Local Visual Feature based Localisation and Mapping by Mobile Robots

Örebro Studies in Technology 31

Henrik Andreasson

Local Visual Feature based Localisation and Mapping by Mobile Robots

Title: Local Visual Feature based Localisation and Mapping by Mobile Robots

Publisher: Örebro University 2008
www.publications.oru.se

Editor: Maria Alsbjer
maria.alsbjer@oru.se

Printer: Intellecta DocuSys, V Frölunda 08/2008

ISSN 1650-8580
ISBN 978-91-7668-614-0

Abstract

This thesis addresses the problems of registration, localisation and simultaneous localisation and mapping (SLAM), relying particularly on local visual features extracted from camera images. These *fundamental problems* in mobile robot navigation are tightly coupled. Localisation requires a representation of the environment (a map) and registration methods to estimate the pose of the robot relative to the map given the robot's sensory readings. To create a map, sensor data must be accumulated into a consistent representation and therefore the pose of the robot needs to be estimated, which is again the problem of localisation.

The major contributions of this thesis are new methods proposed to address the registration, localisation and SLAM problems, considering two different sensor configurations. The first part of the thesis concerns a sensor configuration consisting of an omni-directional camera and odometry, while the second part assumes a standard camera together with a 3D laser range scanner. The main difference is that the former configuration allows for a very inexpensive set-up and (considering the possibility to include visual odometry) the realisation of purely visual navigation approaches. By contrast, the second configuration was chosen to study the usefulness of colour or intensity information in connection with 3D point clouds ("coloured point clouds"), both for improved 3D resolution ("super resolution") and approaches to the fundamental problems of navigation that exploit the complementary strengths of visual and range information.

Considering the omni-directional camera/odometry setup, the first part introduces a new registration method based on a measure of image similarity. This registration method is then used to develop a localisation method, which is robust to the changes in dynamic environments, and a visual approach to metric SLAM, which does not require position estimation of local image features and thus provides a very efficient approach.

The second part, which considers a standard camera together with a 3D laser range scanner, starts with the proposal and evaluation of non-iterative interpolation methods. These methods use colour information from the camera to obtain range information at the resolution of the camera image, or even

with sub-pixel accuracy, from the low resolution range information provided by the range scanner. Based on the ability to determine depth values for local visual features, a new registration method is then introduced, which combines the depth of local image features and variance estimates obtained from the 3D laser range scanner to realise a vision-aided 6D registration method, which does not require an initial pose estimate. This is possible because of the discriminative power of the local image features used to determine point correspondences (data association). The vision-aided registration method is further developed into a 6D SLAM approach where the optimisation constraint is based on distances of paired local visual features. Finally, the methods introduced in the second part are combined with a novel adaptive normal distribution transform (NDT) representation of coloured 3D point clouds into a robotic difference detection system.

Keywords: mobile robotics, registration, localisation, SLAM, mapping, omnidirectional vision, 3D vision, appearance based

Acknowledgements

I had the great luck of having two excellent supervisors. Firstly I thank Dr. Tom Duckett for putting faith in me and giving the opportunity to do my Ph.D. to start with. I'm truly grateful for his enormous capabilities in brainstorming and for sharing his main foundation in how research should be done. Secondly, I thank Dr. Achim Lilienthal for being my supervisor for the latter half of my studies, many thanks for all suggestions, great ideas and of course all the fun during this time.

During my second year, I visited Prof. Dr. Wolfram Burgard's lab (AIS) in Freiburg, Germany, for four months. It is an honour to have been working with him and his group and I thank him for giving me this opportunity. I thank Dr. Rudolph Triebel, whom I worked the most together with during my stay in Freiburg and also occasionally afterwards. I also thank Dr. Maren Benewitz, Dr. Giorgio Grisetti, Dr. Dirk Hähnel, Patrick Pfaff and Dr. Cyrill Stachniss for all the scientific discussions and of course for making my stay there that enjoyable.

I had the chance of working together with very nice people from University of Tübingen, Germany. Many thanks to Dr. André Treptow and Dr. Hashem Tamimi, working in Prof. Dr. Andreas Zell's group (WSI-CS) at that time. Further, I thank Dr. Peter Biber and Sven Fleck from the group (WSI-GRIS) headed by Prof. Dr. Wolfgang Straßer.

Special thanks goes to Prof. David Lowe from the University of British Columbia and Dr. Udo Frese from University of Bremen for providing valuable resources.

A big thanks goes to our excellent research engineers Bo-Lennart Silfverdal and Per Sporrong for the quick response time, their construction and design skills. Also for their patience regarding all fixes on Tjorven due to its poor mechanical design.

Thanks Mathias Broxvall for your unselfish work on maintaining the common server resources here at AASS.

Thanks all people in the Learning Systems Lab, especially the "Tjorven Group"; Martin Magnusson, Martin Persson and Christoffer Wahlgren for taking care of our "precious".

Thanks Martin Persson for prof-reading this thesis and Robert Lundh for giving valuable comments and for kicking my but in our non-regular badminton games.

Thanks all staff at AASS, especially Barbro Alvin for fixing “everything”. Thanks all Ph.D. students past and present for making AASS such a nice place.

Finally the biggest thanks goes to my family. To Malin, for being there for me, for your love and friendship. To Elina, my daily sunshine, for giving me the correct perspective of anything, including this work.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 A World with Robots

Considering the large amount of fiction literature, TV-series and movies containing mobile robots, one could argue that there is no need for an introduction chapter to robotics. Which other research topic has more or less an own index (sci-fi) in the library? This is a very nice property since many people find this topic fascinating. The vast amount of fiction has, however, also raised a quite large misconception of what is the state of the art in mobile robotics. Compared to the robots in science fiction literature, current research is lagging far behind. My aunt, for example, used to say: "I would like to have a personal service robot to help me in the kitchen, something like a C-3PO would be nice since he is also very polite." (C-3PO - gold coloured humanoid from the Star Wars movies, see Fig. 1.1). Obviously there is no C-3PO available on the market. But how far away from a C-3PO are we? What is the current state of the art in mobile robotics?

Most of the work carried out in mobile robotics today is still about finding solutions to the *fundamental problems*. Some of these fundamental problems address the core building blocks required to give a mobile robot the skills to navigate in its environment (go from A to B). The word navigation originates from Latin: *navis*-'ship' and *agere* -'to move' or 'to direct'. The processes involved to move a ship between A and B are indeed similar to the processes required to move a robot. To navigate a ship, the first step would be to get a nautical chart, a map, covering the region of interest. Based on this map, the second step would be to plan the voyage based on the current location and the goal, i.e. to determine a path. The path would typically be represented by a set of way-points or sub-goals. Path following can then be accomplished by moving along the way-points towards the current goal. To determine the heading and distance to way-points, it is beneficial to know the position during the voyage (this problem is called localisation in mobile robotics). Finally, we cannot solely follow the planned path without watching out for other ships or obsta-

Figure 1.1: Robots, both fiction and non-fiction, used as examples in the discussion. Top: C-3PO, the humanoid from Star Wars (fiction). Bottom left: Trilobite - the vacuum cleaner robot from Electrolux. Bottom right: AutoMower - the lawnmower robot by Huskvarna.

cles (obstacle avoidance) during the trip, and consequently it might even be necessary to re-plan parts of the path. In the ship navigation example the maps were available, which is typically true in the case of nautical charts. However, for mobile robots, it is very rare that up-to-date maps exist with the required accuracy. Hence, one large research area in mobile robotics is how to create suitable maps.

An overview of the fundamental building blocks for mobile robot navigation can be seen in Fig. 1.2. How the problems corresponding to these building blocks can be addressed is to a large extent dependent on the environment and the available sensor modalities.

Basically what is achievable nowadays is for a mobile robot to move from position A to B in a planar structured terrain and to perform simple manipulation of objects. This has been done for quite some time in controlled environments, such as car factories, where automatic guided vehicles (AGVs) carry around parts to various assembly stations by following magnetic stripes buried in the floor. More recently it became possible to do the same task without the need to modify the environment. One of the first successful mobile robot application of this kind was a tour guide, back in 1997, that showed visitors around a museum [20]. No modifications (e.g. adding magnetic stripes in the floor) were required in the museum to aid the robot's navigation. Localisation, path planning and obstacle avoidance were done on the robots on-board computer.

Given the current state of the art, a C-3PO seems obviously quite far away, so in what kind of situations is it desirable to use our current mobile robots? In the mobile robotics community, as well as in many other fields, the three D's (dumb, dull and dangerous) are often mentioned. In addition to these three D's, one might want to add other factors such as complexity and cost (two C's). In addition to being suitable for robots, the task must of course also be feasible to perform (complexity) and secondly, at a reasonable cost. Even though, the task of making the bed, for example, might be considered dull, the complexity of the skills involved are far beyond what any robot can perform today. This example highlights the difficulties for consumer robotics. There are enough applications that fit the three D's and are suitable for mobile robots, but most are too difficult for the current state-of-the-art systems, and at the same time the cost of a commercially successful robot needs to be modest, hence the two C's are the main limiting factors. Basically only two consumer applications have been successful so far; vacuum cleaning and lawn mowing, see Fig. 1.1. The reason why these applications are successful is first that neither requires any higher level of navigation and second, both devices are specialised for a single task. Typically both vacuum cleaning and lawn mowing can be performed reasonably well based on random walk, meaning that the robot select the next control command based in a random fashion, and purely reactive obstacle avoidance (if the robot observes an obstacle, it simply does a random rotation and continues straight ahead). Since there is no need for the robot to know where it is by sensing the environment, the sensors utilised in these robots can be very limited

(for example, only a bump sensor, which tells if the robot ran into an obstacle) and the same is true for the computational demands. These robots can be produced at a low cost. These two examples also show ways of simplifying the task (vacuum cleaning and lawn mowing) by designing a robot which solves only a specific sub-task. A generic or multi-purpose robot, such as C-3PO, would be able to perform vacuum cleaning and lawn mowing by using a regular vacuum cleaner and lawnmower. However, trying to create a C-3PO like robot is indeed much more complex. The same approach of simplifying the problem by designing robots for one specific task could, for example, be applied to simplify the problem of making the bed. However, although simplified, it is still extremely complex. Making the bed doesn't require any special device (compared to vacuum cleaning which requires a vacuum cleaner). The step of adding functions to an already existing device (vacuum cleaner becomes a vacuum cleaner robot) seems more straight forward than to introduce a completely new device (bed making robot). Probably there are some additional specific robotic devices that will be used in everyday households in the near future. One could argue though that it is unlikely to be a robot specialised in making the bed.

If we instead look at the commercial areas: production assembly, agriculture, mining, warehouses, stores, harbours, etc., the cost factor has a much lower impact than on the consumer market and also devices already exist (trucks, loaders, forklifts, tractors, harvesters, etc.). There are a large number of applications where going from A to B is an essential step in the production process (i.e. transporting goods, containers, crops, etc. from A to B). What is missing, except for additional sensors and computers, is to accomplish the task autonomously.

Another area that mobile robots can be used in is to collect sensory data with the aim to learn about the environment. Sensor networks are addressed in a research area where the basic idea is to place (many) stationary sensors in an environment and by monitoring the data from all these sensors (humidity, temperature, gas concentration, wind, for example) to extract various properties of the environment such as gas concentration maps. The size of the environment can vary from small scale, for example, gas detection in a building (i.e. fire-alarms) up to a global scale such as the weather. By instead mounting sensors onto mobile robots, larger areas can be covered or fewer sensors may be required. The main motivation for using robots in this context is cost. Each sensor can be very expensive and to cover a reasonable area with a sensor network many sensors are required. Another application within the same context is to use a robot to distribute sensors in a hazardous environment. In the near future, these are the areas (production and environmental monitoring) where mobile robots either will or have just started to be utilised.

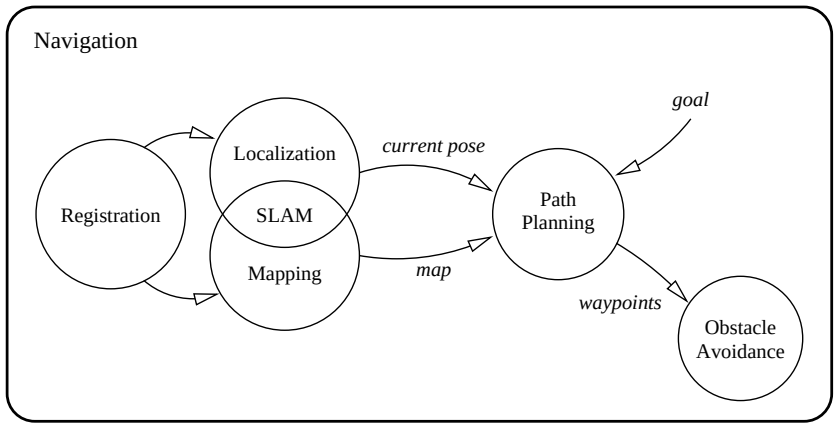

Figure 1.2: Fundamental building blocks for mobile robot navigation. The focus of this thesis is on the building blocks Registration, Localisation, Mapping and SLAM considering different sensor modalities, in particular vision, and non-trivial environments.

1.2 Fundamental Problems

The focus of this thesis is on navigation in a non-trivial environment (essentially a non-planar non-artificial 3D world). In particular, the following fundamental problems or 'building blocks' are addressed (see Fig. 1.2):

- Registration
- Localisation
- Simultaneous Localisation and Mapping (SLAM)
- Mapping

Localisation, is the problem of estimating the current position of the mobile robot. The position estimate can either be relative to a global coordinate frame as in GPS, which will be discussed later, or relative to a given map. Localisation with respect to a map provides an answer to the question "Where am I (given this map)?" [74].

To create a map, called mapping, is another fundamental problem, "What is my map?" [45]. Mapping can be described as to combine a set of sensory readings into a spatial, consistent representation of the environment - a map. Simultaneous Localisation and Mapping (SLAM) [108, 32] is the problem of simultaneously determining the robot's location while constructing the map. SLAM is often referred to as a chicken and egg problem, since, to localise an accurate map and at the same time good estimates of the robot's *pose* (position and orientation), are needed to build the map.

Registration is the problem of determining relative pose estimates between two sensory readings, for example, between two laser range scans. When range data is used, registration is often called scan-matching. Relative poses are used in both localisation and mapping, therefore registration can be seen as an even more fundamental block.

Other fundamental building blocks, not covered in this thesis, are path planning and obstacle avoidance. Path planning corresponds to the question "How do I get there?" [72]. Of course the path planning task typically include a representation of the world (a map) and a position estimate ("I know where I am"), which basically means that localisation and mapping as described above have to be solved. Path planning can also be incorporated within the mapping and localisation process. For example, exploration, to autonomously create a map, requires that the robot both moves to and detects unvisited locations [123]. For a complete autonomous navigation system we need all of the building blocks, see Fig. 1.2.

1.3 Sensors

Previously a few sensors have been mentioned, for example, the 2D laser range scanner. Another common range sensor is the sonar, where both resolution, accuracy and cost are much lower compared to a laser range finder. Generally, time-of-flight (TOF) range sensors work by emitting a signal, then measure the time until the 'echo' bounces back. By knowing the speed of sound for sonar and speed of light for laser, the distance to the reflected surface can be obtained. In addition to the measured time, the phase shift between submitted and received signals are used to improve the resolution in most light based systems. The SwissRange 3000 [1] relies solely on the phase shift of modulated signals. Other non-TOF range sensors works commonly by triangulation, as for example, a stereo camera.

One common sensor that most mobile robots have is odometry. Odometry provides an estimate of the robot pose by estimating the ego-motion, also called dead reckoning. This is most often done by integrating encoder values on the wheels of the robots (most mobile robots nowadays have wheels). The problem is that errors quickly accumulate over time. The benefits are that this kind of sensor is typically accurate over a short distance. Also odometry sensors only give estimates of 2D motion and cannot directly cope with motion in 3D. To address the problem of determining motion in 3D, inertia sensors, gyros and inclinometers can be used. However these sensors, except for the inclinometer (which only measures the pitch and roll angle relative to the gravitation vector) also deteriorate over time.

The Global Positioning System (GPS) gives a position estimate in a global coordinate frame and would ideally solve one of the fundamental blocks directly - localisation. However, GPS has several limitations. First it does not work in many cases, for example, indoor, underground and underwater. Sec-

ond, the accuracy varies heavily depending on the environment, for example in cities (where buildings are blocking and reflecting satellite signals) and other non-open areas. An example of position accuracy, taken from the specification for the GPS receiver located on one of our robots (Novatel ProPak G2), is 1.8 CEP. CEP (Circular Error Probable) measures a horizontal radius from the ground truth position of where half of the position measurements from the GPS are expected to be inside (and half are outside). 1.8 CEP gives approximately a 95% confidence value of 4.5 meters (95% of the position estimates are within 4.5 meters). The vertical accuracy of a GPS is less then the horizontal. Due to these limitations no robot (except for a flying robot or one that operates on the surface of the sea) can solely rely on GPS to localise. Please note that differential GPS (DGPS) and Real Time Kinematic GPS (RTK-GPS), although typcially providing higher accuracy, have the same problems with weak and reflecting signals as the standard GPS.

