consists of sparse 2D pointcloud, which is generated in advance and cached because of performance reasons.

Subsampling technique was used to reduce the required computing power. From each scan, only 50 % evenly spaced points was used.

5.2 World Map

World map is created from architectural plans. An input are architectural plans of the building in human readable form. Preprocessed image of architectural plans was used, but Georgiou et al. show in article [1], that this processing can be automated and thus the architectural plans can be used directly in the human readable form shown in figure 2.2.

The preprocessed image is containing only walls, as shown in figure 5.2. Canny Edge detector, developed by Canny in 1986 [9], is then applied to the input image to retrieve the locations of wall surfaces. Georgiou et al. mention in article [1], that Canny Edge image processing detector has been previously used in the research of construction of SLAM priors from architecture plans.

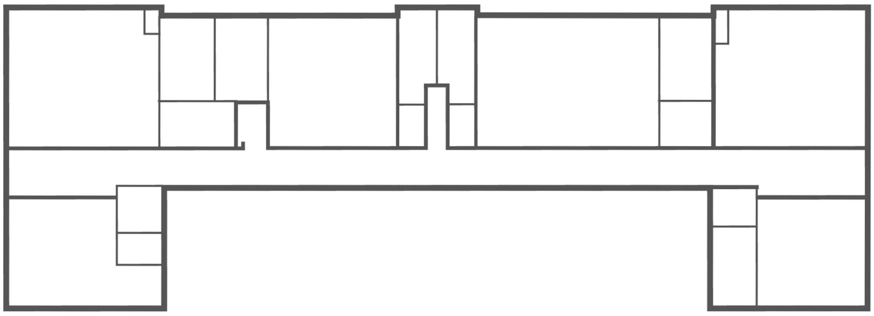

Figure 5.2: Initial floor plan

After initial edge detection step, pixels corresponding to a wall are converted to a sparse set of points. Resulting map is shown in figure 5.3. World map preparation takes non negligible time and is therefore done only once and then cached, as it does not change during the runtime.

The scale of the input drawing of architectural plan must be known precisely. The method proposed here is very sensitive to the right value of the scale. Fortunately it is possible to roughly check if the value corresponds to the reality by laying a scan from laser scanner over the generated world map pointcloud and visually evaluating if they overlap.

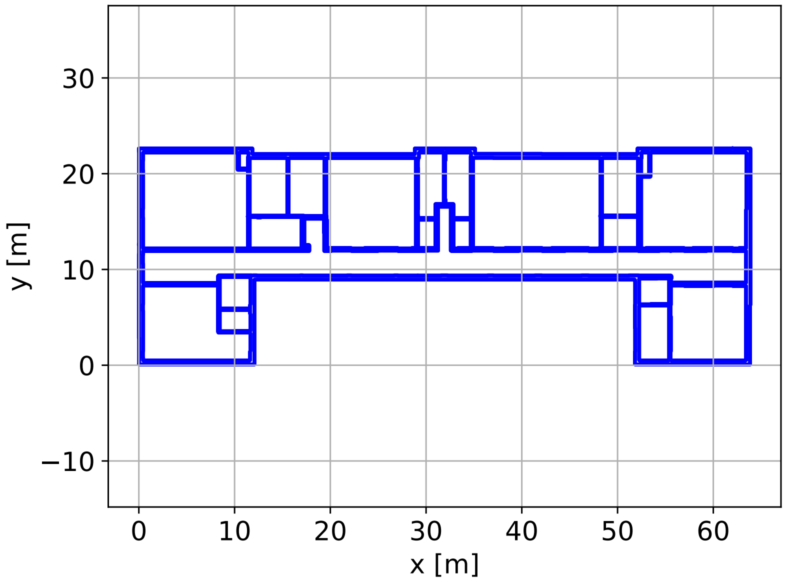

Figure 5.3: World map pointcloud

Pointcloud is used to represent the world map for three reasons. One being that an image, a two dimensional array, provides a fixed limited resolution. Whereas sprase pointcloud is limited only by its float datatype precision. Although arguably, resolution no higher then one of the input image with architectural drawings would be required.

Second reason is that without additional optimalization, filled two dimensional array is more memory expensive than sparse pointcloud.

Finally third reason for using sparse pointcloud for representing the world map generated from architectural plans is that it is the same format to which data from laser scanner can be easily converted. It is preferable to work with and compare two datasets that are use the same representation.

