DLR-IB-RM-OP-2019-38

Robot Localization in Dynamic Environments Using Time-Variant Semantic Cues

Masterarbeit

Oskars Teikmanis

DLR Deutsches Zentrum für Luft- und Raumfahrt

Institut für Robotik und Mechatronik	BJ.: 2019 IB.Nr.: DLR-IB-RM-OP-2019-38

MASTERARBEIT

ROBOT LOCALIZATION IN DYNAMIC ENVIRONMENTS USING TIME-VARIANT SEMANTIC CUES

Freigabe:	Der Bearbeiter:	Unterschriften
	Oskars Teikmanis	
	Betreuer:	
	Dr. Zoltán-Csaba Márton	
	Der Institutsdirektor	
	Prof. Alin Albu-Schäffer	

Dieser Bericht enthält 69 Seiten, 40 Abbildungen und 1 Tabelle

Ort: Oberpfaffenhofen	Datum: 07.02.2019	Bearbeiter: Oskars Teikmanis	Zeichen:

DEPARTMENT OF
MECHANICAL ENGINEERING

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in
Mechatronics and Information Technology

Robot Localization in Dynamic Environments Using Time-Variant Semantic Cues

Oskars Teikmanis

DEPARTMENT OF
MECHANICAL ENGINEERING

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in
Mechatronics and Information Technology

Robot Localization in Dynamic Environments Using Time-Variant Semantic Cues

Roboter Lokalisierung in dynamischen Umgebungen mittels zeitvarianten semantischen Referenzen

Author:	Oskars Teikmanis
Supervisor:	Prof. Dr.-Ing. Darius Burschka
Advisors:	Dr. Zoltán-Csaba Márton, Dipl.-Geogr. Iris Lynne Grixa
Submission Date:	18.12.2018

I confirm that this master's thesis in mechatronics and information technology is my own work and I have documented all sources and material used.

Munich, 18.12.2018 Oskars Teikmanis

Abstract

A challenging issue in the field of robotics is the lost robot problem, in which a robot has to relocalize itself in a previously mapped environment based on current sensor readings. We propose a method for addressing this problem by extending the mapping of an environment with semantically labelled points. These semantic landmarks are processed in an algorithm that registers two labelled point sets in order to obtain the rigid transformation that relates the robot's current frame with the global coordinate system.

The relocalization system was tested on a dataset created for 3D scene analysis, and on self-made scans of several environments with different types of visual interferences, including obstructions to the field of view and severe lighting changes. These tests revealed that the proposed method performs well with realistic levels of noise in the input data. It even outperforms a state-of-the-art visual mapping and relocalization system both in robustness and in accuracy, producing linear and angular errors of less than 10 cm and 1° respectively in successful relocalization attempts.

Kurzfassung

Eine große Herausforderung im Bereich der Robotik ist das Problem des verlorenen Roboters, in dem ein Roboter sich in einer zuvor kartieren Umgebung, basierend auf aktuellen Sensorwerten, lokalisieren soll. Im Rahmen dieser Arbeit wird eine Methode für die Lösung dieses Problems durch das Erweitern der Kartierung einer Umgebung mit semantisch markierten Punkten vorgestellt. Diese semantischen Orientierungspunkte werden in einem Algorithmus benutzt, der zwei solche Punktsätze registriert, um die starre Transformation zu erhalten, die die aktuelle Perspektive des Roboters mit dem globalen Koordinatensystem in Beziehung setzt.

Das Relokalisierungssystem wurde an einem Datensatz getestet, der für die Analyse von 3D Szenen erstellt wurde. Anschließend wurden Experimente in selbstaufgenommenen Umgebungen mit verschiedenen Arten von visuellen Störungen (darunter Hindernisse im Sichtfeld und starke Beleuchtungsänderungen) durchgeführt. Diese Tests haben gezeigt, dass das vorgeschlagene Verfahren bei realistisch verrauschten Eingangsdaten gute Ergebnisse liefert. Es übertrifft sogar ein visuelles Mapping- und Relokalisierungssystem auf dem neusten Stand der Technik, sowohl in Bezug auf Robustheit als auch Genauigkeit. Typischerweise werden bei Erfolgreichen Relokalisierungsversuchen Linear- und Winkelfehler von weniger als 10 cm bzw. 1° erzielt.

Acknowledgments

I would like to thank Prof. Dr.-Ing. Darius Burschka for making this thesis possible and for providing valuable directions.

My greatest thanks go to Iris Lynne Grixa for spending countless hours discussing implementation details of the developed system, and Zoltán-Csaba Márton for his immensely useful advice on the more theoretical aspects of the tackled problem.

Additionally, I would also like to express my gratitude towards the friendly and helpful colleagues at the Institute for Robotics and Mechatronics of the German Aerospace Center for their help with some questions relating to their field of specialization.

Oskars Teikmanis, December 2018

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Description 2

2 Related Work **5**

2.1 Lost Robot Problem 5

2.2 Semantic Exploration 6

3 Fundamentals **9**

3.1 Point Set Registration 9

3.1.1 SVD for Rigid Transformations 9

3.1.2 Iterative Closest Point Algorithm 10

3.1.3 RANSAC 11

3.2 Object Detection 12

4 Semantic Localization **15**

4.1 Overview 15

4.2 Semantic Point Cloud Generation 16

4.2.1 Scene Geometry 16

4.2.2 Object Detection 17

4.2.3 Point Labelling 18

4.3 Semantic Landmark Matching 20

4.3.1 Initial Approximation with RANSAC 21

4.3.2 Refinement with Weighted Least Squares 24

4.3.3 Final Refinement with ICP 24

5 Quantitative Evaluation on a Dataset 27
5.1 About the Dataset 27
5.2 Error Metrics 28
5.3 Single Viewpoint Registration 28
5.4 Combined Viewpoints with Added Noise 31
5.5 Effect of Clustering 35

6 Qualitative Tests in Real Environments 39
6.1 Accuracy Test with Vicon 39
6.1.1 Experiment Setup 39
6.1.2 Relocalization with SPM 40
6.1.3 Comparison with ORB-SLAM2 42
6.2 Relocalization in Dynamic Environments 43
6.2.1 Occlusions to the Sensor's Field of View 43
6.2.2 Strong Lighting Changes 44
6.2.3 Relocalization from Different Perspectives 45

7 Conclusions 49

8 Future Outlook 51

Chapter 1

Introduction

Imagine a simple thought experiment in which a blindfolded person is brought into a randomly selected room of a building. After the blindfold is removed, this person is tasked to locate him- or herself in a map of the building, based on purely visual information. For a human this task is rather trivial. Simply by looking at the surrounding scene and noteworthy objects it contains, one can easily put a dot of his or her location on a model of the room, given that this model is detailed enough. By extension, if the person is familiar with a more general map, they can also determine their location in the whole building. This is largely enabled by the ability of humans to effectively interpret the semantics – or the meaning – of particular elements in the environment, giving more uniqueness to every scene.

This master's thesis explores such a semantics-oriented way of localization in the context of robotics and computer vision. The aim is to create a system that enables a robot to autonomously relocalize itself in a known map after losing track of its surroundings, a circumstance also referred to as the lost robot problem. This goal is achieved by extending a robot's visual map generation routine with a general purpose object detector that would assist in the creation of visual landmarks in the three dimensional reconstruction of the environment.

1.1 Motivation

While traditional (re-)localization approaches exist, these are often over-confident in their estimates, and could be aided by the added robustness offered by deep learning-based semantic segmentation. Adding semantics to the mapping capabilities of robots creates advantages in reliable relocalization. An example showcasing the challenges a robot may have in a dynamic environments is the EU-funded project SPENCER [30]. The purpose of this project was to develop a mobile robotic platform for airport passenger guidance and assistance. An example use-case is showed in Figure 1.1. This robot has to operate in crowded and visually changing environments where its sensors are often occluded. This poses a great challenge for localization. The method developed in this thesis attempts to permit robots to relocalize themselves based on a set of static landmarks in a scene that may also contain moving features. This would be advantageous for home service robots and any other robot that operates in an environment that contains some stationary and reliably detectable objects that can be used as localization landmarks.

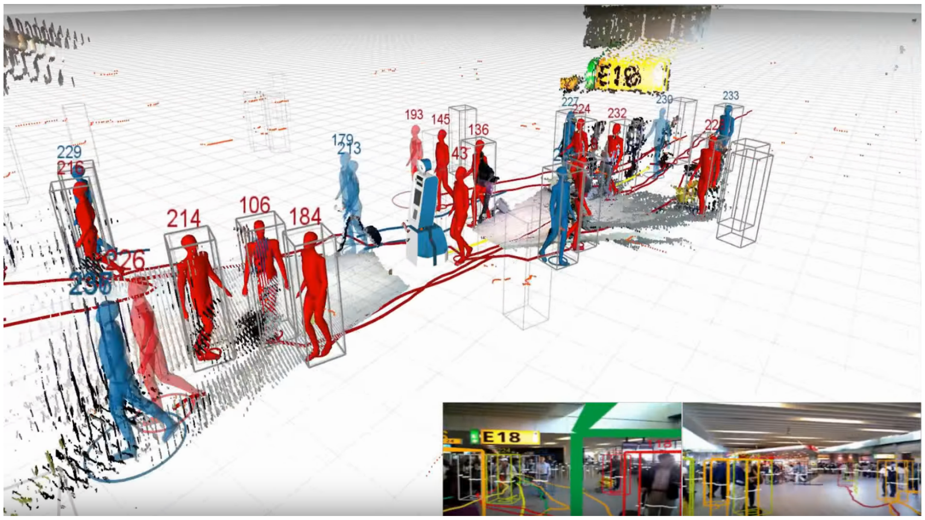


Figure 1.1: Robotic platform developed in the SPENCER project (visible near the centre of the image) operating in a very crowded environment [2, 11], taken from the SPENCER YouTube channel.

There are numerous methods that attempt to solve the lost robot problem using different kinds of sensors, including RGB and depth cameras. Common challenges that have to be dealt with are changes in the local view of the scene as compared to the initial mapping. During relocalization, the sensor may be obstructed, the brightness may have changed substantially or the robot may reanimate in a place where it has not yet physically been. This thesis will show that the proposed semantic method is capable of handling even such extreme cases where other visual localization routines may fail.

1.2 Problem Description

A situation in which a robot is placed in a new location, where it has to localize itself in a pre-existing map, is called the kidnapped (or lost) robot problem. For the semantic relocalization method, we are working with a representation of the environment as a 3D point cloud, created with a simultaneous localization and mapping (SLAM) routine. The sensor used for the point maps and their semantic extensions is a camera with RGB and depth vision. The semantic landmarks are created by placing labelled points in the locations where objects have been detected by using a general purpose object detector. Relocalization is achieved by computing the rigid transformation that registers the labelled point set of a global map with the points detected in a local map. This process is further explained in Chapter 4. The goals of this thesis are the following:

- Develop a system generating semantic landmarks.
- Develop a registration algorithm for semantically labelled point sets.
- Evaluate the performance of the developed systems.

Some limitations of this method can be inferred on a theoretical level. For one, the semantic localization system is highly dependent on the used object detector's accuracy and the presence of objects in the environment it is used in. If a robot is lost in a room which doesn't contain any objects that can be recognized by the chosen detector, the system is sure to fail. In fact, it is highly desirable for objects not only to be present, but to be well distributed inside the room, or else the registration attempt may only be accurate in a small area of the map. Thus the advantages of traditional and deep learning approaches may be combined in order to complement their respective shortcomings.

With these limitations in mind, the developed system is evaluated in Chapters 5 and 6.

Chapter 2

Related Work

This thesis proposes a method that uses semantics in robot localization. To provide some context on the work that has been done in related topics, this chapter contains an overview of contributions made by other researchers in the fields of robot localization and semantic scene analysis.

2.1 Lost Robot Problem

Common sensors employed to help solve the lost robot problem are laser rangefinders (LRF) and cameras. This section describes some noteworthy methods that solve this problem in various environments.

At the time of writing, an example of notable recent work is ORB-SLAM by Mur-Artal *et al.* [16, 18], a visual keyframe-based method for simultaneous localization and mapping, working with monocular, stereo and RGB-D cameras. It has an integrated relocalization module which uses ORB feature matching [22] and a visual bag of words (BoW) [8] approach to find the best candidate for the robot's global pose. This thesis makes use of the pose estimation computed by ORB-SLAM2. Su *et al.* [28] propose a method in which relocalization is enabled by a composite map with LRF data and visual keyframes. In their work, the top scoring Gist [19] descriptors are found using ORB and RANSAC to produce a relocalization pose estimate. The optimal keyframe candidate is found by applying a cost function that also uses the LRF map data.

Another broadly applied method that works well with LRF is Monte Carlo Localization (MCL) [6], in which the position and orientation of a robot is modelled with particles. If no initial guess is available the particles are initialised randomly, after which they are iteratively resampled until convergence, based on the robot's motion model and sensor measurements. MCL provides an accurate estimate of the robot's pose, but it generally fails to efficiently solve kidnapped robot situations, because the new location in which the robot finds itself is very unlikely to be covered by the converged particles. Seow *et al.* [26] attempt to solve this problem by making use of Wifi. In their method, if the robot's position is estimated simultaneously with LRF and a Wifi signal strength sensor. If the mismatch between both estimates is too large, the MCL particles are resampled around the location of the strongest Wifi signal. This leads to faster convergence, since the particles don't have to be resampled randomly. While this method does detect and solve the lost robot problem, it requires a long time to do so according to the authors. The requirement of a Wifi infrastructure and sensors can also be limiting. Still, due to the nature of the problem, it is very difficult to design a solution that is well suited for all environments.

2.2 Semantic Exploration

An idea of a framework for unifying probabilistic and heterogeneous information into a multi modal map is presented by Pronobis *et al.* [20]. The concept is to combine spatial, topological and semantic information into a unified map to provide robots with a more complete view of the world, as illustrated in Figure 2.1. This is done via abstraction of spatial information into set categories like *place*, *path* and *room*, and by integrating relations between objects and scenes to allow the robot to infer knowledge through observations.