Another important sensor is the vision sensor (camera). Cameras have a large potential due to the rich amount of data an image contains. The resolution, accuracy and frame rate of this sensor increases dramatically on a yearly basis while the cost decreases. The cost of a camera is much lower than for a laser range scanner, for example. Typically laser range scanners have a large field of view (FOV) compared to a standard camera. To extend the FOV of a camera, various mirrors and lenses can be used. For example, an omnidirectional lens gives a 360 degrees panoramic view, which, due to the richness of the information, is found to be suitable for localisation tasks. Also, as the eyes are the primary sensor for humans and many other animals, there already exist solutions to the fundamental building blocks, although these solutions are coded in 'wetware' - brain and spinal tissue. This motivates why robots nowadays and in future should rely more on cameras.

1.4 Proposed Approaches

A lot of research has been done, especially in indoor environments, using a 2D laser range scanner on a mobile robot. Much of the current research in mobile robotics is now focusing on developing the fundamental building blocks to fit different sensors such as cameras, 3D laser range scanners, etc. and to move from indoor to outdoor environments. This thesis addresses the fundamental problems of registration, localisation and SLAM by using vision sensors as a foundation of the various proposed methods.

Two groups of different methods are proposed where the difference lies in which sensors are utilised: first, a setup where only camera images are used together with odometry, and second, a combination of vision and a 3D laser range scanner. The latter setup does not require any pose sensor as odometry.

The key part of the work, which is common to all the proposed approaches, is the utilisation of cameras and the application of local visual features. In essence, local features means that the whole image is used at once, but instead

only the interesting parts of an image are looked at. To only look at smaller parts of an image gives several advantages, especially when comparing two images to determine if they were taken at a similar position. For example, if the scene has partly changed, there are still interesting regions in the unchanged area which can be detected. Also, minor changes of the viewpoint (the location of the robot) can be tolerated in that the local features move relative to each other, but their appearance remains similar. Local visual features and their properties will be discussed further in Chapter 3.

An overview of the methods proposed in this thesis can be found in Chapter 2.

1.5 Contributions

This work addresses some of the fundamental problems in mobile robotics by using vision sensors in two completely different set-ups. The proposed approaches can be seen to be at two different ends of the axis representing research in mobile robotics, where the axis represents both complexity in terms of computational requirements and cost in terms of price of the used sensors.

Two new approaches regarding registration are proposed. One is solely based on a measure of how similar two omni-directional images appear using local features together with the robots odometry. The key part and the innovation in this approach is that position estimates of each local feature can be avoided, which can be computationally expensive. The other registration method uses a standard CCD camera and a 3D laser scanner, where the accurate initial pose estimates required in pure 3D laser scanner based methods can be avoided.

Based on these two registration blocks, localisation and SLAM / Mapping methods are proposed. For each type of sensor setup, a SLAM and localisation approach is proposed based on visual appearance. By exploiting the registration method that does not requires any initial position estimate a difference detection systems is also developed, both as an interesting robot sequirity application but also as an evaluation of the proposed methods.

To be able to actively fuse the high resolution images that standard modern cameras can provide with the comparably low resolution of state-of-the-art 3D range scanner sensors, required as a preprocessing step by registration and therefore also the localisation and SLAM methods, yet another building block is presented in this thesis named interpolation. Interpolation is how to actively fuse the depth values obtained from 3D laser scanner and the camera image. The interpolation can also be seen as a separate application, since, by combining these two sensor modalities it is possible to obtain range data at a higher resolution, however, interpolation in this work is used to obtain a depth estimates of local visual features extracted from camera images.

1.6 Publications

Some parts of this thesis work have been presented in a number of journal articles, international conferences, symposia and workshops. The following is a list of publications that have been accomplished during the Ph.D. studies. Each publication that is used within this monograph is marked with a box specifying which chapter it relates to. The publications are available on-line at http://www.aass.oru.se/~han.

Journal Articles

- Henrik Andreasson, Achim Lilienthal. "6D Scan Registration using Depth-Interpolated Local Image Features". *Robotics and Autonomous Systems*, submitted.
  Main part in Chapter 8 and Chapter 9

- Henrik Andreasson, Tom Duckett and Achim Lilienthal. "A Minimalistic Approach to Appearance based Visual SLAM". *IEEE Transaction on Robotics - Special Issue on Visual SLAM*, accepted as a regular paper.
  Main part in Chapter 6 and Chapter 4

- Henrik Andreasson, Rudolph Triebel and Achim Lilienthal. "Non-iterative Vision-based Interpolation of 3D Laser Scans". *Autonomous Robots and Agents, Studies in Computational Intelligence*, Springer-Verlag, 2007.
  Chapter 7

- Henrik Andreasson, André Treptow and Tom Duckett. "Self-Localization in Non-Stationary Environments using Omni-directional Vision". *Robotics and Autonomous Systems*, 2007.
  Main part in Chapter 5 and parts of Chapter 4

- Hashem Tamimi, Henrik Andreasson, André Treptow, Tom Duckett and Andreas Zell. "Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT". *Robotics and Autonomous Systems*, 2006.

Conference Proceedings

- Henrik Andreasson, Martin Magnusson and Achim Lilienthal. "Has Something Changed Here? Autonomous Difference Detection for Security Patrol Robots". *Proc. IEEE/RSJ International Conference on Intelligent Robots and System (IROS07)*, San Diego, CA, USA, 2007.
  Main part in Chapter 10 and Chapter 9

- Henrik Andreasson and Achim Lilienthal. "Vision Aided 3D Laser Based Registration". *Proc. European Conference on Mobile Robots (ECMR07)*, Freiburg, Germany, 2007.
  Main part in Chapter 8 and parts of Chapter 7

- Henrik Andreasson, Tom Duckett and Achim Lilienthal. "Mini-SLAM: Minimalistic Visual SLAM in Large-Scale Environments Based on a New Interpretation of Image Similarity". *Proc. IEEE International Conference on Robotics and Automation (ICRA07)*, Rome, Italy, 2007.
  Main part in Chapter 6 and parts of Chapter 4

- Henrik Andreasson, Rudolph Triebel and Achim Lilienthal. "Vision-based Interpolation of 3D Laser Scans". *Proc. International Conference on Autonomous Robots and Agents (ICARA06)*, Palmerston North, New Zealand, 2006.
  Chapter 7

- Hashem Tamimi, Henrik Andreasson, André Treptow, Tom Duckett and Andreas Zell. "Localization of Mobile Robots with Omnidirectional Vision using Particle Filter and Iterative SIFT". *Proc. European Conference on Mobile Robots (ECMR05)*, Ancona, Italy, 2005.

- Henrik Andreasson, Rudolph Triebel and Wolfram Burgard. "Improving Plane Extraction from 3D Data by Fusing Laser Data and Vision". *Proc. IEEE/RSJ International Conference on Intelligent Robots and System (IROS05)*, Edmonton, Alberta, Canada, 2005.

- Henrik Andreasson, André Treptow and Tom Duckett. "Localization for Mobile Robots using Panoramic Vision, Local Features and Particle Filter". *Proc. IEEE International Conference on Robotics and Automation (ICRA05)*, Barcelona, Spain, 2005.
  Main part in Chapter 5 and parts of Chapter 4

- Sven Fleck, Florian Busch, Peter Biber, Henrik Andreasson and Wolfgang Strasser. "Omnidirectional 3D Modeling on a Mobile Robot using Graph Cuts". *Proc. IEEE International Conference on Robotics and Automation (ICRA05)*, Barcelona, Spain, 2005.

- Peter Biber, Henrik Andreasson, Tom Duckett and Andreas Schilling. "3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera". *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS04)*, Sendai, Japan, 2004.

- Henrik Andreasson and Tom Duckett. "Object Recognition by a Mobile Robot using Omni-directional Vision". *Proc. Eighth Scandinavian Conference on Artificial Intelligence (SCAI03)*, Bergen, Norway, 2003.

Workshop and Symposium Papers

- Henrik Andreasson and Tom Duckett. "Topological Localization for Mobile Robots using Omni-directional Vision and Local Features". *Proc. The 5th Symposium on Intelligent Autonomous Vehicles (IAV04)*, Lisbon, Portugal, 2004.
  Chapter 5

- Tom Duckett, Grzegorz Cielniak, Henrik Andreasson, Li Jun, Achim Lilienthal, Peter Biber and Tomás Martínez. "Robotic Security Guard - Autonomous Surveillance and Remote Perception (abstract)". *Proc. IEEE International Workshop on Safety, Security, and Rescue Robotics*, Bonn, Germany, 2004.

- Tom Duckett, Grzegorz Cielniak, Henrik Andreasson, Li Jun, Achim Lilienthal and Peter Biber. "An Electronic Watchman for Safety, Security and Rescue Operations (abstract)". *Proc. SIMSafe 2004, Improving Public Safety through Modelling and Simulation*, Karlskoga, Sweden, 2004.

1.7 Outline of the Thesis

The thesis is divided into three parts. The first part covers the basic algorithms and methods, which are common for the rest of the thesis. **Part II** covers the proposed omni-directional vision and odometry based approaches, and **Part III** contains methods that use the combination of vision and 3D laser range scanner data.

The remaining chapters are as follows:

- **Chapter 2** gives an overview of all the proposed methods presented in this thesis and how they fit together.

- **Chapter 3** contains vision algorithms and methods regarding local features, which are used in all the presented approaches.

- **Chapter 4** covers the similarity based registration approach using omni-directional vision and odometry.

- **Chapter 5** describes the visual appearance based localisation framework.

- **Chapter 6** describes the omni-directional vision and odometry based SLAM approach (Mini-SLAM).

- **Chapter 7** explains the interpolation process, which actively fuses two different sensor modalities (camera and 3D laser range scanner) to estimate depth values in images.

- **Chapter 8** is about registration in 3D using local visual features to determine correspondences and depth estimates from a 3D laser range finder.
- **Chapter 9** describes the proposed SLAM and localisation methods using vision and 3D laser range scanners.
- **Chapter 10** describes a difference detection application, which detects both structural changes in 3D and changes in colour.
- **Chapter 11** concludes the thesis and discusses future work.

Chapter 2
Overview of Proposed Methods

This chapter gives an overview of the methods presented in the following chapters. It also aims to give a brief introduction to the various problems and concepts. Each of the following chapters will focus more on details, therefore related work is only briefly mentioned in this chapter and can be found in each subsequent chapter. As mentioned in the introduction, this thesis addresses fundamental problems in mobile robot navigation using local image features: registration, localisation, mapping and SLAM considering two different configurations of sensors. An overview of how the different methods relate to each other is given in the following sections.

2.1 Sensory Equipment

Two different sensory configurations have been utilised in this thesis, where the common sensor is the vision sensor (camera), see Fig. 2.1. The motivation for considering different sensor setups is to cover a range of vision-based sensor systems available for mobile robot applications from the complex, expensive, heavy and accurate end of the spectrum to inexpensive, lightweight and small solutions.

The first sensor configuration - denoted *Omnivision*, is considered within Part II. It consists only of a single camera, equipped with an omni-directional lens and odometry. Part II describes how this configuration can be used to create large maps and to perform localisation. For many low cost, lightweight and small robot platforms used in applications relying on moving from A to B, this setup would be suitable.

If we instead have an application where we want to accurately measure the spatial distribution of a certain quality of the environment, for example, to determine the temperature distribution, we would need to resort to a more expensive sensor set-up. In Fig. 10.2 (p. 153) a 3D-thermal map is shown, created by utilising a thermal camera to detect the temperature in different areas, which is used in combination with spatial data obtained from a 3D laser

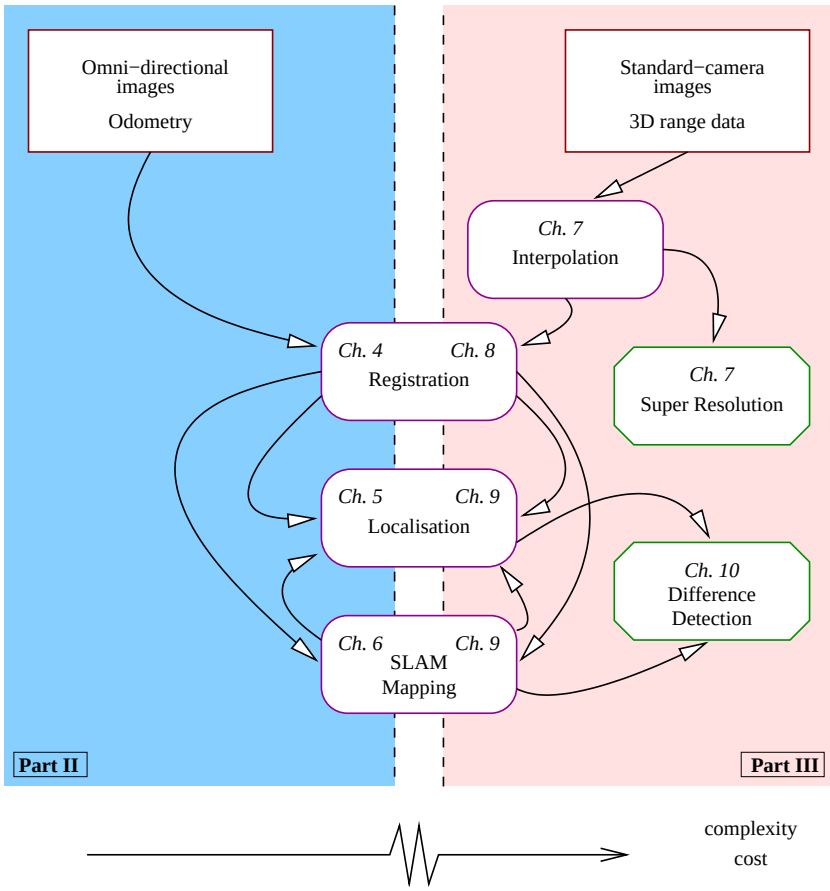

Figure 2.1: Overview of the proposed methods and applications with chapter references. The boxes on top show the type of sensors used for each setup. Boxes with rounded corners are the methods, whereas the boxes with a polygonal shape shows applications. The left area contains the *Omnivision* sensor setup considered in Part II of this thesis. Here the methods only rely on odometry and omni-directional images as input. The right area refers to Part III, where a sensor configuration ($3D - vision$) is considered that delivers standard images and 3D laser range data but no internal pose estimates from odometry are assumed. The left side corresponds to the less expensive sensor configuration and the approaches proposed for this configuration are computationally less complex. The gap between the left and the right area indicates that the approaches developed on the right side are far away from those on the left in terms of cost and complexity.

Figure 2.2: The two mobile robots used for data collection. Left: People-Bot, named PeopleBoy. Right: P3-AT, named Tjorven (from one of the characters of the author Astrid Lindgren). Both robots were used for the experiments with the omni-directional vision sensory system, discussed in Part II, whereas only the robot Tjorven was used for the 3D vision experiments in Part III. Both these robots are manufactured by MobileRobots Inc.

range scanner. Another example is an application where the aim is to detect changes in the environment,see Fig. 10.8 (p. 161). In both examples, a sensor which can obtain 3D range measurements is required, which is used in the second configuration - denoted 3D – *vision* (Part III) together with a standard colour camera.

2.1.1 Omnivision sensor configuration considered in Part II

The two types of sensors considered in Part II are odometry - which gives an internal estimate of the robot's pose by incrementally adding encoder values from the wheels of the robot, and an omni-directional camera - an ordinary camera with a special lens, which gives a 360° field of view (FOV). The odometry values are obtained directly from the on board controller of the robots. The omni-directional lens is manufactured by 0-360.com and is attached to

Figure 2.3: The equipment used in the Omnivision sensor configuration (Part II). Left: odometry, illustrated by an encoder. Middle: the omni-directional lens, produced by 0-360.com. Right: The standard consumer camera (Canon EOS350D) that the omni-directional lens is attached to.

a standard consumer 8 megapixels digital camera (Canon EOS350). Fig. 2.3 shows the sensors used on the mobile robot Tjorven (Fig. 2.2). The mobile robot PeopleBoy (Fig. 2.2), equipped with similar sensors (odometry and an omni-directional camera), was used for the localisation experiments described in Chapter 5. The omni-directional camera is further described in Chapter 4.