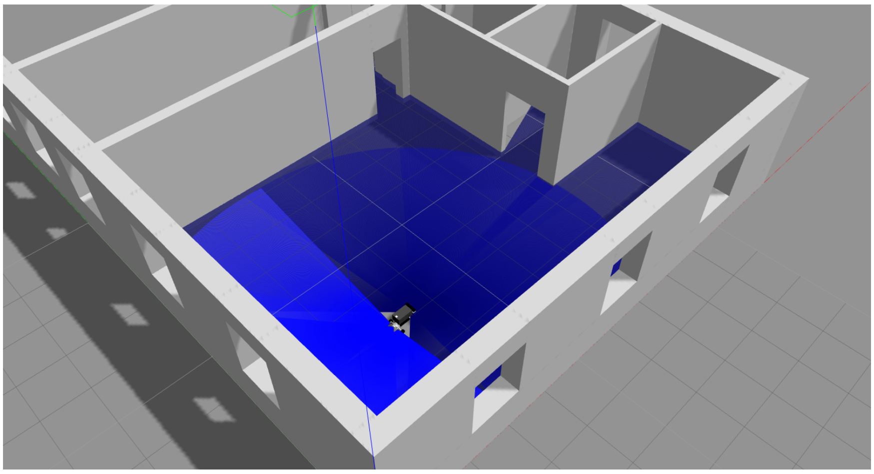

Figure 5.4: Empty room in Gazebo model

Following experiments will be conducted in simulated environment of the Gazebo robot simulation software tool. Figure 5.4 shows detail of particular room in the Gazebo simulation. The room is empty, which is unlikely to be the case in real world and a method that would only work in such environment would perhaps not be that much useful. Therefore this room will be later furnished to simulate a possible real world scenario and the proposed approach of localization of the robot will be validated there.

Although computer models always deviate from reality to some degree, there are several advantages offered by simulation. In general, it reduces costs, because it eliminates the requirement to obtain expensive equipment. It is not necessary to physically produce a prototype when change in the design occurs. It accelerates work, because multiple simulations can run in parallel by multiple researchers. And it is generally more convenient to setup an experiment in the simulated environment.

Researchers can easily share the robot design in the Universal Robotic Description Format (URDF), that is supported by both ROS and Gazebo, which creates great workflow for experimenting with robots in simulated environments.

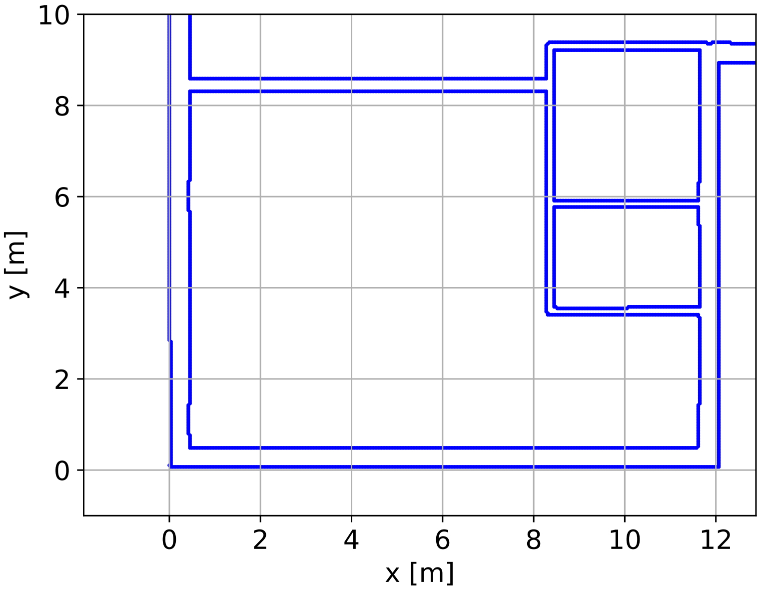

Figure 5.5: Detail of world map pointcloud

In the figure 5.5, the detail of the world map pointcloud is shown. It is focused on the room shown in the figure 5.4. It illustrates that the wall surfaces were correctly identified in the architectural drawing by the edge detector. Usually there are two surfaces recognized for each wall. The inner surface and the outer surface.

5.3 Iterative Closest Point

The ICP algorithm matches two pointclouds. It needs an initial position estimate. The initial position estimate is needed to establish point correspondences between the two pointclouds for the first iteration of the ICP algorithm. Figure 5.6 schematically shows how the ICP algorithm works.