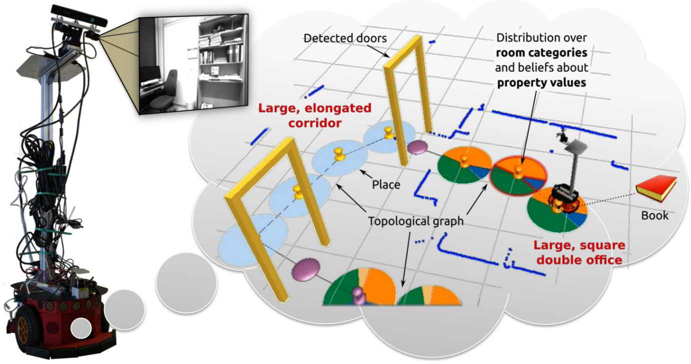


Figure 2.1: Robot platform and semantic map [20], ©2012 IEEE.

Semantics can be introduced in robotic systems in a variety of ways. Brucker *et al.* [4] developed a system that assigns semantic labels to 3D RGB representations of rooms. It does so by analysing a given room as a complete scene and as a set of objects. The room is first labelled by a neural network trained on a dataset containing a large number of different scenes. In order to add more robustness to the method, the scenes are also fed into an object detector, as shown in Figure 2.2. This provides more information that can be used to correctly label a room.

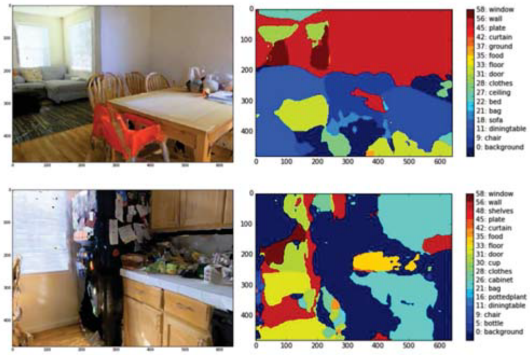
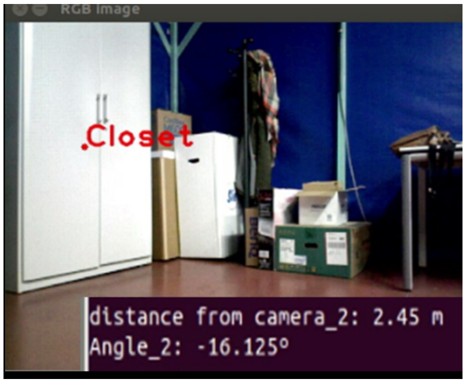


Figure 2.2: Detected objects in a sample scene [4], ©2018 IEEE.

Hernández *et al.* [9] work on detecting and locating objects in unaltered environments. The method uses support vector machines to segment and classify images received from an RGB-D sensor to obtain the position of objects relative to the robot, similar to the example in Figure 2.3.



```
distance from camera_2: 2.45 m
Angle_2: -16.125°
```

Figure 2.3: Detected and localized object [9], ©2016 IEEE.

The extent to which a robot can interact with humans and the environment is largely dependent on its ability to identify its surroundings in a meaningful way. These methods and models are examples of work towards a more complete understanding of an observed scene, which is also one of the aims of this thesis.

Chapter 3

Fundamentals

The robot localization system described in this thesis makes use of a range of tools and concepts stemming from various fields of computer vision. The theory relating to these concepts is outlined in this chapter.

3.1 Point Set Registration

Point set registration has a number of uses in cooperative mapping and navigation among many other applications. One of the tasks of this thesis is to find the robot's pose in an existing three dimensional reconstruction of the environment by using its current view, which can easily be interpreted as a 3D point set registration problem. This section looks at methods designed to align point sets and the mathematics they are based on.

3.1.1 SVD for Rigid Transformations

Singular value decomposition (SVD) can be applied to compute rigid transformations that produce a best-fitting alignment of two equally sized point sets $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ and $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_n\}$ in $\mathbb{R}^d$ [25, 27]. The transformation may be expressed by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ such that

$$(\mathbf{R}, \mathbf{t}) = \underset{\mathbf{R} \in SO(d),\, \mathbf{t} \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{n} w_i \left\| (\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i \right\|^2 .$$

The weights w_i may be set to 1 if they are not applicable in the given setting, resulting in a regular weighted least squares optimization problem. The translation component is obtained by calculating the weighted centroids of the point sets, given by

$$\overline{\mathbf{p}} = \frac{\sum_{i=1}^{n} w_i \mathbf{p}_i}{\sum_{i=1}^{n} w_i} \quad \text{and} \quad \overline{\mathbf{q}} = \frac{\sum_{i=1}^{n} w_i \mathbf{q}_i}{\sum_{i=1}^{n} w_i},$$

which gives us the optimal translation:

$$\mathbf{t} = \overline{\mathbf{p}} - \mathbf{R}\overline{\mathbf{q}}.$$

This allows a reformulation of the problem to

$$\mathbf{R} = \operatorname*{argmin}_{\mathbf{R} \in SO(d)} \sum_{i=1}^{n} w_i \left\| \mathbf{R}\mathbf{x}_i - \mathbf{y}_i \right\|^2,$$

where $\mathbf{x}_i := \mathbf{p}_i - \overline{\mathbf{p}}$ and $\mathbf{y}_i := \mathbf{q}_i - \overline{\mathbf{q}}$. For the rotation component we require the covariance matrix which is defined as

$$\mathbf{H} = \sum_{i=1}^{n} w_i (\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{q}_i - \overline{\mathbf{q}})^T.$$

With singular value decomposition we obtain $\mathbf{H} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T$. The covariance matrix can be used in its decomposed form to calculate the rotation that minimizes the squared distance between corresponding points:

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T.$$

This form of the matrix does not exclude reflections. If the solution has to be limited to rotations, we have to adjust the equation as follows:

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & \det(\mathbf{V}\mathbf{U}^T) \end{bmatrix} \mathbf{U}^T.$$

A more detailed description of SVD for rigid transformations may be found in [27] and related literature. The principle is applied in Section 4.3 of the thesis.

3.1.2 Iterative Closest Point Algorithm

The iterative closest point algorithm by Paul Besl and Neil D. McKay [3] is a significant paradigm in dense point cloud registration. In its core, the algorithm assumes given source and target clouds which have to be aligned, and minimizes the least squares error between corresponding (closest) points of the two inputs. An initial guess of the registration may be computed via principal component analysis (PCA) or by applying any other prior knowledge relating the point clouds. Then, the following steps are repeated until a termination criterion is reached:

1. For each source point, find the closest target point.

2. Calculate the registration (for example, by applying SVD or a quaternion-based transformation).

3. Apply the registration.

The algorithm may interrupt after reaching a user defined number of iterations. According to [3] a number between 30 and 50 is generally sufficient. An alternative termination criterion could be defined by the rate of improvement of the registration. If the next iteration does not improve the cost function by a sufficiently large amount, defined by a threshold, the algorithm may stop. While the convergence of ICP can be mathematically proven, it is prone to local minima, especially if the initial guess is too distant from the true solution and when the overlap between source and target clouds is too small. These issues were addressed in research more than a decade after the inception of ICP to make it more robust with only partially overlapping clouds and to increase its convergence basin [5, 12, 23]. Despite these improvements, ICP generally performs best in local point cloud alignment problems.

An open-source implementation of ICP and its numerous variations can be found in the Point Cloud Library[1] (PCL) [24]. The algorithm is used in this thesis to compute a refined pose after it has been estimated with the proposed semantic method.

3.1.3 RANSAC

Random sampling consensus (RANSAC) by Martin A. Fischler and Robert C. Bolles [7] is a model parameter estimation algorithm working exceptionally well with noisy input data. It does so by processing a subset of the given data in every iteration step, unlike least squares, which generally uses all the inputs. Figure 3.1 shows a typical case where least squares would fail due to the presence of outliers.

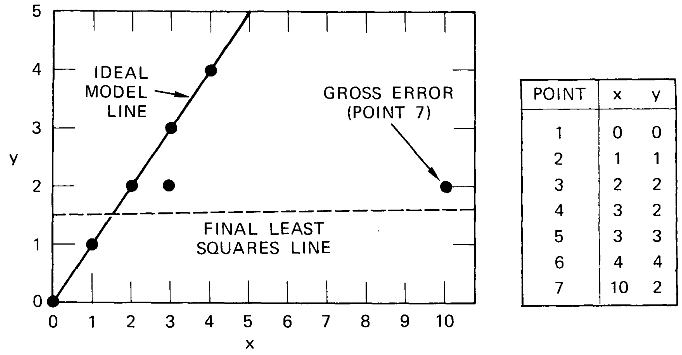
Assuming we wish to estimate a model that requires n data points and a point set consisting of n_p points where $n_p > n$, the model parameters may be estimated via repetition of the following steps:

1. Pick n points at random from the dataset and produce a model.

2. Compute the number of points that are within a chosen error threshold of the current model (inliers).

3. If the current model has more inliers than the previous best one, set it as the new best candidate.

[1] `pointclouds.org/`



POINT	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

Figure 3.1: An illustration of a problem that cannot be solved by a naive least squares parameter estimation, taken from the original publication [7]. This example features an ideal model line and the result of a version of iterative least squares, which deletes outliers of every previous step.

These steps are repeated a set number of times until it is very likely that a sufficiently good model is produced or the algorithm ends in failure. The final best result may be further refined based on the inliers (for example, with least squares). The required number of iterations may be determined probabilistically. Suppose we wish the algorithm to produce an accurate model after N steps with a probability P. We obtain the desired number of iterations as follows:

$$\begin{aligned}(1-p^n)^N &= (1-P)\\ N &= \frac{log(1-P)}{log(1-p^n)},\end{aligned}$$

where p is the probability of a random point being an inlier. This principle is used for the proposed method in this thesis alongside SVD for an estimate of the robot's pose, described in Section 4.3.1.

3.2 Object Detection

Since the work by Krizhevsky *et al.* [13], object detection in RGB images has been predominantly achieved with convolutional neural networks (CNNs). These are types of artificial neural networks that contain specialized layers which greatly reduce the number of parameters required for training. This makes them well suited for applications that work with large numbers of inputs, like image processing. The first CNNs were mainly trained to identify the most likely object visible in a given image, producing a class label and a probability score as outputs. Typical architectures of CNNs consist of several convolutional layers that break the input image into increasingly abstract visual features until it is fed into a set of fully connected layers. These networks are trained and tested on large collections of labelled images, like the *Common Objects in Context* (COCO) dataset [14]. For added robustness, the training data may be augmented via translations, reflections and by altering the RGB channels of the images.

Later work shifted towards multiple object detections that combine region proposals and object detection in order to produce an output vector that contains the coordinates of identified regions, their class label and probability score. An example of a region proposal network is Faster R-CNN by Ren *et al.* [21]. A compact visualization of a Faster R-CNN network in object detection is shown in Figure 3.2.

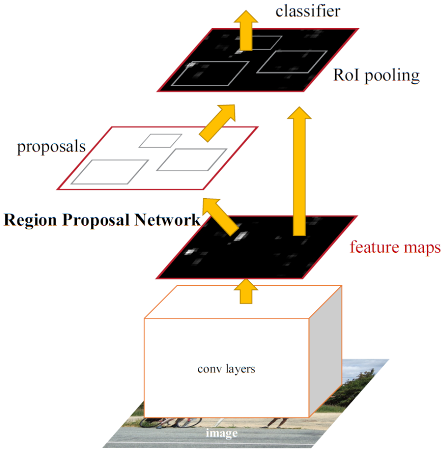


Figure 3.2: Region proposal with Faster R-CNN network [21], ©2017 IEEE.

This thesis makes use of an object detection implementation that combines Faster R-CNN with the Inception V2 [29] object detection architecture. The used model can be found in the Tensorflow[2] object detection API Github repository[3], dating January 1st, 2018.

[2] https://www.tensorflow.org/

[3] https://github.com/tensorflow/models/tree/master/research/object_detection

Chapter 4

Semantic Localization

Localizing oneself in an environment by interpreting the semantics, or meaning, of the surroundings along with its geometry is a primarily human way of perception. This chapter describes how the proposed method attempts to transfer this way of identifying the world in a robotic system.

4.1 Overview

The robot localization system developed in this thesis consists of two main parts: the semantic map generator (SMG), and the semantic point matcher (SPM), both mainly developed in the C++ programming language. The SMG extends a 3D representation of the environment generated by SLAM with a sparse semantically labelled point set. This thesis uses a keyframe-based SLAM system which is described in more detail in Section 4.2.1. The semantic map contains information on the 3D coordinates and class labels of detected objects in the room, along with other information which is covered in Section 4.2.2. A chart of the semantic map generation pipeline is shown in Figures 4.1 and 4.2. The current implementation of the pipeline has an offline and an online setting. In the offline mode, SLAM is run before the labelling step. This allows loop closure to be performed for a globally consistent state of the keyframe poses before they are saved for further use. In the online mode, RGB-D data from the sensor is synchronously processed by the used object detection and SLAM routines. Currently the position of semantic points is not affected by loop closure, meaning that the online mode is not suited for larger maps. It is, however, practical for relocalization.

In order to find itself in the global map, the robot would perform a scan of its current location to generate a local semantic map. The SPM registers the local and global maps created by the SMG and finds the relative transformation between them in a process visualised in Figure 4.3. This registration algorithm makes use of information stored in the semantic maps, including the scene geometry, object labels and positional uncertainties. The lost robot can generally be relocalized if enough matching objects are found in the local and global semantic map. The algorithm itself is covered in depth in Section 4.3.

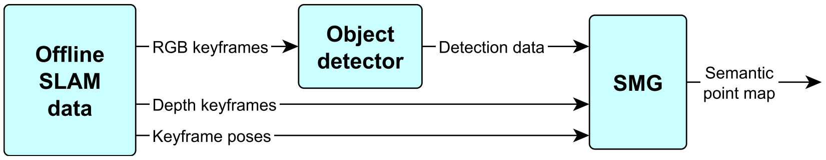


Figure 4.1: The offline map generating pipeline obtains RGB-D keyframes with corresponding poses from files. The RGB keyframes are first processed by the object detector. The detection data is then fed into the SGM along with depth and pose data to create a semantic point map.