All the methods proposed in Part II are, apart from using also the robots odometry, appearance based, meaning that images are match based on their similarity and not by extracting any geometrical properties. Since the proposed methods work without extraction of geometrical properties, a significant benefit is that no calibration of the imaging system is needed.

2.1.2 3D Vision sensor configuration considered in Part III

To obtain 3D range data together with camera images, a 2D laser range scanner is attached to a pan / tilt wrist together with a standard CCD camera, see Fig 2.4. The laser scanner, a SICK LMS-200, has a 180° FOV with a maximum range of 80 meters in good conditions and a range resolution of 10 mm $\pm$ 15 mm. The resolution can be set to either 1, 2 or 4 readings per 1°. In the highest angular resolution, the FOV is reduced to 100°. In addition to the returned range estimates, *remission values* measuring the amount of light reflected back to the sensor can be obtained. The camera is a standard 1 megapixel (1280$\times$1024) CCD camera, ImagingSource DFK 41F02, connected through firewire. The camera has a FOV of 26° using a standard 6 mm lens from Pentax. Both the camera and the laser range scanner are mounted on a pan/tilt wrist, Amtec PW070, which is in turn mounted onto the robot Tjorven, see Fig. 2.2.

Figure 2.4: The different sensors used in the 3D vision sensor configuration, 3D – vision, used in Part III. Left: the 2D laser range scanner, LMS200 produced by SICK GmbH. Middle: the 1 megapixel standard CCD camera (DFK 41F02) by ImagingSource GmbH. Right: the wrist PW070, by Amtec Robotics GmbH, which is used to move the 2D laser to create 3D data and to direct the camera.

An important aspect of combining a 3D laser range scanner and a camera is to determine the geometrical properties of both sensors and their relative position with respect to each other. To obtain these parameters a special calibration routine was developed, which is detailed in Appendix B. An example of the data obtained with the 3D – vision sensor configuration is shown in Fig. 2.10. The data can be described as coloured point clouds where the colour is a possibly multi-dimensional vector, which may contain additional dimensions for temperature or remission values.

2.2 Registration

To enable a mobile robot to perceive the environment *external* sensors, for example laser range finders and cameras, are used. As a robot navigates around, several sensor readings are obtained from different locations. *Registration* addresses the problem of how these measurements are related in terms of position and orientation. Since, as will be shown later, both localisation and SLAM methods rely to some extent on registration, it can be seen as a fundamental problem.

Registration, also called *scan-matching* when range sensors are used, is sometimes further divided into [112]:

- global registration, and
- local registration.

Global registration is related to the mapping or SLAM problem (more specific an approach to mapping / SLAM called *graph* or *relation* based), where

Figure 2.5: An example of two panoramic images illustrating the data association (or correspondence) problem: These two panoramic images were taken at a similar position. However, due to changes in the environment, such as moved objects and occluded persons, it can be difficult to detect that these two images relate to the same physical location (c.f. perceptual aliasing in Fig. 2.6).

robot poses are estimated in a global frame and not only relative to each other as in local registration. This will be discussed further on in section 2.4.

In *local registration* the overlap of the sensor data recorded at different poses is used to determine the relative pose. Local registration typically uses a pair of sensory readings [12, 24, 15] meaning that the relative pose is determined from one set of sensory data to the other. One exception is [16] where multiple (local with overlap) readings are registered. Throughout the rest of this thesis, *registration* refers to local registration.

A closely related issue is to determine which sensor readings are overlapping (whether or not local registration can be performed) known as the *data association* or the *correspondence problem*. Data association aims to find which sensor readings correspond to the same physical object [9], see Fig. 2.5. Hence, if multiple objects (locations) have a similar appearance, also known as *perceptual aliasing*, the perception can fail so that data association becomes very difficult. Perceptual aliasing typically occurs in indoor environments and especially in corridors (Fig. 2.6). Other examples can be observed in hotel or hospital rooms. Both registration and data association depend highly on which sensors are used. For example, cameras seem to be better suited to handle the correspondence problem than laser range based approaches [92], which is probably due to the difference in amount of data provided by the sensors in combination with the extensive research performed in the vision community addressing

Figure 2.6: An example of perceptual aliasing: Although these two panoramic images appear similar, they are in fact obtained at completely different locations.

data association. Also, due to the strong connection to the sensors used, some authors avoid addressing the data association problem [45] to instead focus on simulated data with known correspondences.

Registration can be formulated as: given sensory readings R_a and R_b taken at robot pose x_a and x_b respectively, determine the relative pose $x_{a,b}$ between x_a and x_b. The relative pose $x_{a,b}$ is now known, (this is the registration task to estimate), however, what we might have is an estimate of $x_{a,b}$ denoted $\hat{x}_{a,b}$. This estimate can be determined by odometry or an inertial sensor. An initial relative pose estimate will reduce the search space (by an amount depending on the accuracy of the sensor) of probable relative poses, and therefore will reduce the correspondence problem. However, in some cases there are no initial pose estimates available or the pose estimates have deteriorated to the point where they are not usable, which typically occurs when a robot revisits a location. For example, say the robot takes a tour around a building block and returns to a similar pose x_B compared to the starting pose x_A. The estimate of the pose x_B does not depend on the Euclidean distance from the starting pose x_A but on the distance travelled by the robot (around the building block). Therefore the pose estimate $\hat{x}_B$ and the relative pose estimate $\hat{x}_{A,B}$ may contain large errors. In the robotics literature, to revisit a location (and to detect it) is called to *close the loop* and will be discussed further on.

Registration does not necessarily have to be against another sensor reading, but can also be done relative to a map, which leads us into the next section.

2.3 Localisation

Localisation is to determine the pose relative to a map, which depending on the availability of an initial pose estimate can be divided into [41]:

- pose tracking, and
- global localisation.

Pose tracking or *local localisation* is the problem of determining the robot's pose when the initial pose is known. The problem is to continuously update (track) the pose estimate of the robot while it is navigating around. *Global localisation*, also called the *wake up robot problem*, is when the robot initially does not have any pose knowledge at all and, hence, could be located anywhere within the map. In addition, one can also distinguish a third problem very similar to global localisation: *the kidnapped robot problem* [37] where the robot initially knows its position and then is "blindfolded" and moved to another location (kidnapped). A kidnapped robot also has to re-localise from scratch (global localisation) but in addition needs to detect that it has been moved.

Localisation is often further divided into *topological* and *metric* localisation. Basically the difference lies in that topological localisation refers to a specific place, for example: "the coffee room", "my office", "node 11", etc., while metric localisation output refers to the origin of the coordinate system. For example, a topological localisation result may be : "I'm in the lab" whereas the metric localisation returns "14.33, 123.15, 0.32" meaning that the robot location is 14.33 meters "up" and 123.15 meters to the "right" of the map origin with a heading of 0.32 radians. The type of localisation applied is highly dependent on the map used and can be classified into:

- topological maps,
- metric maps,
- appearance based maps, and
- hybrid maps.

Topological maps consist of a set of locations and *relations* between locations, which can be represented by a connected graph. The nodes in the graph correspond to locations and each link corresponds to a relation between two locations. For example, a map where the relations denote whether two nodes are traversable [125] is suitable for path planning. A typical example is a railroad map where stations correspond to nodes and links correspond to tracks (between stations), see also Fig. 2.7 showing a bus route map. Topological maps can be augmented with metric properties, such as a pose of each node or other properties which, for example, can be used to calculate a cost parameter for evaluating paths.

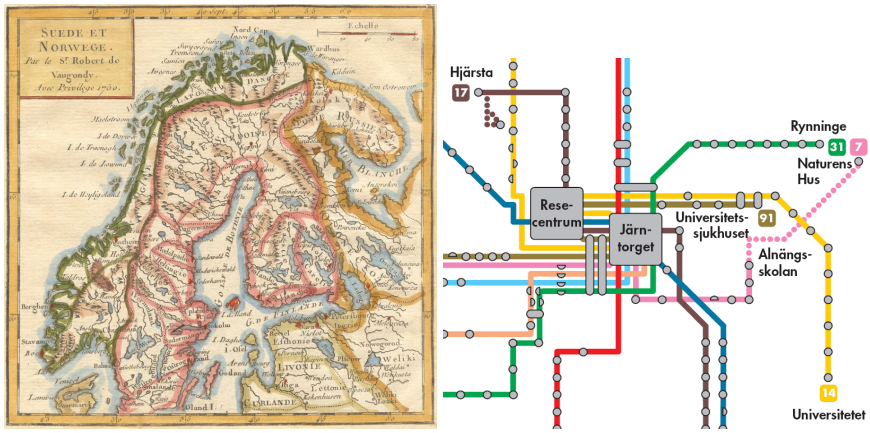

Figure 2.7: Left: Metric map of Sweden and Norway created by Robert de Vaugondy in 1750. Right: Topological map of the local bus routes in Örebro.

Metric maps or *geometric maps* contain geometrical information of the environment, see Fig. 2.7. A typical metric map is a blue print or CAD drawing of a building or a city map. Depending on how the environment is represented, metric maps can further be divided into:

- grid maps, and
- feature maps.

Grid maps [89] are a discrete representation created by dividing the world into (small) cells. Each cell then stores belief about certain properties of the environment. The *occupancy grid* contains the likelihood of the cells being occupied (non-traversable), or empty (traversable). 2D occupancy grids are often created using range sensors such as sonars or lasers, but also stereo imaging has been used as in [61]. Occupancy grid maps are most often used to represent environments in 2D but have also been extended to 3D [114]. Another property represented by the grid cells can be gas concentration [76] or semantic information [86], for example.

Feature maps contain a set of features (or landmarks) that represent the world. Features can either be *natural* (already existing), or *artificial* (added to the environment specifically for the purpose of simplifying the localisation), which means that the environment has to be modified. Artificial landmarks can either be *active* (also called *beacons*), which actively send out signals, or *passive* [73, 29], which do not send out any signals, such as bar-codes or reflective markers. Natural landmarks typically consists of different geometrical properties extracted from the environment, for example, walls (corresponding to lines) and their intersections (corners) are commonly used for sonars and laser

scanners [24]. Cameras often use vertical edges or local feature points (which can be represented as a 3D point [102], or a bearing [62]).

In *appearance maps* [69] the focus is not on extracting geometrical properties of the sensor data (compared with the feature based representation above), but to find a representation that is suitable for matching based on how similar the sensor data (locations) are. An appearance-based map commonly contains metrical [69] or topological [96, 118] information, where typically the topological information is extracted using the appearance-based measures. Cameras and especially panoramic cameras, are often used in appearance-based localisation approaches, due to the richness of the obtained information, however both sonar [30] and laser [18] data have also been exploited.

Finally, a *hybrid map* [21] consists of a combination of other maps, most often a combination of topological and metric maps. Different types of maps have different properties and are therefore suitable for different tasks and may have complementary strengths. By combining different maps strengths can further be exploited. For example, if we have a topological map and a occupancy grid (metric map), the topological map is more suitable for path planning, whereas the occupancy grid can instead be used for (metric) localisation.

2.3.1 Synergies in Maps

By adding metric information to each node in a topological map, i.e. '14.33, 123.15', 0.32 refers to the lab, another example of synergies occurs. Topological localisation can be performed directly from the metric localisation and metric localisation can in a similar way be accomplished, if (and only if) there exists a metric position for each place in the topological map. It is also possible to obtain a higher accuracy in metric localisation using a topological map (with metric information) than the resolution of the nodes in the topological map. An example of metric localisation can be formulated as: imagine the robot is located in the lab and the coffee room is directly connected, and the robot now starts to move towards the coffee room. When the coffee room and the lab are visible at the same time, both are indicated as possible locations. A highly naive approach could then be to draw a line between the metric position of the node "lab" and the metric position of the node "coffee room" and to assume that the robot is located on the middle of this line. Even though this indeed is a naive approach, the metric localisation results are likely to be improved then solely using the metric positions of each place. This basically means that by adding the metrical position of each node in a topological map, metrical localisation can be achieved at a higher resolution than the number of nodes in the map.

The maps used in this thesis are represented using topological, metric and appearance information. Basically the maps are topological, where each node consists of a metric pose, a set of visual features from an image and relations (links) to other nodes. The relations are created from incremental pose esti-

mates (as odometry) and from pairs of nodes with a high appearance similarity, where appearance similarity is measured by matching the visual features.

2.3.2 Global Localisation

The first step in addressing the global localisation problem (either topological or metric) is to be able to evaluate how well a specific sensor reading fits a specific location in the map. This evaluation or similarity measure can directly be used as a global localisation approach by comparing all possible poses in the map with the current sensory reading and selecting the pose which has the best fit. To only take the highest similarity measure has an evident problem: what if two locations have similar appearance? This was described as *perceptual aliasing* in the previous section. Another highly relevant problem in localisation, as-well as in registration, is the *data association* or *correspondence problem*, that is to determine whether or not the current location is in fact the same location within the map, even in the case of occluding persons, robots and other changes. Data association is especially important in localisation since the map was obtained in an earlier stage and is subject to various changes to the environment. Typically humans (and other robots) are not merely "dynamic obstacles" that may occlude the robot's sensors, they also make changes to the world. For example, they may leave temporary objects such as packages, or move the furniture. In addition to these sudden changes, there may be gradual changes such as plants growing, coloured paint fading, etc. Outdoor environments typically have much higher dynamics where the environmental changes are substantial over different seasons and may change very abruptly, for example, during snow fall. The global appearance-based localisation approach of using the highest similarity measured was used in Chapter 5 to compare different methods used to calculate similarities between two locations.

To address the problem of *perceptual aliasing*, the problem of localisation can be considered over a time period with robot movements and multiple sensory readings, and not only a single instance. An example of perceptual aliasing can be seen in Fig. 2.6, where two images taken at two different locations are shown. In this case, due to the high similarity between the images, it would be difficult to infer different locations. By using multiple hypothesis, (that the robot can be at either location) and reevaluating the hypothesis after the robot has moved (updating the new location estimate for each hypothesis using, for example, odometry) the number of likely hypothesis will, unless the environment continues being symmetrical, decrease. For example, in Fig. 2.8, the robot has travelled a distance of 5 meter (forwards) compared to the pose shown in Fig. 2.6, and the locations are now distinguishable due to the low similarity of the images. Hence, given a non symmetrical environment, it is possible to determine a single location hypothesis, see Fig. 2.9, where a particle filter is used to handle multiple hypotheses, where each hypotesis is a cluster of particles. If we instead, for example, have a symmetric corridor two hypotheses will persist,

Figure 2.8: Addressing the perceptual aliasing problem using multiple sensory readings over time: These figures relate to Fig. 2.6 with the difference that the robot in these figures has moved 5 meters (relative to Fig. 2.6) and are now likely to be classified as different positions.

since there is no possible way to separate or distinguish the two ends of the corridor. Also *data association* can benefit from using multiple sensory readings, for example, if the robot receives a high similarity measure relative to the map at a certain position, and at the next step the field of view of the camera is blocked by a number of persons. Even though appearance-based localisation fails to give a location, the robot is probably at a similar location as before and can use its odometry (metric information) to update the location hypotheses.

Localisation, using multiple sensory readings over time, has successfully been approached with probabilistic techniques [110]. A common method is *Monte-Carlo localisation* [26], which utilises a *particle filter* [3] to maintain a probability distribution of the robot position, see Fig. 2.9. Another probabilistic approach, is *Markov-localisation* [43], which also has successfully been utilised in mobile robot applications [20]. The main difference is that Markov localisation typically maintain the location hypotheses on a discrete grid space to allow multi-modal distributions whereas Monte-Carlo methods instead use the distribution of the samples directly.