As the name suggests, ICP works in iterations. At the begining of every iteration, new set of correspondences is established, based on the last estimate of rigid transformation between the two pointclouds. Correspondences are formed based on the distance between the points from both pointclouds after transforming the scan by the last estimate or the initial estimate. For point in scan, the closest point from world map pointcloud is choosed as a correspondence.

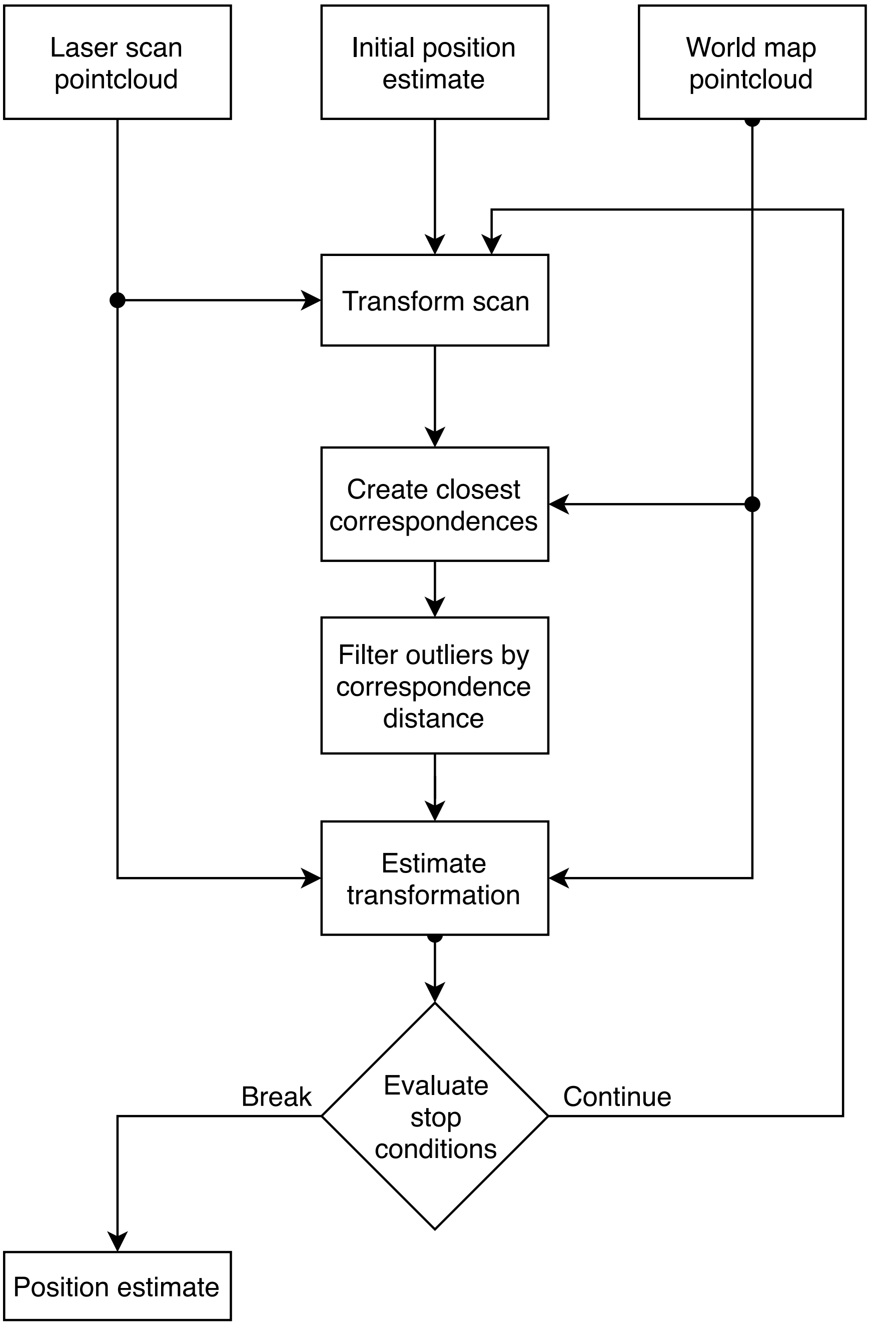

Figure 5.6: Iterative closest point algorithm

It is always useful to filter outliers from the dataset if possible. Here the outliers are filtered based on the maximum allowed distance of the correspondence. Original scan is used as a source for outlier filtration during every iteration. The set correspondences is reduced by this step.