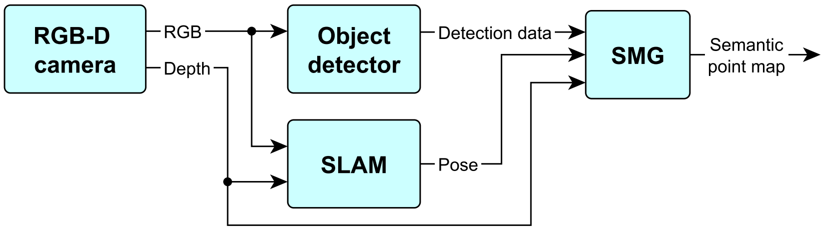


Figure 4.2: The online map generating pipeline receives RGB and depth data from an RGB-D sensor. Those are initially processed by SLAM and object detecting systems. Their outputs and the corresponding depth frame are fed into the SMG, which in turn produces a semantic point map.

4.2 Semantic Point Cloud Generation

This Section focuses on the implementation and theory behind the generation of semantic point clouds, as well as the inputs, outputs and individual components of the system.

4.2.1 Scene Geometry

The semantic point cloud generator uses a slightly adapted version of an open-source implementation of SLAM called ORB-SLAM2 [18]: a keyframe-based SLAM system that uses ORB keypoint detection for patch-tracking and mapping [22], and visual bags of words [8] for loop closure and relocalization. The adaptations of our system include a function to save point maps in various formats and keyframes (RGB and depth) with their corresponding poses. It can

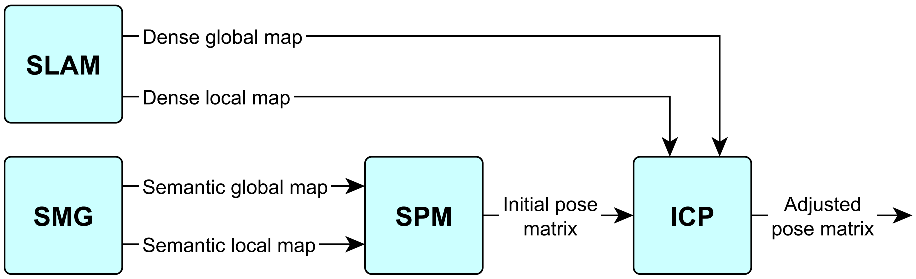


Figure 4.3: The inputs for the semantic point matching system are a pair of labelled point maps: the global map in which we wish to localize the robot and the local map resulting from a relocalization scan (see Section 4.2.3). A RANSAC-based algorithm computes an initial estimate for the registration. The dense local point cloud is transformed based on this estimate and fed into an ICP routine along with the global map. Ideally, this results in an improved relative transformation between the local and global coordinate frame.

also publish various outputs over Links and Nodes (LN), an inter-process communication framework developed at the DLR.

The 3D geometry of the scene is reconstructed by ORB-SLAM2 and saved in a sparse map along with a set of keyframes (depth and RGB) and their corresponding poses. The pose information is essential for the SMG during the creation of the labelled point map. In a post-processing step, a dense reconstruction is computed from the stored keyframes, and it is used in a pose refinement step involving ICP. The outputs of SLAM are processed differently in the offline and online settings. In the offline mode, the RGB-D keyframes are stored as separate PNG files, ideally after loop closure. The pose matrices corresponding to each frame are stored in a single data file containing a frame number coupled with each matrix. The online implementation uses LN to transfer data: once a new pose has been computed it is published along with a frame counter.

4.2.2 Object Detection

One of the goals of this thesis is to make a localization system that obtains useful information from general purpose detectors, instead of using a specialized architecture. The semantic labelling system was tested with the COCO trained Faster R-CNN [21] with Inception V2 [29]. This model was chosen due to its reasonable balance between frame rate and detection accuracy.

The outputs of an object detector include bounding boxes around detected objects in each frame, their labels and a probability score for each detection. This data is sent to the semantic labeller using LN, similarly as ORB-SLAM2. The bounding boxes are defined by a pair of x and y coordinates representing the upper left and lower right corners. In the COCO dataset the label is defined by a number from 1 to 90, each referring to an object class. The detection score is the probability of the top candidate label for a given object and is represented by a floating point value ranging from 0 to 1. An example image with detected objects is shown in Figure 4.4. Detector outputs are prone to various types of noise, including false and overlapping detections as well as deviations in bounding box size and placement. The latter are often due to asymmetric objects, like chairs, and the sensor's changing orientation relative to the scene.

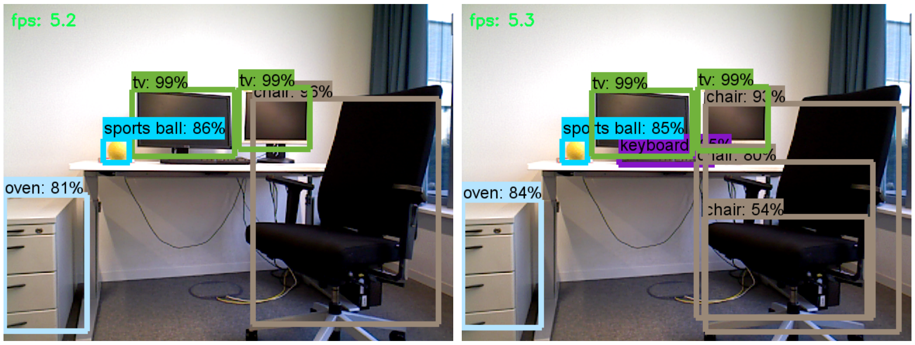


Figure 4.4: Two sample images with detected objects in an office environment. The detections differ in both images due to camera noise. Some objects like the office chair may be detected several times. Due to the angle and distance of the camera, objects like the keyboard are not always detected. In some cases, objects are consistently assigned false labels (office drawers labelled as oven).

4.2.3 Point Labelling

The SMG, which creates a 3D map containing semantically classified points, is the central part of the semantic map generating pipeline. It receives a depth image, its pose as computed by SLAM and detected object data (bounding boxes, labels and scores) as inputs and creates a sparse map containing (ideally) one labelled point per detected object. The program starts by reading a depth image (either form a camera or from files, depending on the mode) and object detector outputs from the corresponding RGB image. Next, each bounding box is used to obtain the region of each detection within the depth image. From this, the x, y and z-coordinates of every detected object in the camera frame can be obtained. They are given by

$$x = d\frac{(u - c_x)}{f_x}, \tag{4.1}$$

$$y = d\frac{(v - c_y)}{f_y}, \tag{4.2}$$

$$z = d, \tag{4.3}$$

where d is the representative depth of the object, given by the median depth of the cropped region. The remaining symbols are explained below.

- u, v: horizontal and vertical pixel positions of the region's centre
- c_x, c_y: optical centre of the Asus Xtion sensor (in pixels)
- f_x, f_y: focal lengths of the Asus Xtion sensor (in pixels)

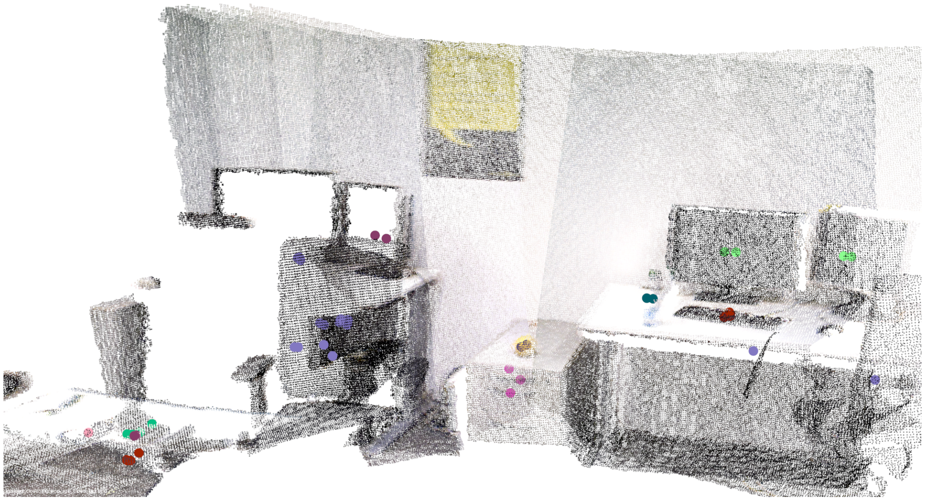


Figure 4.5: Repeated detections of objects in a partial scan of a room. Every colored blob represents a detection that was made in one of the captured frames to build this semantically extended map.

While scanning a whole room it is highly likely (and expected) that multiple objects are observed more than once, resulting in a cluster of points characterizing the same detected object, as can be seen in Figure 4.5. We are interested in reducing each cluster to one point. This is achieved with Euclidean clustering applied on every set of points with the same label, implemented in PCL. This algorithm searches for neighbours for every point and adds them to a cluster if they are closer than a given radius. This search radius was empirically set to 10 cm, which permits some noise in object positioning while preventing several clusters from merging into one. Additionally, in order to reduce noise from spuriously detected objects, the minimum cluster size was set to three points. After the clustering is complete, the program generates a list of visual landmarks, an example of which is shown in Figure 4.6. Each of these landmarks contains the following information:

- Mean x, y and z coordinates of each cluster
- Positional variance of each cluster, based on distance from mean

- Number of points in each cluster
- Class label
- Class probability

The class probability component is given by

$$P_{class} = 1 - \prod_{i=1}^{N}(1 - P_i) \tag{4.4}$$

where P_i is the class probability of each detection instance in the cluster and N is the number of points in the cluster. This adjusted probability score is motivated by the idea that an object that has been consistently identified several times with a probability higher than 0.5 is likely to be identified correctly. The semantic landmarks are now ready to be used for relocalization. The method accomplishing this is explained in the next section.

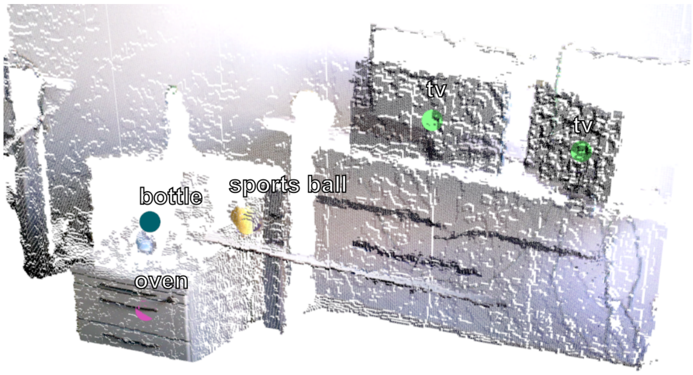


Figure 4.6: RGB point cloud and semantic landmarks of a partial office view. The objects are not always detected correctly, an example being cabinet drawers that are interpreted as an oven. This is generally not a problem: if the object is detected falsely, but consistently, it may still be a useful landmark. Otherwise it adds to noise which is ignored by the semantic landmark matcher.

4.3 Semantic Landmark Matching

This section covers the proposed method for relocalizing a robot with a pair of semantically extended 3D point maps. It includes details on how this is achieved with a combination of RANSAC, SVD and ICP. To differentiate between the two point sets more easily, the convention of *target* and *source* map shall be used for the initial scan and the relocalization scan respectively.

4.3.1 Initial Approximation with RANSAC

Due to the very sparse nature of the semantic point sets (often in the order of ten objects in a room), random sampling with an informed point selection heuristic is a reasonable method for finding an estimate of their relative transformation. The method described here is inspired by Chapter 3 of [15].

Algorithm 1 Random sampling for semantic point set matching

```
while I < Imax do
   P = sourceTriplet()
   Q = targetTriplet()
   if validTriplets(P, Q) then
      T = rigidTransform(P, Q)
      [fitness, inliers] = fitnessScore(P, Q, T)
      if fitness > bestFitness AND inliers >= bestInliers then
         bestFitness = fitness
         bestInliers = inliers
         bestT = T
         Imax = updateIterations(inliers)
      I = I + 1
return bestT
```

First, a set of three different points $\mathcal{P}$ is randomly chosen from the source point set. Next, a sample $\mathcal{Q}$ is searched in the target set such that both triplets have corresponding labels, as shown in Figure 4.7. Before a transformation between $\mathcal{P}$ and $\mathcal{Q}$ is computed, the triplets are checked for similarity. This is done by calculating the side lengths of the triangles spanned by both triplets. If all corresponding lengths are similar within a threshold (20 cm were chosen as a reasonable value), the two samples are valid. If this check fails, the iteration is skipped and the process is repeated on a new pair of samples. If the samples are valid, the rigid transformation relating them is calculated via weighted least-squares with singular value decomposition [25], as described in [27]. We are looking for a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ that minimize

$$\sum_{i=1}^{3} w_i \left\|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\right\|^2$$

where $\mathbf{p}_i$ and $\mathbf{q}_i$ are points in $\mathcal{P}$ and $\mathcal{Q}$ respectively, and w_i is an associated weight. For our purposes the weights are defined as follows:

$$w = P_{class}\, e^{-\frac{\sigma^2}{\lambda^2}},$$

where P_{class} is the adjusted detection probability and σ^2 is the positional variance of a labelled point, both of which are calculated during the semantic point map generation step, as described in Section 4.2.3. This definition is motivated by the idea that points should be less reliable for relocalization if they are detected improperly and have a large positional uncertainty. The parameter λ adjusts the decline rate of the exponential function and was set to 10 cm, which worked well to reduce the impact of less accurate points.

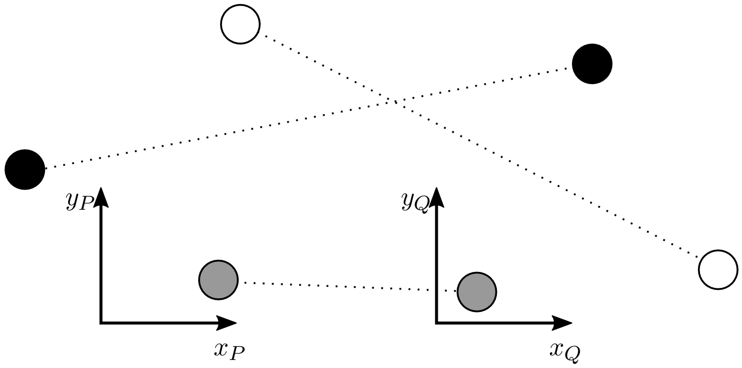


Figure 4.7: Two labelled point triplets with 1:1 correspondences.