The method proposed in Chapter 5 incorporates odometry together with a Monte-Carlo scheme to accomplish metric localisation against a given map. The map representation is topological with metric pose information and visual features (for appearance-based similarity matching) for each node. The registration between the map and the current sensory reading uses the visual features and returns an estimate of the robot's relative orientation. The general idea is

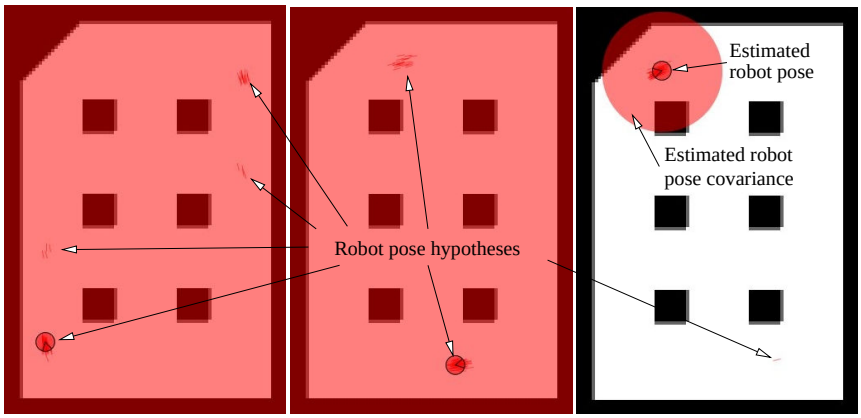

Figure 2.9: Monte-Carlo localisation (MCL) - localisation using multiple sensory readings and robot motion within a particle filter framework. The map (occupancy grid) consist of an almost symmetric environment. The images shows the location hypotheses, drawn as clusters of red lines (dark grey) representing particles, at different times whereas the pink area represents the covariance (light grey). Left: The robot sees a wall straight ahead, removes two unlikely hypotheses and the number of hypothesis is about to be reduced from 4 to 2. Middle: Directly before the diagonal wall is seen, two hypotheses remains. Right: The diagonal wall is seen and the pose is about to uniquely be determined and (only one weak hypothesis still remains in the bottom right corner). Note: the displayed robot position in the first 2 frames are incorrectly guessed by the system.

to extend the strong correlation that can be achieved with similarity based image matching using local features with metric localisation but to also address perceptual aliasing.

The method proposed in Part III relying on the $3D-vision$ sensory configuration do not rely on an internal pose estimation sensor, like odometry, for example. Instead, the assumption is that accurate range measurements are available from a 3D laser range scanner. The metric localisation is possible with this configuration due to the registration method, which relies on the visual features to handle the correspondence problem. As mentioned in the previous section, visual features enable the registration to be handled without initial position estimates. Also, based on the similarity measures it is possible to determine which node is most similar to the sensory readings from the current position. The registration can thereafter be done from the current position to the position determined by the global topological localisation. Note, even though there is no incremental sensor used in Part III, there is a possibility to obtain incremental pose estimates from the registration of subsequently measured data. This basic principle makes it possible to use all methods proposed in Part II to be applicable to the sensor setup $3D-vision$ used in Part III.

2.4 Mapping / SLAM

Registration can be used not only for localisation but also while building a representation of the environment, i.e. a map. The registration method can (in small scales) be applied in an incremental way to create a map. This works in principle as an incremental sensor (for example, an odometry sensor), if the current sensory reading R_t at time t is registered relative to the previous sensor reading R_{t-1}. This method has the same drawback as any other incremental sensor since the error of subsequent registrations accumulates and the pose estimate deteriorates over time. With accurate sensors, such as a laser range finder, the error made in each registration is typically much lower than odometry or many other incremental sensors directly. However, it is simply a matter of the scale of the environment before the errors increase to an extent where the map becomes useless.

What mapping is all about is to create as correct maps as possible and therefore these errors have to be reduced, but how is this achieved? The answer to this question is to *close the loop*. The only way to decrease the uncertainty in the position estimates is to visit a previously visited place [109] and, equally important, to detect that it is the same place. There are two main problems here: the first question to address is how to incorporate the knowledge that the robot has revisited a location, and secondly to enable the robot to be able to recognise previously visited places (data association). These two problems can be distinguished as: *continuous* and *discrete* components of SLAM [28, 109]. The continuous component consists of determining the robot poses (and landmarks in case they are used) given a certain data association and the discrete

component is the data association. Both these problems are highly depending on the representation, or map, used. Therefore the discussion that follows will focus on the representation used in this thesis. As previously described, the map consists of a set of nodes (associated with a set of local visual features with or without depth estimates) and links (relative position estimates). This map representation is a graph and SLAM / Mapping approaches working on this type of structure are called *graph based SLAM* methods, which will be the topic here.

The *discrete* aspect expressed in the question: "How to detect a previously visited place?", is highly related to localisation. Hence the same difficulties (as with localisation) will also occur. If the robot detects that the current node is very similar to a node previous visited, this may be due to the fact that the robot is at a similar position or due to *perceptual aliasing*. On the other hand, given the fact that the robot is truly located at a similar position as a previously visited one, the robot might still fail to detect this (*correspondence problem* or *data association*). The problem of perceptual aliasing is addressed in the Omnivision sensory configuration (Part II) utilising the metric properties of the map. By only evaluating the similarity of nodes in the map in the proximity of the current location, only a subset of all nodes has to be compared to the current pose. Since only a subset of all nodes is used, the approach can handle perceptually similar locations if they are not part of the selected subset. An additional benefit is that the approach becomes faster.

The *continuous* aspect of the SLAM problem instead relates to the question: "How to incorporate knowledge about loop-closing?". Before any loop is closed, the position of each node has to be directly determined from the position of the previous node and the relative pose estimate. What happens when the robot detects a previously visited place for the first time is that the graph will contain more links (relative pose estimates) than nodes (locations). This means that when a loop is detected, there is not a direct method to determine the pose of each node, since there are links which typically do not agree on where the nodes should be located. What we have obtained is an overdetermined equation system, and this is basically what graph-based SLAM methods need to solve. The key issue here (since solving large equation systems is very time consuming) is to exploit various properties of the equation system to lower the computational burden (computational time and memory storage) while retaining consistence and accuracy of the map. See Chapter 9 for more details.

2.5 Interpolation

An important issue when combining cameras and 3D laser scanners is how to actively fuse the depth values obtained from the 3D laser scanner with the camera image, that is to utilise the colour / intensity pixel data from the image to select how the laser range data should be interpolated. In this thesis interpolation is the problem of determining a depth estimate for any given pixel (or sub-pixel)

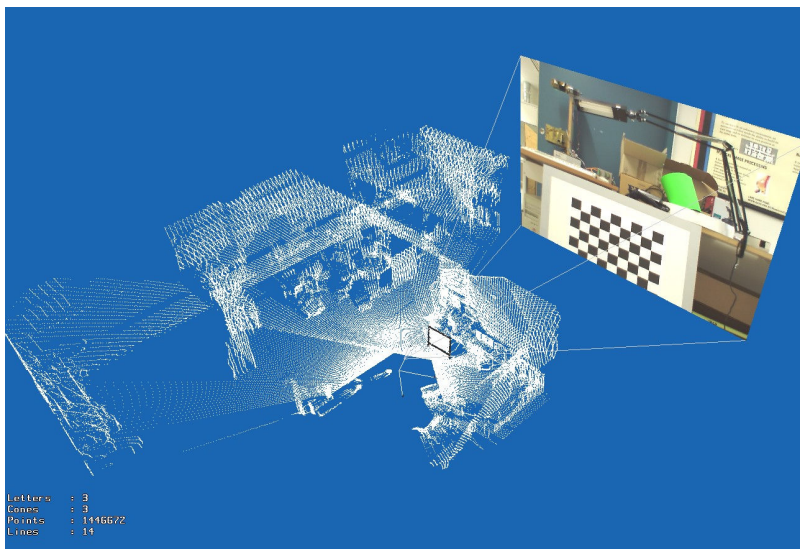

Figure 2.10: Difference in resolution comparing 3D laser range data and image data from a CCD camera from the 3D – *vision* sensory setup, visualised in 3D. The image data are projected from the image plane to a plane at a distance of 10 meters from the robot centre (located roughly in the middle of the image) to illustrate the difference in spatial resolution.

in the image. All the methods in Part III rely on the interpolation method, see Fig. 2.1, which make the interpolation method highly important. For example, the registration method utilises extracted local visual features from the image with the estimated depth obtained from the interpolation. Note that the resolution of 3D laser scanner data is typically much lower than the resolution of a camera image, see Fig. 2.10.

If vision-based interpolation is applied to determine a depth estimate for all pixels (or even at the sub-pixel level) we obtain Super Resolution [27].

Notice that interpolation is not used in Part II where 3D range data are not available.

2.6 Non-navigation Methods

The overview figure 2.1 shows two methods, which do not classify as navigation methods, namely "Difference detection" and "Super Resolution". Super Resolution is mentioned in the previous section.

2.6.1 Difference Detection

The difference detection method developed in Chapter 10 uses all methods proposed in Part III and is a vital part in security applications. A typical security scenario is to detect differences between the current environment and a previous state defined as "normal".

This method can be seen as a "find five error" puzzle where the task is to determine 5 differences in two images. The task in the proposed difference detection method is somewhat similar but with the exception that the method runs in real 3D environments. To detect differences in 3D is obviously a more challenging problem for a person compared to looking at two images, mostly because there is physically no possibility to alternate between the unchanged and the changed environment. Hence, there is no simple way of directly comparing the two environments and unless the human is equiped with additional sensors, the unchanged environment only exists in the person's own memory.

The method combines both range data from a 3D laser range scanner and vision, which makes it possible to detect changes which is not possible to determine in range data alone. For example, if a poster is mounted onto a wall, it will not be visible in the range data but will indicate difference in colour in the camera data.

Part I

Basic Methods

Chapter 3
Image Matching Algorithms

All methods proposed in this thesis have the vision sensor in common. One of the key elements is to utilise local features to address data association and to determine a measure of similarity between two images. This chapter presents the image matching methods covering both the calculation and matching of local features and the computation of a similarity measure from the number of matches.

3.1 Introduction

To be able to compare two images, all methods (except pixel by pixel comparison) start with transforming each image into a specific description - either a single descriptor or a set of descriptors. This description should be compact, unique and at the same time invariant to changes in contrast, illumination and also to small pose changes between the images (that is, if two images were taken at slightly different poses, the description should only undergo minor changes). The description is typically much more compact than the complete image and enables an easy comparison, e.g. Euclidean distance, between two images. The description of an image can be divided into two categories:

- global features, and
- local features.

Global features condense information from the whole image into a single descriptor. Hence, for each image only one descriptor is created for each global feature. Several global features have been proposed, for example colour histogram, see Section 5.3.

Local features are instead calculated from (many) different salient subregions in the image. For each sub-region a descriptor is obtained, see Fig 3.1. To compare two images using local features is therefore typically more time

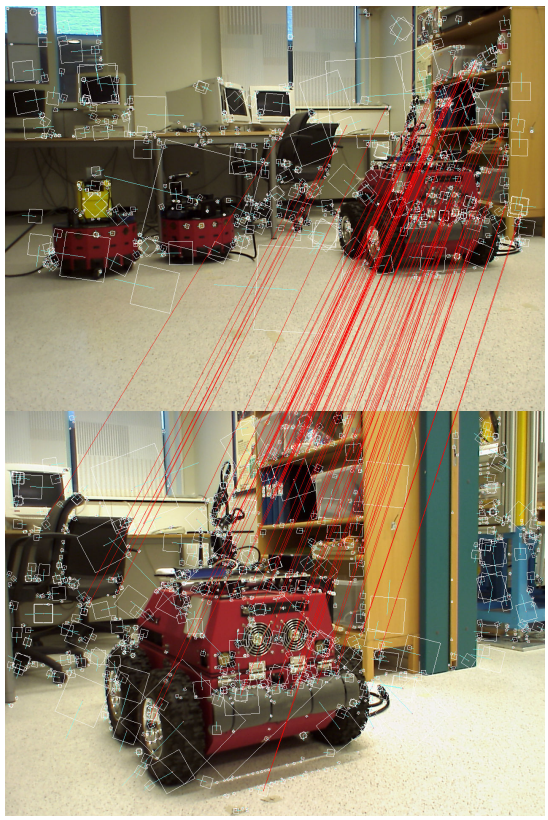

Figure 3.1: Two images with extracted local features (SIFT) and corresponding matches indicated by connecting lines. The orientation of the squares relates to the assigned orientation of the interest points and the size relates to the scale σ. That is, the descriptor window in Fig. 3.4b) used to calculate a descriptor, is illustrated in this figure by squares with different sizes and orientations.

consuming than matching a single global descriptor, since each feature has to be compared to the many features extracted from the other image.

The motivation for using local features in this work is the better invariance and occlusion performance obtained compared to global methods. This difference follows directly from the fact that a global feature is computed from the whole image. For example, if parts of the image are occluded the global feature will be affected. With local features, many feature points with corresponding descriptors are extracted both in the occluded and non-occluded areas in the image. The difference lies in the fact that many of the local features will remain similar for the descriptors for which the salient sub-region is not affected and therefore local features typically shows better robustness towards dynamic environments compared to global features. Also, as seen in Fig. 3.1, where the distance between the two matched images is several meters, local feature based approaches that considers scale are not as sensitive to changes in the camera view point. Local features are also suitable for handling data association, that is, to determine which sensory readings (parts of the camera images) overlap with the same physical region, which are especially utilised in the registration method presented in Chapter 8.

The remaining parts of this chapter cover the local feature extraction, the image matching and the similarity calculations used in the rest of this thesis. A basic work flow of the methods described in this chapter can be seen in Fig. 3.2.

3.2 Local Features

As a local feature the famous SIFT method developed by Lowe [81] is used throughout this thesis with the exception of the omni-vision localisation chapter (Ch. 5) where a modified SIFT (MSIFT) version is used. The MSIFT method is however based upon parts of the standard SIFT method.

3.2.1 Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT), developed by Lowe [81], is a local feature extraction method invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine 3D projection. The extraction of SIFT features relies on the following stages:

1. Creation of scale-space [77]: The scale-space is created by repeatedly smoothing the original image with a Gaussian kernel.

2. Detection of scale-space extrema (*interest point detection*): This is done to find peaks in the scale space of image (pixel) positions $p = [x, y]$, and the scales σ. This is done by searching the (x, y, σ) space for extrema, which are filtered using stability criteria (step 4).

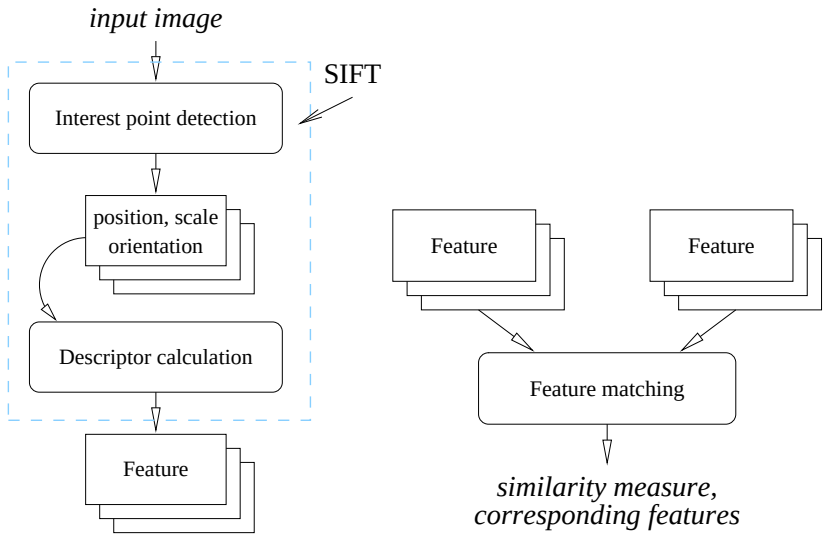

Figure 3.2: Left: Extraction of local features; interest point detection and descriptor calculation. The dashed box represent the different stages that are included within the SIFT method. Right: Feature matching.

3. Accurate interest point localisation: In the previous step, the interest points were detected in a discrete space. This step determines the location of interest points with sub-pixel and sub-scale accuracy.

4. Rejection of weak interest points: All interest points that have low contrast and are lying on an edge are removed.

5. Orientation assignment: To obtain rotational invariance, each interest point is assigned an orientation determined from the image gradients of the surrounding patch. The size of the patch is determined by the selected scale.

6. Calculation of descriptor histogram: Given the position, scale and orientation of each interest point, a patch is selected where magnitude and orientation of gradient is used to create a representation which allows, to some extent, affine and illumination changes.

These steps are discussed in more detail below.

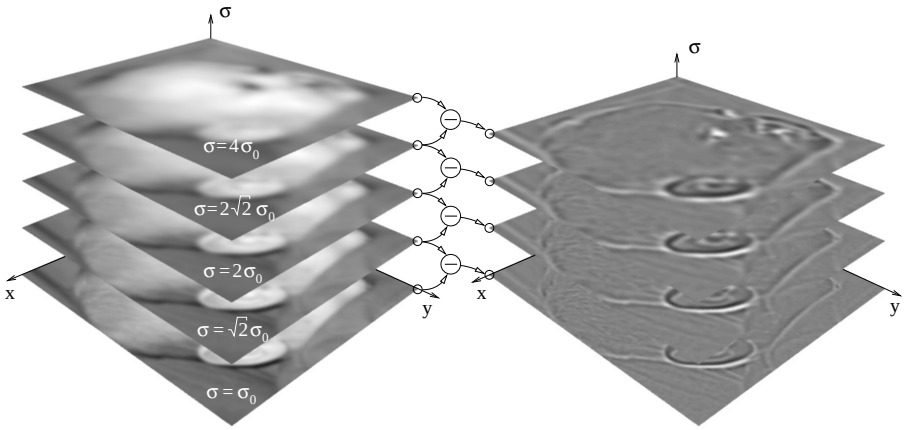

Figure 3.3: Creation of the DoG: The stack of blurred images, the scale-space representation, created by convoluting the Gaussian function G with different scales σ with the original image (to the left). These blurred images are subtracted from each other to create the DoG stack (to the right). This figure shows an octave consisting of 2 scales ($s = 2$), where the scale of the lowest image is σ_0.