The rigid transformation between the two sets of corresponding points is then computed. The problem of computing optimal ridig transformation between two pointclouds is defined by OpenCV Documentation[10] as follows.

$$\left[\boldsymbol{A}^{*}|\vec{b}^{*}\right] = arg\,\min_{\left[\boldsymbol{A}|\vec{b}\right]} \sum_{i} \left\| D_i - \boldsymbol{A}S_i^T - \vec{b} \right\|^2 \tag{5.1}$$

Matrix $\boldsymbol{A}$ in the equation 5.1 is the rotation matrix. Vector $\vec{b}$ is the translation vector. The matrix $\boldsymbol{A}$ and vector $\vec{b}$ have forms shown in equations 5.2 and 5.3 respectively. Symbols D_i and S_i represent i-th corresponding points from respective pointclouds. Matrix $\boldsymbol{A}^*$ and vector $\vec{b}^*$ are optimized results with same forms as matrix $\boldsymbol{A}$ and vector $\vec{b}$ respectively.

$$\boldsymbol{A} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{5.2}$$

$$\vec{b} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{5.3}$$

Figure 5.7 shows green points from the scan pointcloud matched against blue world pointcloud by the ICP algorithm. The red arrow represents odometry position of the robot, that was used as an initial estimate for the ICP algorithm. It is overlaying green arrow, that is representing position estimated by the ICP algorithm.

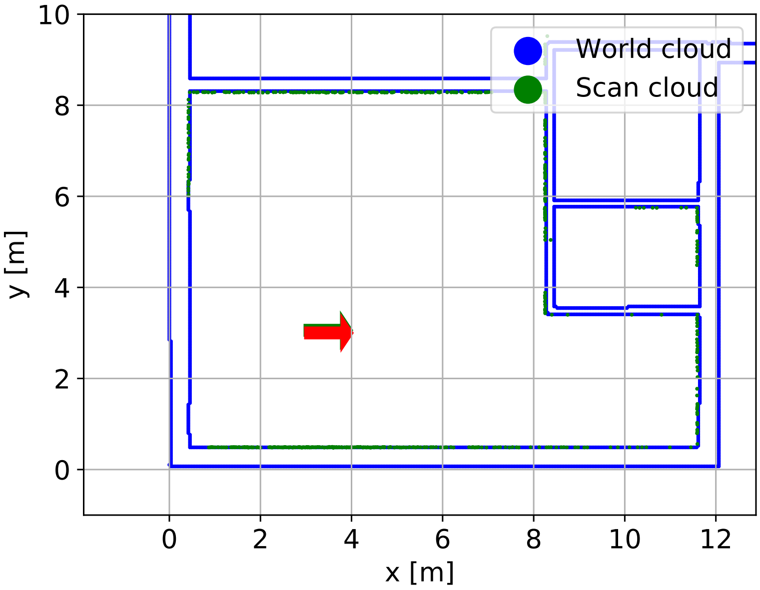

Figure 5.7: Matching scan to the world pointcloud

With the new rigid transformation estimate, whole process is repeated, until a stop condition is reached. It was decided to stop the ICP algorithm after fixed maximum number of steps. Other options might include evaluating quality of the estimate from sum of correspondence distances, either in absolute value or as a change over time. As the ICP algorithm iterates, the estimate converges to a local minimum.

5.4 Outlier Rejection

As previously stated, removing outliers from the dataset can greatly help in cases when the data contain additional noise. In the real world scenario, the room will likely not be empty. It may contain static obstacles that may not be part of the architectural plans, such as closets or tables. Also, the robot can encounter movable obstacles, such as chairs, other robots or people.

To approximate the real world scenario, the room was furnished with several closets around the perimeter of the room, one solid obstacle in the free space and three tables in the middle of the room. The furnished room is shown in figure 5.8. Each table has a single leg, that the robot is able to register with the laser scanner.