To obtain the transformation itself, we first calculate the weighted centroids of the point sets:

$$\overline{\mathbf{p}} = \frac{\sum_{i=1}^{3} w_i \mathbf{p}_i}{\sum_{i=1}^{3} w_i}, \quad \overline{\mathbf{q}} = \frac{\sum_{i=1}^{3} w_i \mathbf{q}_i}{\sum_{i=1}^{3} w_i}.$$

The covariance matrix is defined as

$$\mathbf{H} = \sum_{i=1}^{3} w_i (\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{q}_i - \overline{\mathbf{q}})^T.$$

With the singular value decomposition $\mathbf{H} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T$ we obtain the required rotation and translation as follows:

$$\begin{aligned} \mathbf{R} &= \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}\mathbf{U}^T) \end{bmatrix} \mathbf{U}^T, \\ \mathbf{t} &= \overline{\mathbf{p}} - \mathbf{R}\overline{\mathbf{q}}. \end{aligned}$$

Figure 4.8 visualizes a stepwise application of the transformation to the sample points. Since the points contain positional uncertainties, it is possible that the samples don't match perfectly.

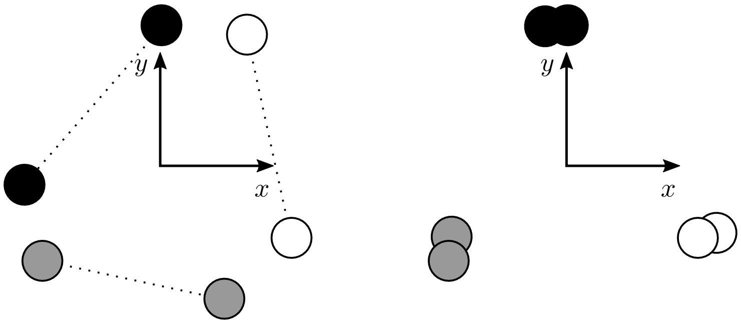


Figure 4.8: The point triplet registration consists of a translation of the two point triplets to the position where their centroids match (left) and a rotation that minimizes the sum of squared distances between corresponding points (right).

The algorithm continues by applying the resulting transformation to all the points in the source set. Afterwards, a fitness function is applied to calculate the number of inliers. Here, inliers are defined as points in the source set that have corresponding points in the target set after the transformation. Correspondences have the same label and are within a small radius of each other. They are located by means of a nearest neighbour search algorithm using a kD-tree. The search radius for correspondences was chosen to be 50 cm. The closest point with the same label is selected within this radius and added as an inlier. This is repeated for every point in the source set. The algorithm keeps track of points that have been added as inliers in order to avoid 2:1 correspondences. The fitness function also sums up the squared distances between matching points. This sum is then divided by the total number of inliers. The resulting value serves as a fitness score for a given transformation of the point set. The RANSAC search updates its estimation of the current best transformation matrix if both of the following conditions are met:

- The current number of inliers is greater or equal to the highest inlier count so far.
- The corresponding fitness score is lower than the previous best value.

The second criterion allows the estimation to improve, even if the number of inliers has not changed.

The process described in this chapter is repeated a set number of times which is adjusted whenever the number of inliers increases. The transformation resulting in the best overall fitness score along with the most inliers becomes the output. The number of iterations required to reach a satisfactory transformation within a certain degree of confidence is determined probabilistically. We intend the algorithm to provide a good result with a probability $P_{desired}$ close to 1. Let N_I be the number of inliers. It can be at most equal to the number of source points N_S, assuming that the source map is a subset of the target map. Based on an adaptation of [7], the probability of a failed match is given by

$$\begin{aligned} P_{failure} &= \left(1 - \left(\frac{N_I}{N_S}\right)^3\right)^{N_{iters}} \\ &= 1 - P_{desired}. \end{aligned}$$

From this relation we can obtain the number of iterations required to compute a successful match, assuming it exists, with the desired probability of success:

$$N_{iters} = \frac{\log(1 - P_{desired})}{\log\left(1 - \left(\frac{N_I}{N_S}\right)^3\right)}. \tag{4.5}$$

Every time a better transformation candidate is found, the iterations decrease based on Equation 4.5. Once the current iteration counter has reached the adjusted number of remaining iterations, the random sampling step of the matching algorithm is completed.

4.3.2 Refinement with Weighted Least Squares

The estimated transformation matrix generated by RANSAC is by no means the final solution to the relocalization problem, since it is merely based on a pair of sample triplets. Within the context of labelled points, the globally optimal solution has to take into account all the points that have been marked as inliers. This additional refinement is done via weighted least squares optimization with SVD, similar to the way it was done in the previous section. The optimal registration of the semantic landmarks requires two equally sized point sets that contain all the correlating points from the target and source maps. This correlation map is created from all the inlier pairs found during the previous step. When SVD is applied to the new pair of point sets, the refinement of the transformation is complete. A comparison of inlier positions before and after this step can be seen in Figure 4.9.

4.3.3 Final Refinement with ICP

So far, the matching algorithm has only taken semantically labelled points into account. Even the refined alignment may not represent the best transformation relating the current pose of the camera to the initial frame of reference. This is due to the fact that the semantic points are very sparse compared to a typical 3D point cloud used for reconstructing an environment. They are also prone to positional noise caused largely the object detector. In addition, objects with very uneven and perspective dependent features tend to add even more noise to their final position in the semantic map.

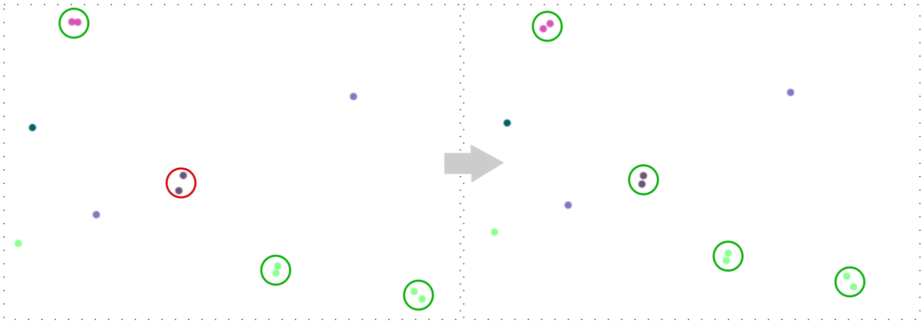


Figure 4.9: Correlations before (left) and after (righth) the global weighted least squares adjustment. The most successful RANSAC iteration returns a transformation in which a point triplet is well matched (green circles) while the remaining points are not (red circle). The refinement step adjusts the positions of all correlation candidates for a smaller total error. All other points are ignored in the process.

The iterative closest point algorithm [3] may be applied in this situation for a final refinement of the point registration, if a dense representation of the scene is available. Assuming that the initial transformation produced by the SPM is accurate enough for ICP to converge, this final refinement step may improve the estimation of the robot's position while relocalizing. The variant of ICP applied in this thesis makes use of normal estimation in point clouds [10, 23]. A simplified version of the code is given below:

Algorithm 2 ICP with normal estimation

```
src = getSourcePointCloud()
tgt = getTargetPointCloud()
srcNormals = normalEstimation(src)
tgtNormals = normalEstimation(tgt)
Ti = identityMatrix()
while iters < N do
   Ti = ICPWithNormals(srcNormals, tgtNormals) * Ti
   iters = iters + 1
return Ti
```

The value N was set to 30 to ensure that most input clouds register successfully, given that they are within the convergence radius of ICP. The effect of the algorithm on the accuracy of relocalization attempts, along with other system performance tests, are covered in the next chapter.

Chapter 5

Quantitative Evaluation on a Dataset

In order to determine the validity of the proposed method, the semantic matching algorithm was put through a series of tests based on an existing dataset designed for computer vision applications. The computing was done on an 8 core Intel Xeon E5-1630 3.70GHz processor with an Nvidia Quadro M-2000 graphics card. For most testing purposes, the algorithm itself was set to execute a minimum of 500 random sampling iterations for an increased chance to find better matches. The exact means of testing and the results are covered in this chapter.

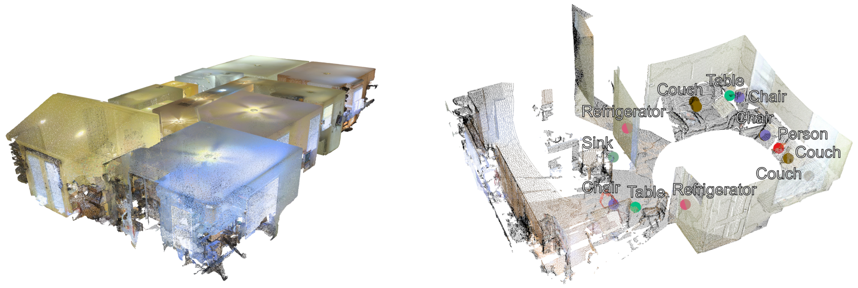
5.1 About the Dataset

The "Bosch Semantic Interpretation Challenge (Indoor)" dataset [1] is a collection of 3D reconstructions of ten buildings from a varying number of viewpoints. The individual viewpoints are a set of 36 colour and depth frames captured in a circular motion with an offset of 10° between each view. The corresponding 3D point clouds are also available, a sample of which is shown in Figure 5.1a. This dataset was chosen for an initial test of the semantic localization system due to its broad applicability. It can be used to generate simple inputs for system sanity checks, and it can be easily modified to obtain more realistic scenarios.

The generation of semantic point map data is done similarly to the method described in Section 4.2.3 with one key difference: the depth values are taken directly from the point cloud instead of the depth image. This is because the point clouds were created with a Matterport[1] system which fuses data from several sensors, thus producing point clouds that contain more complete information than a single depth frame. This method is also independent of intrinsic

[1] https://matterport.com/

sensor parameters, which are necessary for the optical projection of the labelled points in 3D space. An example of a semantically extended viewpoint from the dataset can be seen in Figure 5.1b.



(a) 3D reconstruction of building 1, which is part of the Bosch dataset.

(b) Detected objects in viewpoint 1 of building 1, depicted by the larger dots.

Figure 5.1: Sample 3D data from the Bosch dataset.

5.2 Error Metrics

In order to evaluate the accuracy of the rigid transformation matrix computed by a relocalization method, we need to compare it with a ground truth. The errors are expressed as translational and rotational deviations from this ground truth transformation. The linear error is given by

$$e_{trans} = \left\| \mathbf{t}_{GT} - \mathbf{t} \right\|,$$

where $\mathbf{t}_{GT}$ and $\mathbf{t}$ are the ground truth and estimated translations respectively. The angular deviation is calculated with the trace of the rotation matrix:

$$|e_{rot}| = arccos\left(\frac{\mathrm{Tr}(\mathbf{R}^{-1}\mathbf{R}_{GT}) - 1}{2}\right),$$

where $\mathbf{R}$ is the calculated rotation matrix and $\mathbf{R}_{GT}$ is the ground truth. These translation and rotation errors are referred to throughout the remaining thesis for system performance evaluation. Useful information is also obtained by examining the cost fitness score of the registration result, as well as other information stored in the semantic maps, including the number of labelled in different viewpoints points and their positional variances.

5.3 Single Viewpoint Registration

In this test, every viewpoint of a sample building is made into a pair of semantic maps. The first consists of the complete set of 36 frames and represents the target map. The source map is created from every second frame and is added Gaussian noise with zero mean and a standard deviation of 1 cm. The noise renders the data slightly more realistic, since the camera is very unlikely to observe two identical frames in a real use-case. Both maps are aligned by default, meaning that the semantic point matcher should produce an identity transformation matrix in an ideally successful case.

The resulting errors in Figure 5.2 show a possible output of the algorithm when applied on sample scans within the first five buildings. The runtime of a single registration is generally in the order of tens of milliseconds. Several viewpoints were considered invalid due to an insufficient number of objects in the relocalization map, resulting in 40 invalid maps out of a possible 61. Among the 21 valid scans all had linear errors of less than 20 cm, though it is not always the case. To demonstrate this, scan 16 (corresponding to viewpoint 12 in building 3) is worth a closer look. After running the algorithm several additional times, it occasionally resulted in very large errors, with translations in the order of 4 m and rotations around 180°. A failed and successful attempt can be seen in Figure 5.3. The incorrect match found 5 out of 7 possible inliers with an average distance around 8 cm between corresponding points. The correct match had 7 inliers with correspondences 11 cm apart. This shows that a better fitness function does not necessarily hint at a better match.

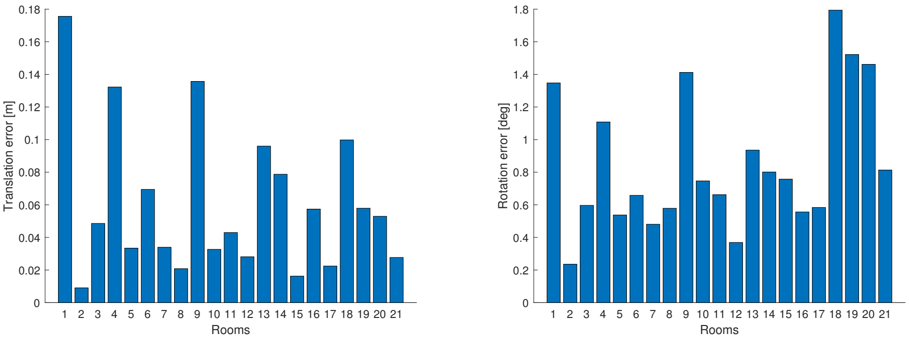


Figure 5.2: Translation and rotation errors after running the SPM on valid scans within the first three buildings of the Bosch dataset. From a total of 61 viewpoints in 3 buildings, 21 were valid for this test (had at least 3 detections in the source map).

In order to get a more general understanding of these errors, the algorithm was repeated 100 times on all the valid viewpoints of building 3. The resulting histograms are shown in Figure 5.4. For viewpoint 12, the large linear errors of several metres occur in about 24% of the trials. These are the only outliers with errors larger than 1 m when looking at the combined picture. Among the 600 samples, 73% have errors less than 10 cm.