Creation of scale-space

Given an original grey-scale image I, the scale space is defined as a function L, which is calculated as:

$$L(x, y, \sigma) = G(x, y, k\sigma) * I(x, y), \tag{3.1}$$

where $*$ is the convolution operation in x and y, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2 + y^2)}{\sigma^2}} \tag{3.2}$$

is a variable-scale Gaussian.

The scale space is represented as a set of smoothed images, see Fig. 3.3 (left). To discretise the resolution in scale σ (the pixel coordinates x, y are already discretised from the image), a constant factor $k = 2^{1/s}$ is used, where s is the number of images until the scale parameter σ is doubled. The scale for the first smoothed images $i = [1..n]$ can be created by convolving with $G(x, y, k^i\sigma)$.

To decrease both the computational cost and storage, the image is resampled by taking every second pixel in each row and column when σ is doubled, that is, every s images. Each block of images with the same size is denoted as an *octave*, hence for each octave s scales are used. Typically 3 scales per octave are used.

Detection of scale-space extrema

One important aspect of the SIFT method is the detection of interest points (x, y, σ) in the scale space. This is done efficiently by directly using the calculated scale space representation to create Difference of Gaussian (DoG) images. The scale space described above is not only used for creating the DoG but is also needed to both calculate the orientation and the descriptor of each feature. The DoG is calculated as:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y), \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \tag{3.3}$$

To determine the extrema locations, each pixel in the scale space is compared with its 26 neighbours in a 3x3x3 block (8 neighbours in the current scale-space image and 9 neighbours in the scale-space image above and below). A candidate interest point location is selected if the DoG value is smaller or larger than all of its neighbours. To be able to determine interest points in n different scales, $n + 3$ blurred images are required in total. Two additional images (above and below) are needed in the extrema location detection and one additional to determine the DoG. Hence, each octave has to contain $s + 3$ images.

Accurate interest point localisation

From the previous step, an interest point is found in a discrete space at a pixel position (x, y) and at a specific scale σ. To obtain a higher resolution the interest point position is determined at sub-pixel and sub-scale accuracy by fitting a 3D quadratic function to the DoG function and determining the interpolated maxima. The quadratic function to fit is the second order Taylor-expansion of the DoG function $D(x, y, \sigma)$:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^w}\mathbf{x}, \tag{3.4}$$

where $\mathbf{x}$ is the discrete interest point position. The location of the extrema $\hat{\mathbf{x}}$ is determined by $\partial D(\mathbf{x})/\partial \mathbf{x} = 0$, giving

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}}. \tag{3.5}$$

The final position of the interest point is determined as $\mathbf{x} + \hat{\mathbf{x}}$.

Rejection of weak interest points

The key objective of extracting interest points is to obtain a set of stable points that can repeatedly be detected in other images, therefore each interest point

is checked to make sure it fulfils this objective. Two criteria are used to reject interest points: weak contrast and interest points with an edge response. As a contrast measure, the interpolated function value at the interest point, $D(\hat{\mathbf{x}})$, is obtained as:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}}. \tag{3.6}$$

An interest point with a value of $|D(\hat{\mathbf{x}})| < 0.03$ is rejected, the threshold value which is also used by Lowe [81].

To determine edge responses, the Hessian matrix $\mathbf{H}$ is used:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}, \tag{3.7}$$

where the derivatives are calculated from neighbouring points. By looking at the eigenvalues of $\mathbf{H}$ the property of the interest point can be determined. For example a corner is found if both eigenvalues are large, whereas an edge is found if only one of the eigenvalues is large. Eigenvalues, however, are expensive to compute and only the ratio r between the larger eigenvalue denoted α and the smaller eigenvalue denoted β is required. To determine the ratio r only the determinant and trace of $\mathbf{H}$ are required. The sum of the eigenvalues can be computed from the trace of $\mathbf{H}$, whereas the product of the eigenvalues is obtained from the determinant of $\mathbf{H}$:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \tag{3.8}$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \tag{3.9}$$

By using the equations above and $\alpha = r\beta$, the following quadratic equation is obtained:

$$\frac{Tr(H)^2}{Det(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}. \tag{3.10}$$

To check the ratio to determine the edge properties of the interest point, only the determinant and trace of H is required. Interest points are discarded if f is lower than 10, the threshold value which is also used in [81].

Orientation assignment

To determine the interest point's orientation, a gradient orientation histogram is computed over the neighbourhood of the interest point. The idea is to determine a consistent orientation for each interest point to obtain image rotation

invariance. Based on the interest point scale σ, the corresponding smoothed image $L(x, y) = L(x, y, \sigma)$ is selected. For each pixel $[x, y]$ the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ is calculated as:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.11)$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}. \quad (3.12)$$

The contribution of each neighbouring pixel is weighted by the gradient magnitude m and a Gaussian window G_o with a value of $\sigma_o = 1.5\sigma$ where σ is the scale of the interest point. Peaks in the histogram correspond to dominant orientations. For all bins peaks that reach at least 80% of the maximum value, additional interest points are created at the same location and scale, but with an orientation according to the respective peak.

The orientation histogram contains 36 bins, i.e. 10 degrees per bin. To increase the orientation resolution, a parabola is fitted to three bins; the bin of the peak and the bins to the left and to the right.

Descriptor Calculation

The descriptor is used as a signature for each interest point. Descriptors are used to determine whether two interest points correspond to the same physical location, i.e. for data association. The ideal descriptor should therefore be highly distinctive as well as invariant to changes in view point and illuminance. Note that the scale, pose and orientation are obtained in the interest point selection operations.

Around each interest point, a window is selected where the size and orientation depend on the scale and orientation of the interest point. Gradient magnitude m and orientation θ, see Fig. 3.4a, are calculated (Eq. 3.11, 3.12) for each pixel using the scale space image with the most similar scale as the interest point.

The magnitude m is weighted with a Gaussian function to reduce changes in the descriptor due to small changes of the position of the sub-window and to reduce the impact of magnitudes far from the centre of the interest point.

The weighted magnitude and orientation values are then accumulated into 16 orientation histograms summarising the contents of the whole sub-window corresponding to the 16 squares in Fig. 3.4b (these histograms constitute the so-called interest point descriptor or feature vector used for matching features). Each histogram has 8 bins, i.e. the orientation information is discretised into 8 possible values. Each bin accumulates the total weighted magnitude information for a particular orientation. To avoid boundary effects when gradients are changing from one histogram area to another, each pixel gradient is stored in

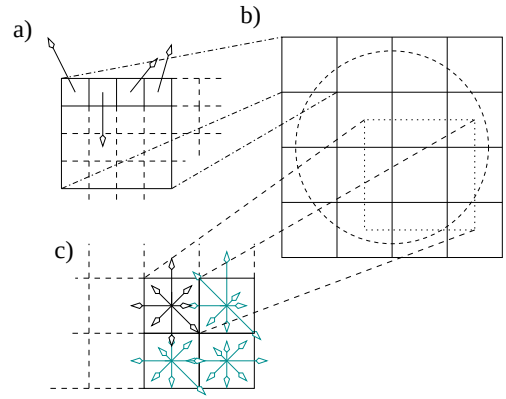

Figure 3.4: Descriptor used in SIFT features. a) Rotation θ and gradient magnitude m for each pixel. The circle in b) represents the Gaussian weighting function. Note that the histograms in c) are created by bi-linear interpolation from surrounding pixels represented as a dashed square in b).

the four closest histograms with bi-linear interpolation using the distance to the centre of each histogram (Fig. 3.4c). The descriptor contains 4×4 histograms with 8 bins, which gives a descriptor histogram vector H of 128 elements.

The descriptor vector is then normalised to a unit length. To further increase robustness towards non-linear illumination changes, each feature vector bin is limited to 0.2 which is followed by another normalisation step.

3.2.2 Modified SIFT (MSIFT)

This section describes the modified SIFT (MSIFT) feature approach used in the localisation approach in Chapter 5.

The SIFT method, described above, generates features which are invariant to image translation, scaling and rotation, and partially invariant to illumination changes and affine or 3D projection. These properties make SIFT very suitable for mobile robots because landmarks can be observed from different angles, distances and illumination [102].

However, for the purpose of self-localisation, we actually do not want full invariance to translation and scale: we would like view matching to be successful only in the vicinity of the location where the original image was recorded in the database. Therefore, a number of changes were introduced, described as follows.

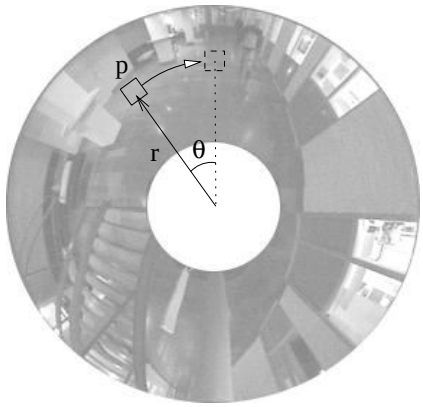

Figure 3.5: Creation of the neighbourhood $\mathcal{N}(\mathcal{F})$ drawn as a dashed square. Before matching, the feature F and the surrounding pixels are rotated by angle θ to a fixed heading (facing forwards relative to the robot). This means that matching of two features in different images will be independent of the heading of those points.

Selection of Interest Points

The selection of interest points in MSIFT is not done in scale space. Instead it is based on the eigenvalues of the Hessian matrix $\mathbf{H}$ (Eq. 3.7). A large smallest eigenvalue β indicates that the intensity changes substantially in both directions, e.g. corners. A non-maxima suppression is performed and the pixels with the highest values of β are then selected as interest points. If two interest points are closer than a distance of 5 pixels the weakest point is removed, see also [105] or the function *cvGoodFeaturesToTrack* in the OpenCV library [19].

The number of features extracted from each images is set to a constant. By sorting the features using β as an indicator, only the strongest features will be used. By using a constant number of features an approximate constant processing time required to match two images is obtained. For example, in the localisation experiments in Chapter 5 the number of features was set to 100.

Descriptor Calculation

To create a rotationally invariant descriptor in the image plane, the sub-window $\mathcal{N}(\mathcal{F})$ is created by rotating the surrounding pixels the same angle θ as the feature F is rotated, see Fig. 3.5, using bi-linear interpolation. By doing this, $\mathcal{N}$ will be independent of the rotation θ. This step is necessary since MSIFT is used directly on an omni-directional image.

The descriptor is based upon the standard SIFT descriptor. Since no scale is selected, the size of the sub-window $\mathcal{N}$ is set as a fixed value, which was 32×32 in the experiments reported here.

In the SIFT algorithm, the histogram H is normalised to have unit length. However, for MSIFT the localisation performance was improved on our data by omitting the normalisation step, which is used partly to reduce illuminance changes but mostly to cancel out effects due to different scales. Since scale invariance has been removed, the normalisation factor is no longer needed for this purpose, and the magnitude of the histograms can provide additional useful information for matching.

3.3 Image Matching Using the Features

The described descriptors constitute the features used for matching images. Consider two images, I_a for frame a, and I_b for frame b. For both images, local features are extracted (using one of the methods described above), which results in two sets of features, F_a and F_b. Each feature $F = [x, y], H$ comprises the pixel position $[x, y]$ and a histogram H containing the SIFT descriptor. The *similarity measure* $S_{a,b}$ is based on the number of features that match between F_a and F_b.

The feature matching algorithm calculates the Euclidean distance between each feature in image I_a and all the features in image I_b. A potential match is found if the smallest distance is smaller than 60% of the second smallest distance. This criterion was found empirically and also used in [49]. It guarantees that interest points match substantially better compared to the other feature pairs, see Fig. 3.1. In addition, no feature is allowed to be matched against more than one other feature. If a feature has more than one candidate match, the match with the lowest Euclidean distance among the candidate matches is selected. Note that the number of matched features will depend on the order that the features are matched, that is, if each feature in I_b is instead matched with all features in I_a the number of matches may differs. This can be avoided if the matching is done in both ways, where a match is only considered valid if the match occurs twice. The feature matching step results in a set of matched feature pairs $P_{a,b}$, with a total number of $M_{a,b}$.

To decrease the matching time more efficient search structures are commonly applied. Due to the high dimensionality of the descriptor vector an approximate search is often used, for example the Best Bin First (BBF) search [11].

3.3.1 Similarity Measure

A high similarity measure gives an indication that we are at a perceptually similar position. Since the number of extracted features varies heavily depending on the image if SIFT is used (not MSIFT), the number of matches is normalised, hence the similarity measure $S_{a,b} \in [0, 1]$ is defined as:

$$S_{a,b} = \frac{M_{a,b}}{\frac{1}{2}(n_{F_a} + n_{F_b})} \tag{3.13}$$

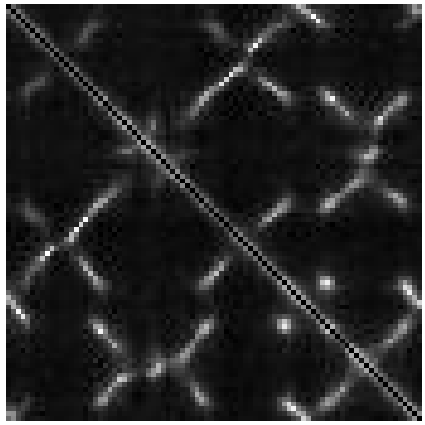

Figure 3.6: Similarity matrix, brighter elements indicate higher similarities. Diagonal elements are black, self-similarities ($S_{a,a}$) are not considered.

where n_{F_a} and n_{F_b} are the number of features in F_a and F_b respectively.

3.3.2 Similarity Matrix

The mutual similarity measures in a set of images $I_{1..N}$ constitutes the *similarity matrix* $\mathbf{M}$. Each element $\mathbf{M}_{a,b}$ corresponds to the similarity measure $S_{a,b}$ for a pair of images a and b, see Fig. 3.6 containing a similarity matrix obtained from the $\mathtt{lab}$ data set in Chapter 6. The similarity of the image I_a with itself ($S_{a,a}$) does not provide any useful information and is therefore not considered. The same type of similarity matrix can be generated for multiple sequences of images, see Fig. 3.7 where the similarity matrix $\mathbf{M}$ is shown for three different image sequences (data sets $\mathtt{lab}$, $\mathtt{studarea}$ and $\mathtt{lab{-}studarea}$ from Chapter 6). In each sequence of images, each successive image is collected at a position close to the position of the previous image (and might contain relative pose estimates from odometry) and each sequence of images was collected using the same camera. In Fig. 3.7, the complete similarity matrix between all three data set is shown with lines drawn to separate each data set, for example, $\mathbf{M}^{\mathrm{i+iii}}$ indicates the similarity matrix between the data set $\mathtt{lab}(\mathrm{i})$ and $\mathtt{lab} - \mathtt{studarea}(\mathrm{iii})$.

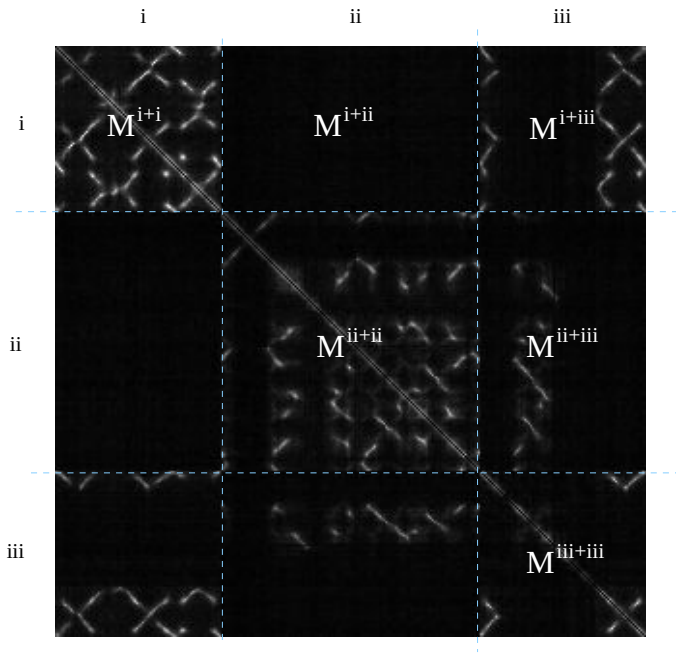

Figure 3.7: Similarity matrix of three different data sets indicated with i, ii and iii. $\mathbf{M}^{i+i}$ is the similarity matrix of the data set i with itself, whereas $\mathbf{M}^{i+ii}$ is the similarity matrix between data set i and ii, etc.