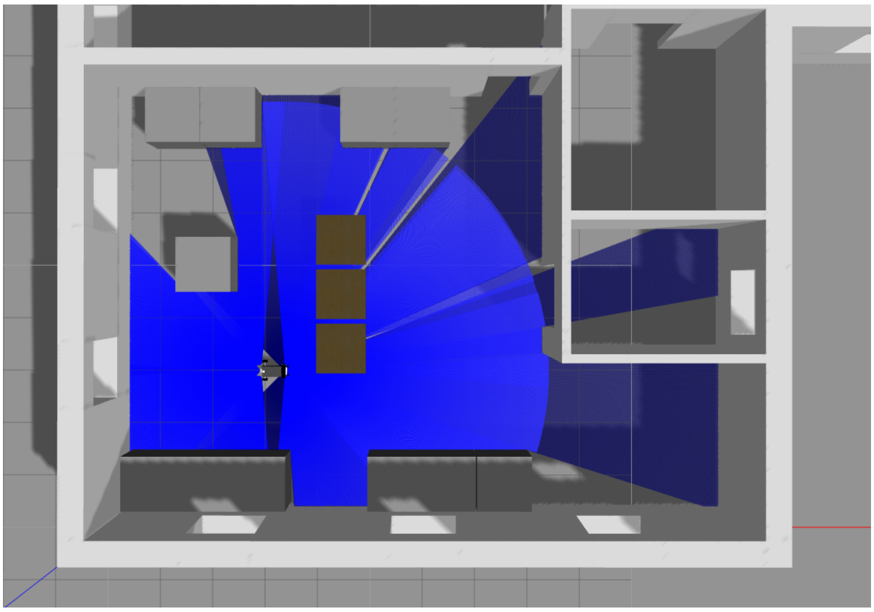

Figure 5.8: Furnished room in Gazebo model

Those deviations from the architectural plan are noise from the ICP algorithm perspective. If there is too much noise, the algorithm may fail and find an incorrect transformation. Incorrect match of scan with high quantity of noise without outlier rejection is shown in figure 5.9.

Outlier rejection was used to mitigate this behaviour. The correspondences are establisted using entire scan pointcloud at the beginning of every iteration. During the iteration, after establishment of correspondences, any correspondence with distance between corresponding points bigger, than a threshold value, is removed.

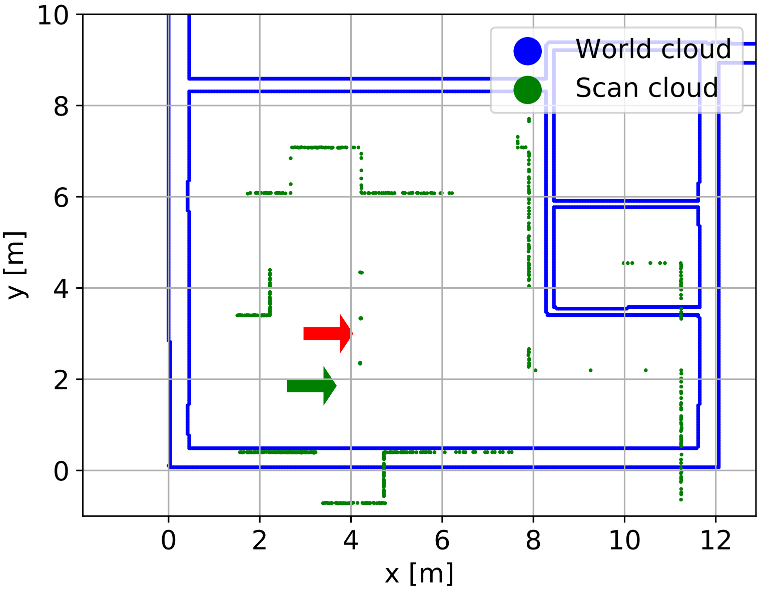

Figure 5.9: Matching scan without outlier rejection

When the outlier filtration is used, the ICP algorithm produces correct results, as shown in figure 5.10, where the scan pointcloud was split to green inlier points and black outlier points. Treshold value used to filter outliers was 25 cm.