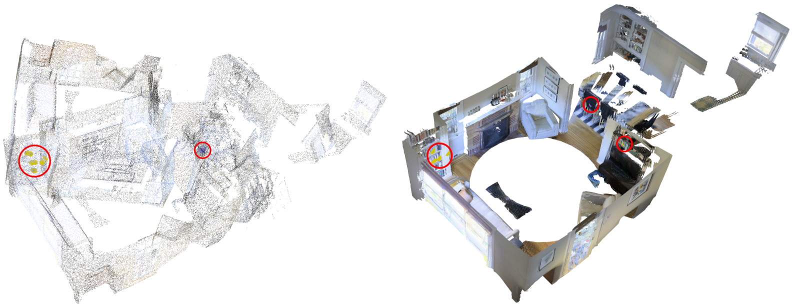


Figure 5.3: A failed (left) and successful (right) registration of a sample viewpoint. The red circles show the locations of labelled points that were successfully matched. The left image was downsampled to make the labelled points more visible.

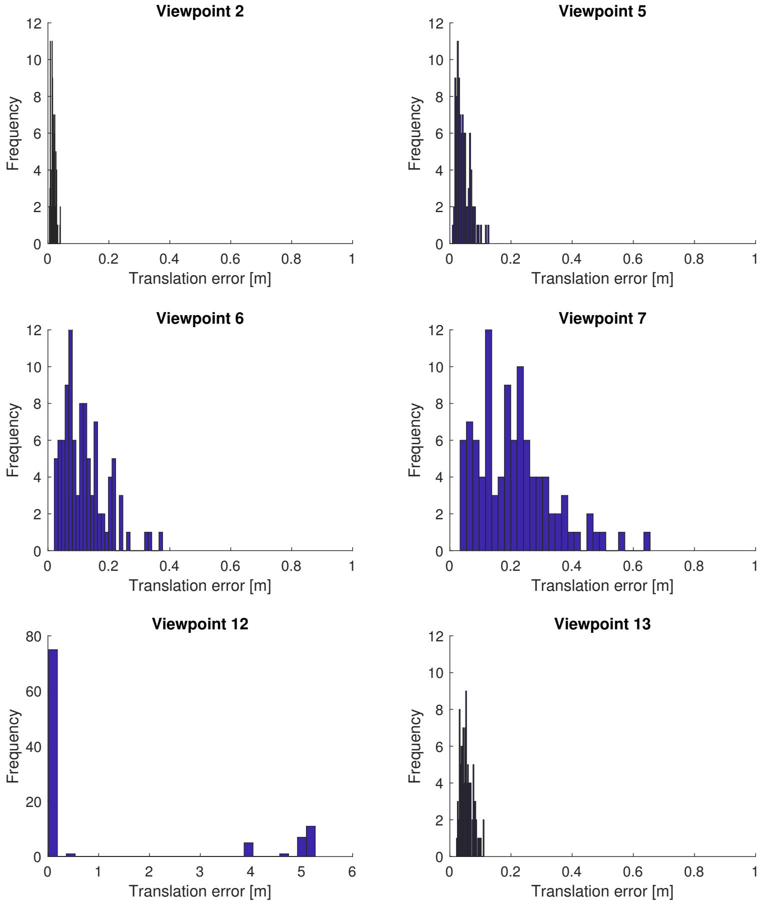


Figure 5.4: Histograms of translational errors for all valid viewpoints in building 3. Viewpoint 12 is scaled differently on both axes due to outliers.

5.4 Combined Viewpoints with Added Noise

Complexity is added to the test in a different way by combining all the complete semantic maps of the different viewpoints into a single map of the whole building to form a new target map. The new source maps are semantically extended reconstructions of the individual viewpoints, using all 36 frames. Again, the quality of the registration is judged by the amount of deviation between the system output and an identity transformation. Additionally to the previous test, the amount of noise is increased over several runs of the algorithm, with the aim to determine to what extent unreliable data can influence the accuracy of the SPM and its ability to relocalize in the correct room among many others.

Positional noise was added to the points of the relocalization map with zero mean and standard deviations of 1, 5, 10, 15 and 20 cm. Figures 5.5 and 5.6 show the translation and rotation errors respectively. They are depicted as functions of the added noise in 5 different buildings for single registration attempts. Random sampling tends to produce occasional incorrect values sporadically for some viewpoints while other registrations are still accurate. For this reason, the aforementioned figures also feature plots of median errors. It can be observed that the point matching results are rather robust in most cases with noise below 10 cm, with values that generally deviate by less than 50 cm. An examination of the semantic maps of the analysed buildings reveals that the average positional standard deviation of labelled points is in the order of 5 cm or less. Within this range, the median errors are 20 cm and 1°. Buildings 4 and 5 show some relatively large errors with small amounts of noise, especially in the rotation error. This oddity can in part be explained by the fact that a total of 48 objects contributed to the semantic map of building 4, while the other buildings had 74 to 112 objects. A lower number implies fewer registration candidates per viewpoint, which can cause the SPM to yield less accurate results. The relationship between errors and the number of objects is considered further in Figure 5.7. The largest errors originate from the buildings 4 and 5 which also have the least objects in the corresponding viewpoints. Among the viewpoints that have more than 10 detections, only one case resulted in an error greater than 20 cm.

For an additional impression of the semantic matcher's performance, Figure 5.8 demonstrates several attempts at reconstructing building 4 with increasing noise, in comparison with the actual reconstruction. Overall, the results from this test show a good performance of the algorithm, even with unreliable data.

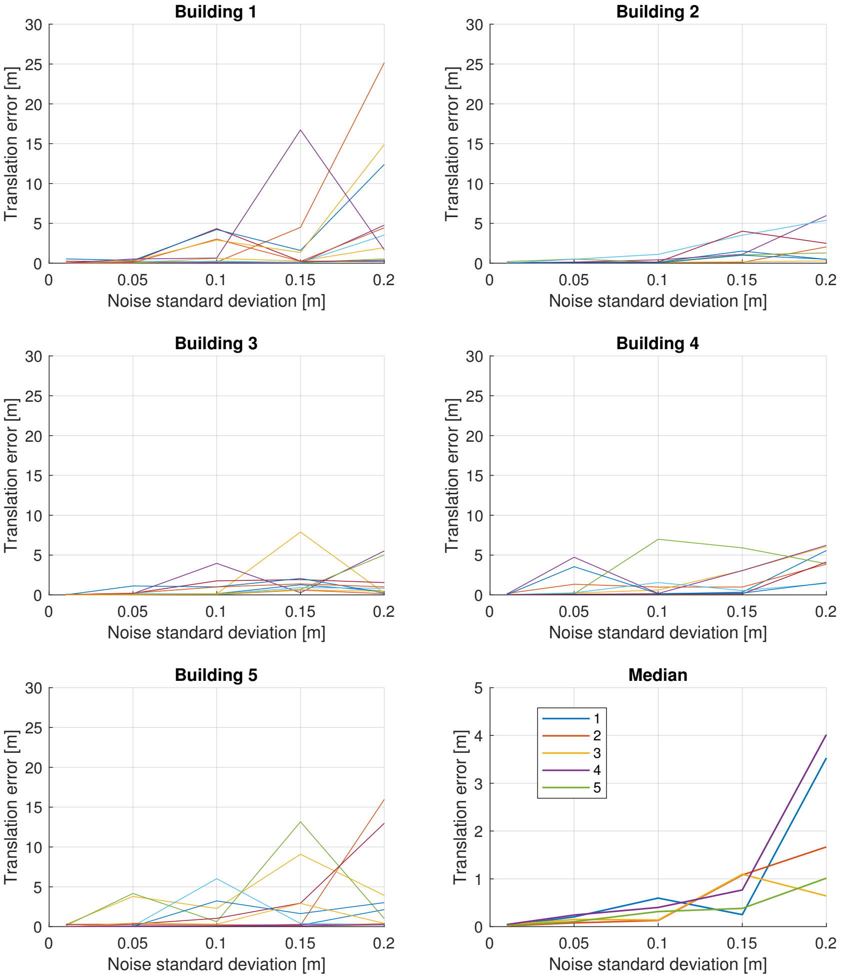
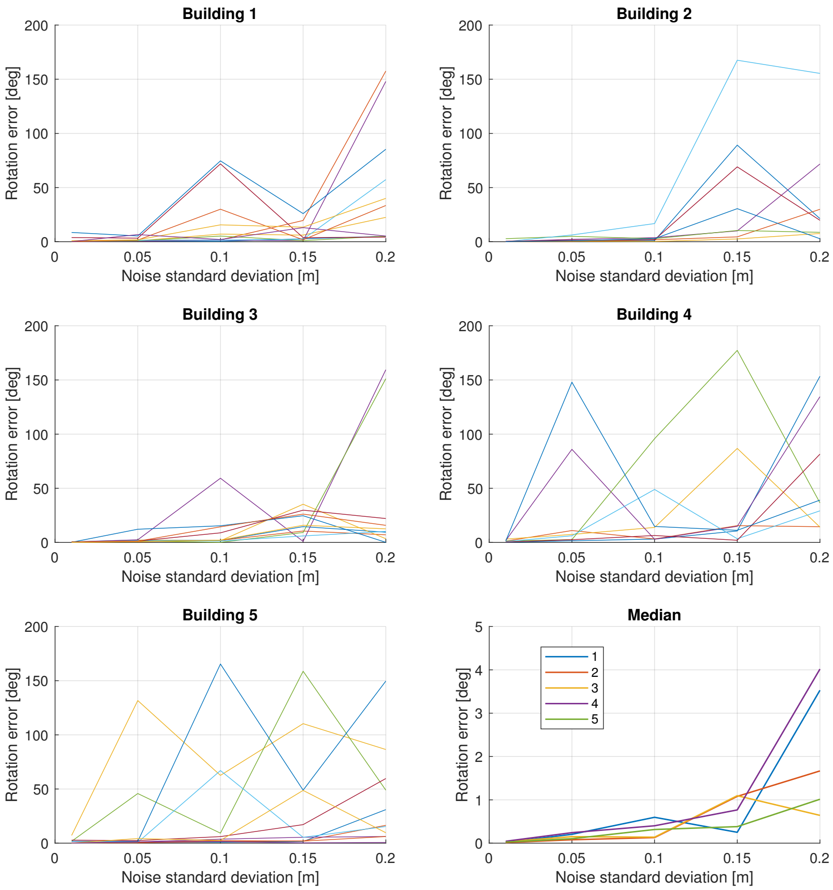


Figure 5.5: Translation errors after running the SPM once on valid viewpoints in the first 5 buildings with increasing positional noise for the labelled points. The bottom right plot shows median errors of each building.


Building 1

Building 2

Building 3

Building 4

Building 5

Median

Figure 5.6: Rotation errors and median after running the SPM.

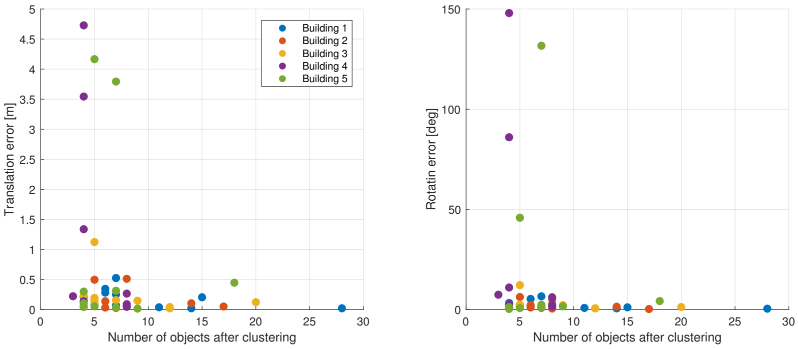
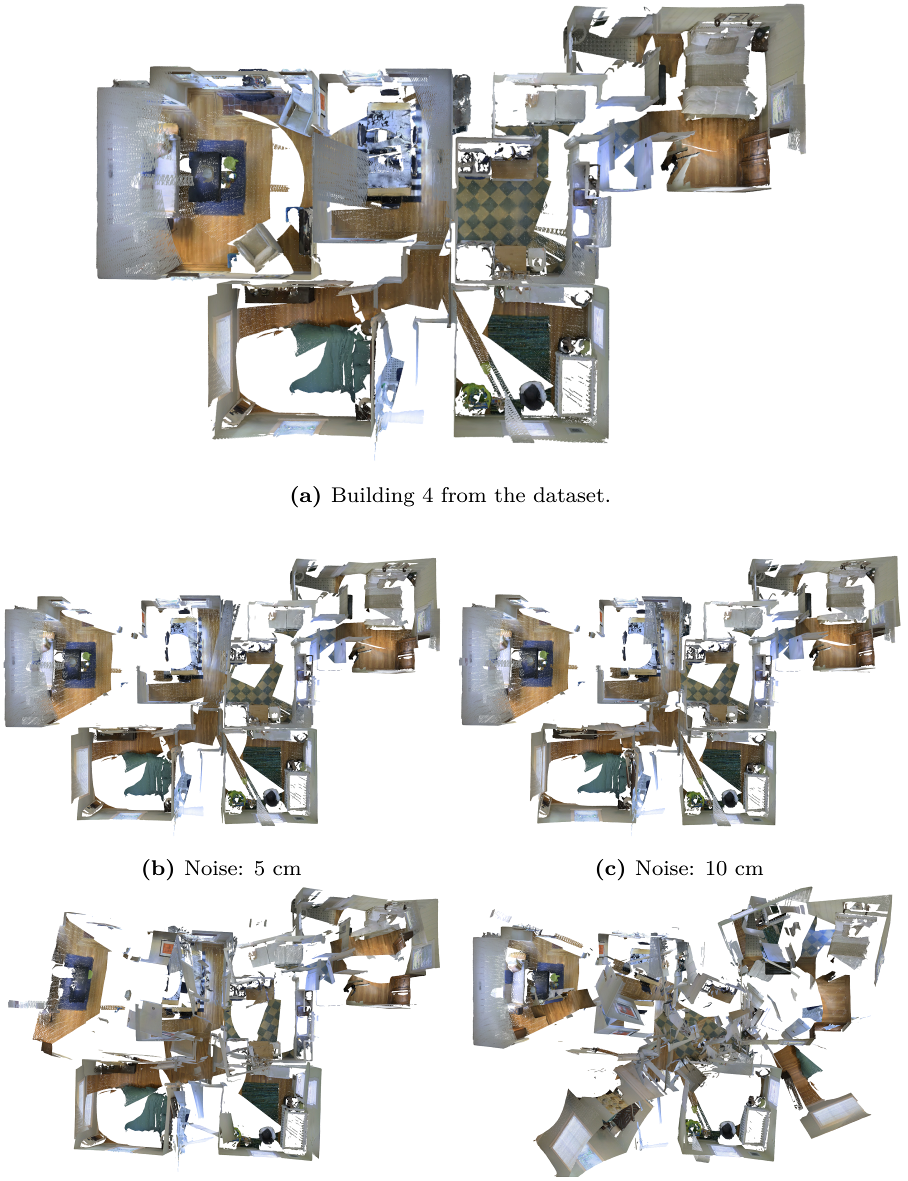


Figure 5.7: Translation and rotation errors resulting from the SPM in relation to the number of objects in each viewpoint of the tested buildings. The errors are taken from the combined viewpoint registration test with a noise standard deviation of 5 cm.