Part II

Omni-directional Vision Sensor Configuration

Chapter 4
Omni-vision Based Registration

This chapter describes the vision-based registration process to determine the relative pose between two robot poses using an omni-directional camera. The process is divided into:

1. Feature extraction (MSIFT or SIFT).
2. Feature matching.
3. Estimation of relative pose and pose uncertainty.

Contrary to most other vision-based registration methods, the presented approach does not rely on tracking and position estimation of each feature. Please note that different feature extraction methods are used for the localisation approach in Chapter 5 and the mapping approach in Chapter 6. In the localisation experiments scale-invariance was not considered (MSIFT) whereas it was used in the mapping experiments (SIFT). The reason is that longer distances were travelled between successive images in the mapping experiments and the omni-directional lens had a larger field of view (FOV), which results in a larger variance in the appearance of the features in different images compared to the localisation setup. SIFT and MSIFT, together with the feature matching method, are described in the previous chapter (Ch. 3).

4.1 Sensors

The registration method described in this chapter relies on an omni-directional vision system combined with odometry as the egomotion sensor. In our setup, the omni-directional image is created by using a concave mirror placed on top of a standard digital camera, see Fig. 2.3. Typically the concave mirror is designed to create a *spherical image*, meaning that emitted light into the camera

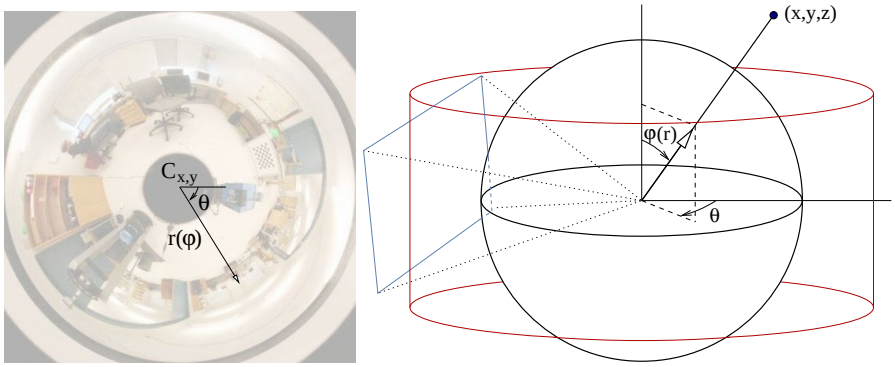

Figure 4.1: Left: The polar representation of an omni-directional image. Right: The spherical representation, where each point in the world (x, y, z) can be projected down to the surface of a sphere. The corresponding pixel coordinate is then obtained using the $r(\varphi)$ function. For example, by projecting points lying on a plane a planar image can be obtained, see also Fig. 4.2.

will be towards a single point (physically located inside the concave mirror approximately 20 cm above the camera lens in our setup). This single point can be seen as the centre of a sphere from where the camera image pixels can be projected onto the sphere's surface. To convert the pixel coordinates into spherical coordinates both the centre $C_{x,y}$ of the omni-directional image and a transformation from the radius r to the angle φ has to be known, see Fig. 4.1. Examples of the raw omni-directional image together with two *unwrapped* or reprojected images can be seen in Fig. 4.2.

In addition to the omni-directional images, odometry is used in the estimation of the relative pose uncertainty. Odometry works by incrementally integrating the rotations of the wheels obtained from encoder readings over time. This is often provided by the robot platform directly, however typically a few calibration parameters have to be set such as the number of encoder ticks per mm, the number of differential encoder ticks to finish a complete turn of the robot and a drift factor to compensate for a size difference between the right and the left wheels. Odometry deteriorates over time, but is accurate over short distances, see for example Fig. 5.6 (p. 69).

4.2 Estimating the Relative Pose and Uncertainty

The relative pose between two frames (omni-directional images) μ is divided into relative position $\mu_{x,y}$ and relative orientation μ_θ. The uncertainty of the relative robot pose covariance matrix C is decomposed into a covariance matrix of the robot position $C_{x,y}$ and a variance of the orientation σ^2_θ, i.e. the

Figure 4.2: Top left: An omni-directional image created by using the lens and camera shown in Fig. 2.3. Top right: An image generated by projecting a part of the image data onto a plane. Bottom: A panoramic image created by using pixel coordinates $x = k\theta$ and $y = k\varphi$, where both θ and φ are defined in Fig. **4.1** and k is a scaling factor.

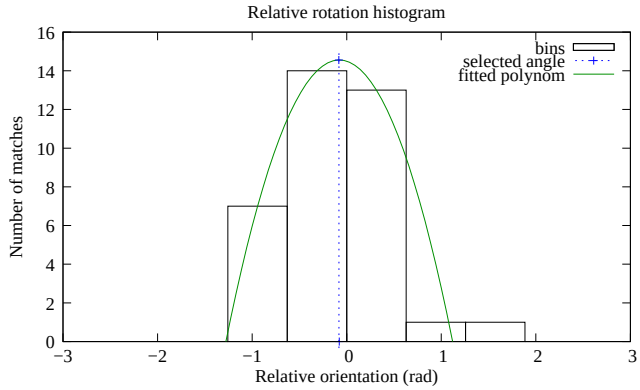

Figure 4.3: Relative orientation histogram created from two omnidirectional images taken 2 meter apart. The dotted line marks the relative orientation estimate μ_θ.

covariance matrix is assumed to have a block structure. This section describes how both the relative pose μ and the uncertainty C of the pose is estimated.

4.2.1 Estimating the Relative Rotation and Uncertainty

The relative rotation between two panoramic images I_a and I_b can be estimated directly from the horizontal displacement of the matched feature pairs $P_{a,b}$ since the width of a panoramic image encloses a complete revolution of the scene. When an omni-directional image is used directly, as in MSIFT, the relative rotation is obtained from the difference in orientation θ as described in Section 3.2.2. Here, the relative rotations θ_p for all matched pairs $p \in P_{a,b}$ are accumulated in a 10 bins histogram and the relative rotation μ_θ is determined as the maximum point of a parabola fitted to the largest bin and its left and right neighbour, see Fig. 4.3.

To evaluate the accuracy of the relative rotation estimate θ_p, we collected panoramic images in an indoor laboratory environment and computed the relative orientation with respect to a reference image I_0. Panoramic images were recorded at a translational distance of 0.5, 1.0 and 2.0 meters to the reference image I_0. The ground truth rotation was obtained by manually measuring the displacement of corresponding pixels in areas along the displacement of the camera. The results in Table 4.1 demonstrate the good accuracy obtained. Even at a displacement of 2 meters the mean error is only 7.15 degrees.

Table 4.1: Errors of relative rotation θ estimate in radians.

transl (m)	error_θ	$\sigma_{\text{error}_\theta}$
0.5	0.100	0.0630
1.0	0.104	0.0500
2.0	0.125	0.0903

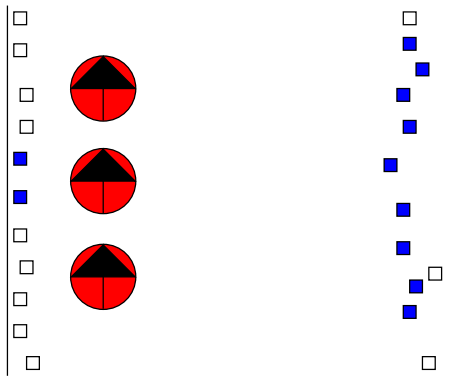

Figure 4.4: The physical distance to the features will influence the number of features that can be detected from different poses of the robot. The filled squares represent features that could be matched in all three robot poses while the unfilled squares represent the features were the correspondence cannot be found from all poses. The left wall in the figure is closer to the robot. Thus, due to the faster change in appearance, the number of features of the left wall, which can be matched over successive images, tends to be less compared to the number of matched features of the right wall.

The relative rotation variance σ_θ^2 is estimated by the sum of squared differences between the estimate of the relative rotation μ_θ and the relative rotation of the matched pairs $P_{a,b}$ as

$$\sigma_\theta^2 = \frac{1}{M_{a,b} - 1} \sum_{p \in P_{a,b}} (\mu_\theta - \theta_p)^2, \tag{4.1}$$

where $M_{a,b}$ is the total number of matched pairs. To increase the robustness towards outliers, a 10% Winsorized mean is applied. For the evaluated data this only had a minor effect on the results compared to using an un-truncated mean.

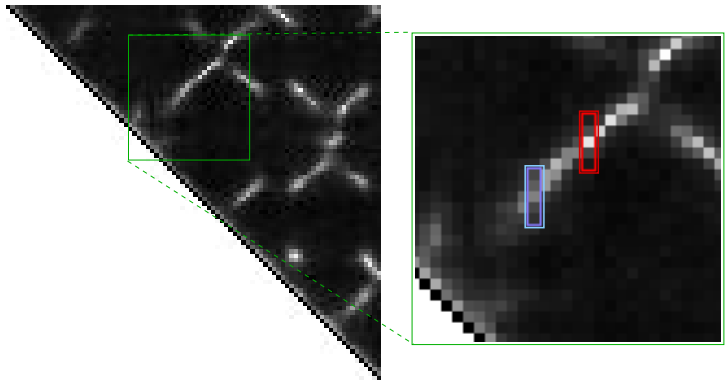

Figure 4.5: Left: Full similarity matrix for the lab data set (Sec. 6.3.3). Brighter entries indicate a higher similarity measure S. Right: Zoomed in image, the left area (enclosed in a the blue frame) corresponds to a sequence of similarity measures that give a larger position covariance than the right sequence (dark red frame).

4.2.2 Estimating the Relative Position and Uncertainty

The registration approach proposed in this chapter does not attempt to determine the position of the detected features. Therefore, the relative position between two frames a and b cannot be determined very accurately. Instead we use only image similarity of the surrounding images to estimate the relative position $[\mu_x, \mu_y]$ as described below. It would be possible to use an estimate based on multiple view geometry, for example, but this would introduce additional complexity that we want to avoid.

Instead, geometric information is obtained from an estimate of the covariance of the relative position between a current frame b and a previously recorded frame a. This covariance estimate is computed using only the similarity measures S (described in Sec. 3.3.1) of frame b with a and the neighbouring frames of a.

The number of matched features between successive frames will vary depending on the physical distance of the extracted features, see Figs. 4.4, 6.2 (p. 86) and 6.3 (p. 87). Consider, for example, a robot located in an empty car park where the physical distance to the features is large and therefore the appearance of the environment does not change quickly if the robot is moved a certain distance. If, on the other hand, the robot is located in a narrow corridor where the physical distance to the extracted features is small, the number of feature matches in successive frames tends to be smaller if the robot was moved the same distance.

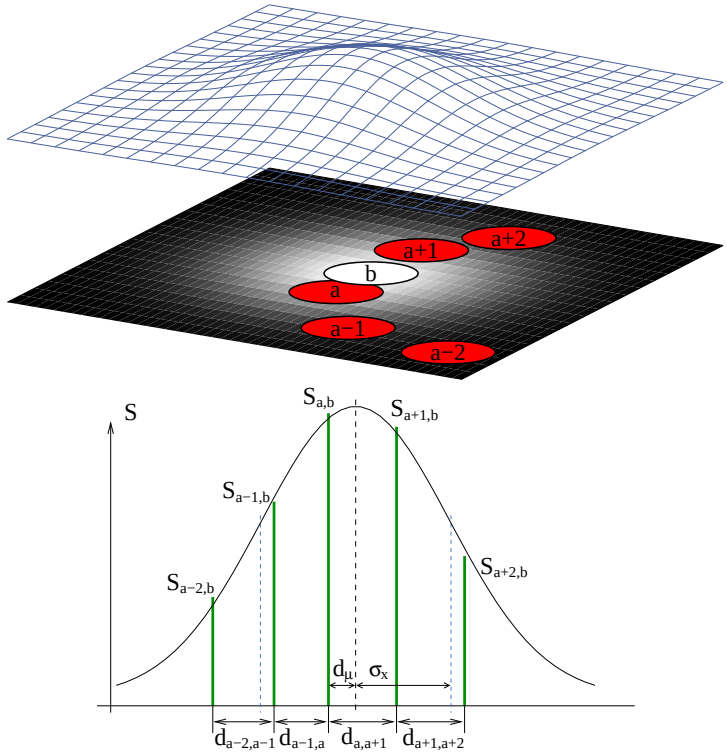

Figure 4.6: Gaussian fitted to the distance travelled d (as obtained from odometry) and the similarity measures between frame b and the frames of the neighbourhood $N(a) = \{a-2, a-1, a, a+1, a+2\}$. From the similarity measures, both a relative pose estimate μ and a covariance estimate C are calculated between node a and node b. The orientation and orientation variance are not visualised in this figure.

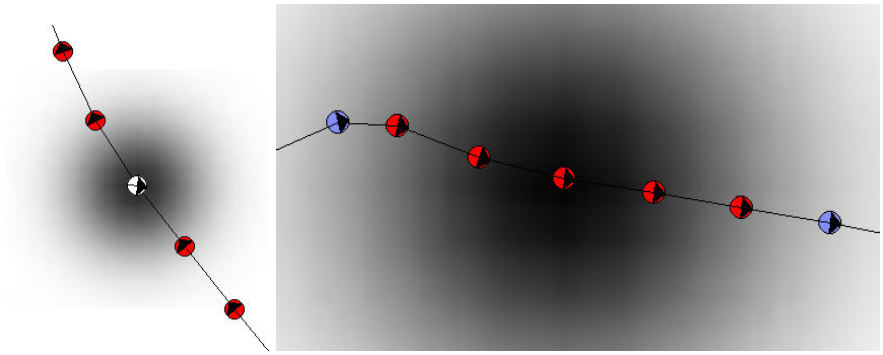

Figure 4.7: Examples of different position covariances created by fitting a 2D Gaussian using the similarity measures of the frames $N(a)$ (drawn in red). The estimated robot pose is drawn with white background in the left figure to illustrate the possibility to estimate an orientation. The orientation variance is not shown in these figures.

The covariance of the robot position estimate

$$C_{x,y} = \begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y \\ \sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \tag{4.2}$$

is computed based on how the similarity measure varies over the set $N(a)$, which contains frame a and its neighbouring frames. The analysed sequence of similarity measures is indicated in the zoomed in visualisation of a similarity matrix shown in Fig. 4.5. In order to avoid issues estimating the covariance orthogonal to the path of the robot if the robot was driven along a straight path, the covariance is simplified by setting $\sigma_x^2 = \sigma_y^2$ and $\sigma_x\sigma_y = 0$. The remaining covariance parameter is estimated by fitting a 1D Gaussian to the similarity measures $S_{N(a),b}$ and the distance travelled as obtained from odometry, see Figs. 4.6 and 4.7. Two parameters are determined from the nonlinear least squares fitting process: mean (d_μ) and variance (σ_x^2). An estimate of the relative position $[\mu_x, \mu_y]$ is calculated as

$$\mu_x = \cos(\mu_\theta)d_\mu \tag{4.3}$$
$$\mu_y = \sin(\mu_\theta)d_\mu, \tag{4.4}$$

where d_μ is the calculated mean of the fitted Gaussian and μ_θ the estimated relative orientation (Sec. 4.2.1).

In the experimental evaluation the Gaussian was estimated using five consecutive frames, e.g. using $[S_{a-2,b}, S_{a-1,b}...S_{a+2,b}]$. In addition, a Gaussian was only fitted to a neighbour $N(a)$ if the similarity measure had its peak in $S_{a,b}$ and the similarity measure for frames further away were smaller than closer ones.

Table 4.2: Statistics of the error ϵ between the Gaussian fit and the similarity measure $S_{a-2,b}, S_{a-1,b} ... S_{a+2,b}$ for each node for which the fit was performed in the outdoor / indoor data set.

node pair	ϵ	σ_ϵ
$< a-2, b >$	0.031	0.0441
$< a-1, b >$	0.029	0.0348
$< a, b >$	0.033	0.0601
$< a+1, b >$	0.026	0.0317
$< a+2, b >$	0.028	0.0388

To evaluate whether the evolution of the similarity measure in the vicinity of a visual relation can be reasonably approximated by a Gaussian, the mean error between the 5 similarity measures and the fitted Gaussian was calculated for the outdoor / indoor data set, which is described in Section 6.3.1). The results in Table 4.2 indicate that the Gaussian represents the evolution of the similarity in a reasonable way. Please note that frame b is taken at a later time than frame a meaning that the covariance estimate $C^{a,b}$ can be calculated directly without any time lag.