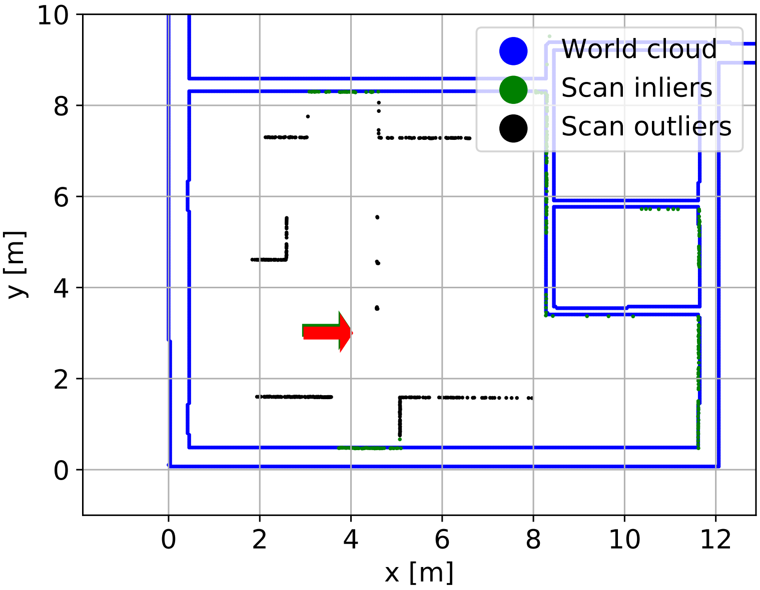

Figure 5.10: Matching scan with outlier rejection

5.5 Dynamic Behavior

To illustrate robustness of the proposed solution against drift in the localization process an arbitrary static error was introduced to the odometry measurement by error generator. Figure 5.1 shows where the error is introduced into the system. Odometry measurements are published with a frequency of 100 Hz. Every time the odometry measurement is published, its translation is increased by $5 \cdot 10^{-5}\ m$ and its rotation is increased by $5 \cdot 10^{-5}\ rad$.

While the ICP algorithm still uses the relative position change from the odometry measurement (together with last position estimate) as an initial position estimate for new estimation, it aligns with the world map pointcloud generated from architectural plans. The figure 5.11 shows euclidean distance deviation of odometry measurement and the ICP estimate from ground truth. The figure 5.12 shows angle deviation of odometry measurement and the ICP estimate from ground truth.

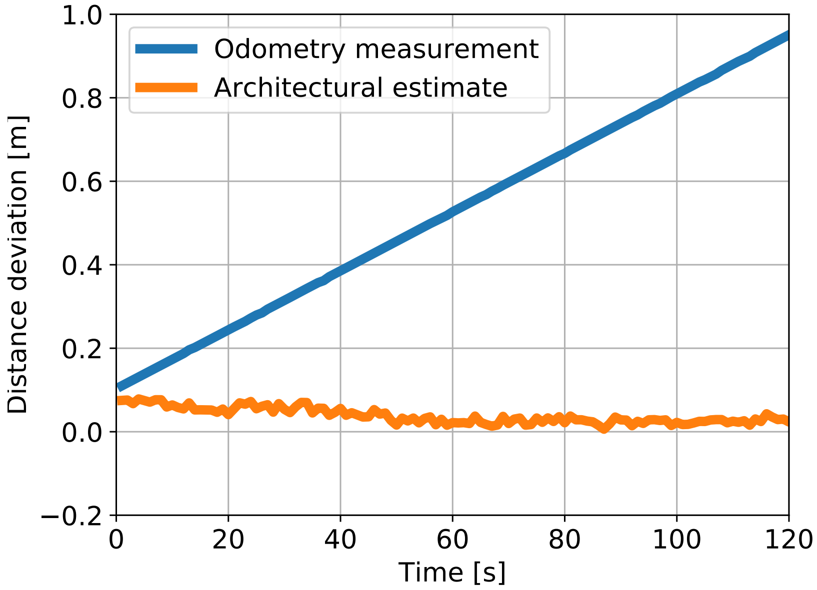

Figure 5.11: Distance deviation with static error

This experiment shows, that proposed method can be used as corrective step to prevent localization drift. While the deviation of the odometry position is constantly increasing, the deviation of architectural plans based estimate, produced by the ICP algorithm, remains constant.

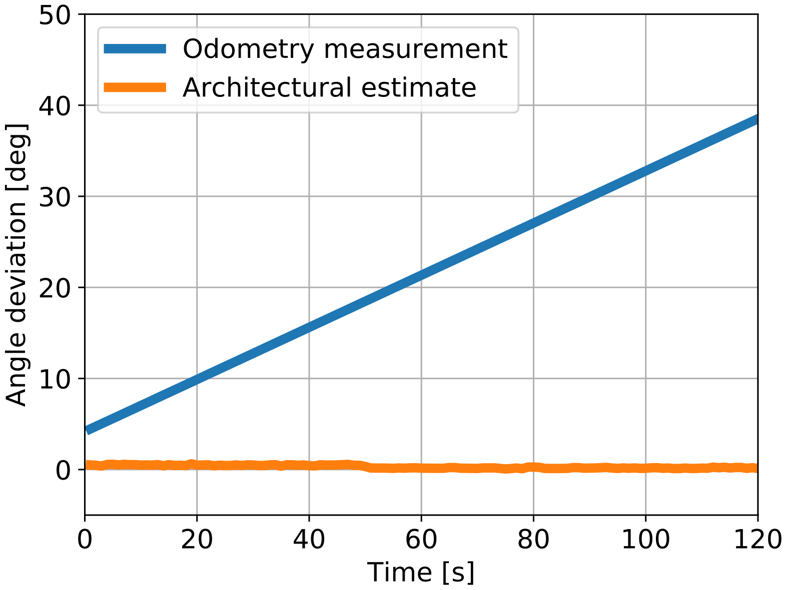

Figure 5.12: Rotation deviation with static error

5.5.1 Enhanced Occupancy Grid

No.	Color	Label
1	Grey	Unknown
2	White	Free
3	Black	Occupied
4	Blue	Architectural