(a) Building 4 from the dataset.

(b) Noise: 5 cm

(c) Noise: 10 cm

(d) Noise: 15 cm

(e) Noise: 20 cm

Figure 5.8: Relocalized and combined viewpoints with various amounts of noise compared to the 3D reconstruction from the dataset.

5.5 Effect of Clustering

So far, every registration has been the result of a match between two semantic maps that have been compressed via clustering, as explained in Section 4.2.3. Clustering has the advantage that outliers resulting from incorrect detections are removed, and the final position of objects in the map is improved via averaging. However, the first two tests revealed that this process also removes a number of potentially useful detections from the semantic map, especially if every object is observed in a limited number of frames. This problem is clarified in Figure 5.9 where a bag is seen a total of three times during a scan of 20° intervals. For cameras field of view of about 60°, this can occur for many objects. If the detector fails to perceive the object in even one of these frames, the clustering based on a minimum of three points will also fail.

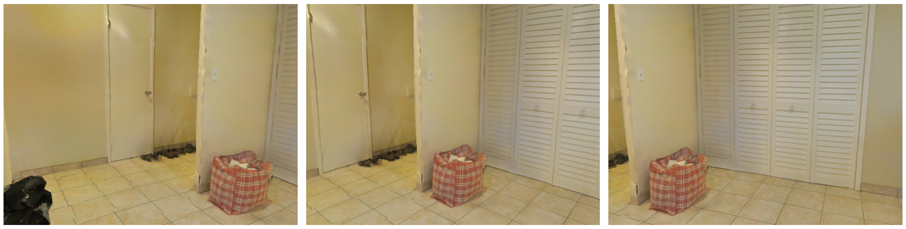


Figure 5.9: Three views on the same object with an angular offset of 20°. Images are taken from the Bosch dataset.

The strong decline in the number of detected objects is additionally shown in Figure 5.10. If every available frame of the dataset is fed into the SMG, a total of 1135 objects is found in the ten buildings. If every second frame is used for the map generation, this number drops to 430. This substantial difference suggests that there may be advantages in ignoring the clustering step while processing the semantic points.

This idea was tested by creating a pair of clustered and non-clustered semantic maps from the buildings 6 to 10 by using all even-numbered frames for the source map and all odd-numbered frame for the target. No noise was added this time, since both maps were based on a disjoint set of inputs. The registration algorithm was then applied on the clustered versions of the input maps as well as the full maps containing all detections. The resulting translation and rotation errors are compiled in Figures 5.11 and 5.12. It can immediately be seen that the SPM did not have problems finding a registration estimate for nearly all test cases, producing an average deviation of 29.6 cm and 4.6° with two cases having an error in the order of a few metres. Without these outliers, the average error becomes 13.1 cm and 1.4°. The non-clustered map registration was successful in each viewpoint of each tested building, except for viewpoints 20 and 21 of building 10, which didn't have enough detections to make a correct

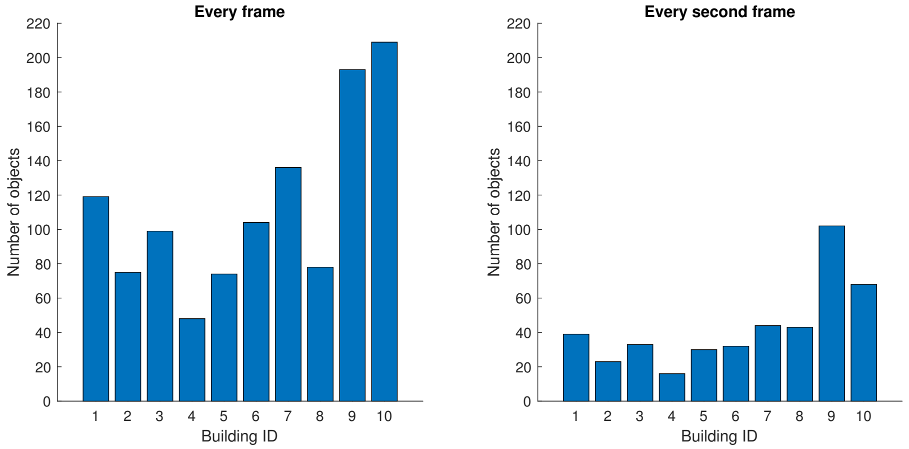


Figure 5.10: Number of objects in the different buildings based on an angular offset of 10° (left) and 20° (right) between each frame of the scan.

registration, even before clustering. In contrast, the clustered registration only yielded results in 29 out of 66 cases with three instances of errors greater than 1 m. If these outliers are ignored, the mean error of clustered matching is 11.6 cm and 1.5° (otherwise it jumps to 1.13 m and 7.6°).

When comparing runtime performance, the clustered variant has an advantage in the RANSAC search. On the used machine, the average runtime of the clustered variant is 7.3 ms with a maximum value of 99.6 ms. Without clustering, the average rose to 32.2 ms with a maximum of 244.1 ms. Since the compressed map performs about 4 times faster than the full map on average, it can be assumed that clustering has advantages when working with maps of much larger scale. The input data in tests with the Bosch dataset consisted of 36 frames at most, meaning that even the non-clustered maps were rather sparse. The next section considers maps that were generated from more than a thousand frames for the relocalization algorithm.

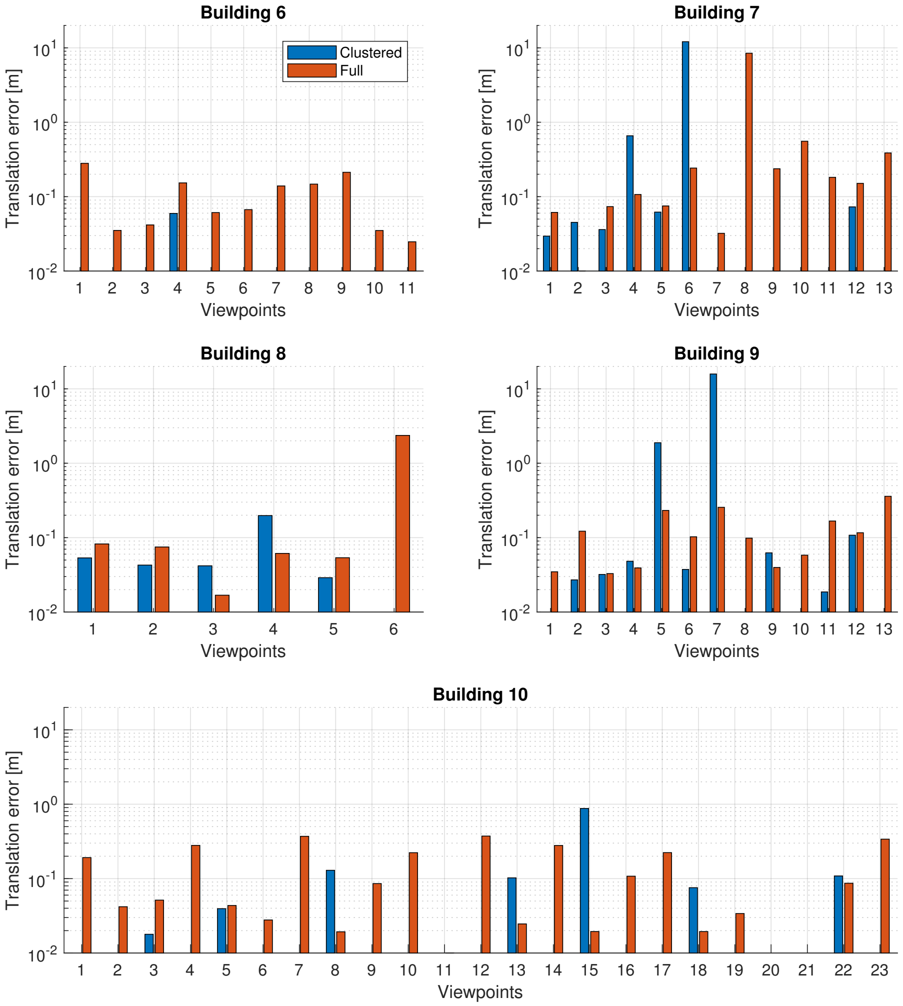


Figure 5.11: Translation errors resulting from registrations based on clustered and non-clustered sets of detections. The y-axis is set on a logarithmic scale for better visibility of large error differences. The missing bins stem from matching attempts that failed or did not have enough input points.

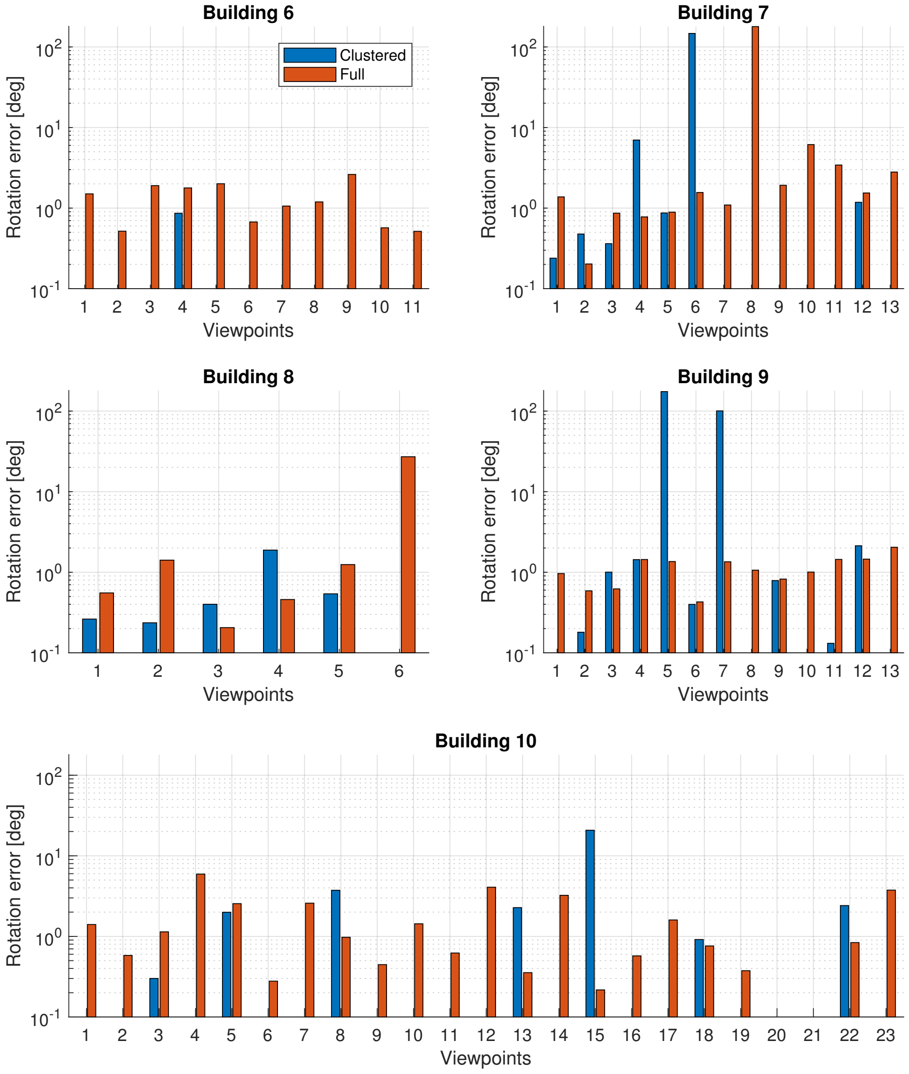


Figure 5.12: Rotation errors resulting from registrations based on clustered and non-clustered sets of detections, with a logarithmic y-axis.

Chapter 6

Qualitative Tests in Real Environments

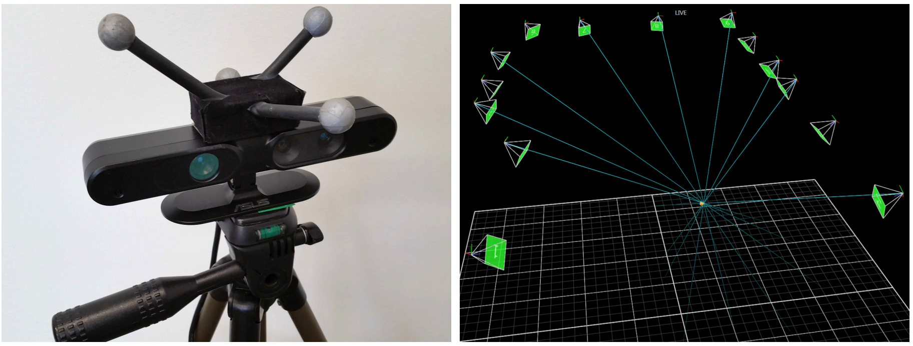
After validating the proposed approach we performed mapping and relocalization experiments on self-acquired data in different environments in a variety of circumstances to compare the performance of the SPM with the relocalization routine of ORB-SLAM2. The experiments described in this Chapter were completed in the Institute for Robotics and Mechatronics at the German Aerospace Center in a typical office environment and in the mobile robotics laboratory.

6.1 Accuracy Test with Vicon

When moving from datasets to tests with real environments, it is highly valuable to have a ground truth measurement for reference. In the case of this relocalization problem, the ground truth was provided by a Vicon motion capture sensor array in the mobile robotics laboratory (Figure 6.1). The goal of this experiment is to determine the accuracy of the semantic point matcher in a relatively large environment with realistic data.