The complete covariance matrix C of the relative pose ($\mu = [\mu_x, \mu_y, \mu_\theta]$) is calculated as

$$C = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}. \tag{4.5}$$

4.3 Determining the Image Density

The distance between the successive frames $[d_{a-2,a-1}, d_{a-1,a}, ..., d_{a+1,a+2}]$ has previously been used to determine the relative pose estimate. This distance can either be kept constant by taking pictures whenever the robot has travelled a certain distance, which is the approach used in the experiments presented in Chapters 5 and 6. However, by utilising the similarity measure between the previously taken image and the current image the system could determine whether to add the current image. This would result in fewer images to represent the environment in open areas where the scene is typically rather stable compared to narrow areas where more images should be used.

4.4 Conclusion

This chapter has given a description of how relative pose estimates and the corresponding uncertainty can be calculated without using any depth information. The experimental verification of the proposed approach is described in

the localisation chapter (Ch. 5) and the SLAM chapter (Ch. 6). The main advantage of this registration method is that it uses only a small set of similarity measures in combination with metric data obtained from relative odometry readings as input. Only relative odometry readings are used between successive nodes or between the neighbours N. Odometry is fairly accurate over the relatively short distances that are considered here. Although local features are used in this chapter, it would be possible to extend the proposed registration method to use global feature based similarity meassures.

Chapter 5
Omni-vision Based Localisation

This chapter presents an image-based approach for localisation in non-static environments using local feature descriptors, and its experimental evaluation in a large, dynamic, populated environment where the time interval between collecting the data was up to two months. By using local features together with panoramic images, substantial changes in the environment can be handled. Results from global place recognition with no evidence accumulation and a Monte Carlo localisation method are presented. To test the robustness of the approach, experiments were conducted with up to 90% virtual occlusion in addition to the dynamic changes in the environment.

5.1 Introduction

5.1.1 Background

There has been much research on using accumulated sensory evidence to improve localisation performance, including a highly successful class of algorithms that estimate posterior probability distributions over the space of possible locations [59, 22, 107, 31, 42, 26]. These approaches enable both position tracking and localisation from scratch, for example, when the robot is started (no prior knowledge of its position) or becomes lost or "kidnapped" (incorrect prior knowledge).

In vision based approaches, panoramic or omni-directional cameras have become popular for self-localisation because of their relatively low cost and large field of view. This makes it possible to extract features that are invariant to the robot's orientation, for example, using various colour histograms [117, 17, 49] or Eigenspace models [69, 120, 6]. Other approaches create multiple images from the same location by shifting the panoramic view [119] or by rotation [63], which increases the amount of data several times. Another approach

is to only take pictures at the same orientation [6], e.g., when the robot is facing north.

Other innovations include methods to increase robustness to lighting variations [120] and multi-view registration to deal with occlusions [95]. Some authors have combined panoramic vision with particle filters for global localisation, including feature matching using Fourier transform [33], PCA [119] and colour histograms [54].

5.1.2 Dynamic Environments

Most previous work on robot mapping and localisation assumes a static world [14], i.e. that there are no changes to the environment between the time of mapping and the time of using the map. However, this assumption does not hold for typical populated environments. Humans (and other robots) are not merely "dynamic obstacles" that may occlude the robot's sensors – they also make changes to the world. For example, they may leave temporary objects such as packages, or move the furniture. In addition to these sudden changes, there may be gradual changes such as plants growing, coloured paint fading, etc.

Our approach to self-localisation in non-static environments uses an image matching scheme (Ch. 3) that is robust to many of the changes that occur under natural conditions. The robustness is achieved by looking at parts of the images (local features) instead of the whole images at once, meaning that partial occlusions or changes will only affect a subset of all features. Our hypothesis is that a map that is out of date can still contain much useful information. Thus the important question is how to extract features that can be used for matching new sensor data to a map that is only partially correct. This chapter presents an appearance-based approach to matching panoramic images that does not require calibration or geometric modelling of the scene or imaging system, thus it should be applicable to any mobile robot using omni-directional vision for self-localisation. The hypothesis is validated through experiments using sensor data collected by a mobile robot, PeopleBoy (see Sec. 2.1.1), in a real dynamic environment over a period of two months.

5.1.3 Overview

The image matching algorithm, described in Chapter 3, uses local features extracted from many small subregions of the image rather than global features extracted from the whole image, which makes the method very robust to variations and occlusions. For example, Fig. 5.1 shows some of the local features that were matched between two different panoramic images of a laboratory environment, recorded 56 days apart, despite changes such as a television appearing, chairs moving and people working. Our proposed method is similar to other approaches that use local features for self-localisation [121, 7, 68], with

Figure 5.1: Matching a new image (left) against the corresponding database image recorded 56 days earlier (right).

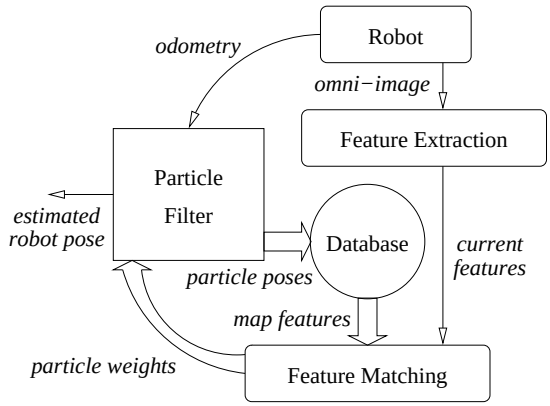

Figure 5.2: An overview of the proposed method. Input: the current omni-image and the current odometry reading. The database consists of poses (x, y, θ) of the database images together with the extracted features. For every pair of sensory inputs (panoramic image and odometry) features are extracted and the poses of the particles are updated with the odometry. Thereafter the weight of each particle is updated with the feature match value of the closest database location. Output: the current estimate of the robot position based on the weight and distribution of particles.

the differences that the method proposed here is adapted for panoramic images and was specifically designed and tested to work in long-term experiments conducted in a real dynamic environment. The results demonstrate that the robot is able to localise itself from scratch, including experiments in "kidnapping", and that the performance shows a graceful degradation to occlusions (validated for up to 90% of the robot's field of view).

Our localisation methods assume that the database (map) has been already created. To be able to match the current image with the images stored in the database, each image is converted into a set of features. The matching is done by comparing the features in the database with the features created from the current image, see Chapter 3. The database is constructed from another set of images collected by the same robot. For each stored image, the database contains a set of extracted local feature descriptors (Sec. 3.2.2) and the corresponding location (x, y, θ) of the robot, which in our case was estimated using a SLAM method [47] (see Sec. 5.4 for further details). In our experiments, 100 features were selected for each panoramic image. The diameter of the panoramic view is approximately 576 pixels. For each feature, the radius r and orientation θ are also stored (Fig. 3.5) for subsequent processing.

A novel scheme is introduced for combining local feature matching with a particle filter for global localisation (Sec. 5.2), which minimises computational costs as the filter converges. See also Fig. 5.2 for a brief overview of the method. To evaluate the method, we also compare the performance with several other types of features, including both global and local features (Sec. 5.3). How image matching performance can be further improved by incorporating information about the relative orientation of corresponding features between images is shown in Section 5.3.1. Our experiments were designed to test the system under a wide variety of conditions, including results in a large populated indoor environment (up to 5 persons visible) on different days under different lighting conditions (Sec. 5.4).

5.2 Monte Carlo Localisation

Monte Carlo methods such as particle filters [3] have become popular in recent years for estimating the state of a system at a certain time based on the current and past measurements. The probability $p(X_t|Z_{1:t})$ of a system being in state X_t given a history of measurements $Z_{1:t} = \{z_0, ..., z_t\}$ is approximated by a set of N weighted particles:

$$S_t = \{x_t^{(i)}, \pi_t^{(i)}\},\ i = 1...N. \tag{5.1}$$

Each particle $x_t^{(i)}$ describes a possible state together with a weight $\pi_t^{(i)}$, which is proportional to the likelihood that the system is in this state. Particle filtering consists of four main steps:

1. Create a new particle set S_{t+1} by resampling from the old particle set S_t based on the particle weights $\pi_t^{(i)}, i = 1...N$
2. Predict the next particle states based on the dynamic model $p(x_{t+1}^{(i)}|x_t^{(i)}, u_t)$ with incremental pose estimate (odometry) u_t, $i = 1...N$
3. Calculate the new weights by application of the measurement model: $\pi_{t+1}^{(i)} \propto p(z_{t+1}|X_{t+1} = x_{t+1}^{(i)}), i = 1...N$.
4. Normalise the weights $\pi_t^{(i)}$ so that $\sum_{i=1}^{N} \pi_t^{(i)} = 1$.

The estimate of the system state at time t is the weighted mean over all particle states:

$$\hat{X}_t = E(S_t) = \sum_{i=1}^{N} \pi_t^{(i)} x_t^{(i)}. \tag{5.2}$$

In our case the state is described by a three dimensional vector $x_t = (x, y, \theta)_t$ containing the robot pose consisting of the position (x, y) and the orientation θ of the robot. The x and y coordinates are initialised randomly within a radius of one meter around a randomly selected database pose to assure that the robot pose can be in the surrounding of the database poses, for example, between two database locations or orthogonal to the path the robot was driven during creation of the database. The orientation θ is calculated as the orientation of the database pose added to the relative orientation between the current frame and the selected database frame. The relative orientation is estimated as described in Section 4.2.1 with an added random value drawn from a normal distribution with standard deviation $\pi/8$ radians to allow particles located close to the same database pose to have different orientations. The prediction and measurement steps are described in the following sections.

5.2.1 Dynamic Model

All state variables $x_t^{(i)} = (x, y, \theta)_t$ are updated according to the odometry readings u_t from the robot. To cope with the additional uncertainty due to odometry error (the magnitude of the odometry error can be seen in Fig. 5.6), the odometry values are updated with small random values drawn from a normal distribution, using a standard deviation of 0.1 radians for the rotation and a standard deviation of 2% of the measured distance for the translation. To use a standard deviation which is not dependent on the rotation avoids that the model underestimates the rotational error while the robot according to the odometry follows a straight path.

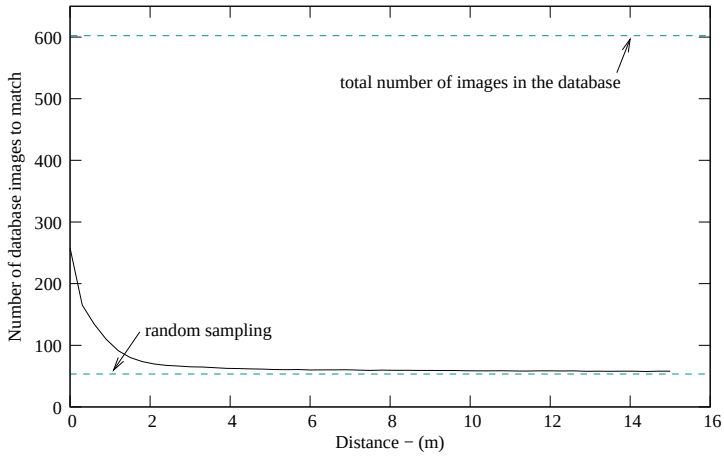

Figure 5.3: Number of database locations (in total 603) to match against distance travelled for Run1 with 50% occlusion.

5.2.2 Measurement Model

To calculate the weight of particles only the location of the database image that is closest to the current particle is used (see also the experimental setup, Sec. 5.4.1). This means that the computation time will decrease as the particles converge around the true location of the robot, since fewer images in the database need to be matched to the current image. Fig. 5.3 shows the decreasing number of matched database locations against distance travelled after initialisation of the particle filter. In the experimental evaluation the number of database locations was 603.

The particle weight $\pi^{(i)}$ is based on the similarity measure $S^{(i)} = S_{a,b}$ between the current image a and the closest database image b for particle $x_t^{(i)}$ (see Sec. 3.3). Hence, all particles that are closest to the same database location will have the same match value independent of the actual distance to the database position. Therefore, to avoid that the particles drift away from the mapped area, the weighting function $f_w(d)$ is applied:

$$f_w(d) = \begin{cases} \exp\left(-\frac{(d-\sigma)^2}{\tau^2}\right) & (d > \sigma) \\ 1 & (d \leqslant \sigma) \end{cases} \tag{5.3}$$

where d is the Euclidean distance between the particle and the database position. In the experiments, σ and τ were set to $2T$ where T is the minimum

distance between database positions (see Sec. 5.4.1). The new weight is then calculated as

$$\pi_t^{(i)} = f_w(d) \cdot S^{(i)}. \tag{5.4}$$

5.2.3 Inertia Models

For images with a large amount of occlusion, the number of matches could be low and possibly zero, even for the correct position. To increase the inertia of particles in order to survive through sections of the map with few matches, two additional approaches were evaluated in addition to the standard method, see Fig. 5.4.

Standard Method (No Inertia) In the standard approach, the weights $\pi_{t+1}^{(i)}$ of the new set of particles S_{t+1} are assigned from the measurement model without using the previous weight. If the number of matches is low for the correct location (for example, due to major changes in the environment), the number of particles at the correct location will decrease rapidly.

Forgetting Factor By only updating the weight of the particle if it is similar or better compared to the previous iteration, the "inertia" of the distribution is increased. To allow the weight to decrease a 'forgetting factor' f_{forget} is used. The new weight is used only if

$$\pi_{t+1}^{(i)} > \pi_t^{(i)} \cdot f_{forget} \cdot f_w(d),$$

otherwise

$$\pi_t^{(i)} \cdot f_{forget} \cdot f_w(d)$$

is used instead. To evaluate the performance of the method, different values were used, with the best results obtained with $f_{forget} \in [0.8, 0.9]$.

Keep Random By randomly keeping weights for some particles from the previous iteration, the inertia of the distribution is increased. With a probability $p(f_{keep})$, the weight of a particle is not updated. Instead the weight of that particle keeps its previous value,

$$\pi_t^{(i+1)} = \pi_t^{(i)}.$$

The performance was evaluated using different values of f_{keep}. The best results were obtained with $f_{keep} \in [0.2, 0.3]$.

5.3 Alternative Feature Methods

Together with MSIFT, three other features for image matching were evaluated, one global feature and two local features.

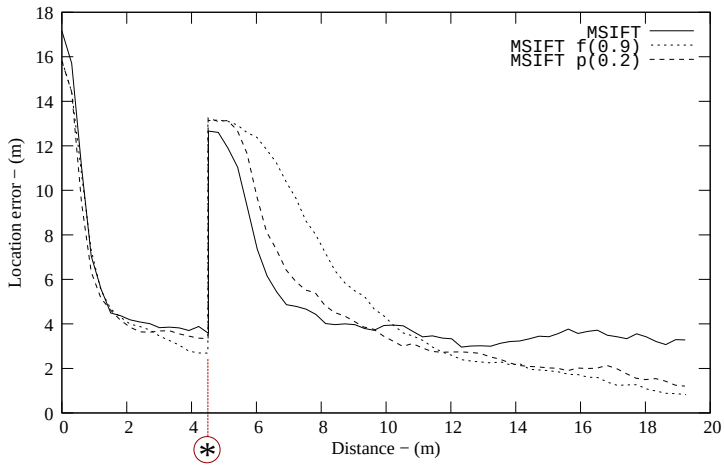

Figure 5.4: Localisation errors for the different inertia models with 50% occlusion and a kidnapped robot scenario where the robot was virtually moved to a new random position after approx. 4.5 meters, marked as an '*'. f - "Forgetting Factor" (f_{forget}), p - "Keep Random" (f_{keep}). Note, for the increased inertia models the convergence is slower but the performance is better.

Normalised Colour Histogram - NCH NCH is a global feature and is included for performance comparison with the other methods. All pixels covered by the reflective mirror (i.e. not the black centre and outer parts seen in Fig. 5.1) in the omni-directional image are used. The RGB values of each pixel are normalised, i.e. $R_{norm} = \frac{R}{R+G+B}$. Three histograms of normalised intensity are created, one for each colour. The histograms are matched by squared Euclidean distance.

Average Squared Difference Correlation - ASD ASD matches two local features F and F′ in two different images I and I′ by the sum of the squared differences of intensity in a surrounding neighbourhood window $\mathcal{N}$ as

$$ASD(F, F') = \frac{1}{N} \sum_{n \in \mathcal{N}(F), n' \in \mathcal{N}(F')} [I(n) - I'(n')]^2, \tag{5.5}$$

where N is the size of the neighbourhood $\mathcal{N}$.