Table 5.1: enhanced occupancy grid states

Surrounding environment can be represented in a form of occupancy grid. Occupancy grid divides the space into regular square shaped cells and assigns a state to each cell. Commonly used states are unknown, free and occupied. It is proposed to fuse the architectural plan with the occupancy grid by creating an enhanced occupancy grid (EOG) that would operate with additional architectural state. Table 5.1 lists all possible states of enhanced occupancy grid (EOG) and their assigned colors.

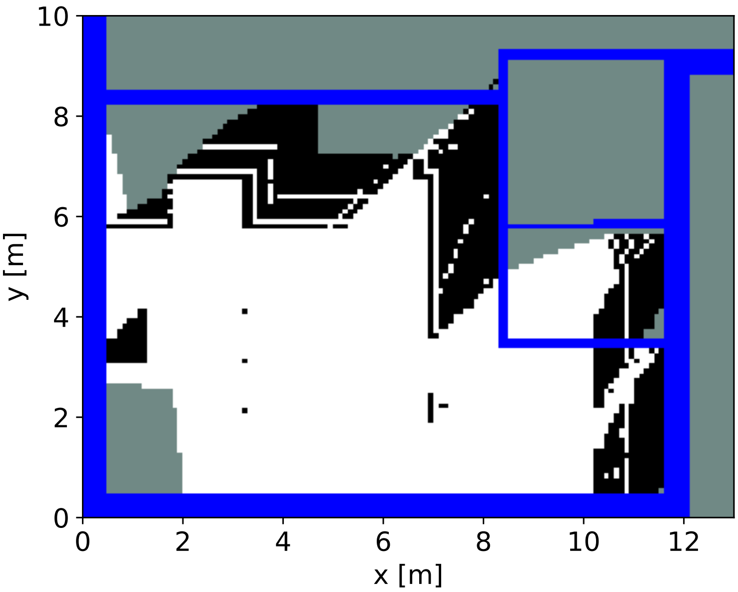

Figure 5.13: Enhanced occupancy grid without iterative closest point applied

Figures 5.13 and 5.14 compare an EOG created only by the erroneous odometry measurement with an EOG created by ICP method. The odometry measurement is affected by the static artificial error in both cases.

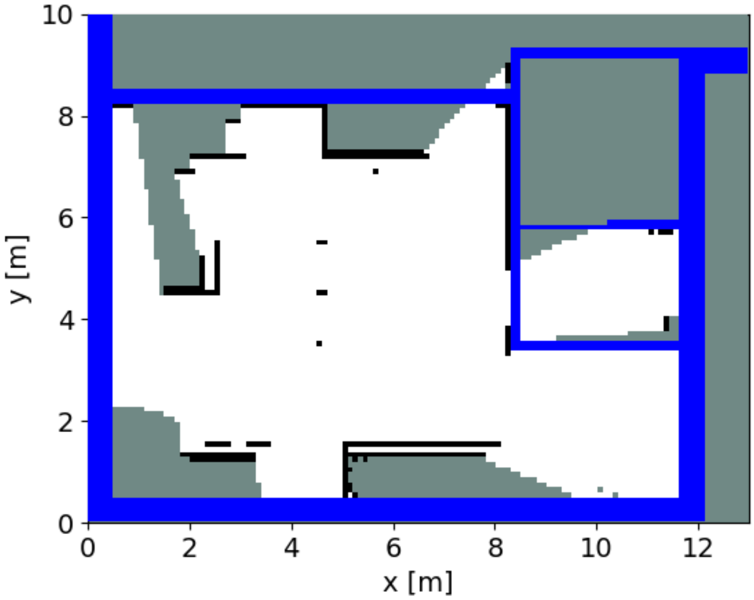

Figure 5.14: Enhanced occupancy grid with iterative closest point applied

Chapter 6

Outlook

Aside from of transparent surfaces, the laser beam of the laser rangefinder cannot pass through wall surface. Therefore outer wall surfaces could be filtered out, which might further improve fidelity of an approach proposed in this thesis. Outer wall surfaces could be filtered out by modelling the laser beam and finding what wall surfaces could the robot see from its current position in the room.