6.1.1 Experiment Setup

The Xtion sensor was first equipped with a set of markers. Then it was placed at a chosen location and ORB-SLAM2 was started. The initial position of the sensor was recorded from two perspectives: the Vicon coordinate system and the local camera frame via ORB-SLAM2. The 3D reconstruction of the laboratory (from stored keyframes) is visualized in Figure 6.2. In the semantic map, the detected objects include a number of screens, chairs and keyboards, as well as a bottle, cup, suitcase and remote among a total of 59 detections. Since the detector was not trained for this environment, the semantics do not necessarily hint at a robotics laboratory. Figure 6.3 reveals that the density of detections



(a) Asus Xtion camera with markers.

(b) Vicon sensor array.

Figure 6.1: Experiment setup for acquiring the ground truth.

is much higher in the bottom area where all the monitors are located, while the top area failed to locate a particularly high number of objects. This tells us that a relocalization attempt is less likely to make use of the upper area of the lab and risks finding faulty registrations.

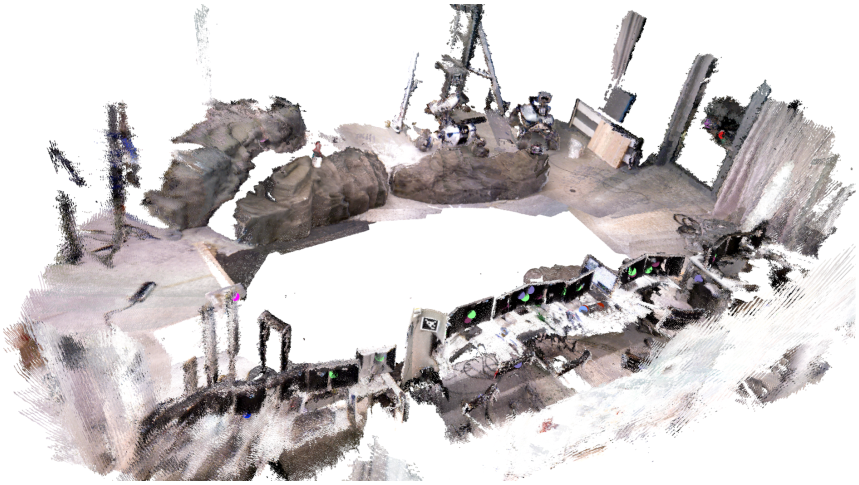


Figure 6.2: Semantically extended 3D point cloud of the mobile robotics laboratory.

6.1.2 Relocalization with SPM

To test the effectiveness of the semantic point registration algorithm, a set of three relocalization scans was made in three different locations within the working volume of the Vicon sensor array. The point maps from the partial views were then matched against the complete map.

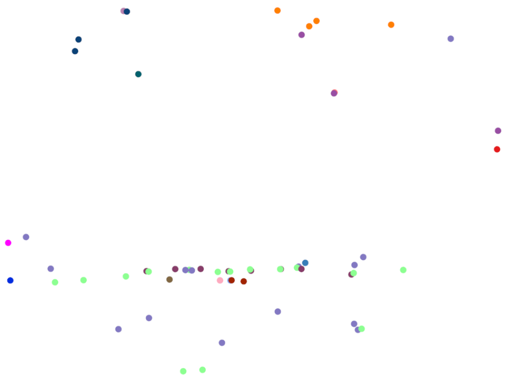


Figure 6.3: Top-down perspective on the semantic points of the mobile robotics laboratory.

During the first relocalization attempt, the algorithm failed to find common landmarks in the rocky area of the laboratory. In fact, all of the inliers consisted of approximately collinear objects around the desks, including screens, chairs and a cup. Due to this, the proposed match was only correct in that region, as seen in Figure 6.4.

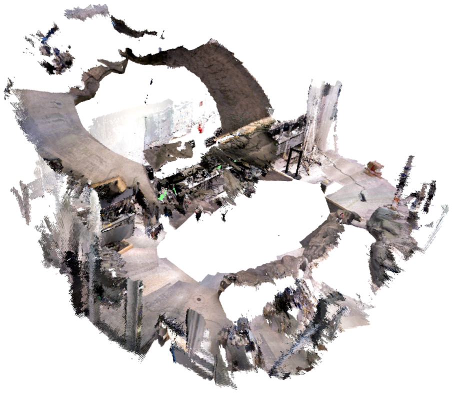


Figure 6.4: Failed registration attempt due to common landmarks being nearly collinear in one area.

The second attempt was more successful, though the reference objects were still exclusively in the desk area (Figure 6.5). The SPM estimation was accurate enough to be a valid initial guess for ICP, having a translation and rotation errors of 37.62 cm and 3.73° respectively, compared to the transformation measured by Vicon. After ICP, the deviations went down to 5.29 cm and 0.09°.

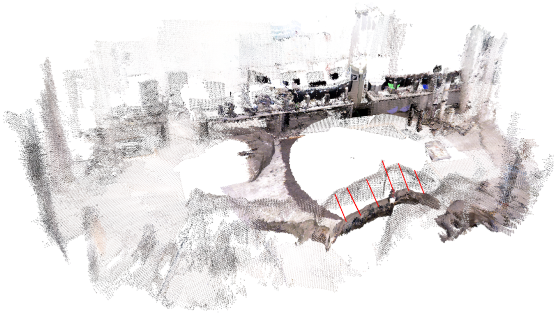


Figure 6.5: Registration attempt of test 2. The red lines mark corresponding areas that have not been properly registered by SPM.

The third attempt was the most successful one, resulting in a quite close match between the two views with deviations of 15.11 cm and 0.15°, which changed to 9.7 cm and 0.83° after applying ICP.

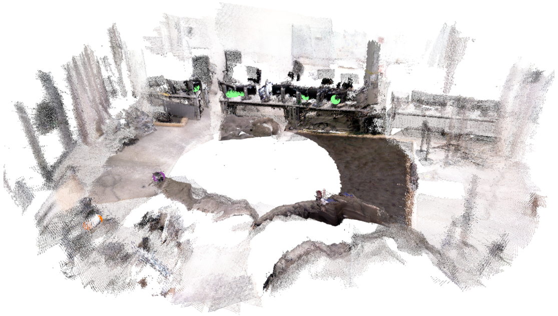


Figure 6.6: Registration attempt of test 3.

These experiments illustrate the initial considerations of Section 1.2: the accuracy of SPM depends not only on the existence of objects, but also their distribution around the room.

6.1.3 Comparison with ORB-SLAM2

For this test, the camera was placed in three new locations that successfully triggered the relocalization routine of ORB-SLAM2. The results are compiled in Table 6.1, which shows translation and rotation errors that are in a similar order of magnitude as the previously obtained results from the SPM.

Position	e_{trans} [cm]	e_{rot} [deg]
1	7.9	1.1
2	21.6	1.5
3	17.1	2.0

Table 6.1: Translation and rotation errors of the relocalization by ORB-SLAM2.

During these relocalization attempts with ORB-SLAM2, it was made apparent that the keypoint tracker is unable to find matches reliably from numerous positions. The camera had to be moved around for a while until it was relocalized successfully. A simple rotatory motion of the tripod was often insufficient for the task. This may not be an issue with flying robots that can scan the environment very fast, but less dynamic robots like rovers may require a lot of time to properly relocalize themselves using ORB-SLAM2.

6.2 Relocalization in Dynamic Environments

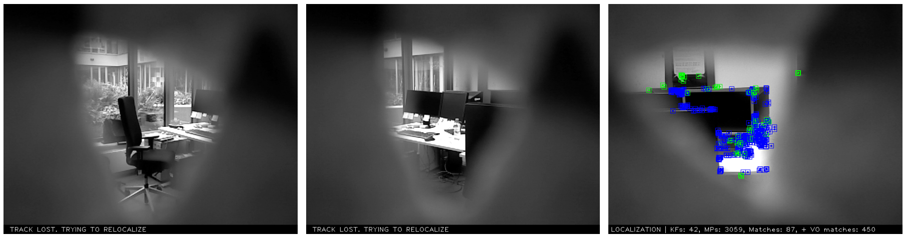
To demonstrate the benefits of semantic maps, we compare the SPM with the relocalization routine of ORB-SLAM2 in different office scenarios that showcase the robustness of object detections against visual keypoints. According to Mur-Artal *et al.* [17], their method can handle keyframe scale changes between 0.36 and 2.93 and an angle difference of 59° in the optical axis. While this allows for a rather wide range of cases where the sensor can be successfully tracked, there are some scenarios where the visual feature matcher may fail compared to a semantic method. This section discusses such cases.

6.2.1 Occlusions to the Sensor's Field of View

The first example is a situation where the sensor has a heavily impaired field of vision during the relocalization attempt. This effect was achieved by putting bits of paper around the camera lens and doing a partial scan (about 270°) of the room by using a tripod. The initial map was produced with a 360° scan in the same position without any obstructions. Afterwards, the semantic point matcher and the place recognition module of ORB-SLAM2 were evaluated in these conditions.

The relocalization routine of ORB-SLAM2 failed to find the obstructed sensor's location over a considerable range of frames, but it did succeed in visually denser areas, as depicted in Figure 6.7. For the initial position of the camera $(0, 0, 0)$, ORB-SLAM2 produced a positional error of 59.5 cm. It is also noteworthy that the results were worse when attempting the same with the obstructed map in a "clean" environment, where very few of the frames returned a match. The same conditions were tried on SPM, which resulted in the estimated position shown in Figure 6.8 on the left. This estimation is quite close, and it is based on 6 reference points. The pose after applying ICP can be seen on the right. Before

ICP, the positional error was 7.0 cm and it changed to 9.0 cm after the adjustment, showing that ICP may not always result in more accurate localization of the sensor.



```
TRACK LOST. TRYING TO RELOCALIZE
TRACK LOST. TRYING TO RELOCALIZE
LOCALIZATION | KFs: 42, MPs: 3059, Matches: 87, + VO matches: 450
```

Figure 6.7: Partially successful attempts to relocate by ORB-SLAM2.

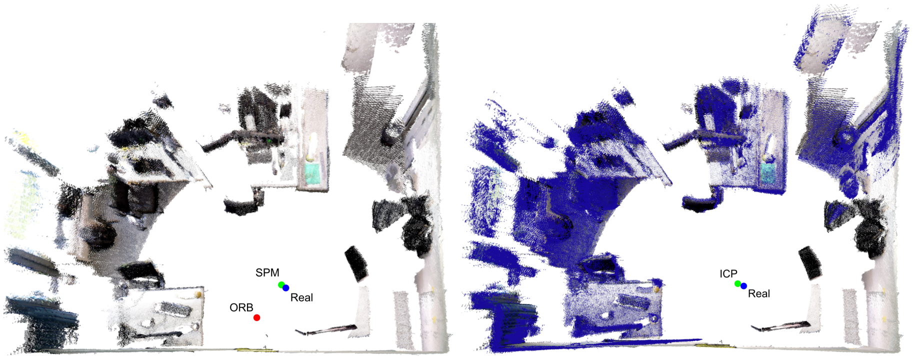


Figure 6.8: Overlapping 3D maps of the office after SPM (left) and ICP (right). The larger coloured dots represent the position estimation by ORB-SLAM2 (red), SPM (green) and the real position of the sensor (blue).

We can observe that object detection is comparatively less affected by visual impairment than keypoint matching. Additionally, we can see that the semantic point matcher can be accurate, even when the number of inliers is rather small.

6.2.2 Strong Lighting Changes

Another example is a situation where the relocalization environment has dramatically different lighting conditions than the initial map. To test this, two revolutions of the camera were made at the same viewpoint – once with a moderately high level of brightness and once in a much darker setting. For a visual impression of the illumination differences, refer to Figure 6.9.

After the captures were made, the relocalization routine of ORB-SLAM2 was tested by using the darker map to locate the sensor within the bright room and

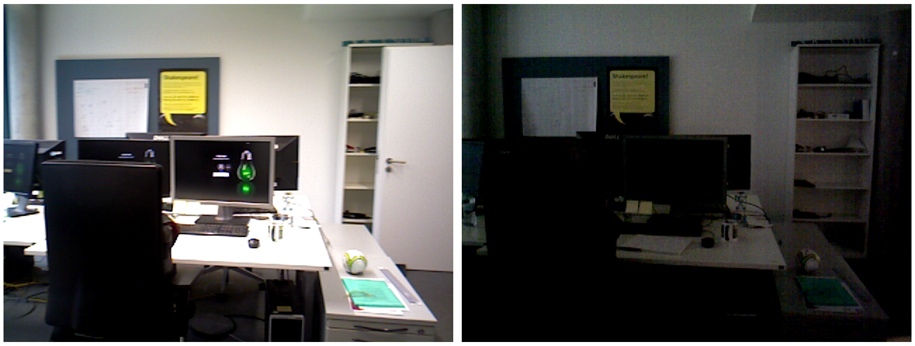


Figure 6.9: Brightness difference in the two scans of the scene.

vice versa. ORB-SLAM2 failed to relocate the camera in both cases. SPM, on the other hand, found a rough estimate of a transformation relating both maps of the office, as depicted in Figure 6.10. Similarly to the previous test, the overlap between the two maps is once again well within the convergence range of ICP. The estimated pose matrices before and after ICP are the following:

$$\begin{aligned}
\mathbf{P}_{SPM} &= \begin{bmatrix} 0.9999 & 0.00968 & -0.00583 & -0.0529 \\ -0.0097 & 0.9999 & 0.0041 & -0.0438 \\ 0.0059 & -0.0040 & 0.9999 & -0.0599 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{P}_{ICP} &= \begin{bmatrix} 0.9999 & 0.0015 & -0.0114 & -0.0419 \\ -0.0014 & 0.9999 & 0.0088 & -0.0251 \\ 0.0114 & -0.0088 & 0.9999 & -0.0429 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

The pose estimated by SPM deviates from the unit matrix by a distance of 9.11 cm and an angle of 0.69°. After ICP the deviation changed to 6.50 cm and 0.83°. While ICP did not improve the orientation error, this experiment still resulted in a successful relocalization in an environment where feature matching had failed.