Transparent or translucent surfaces present specific challenge for the laser rangefinder. Perhaps other types of sensors, working on different principle, could be used jointly with laser rangefinder to address this challenge. This possibility was not further investigated in this thesis and could be further worked out.

Only walls were extracted from the architectural plans in this work. But architectural plans can contain more details, such as doors. Extracting additional features from architectural plans, their representation and use in the SLAM process can be a direction of continuing research.

In public buildings, the rooms are commonly numbered to facilitate orientation of the people. If the room numbers were included in the architectural plans, perhaps they could be identified from the wall signs, using the camera, and taken into account in the SLAM process. This may be the subject of further research.

Chapter 7

Conclusion

In this work, it was demonstrated, that it is possible to extract information from architectural plans that can be used to improve the quality of the SLAM process. Positions of walls were extracted from architectural plans and used to construct a world map representing the architectural plans. The ICP algorithm was used to match scans from laser rangefinder to the world map. The enhanced occupancy grid (EOG) concept was proposed to hold a fusion of architectural plans and OG from the SLAM process, as shown in figure 5.14. ROS node for generating the EOG was created.

ROS node for position estimation based on the ICP algorithm was prepared. The ICP algorithm has been extended with outlier rejection to compensate for the effect of noise created by obstacles not included in the architectural plans. The robot was localized relative to the architectural plans using this approach, as shown in figure 5.10.

ROS node introducing arbitrary error to odometry measurements was formed to simulate the reaction of the proposed solution to a small incremental error. Two auxiliary ROS nodes were created to store and display the course of the experiment. The experiments were conducted in Gazebo simulator. It was shown in figure 5.11, that the proposed solution eliminates the effects of small incremental error in odometry measurements and does not display the tendency to drift. Thus improving the long-term stability of the SLAM process.

Bibliography

[1] C. Georgiou, S. Anderson, and T. Dodd, “Constructing contextual SLAM priors using architectural drawings,” in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. Queenstown, New Zealand: IEEE, Feb. 2015, pp. 50–56. [Online]. Available: http://ieeexplore.ieee.org/document/7081124/

[2] “Flucht-und Rettungsplan,” *Staatliches Bauamt Aschaffenburg*, Jan. 2017, photographed 2019-05-24.

[3] L. Gaget, “Top 10 of the best 3d modeling software for architecture,” accessed 2019-05-21. [Online]. Available: https://www.sculpteo.com/blog/2017/10/23/top-10-of-the-best-3d-software-for-architecture/

[4] “25 Lovely Blueprint Home Plans,” accessed 2019-05-21. [Online]. Available: http://hocafersan.com/blueprint-home-plans/

[5] D. Fellhauer, “Kooperationsstrategien für autonome mobile Roboter,” Projektphasenbericht I, Hochschule Aschaffenburg, 2018.

[6] D. Förter-Grauel, “Kooperative Exploration mit autonom agierenden mobilen Robotern,” Masterarbeit, Hochschule Aschaffenburg, 2018.

[7] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, “A pose graph-based localization system for long-term navigation in CAD floor plans,” *Robotics and Autonomous Systems*, vol. 112, pp. 84–97, 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889018306092

[8] F. Boniardi, T. Caselitz, R. Kummerle, and W. Burgard, “Robust LiDAR-based localization in architectural floor plans,” in *2017*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, BC: IEEE, Sep. 2017, pp. 3318–3324. [Online]. Available: http://ieeexplore.ieee.org/document/8206168/

[9] "Canny Edge Detector — OpenCV 2.4.13.7 documentation," accessed 2019-05-23. [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html

[10] "Motion Analysis and Object Tracking — OpenCV 2.4.13.7 documentation," accessed 2019-05-23. [Online]. Available: https://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html

Appendix A

Acronyms

CAD computer-aided design. 5, 6

EOG enhanced occupancy grid. ix, 23, 27

ICP iterative closest point. 9, 10, 12, 16, 18–23, 27

IMU inertial measurement unit. 8

OG occupancy grid. 2, 27

ROS Robot Operating System. 3, 12, 15, 27

SLAM simultaneous localization and mapping. 2, 6, 10, 13, 25, 27

URDF Universal Robotic Description Format. 15