6.2.3 Relocalization from Different Perspectives

The third test discussed in this section involves relocalization in partially overlapping maps taken from two different perspectives. It serves to verify the robustness of the algorithm in situations where a complete map is not available. While visual feature matching is robust to scale and rotation changes, we expected this case to be too extreme for the tracking module of ORB-SLAM2 to handle. The semantic localization system, on the other hand, only requires to detect a small common set of similar objects in both views to find a match. With

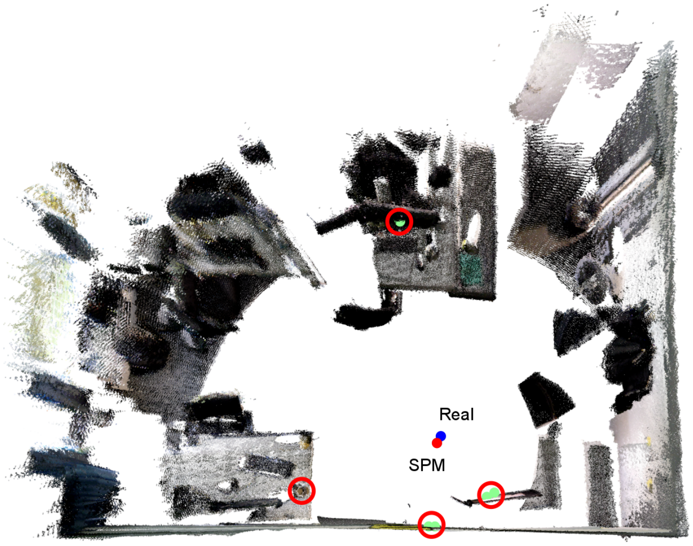


Figure 6.10: Target point cloud (light) and transformed source point cloud (dark) overlapped and visualized as RGB point clouds. The circled pairs of coloured blobs represent the semantic landmarks that were successfully correlated by the matching algorithm: one cup and three screens. The blue and red dots represent the real and estimated positions of the sensor respectively.

this thought, a pair of partial maps was generated based on the plan shown in Figure 6.11. The 3D maps themselves are shown in Figure 6.12.

After the recordings, the camera was put in position 1 with the map of position 2 and vice versa. The results of relocalization attempts were similar to the previous test. ORB-SLAM2 failed to find a match along the whole field of view from both perspectives. SPM produced the pose estimation depicted by the overlapping point clouds in Figure 6.13. This specific matching attempt found 9 semantic correlation candidates. Since the exact location of the camera from one frame to the other could not be measured, the accuracy was estimated by comparing the transformations implicitly by computing the transformation from one perspective and applying the inverse to project it back to the initial pose. These two poses differed from one another by 0.79 cm and 0.17°.

Overall, after completing these experiments we can observe that the object detector shows more robustness towards harsh visual changes in the environment. As long as there are detectable objects in the camera's field of vision, the semantic point matcher can produce pose estimates that are at least in the range of ICP and may even have linear accuracies up to a few centimetres and deviations in the orientation of less than 1°.

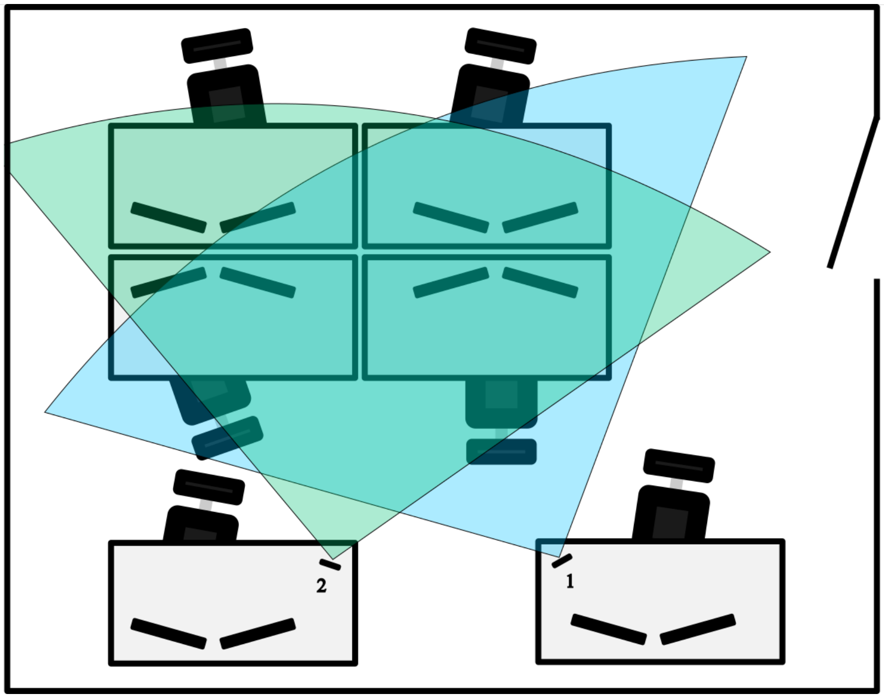


Figure 6.11: Simplified top-down plan of the office where the tests were carried out. The two circular sectors represent the approximate viewing angles of both scans.

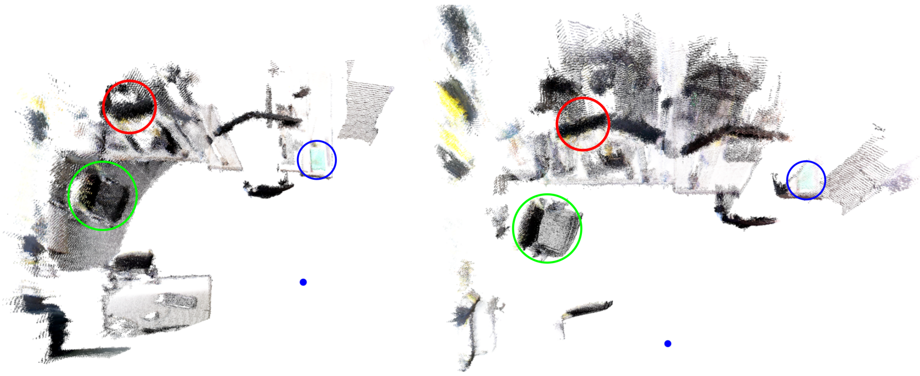


Figure 6.12: Top view of 3D reconstructions from both perspectives. Some elements are marked to make the similarities clearer. The blue dot represents the position of the camera in both frames of reference.

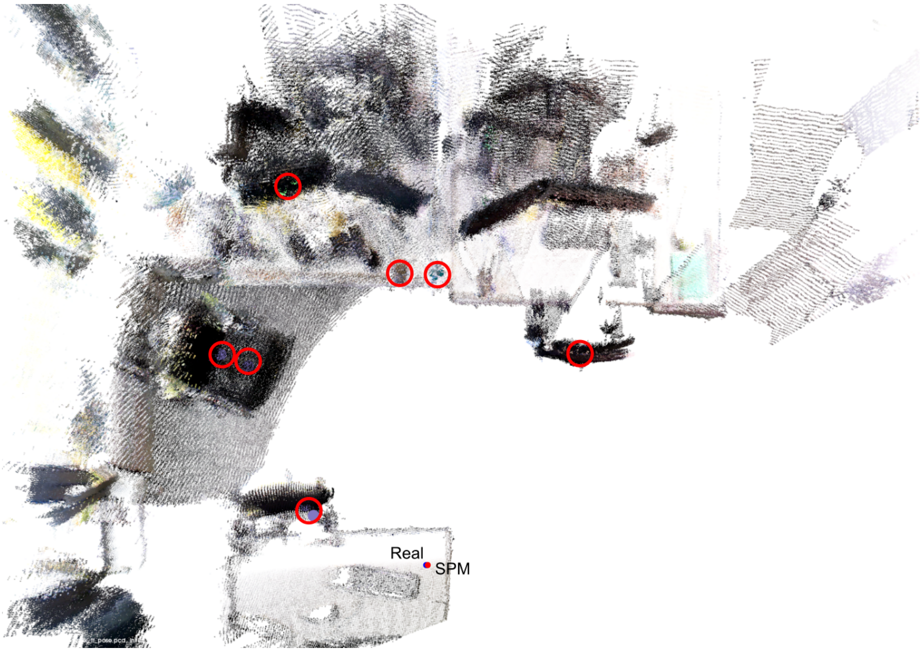


Figure 6.13: Registration of two semantically extended point clouds of the office from two different perspectives. The blue dot represents the estimated position by SPM and the red dot is the actual position of the sensor. The red circles mark the visible semantic correlations that resulted in the registration result. The landmarks include chairs, screens, a bag, a cup and a bottle among other items.

Chapter 7

Conclusions

The semantic point matcher (SPM) developed and tested in this thesis is a method for robust and accurate 6 DoF relocalization of lost robots in an existing 3D point map, created with the semantic point map generator (SMG). The algorithm can reliably achieve linear and angular accuracies of less than 10 cm and 1° respectively, as shown both by quantitative tests with the Bosch Semantic Interpretation Challenge dataset, as well as experiments on self-made scans of an office environment and the mobile robotics laboratory in the Institute for Robotics and Mechatronics of the German Aerospace Center (DLR).

The quantitative tests showed that the proposed method can handle unreliable point data with positional noise up to 10 cm successfully in the majority of cases. The SMG is also capable of successful relocalization when only a very limited number of common objects have been detected between the global and local map. The tests demonstrated some benefits and limitations of clustering the semantic point data before applying the SPM. In both cases the accuracy is comparable, but skipping the clustering step permits relocalization in very sparse maps where many detections would be otherwise filtered out. However, compressing the semantic map does improve runtime performance and is well suited for larger maps.

The experiments on real data revealed more favourable features of the SPM in a test where it was compared with the relocalization routine of ORB-SLAM2. One of the tests compared both methods in an office environment where the original map was drastically different from the current view, either due to obstructions, lighting differences or perspective changes. The semantic matcher consistently outperformed its counterpart, both in accuracy and in the rate of successful registration attempts. The mobile robotics laboratory presented a challenge for relocalization due to its lack of commonly detectable objects, but the successful pose estimations deviated by less than 10 cm and 1° from the ground truth measured by a Vicon motion capture sensor array.

The proposed method is not devoid of limitations. An obvious weakness of the SPM is its dependency on correctly detected objects in a room. When working with sparse maps, the algorithm may find a solution that is supported by a good fitness score, but does not relocalize the robot properly. Because of this, a reasonable way to apply the algorithm is by integrating it in existing SLAM systems. If several different relocalization routines work in cohesion, we may increase the likelihood of a successful pose estimation. Many robots use object detectors for various purposes, and extending them with the semantic map generator would not add a lot of weight to the existing processes, and in turn provide the robot with additional tools to understand its environment on a more detailed level.

Chapter 8

Future Outlook

Since the SMG and SPM show promise as extensions of SLAM, a reasonable development would be to implement a purely online relocalization routine that can make proposals and adjustments to existing methods to increase the accuracy of pose estimations in real time.

Other desirable improvements can focus on the runtime performance of the SPM. A rather simple yet effective addition could be GPU acceleration for the RANSAC step of the relocalization routine, which would allow the various transformation candidates to be processed in parallel for considerably faster performance. Another way of potentially improving the point matching speed is to add a long term reliability measure to the objects in a room based on their displacement over several relocalization attempts. If such points are favoured during the RANSAC pose estimation step, the correct transformation may be found in fewer iterations.

For environments that are densely populated with just one type of object, it may be beneficial to extend the labelled points with visual feature descriptors in a small region around them. This would add more uniqueness to each point in the semantic map, allowing more certainty in proposed pose estimations. Reliability may be added to the position of every object by using a detector that also produces segmentation masks. A closely cropped region opens up new possibilities for a more accurate mapping of every detection.

Bibliography

[1] Rares Ambrus, Sebastian Claici, and Axel Wendt. Automatic room segmentation from unstructured 3-d data of indoor environments. *IEEE Robotics and Automation Letters*, 2(2):749–756, 2017.

[2] Kai O Arras, Oscar Martinez Mozos, and Wolfram Burgard. Using boosted features for the detection of people in 2d range data. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3402–3407. IEEE, 2007.

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[4] Manuel Brucker, Maximilian Durner, Rares Ambrus, Zoltán Csaba Márton, Axel Wendt, Patric Jensfelt, Kai O Arras, and Rudolph Triebel. Semantic labeling of indoor environments from 3d rgb maps.

[5] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005.

[6] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[9] Alejandra C Hernández, Clara Gómez, Jonathan Crespo, and Ramón Barber. Object classification in natural environments for mobile robot navigation. In *Autonomous Robot Systems and Competitions (ICARSC), 2016 International Conference on*, pages 217–222. IEEE, 2016.

[10] Stefan Holzer, Radu Bogdan Rusu, Michael Dixon, Suat Gedikli, and Nassir Navab. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2684–2689. IEEE, 2012.

[11] Omid Hosseini Jafari, Dennis Mitzel, and Bastian Leibe. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5636–5643. IEEE, 2014.

[12] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645, 2011.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Zoltán Csaba Márton, Serkan Türker, Christian Rink, Manuel Brucker, Simon Kriegel, Tim Bodenmüller, and Sebastian Riedel. Improving object orientation estimates by considering multiple viewpoints. *Autonomous Robots*, 42(2):423–442, 2018.

[16] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[17] Raúl Mur-Artal and Juan D Tardós. Fast relocalisation and loop closing in keyframe-based slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 846–853. IEEE, 2014.

[18] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[19] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[20] Andrzej Pronobis and Patric Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3515–3522. IEEE, 2012.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1137–1149, 2017.

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[23] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[24] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[25] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[26] Yiploon Seow, Renato Miyagusuku, Atsushi Yamashita, and Hajime Asama. Detecting and solving the kidnapped robot problem using laser range finder and wifi signal. In *Real-time Computing and Robotics (RCAR), 2017 IEEE International Conference on*, pages 303–308. IEEE, 2017.

[27] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *no*, 3:1–5, 2017.

[28] Zerong Su, Xuefeng Zhou, Taobo Cheng, Hong Zhang, Baolai Xu, and Weinan Chen. Global localization of a mobile robot using lidar and visual features. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pages 2377–2383. IEEE, 2017.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[30] Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, et al. Spencer: A socially aware service robot for passenger guidance and help in busy airports. In *Field and service robotics*, pages 607–622. Springer, 2016.