Variance Normalised Correlation - VNC VNC is defined for local features F and F' as

$$VNC(F,F') = \frac{1}{N\sqrt{\sigma_I^2(F)\sigma_{I'}^2(F')}} \sum_{n\in\mathcal{N}(F),n'\in\mathcal{N}(F')} [I(n) - \overline{I(F)}][I'(n') - \overline{I'(F')}], \tag{5.6}$$

where $\overline{I(F)}$ and $\sigma_I^2(F)$ are the mean and variance over intensity of the neighbourhood of $\mathcal{N}$. VNC was shown to give high invariance to changes in view point orientation [38].

Matching of local descriptors was done as described in Section 3.3, where the match score is based on the total number of matched features between two images. The neighbourhood $\mathcal{N}$ was set to 12×12 pixels giving a feature vector of length 144. To determine the interest point, the same feature detection approach as in MSIFT was used, see Section 3.2.2.

5.3.1 Improved Image Matching

To improve image matching, the position of each point in the image could also be used. This could be extended all the way to extracting and matching full 3D information between images, but this would require geometric modelling and a higher computational cost, as in [13]. However, one common property of omni-directional imaging systems for mobile robot navigation is symmetry about the z-axis, meaning that it should be possible to incorporate information about the relative orientation of matched features between images without loss of generality.

As described in Section 3.3, the rotation ϕ between matched images is estimated using a histogram of the respective rotations $\theta_F - \theta_{F'}$ between matched features F and F', where θ is the direction to the feature shown in Fig. 3.5. This histogram is used here to obtain an additional certainty measure, by counting the number of points that lie in the winning bin and the two adjacent bins to the left and right. For each feature match where $|\theta_F - \theta_{F'} - \phi| < \frac{\pi}{16}$, one extra match is added and therefore the similarity measure $S^{(i)}$ is increased.

This method can only be applied to local features. In the result section the results with the improved matching are denoted MSIFT*, ASD* and VNC*. In the Monte Carlo localisation using MSIFT*, the improved similarity measure is used to update the weights of particles.

5.4 Results

The results are divided into two parts, one considering the location and orientation recognition performance with no prior knowledge (Sec. 5.4.2) and the other evaluates the full Monte Carlo localisation scheme (Sec. 5.4.3).

Figure 5.5: Robot platform in the test environment.

5.4.1 Experimental Set-up

Robotic platform and environment

The robot used in the experiments is an ActivMedia PeopleBot called PeopleBoy equipped with a LMS-200 SICK laser range scanner and an omni-directional lens from RemoteReality together with an ordinary CCD camera mounted on top of the robot, see Fig. 5.5 and 2.2. The localisation system consists of a database of features where one set of features is stored for each database position. The features were calculated from images taken at known positions. To obtain these positions and the ground truth data for performance evaluation, a SLAM implementation was applied as described in [47]. A total of 603 images were collected covering an area of approximately 60×55 meters, as shown in Fig. 5.6. New laser scans and images were recorded if the rotation since the previous image exceeded 15 degrees or when the translation exceeded 0.5 meters. For each image the corresponding pose estimate from the SLAM algorithm was stored.

All data sets used in the experiments are visualised as occupancy maps in Fig. 5.6 and Fig. 5.7. Run_1 and Run_4, which mainly cover the student area and the labs. Run_2 is from a corridor, see Fig. 5.8, which contains a lot of similar features, e.g., doors to office rooms, and a lack of furniture or objects. Run_3 passes parts of the office corridor, the coffee room and the secretary offices. Each run was recorded at a different time compared to the database. Run_1 was recorded 2 days before the database, both Run_2 and Run_3 were recorded 56 days later, while Run_4 was recorded 64 days later. Run_4 and most of Run_3, see Fig. 5.7, right, were collected with the robot driving in the opposite direction compared to when collecting the images for the database. The distance between each successive image for the test runs was 0.25 meters.

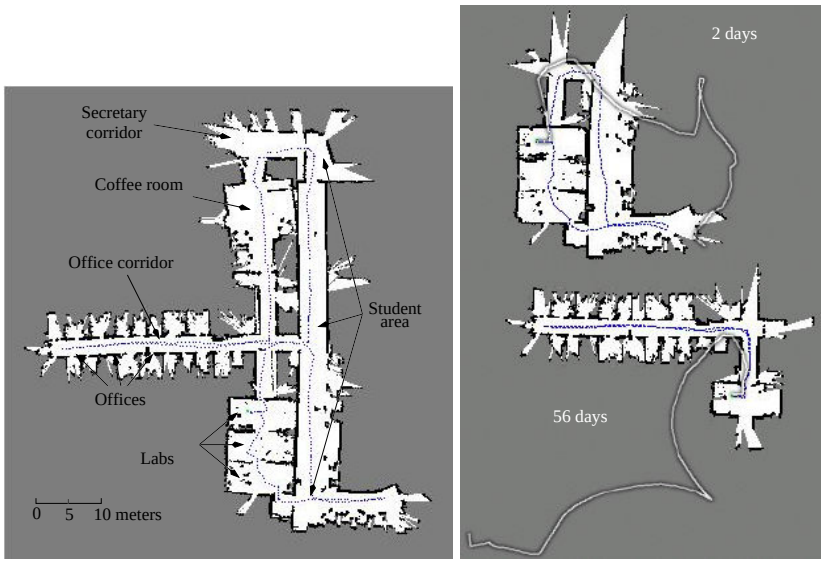

Figure 5.6: Left: Area covered by the database. Right: Two of the test sequences with ground truth and raw odometry data, Run_1 (above) and Run_2 (below) with the number of days between each run was collected compared to the database.

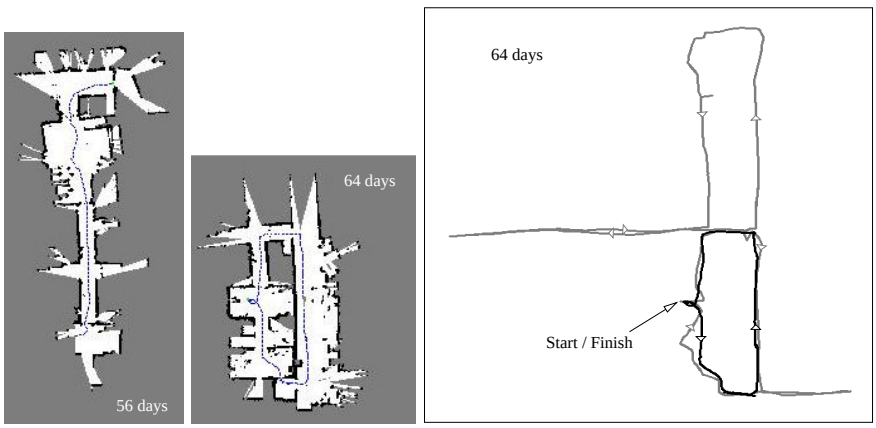

Figure 5.7: Left: Test sequence Run_3. Middle: Test sequence Run_4. Right: Paths travelled by the robot, Run_4 (black) and the database (grey), the arrows indicate the path direction.

Building the database

Since image feature matching does not depend on the orientation of the images, it is only necessary to use images with different location, i.e. when the robot is travelling back and forward along a corridor it is sufficient to save the data in one direction. The building of the database starts after the run is completed and optimised with SLAM. The images are used in the same order as they were recorded. An image is added to the database if the metric distance to the nearest stored image exceeds a threshold T. In this chapter, a value of $T = 0.4$ meters was used. For each image included in the database, a feature set is calculated and the 100 strongest features are stored, as described in Sections 3.2.2 and 5.3 except for the global NCH method, see Section 5.3.

5.4.2 Location and Orientation Recognition

Different features were evaluated using the described method on a set of test runs that overlap with the area covered by the database, as shown in Fig. 5.6. Comparative performance statistics for the different matching algorithms are given in Tables 5.1 – 5.4. Note that all results presented are for the global localisation robot problem without any prior knowledge of the robot's location.

To calculate performance, the positions of the test image and the image found in the database estimated from SLAM were compared. Only matches that had a Euclidean distance smaller than a certain value r_{max} were considered to be a correct match. In some experiments the accuracy of the rotation estimates ϕ was also considered (see the rightmost columns of Tables 5.1– 5.4). The results show that the performance of all algorithms in all runs was significantly better using the improved image matching scheme (a paired t-test with $p < 0.05$ was used for the statistical evaluation), and that the best performance was achieved by MSIFT*.

Table 5.1: Run_1, parts of the student area and the labs.

	r_{max}			ϕ_{max} ($r_{max} = 4.0$)		
	2.0	4.0	6.0	$\pm\frac{\pi}{2}$	$\pm\frac{\pi}{4}$	$\pm\frac{\pi}{8}$
NCH	2.2%	4.4%	8.6%	n/a	n/a	n/a
ASD	49.7%	57.9%	62.7%	55.4%	53.2%	51.3%
VNC	3.8%	8.2%	11.4%	7.6%	5.4%	2.5%
MSIFT	96.2%	98.1%	99.7%	98.1%	98.1%	98.1%
ASD*	70.0%	75.6%	77.5%	73.7%	71.8%	71.8%
VNC*	3.5%	7.9%	13.3%	7.0%	5.4%	3.2%
MSIFT*	98.1%	99.4%	100.0%	99.4%	99.4%	99.4%

Table 5.2: Run_2, Ph.D. corridor, part of the labs.

	r_{max}			ϕ_{max} ($r_{max} = 4.0$)		
	2.0	4.0	6.0	$\pm\frac{\pi}{2}$	$\pm\frac{\pi}{4}$	$\pm\frac{\pi}{8}$
NCH	15.4%	23.0%	28.7%	n/a	n/a	n/a
ASD	28.0%	36.6%	40.9%	30.2%	27.7%	25.0%
VNC	0.6%	3.7%	4.9%	0.6%	0.3%	0.3%
MSIFT	60.4%	71.0%	74.7%	64.0%	62.5%	59.4%
ASD*	39.3%	47.6%	52.7%	41.8%	39.6%	35.7%
VNC*	1.2%	3.4%	4.6%	1.5%	0.3%	0.0%
MSIFT*	67.0%	76.2%	80.2%	71.3%	71.0%	68.3%

Table 5.3: Run_3, parts of the labs and Ph.D. corridor, coffee room, secretary corridor.

	r_{max}			ϕ_{max} ($r_{max} = 4.0$)		
	2.0	4.0	6.0	$\pm\frac{\pi}{2}$	$\pm\frac{\pi}{4}$	$\pm\frac{\pi}{8}$
NCH	10.5%	16.4%	20.2%	n/a	n/a	n/a
ASD	40.6%	50.9%	54.3%	45.7%	42.3%	40.6%
VNC	1.7%	5.1%	12.6%	3.4%	2.3%	1.7%
MSIFT	56.6%	70.3%	74.3%	65.7%	62.3%	57.7%
ASD*	56.0%	63.4%	67.4%	60.0%	59.4%	56.6%
VNC*	0.6%	5.1%	11.4%	3.9%	1.7%	1.1%
MSIFT*	68.0%	79.4%	83.4%	74.9%	72.0%	68.6%

Table 5.4: Run_4, parts of student area and the labs.

	r_{max}			ϕ_{max} ($r_{max} = 4.0$)		
	2.0	4.0	6.0	$\pm\frac{\pi}{2}$	$\pm\frac{\pi}{4}$	$\pm\frac{\pi}{8}$
NCH	10.5%	16.4%	20.2%	n/a	n/a	n/a
ASD	15.4%	24.0%	26.3%	18.9%	17.7%	13.1%
VNC	3.4%	6.9%	13.7%	4.6%	2.3%	1.1%
MSIFT	81.1%	85.7%	88.6%	85.1%	85.1%	83.4%
ASD*	27.4%	37.1%	38.9%	33.7%	32.6%	28.6%
VNC*	4.6%	6.3%	12.0%	4.6%	3.4%	2.3%
MSIFT*	87.4%	92.6%	93.1%	92.0%	92.0%	90.9%

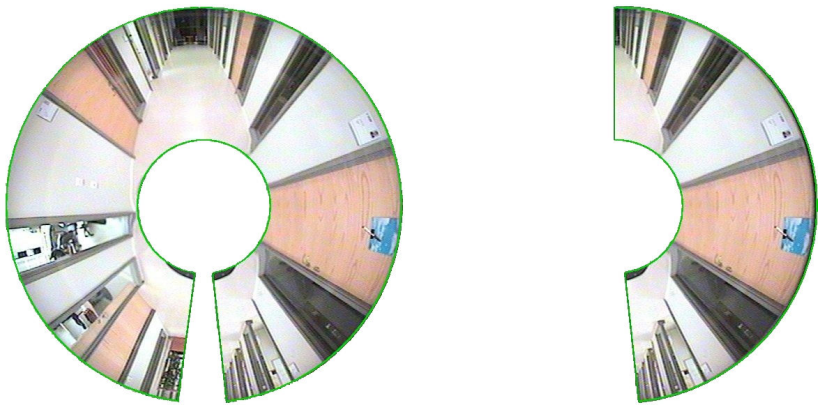

Figure 5.8: Virtual occlusion: valid regions for extracting features. Left: no added occlusion (a small sector of the omni-image is removed because of occlusion by the stand on which the camera is mounted). Right : 50% added occlusion. The image is taken from Run_2.

5.4.3 Monte Carlo Localisation

In the experiments, a total of 500 particles was used and the 10% of the particles with the lowest weights were reinitialised (see Sec. 5.2) at each iteration to enable localisation. The inertia model used is the Forgetting Factor with $f_{factor} = 0.9$, see Section 5.2.3. To calculate performance, the Euclidean distance between positions of the test image and the median value of 90% of the particles with the highest fitness value was used, in order to increase robustness against outliers.

The database map and the maps for the different runs (see Fig. 5.6) were manually fitted and 'placed' on top of each other. To obtain more evaluation data, each dataset was used multiple times by dividing it into smaller runs. The new runs contained 30 images each covering approximately 9 meters, where each run has a different starting position. To test the robustness, additional levels of occlusion were simulated by removing features. The occlusion percentage indicates the proportion of the current image (field of view) where features were deleted (see Fig. 5.8). For the global localisation problem, the particles were reinitialised after each completed run (see Fig. 5.9 and 5.10). To evaluate the kidnapped robot scenario a randomly selected run was used to accumulate evidence before the robot was 'virtually' moved by randomly selecting another run (see Fig. 5.11 and 5.12).

A comparison between MSIFT and MSIFT* can be seen in Table 5.5, where the distance travelled until the localisation error falls below 2 and 5 meters respectively is shown together with the standard deviation. Interestingly, in con-

Table 5.5: Distance travelled in meters until localisation error is less than 2 and 5 meters, no occlusion.

	error < 2m		error < 5m		No. of
	MSIFT	MSIFT*	MSIFT	MSIFT*	expts.
Run$_1$	0.48±0.56	0.42±0.50	0.34±0.47	0.30±0.41	265
Run$_2$	3.16±1.55	2.91±1.50	1.88±1.26	1.90±1.28	277
Run$_3$	3.59±1.62	3.18±1.52	2.10±1.27	1.90±1.21	124
Run$_4$	0.90±0.77	0.85±0.73	0.64±0.64	0.63±0.63	229

Table 5.6: Distance travelled in meters until localisation error is less than 5 and 10 meters with 50% occlusion.

	error < 5m		error < 10m		No. of
	MSIFT	MSIFT*	MSIFT	MSIFT*	expts.
Run$_1$	1.20±1.07	0.63±0.90	0.83±0.92	0.43±0.77	265
Run$_2$	3.29±1.59	2.31±1.49	2.14±1.45	1.53±1.31	277
Run$_3$	3.87±1.78	2.86±1.64	3.12±1.72	2.56±1.61	124
Run$_4$	2.18±1.46	1.02±1.09	1.71±1.34	0.82±1.02	229

trast to the previous experiments in location and orientation recognition, the differences in performance between the two systems using Monte Carlo localisation were not significant ($p < 0.05$, unpaired t-test), except for one case (Run3, error < 2m).

As an additional evaluation on the difference between MSIFT and MSIFT* the same comparison was done but with an occlusion rate of 50% and localisation error threshold set to 5 and 10 meters. The results are shown in Table. 5.6, where all the differences in performance between MSIFT and MSIFT* were statistically significant.

This would suggest that the accumulation of sensory evidence, including relative odometry, in the non-occlusion case has a greater influence on overall localisation performance than the improved measurement model using rotation information. The improved measurement model gave better performance regarding convergence with a higher occlusion rate.

Localisation errors against distance travelled for MSIFT and MSIFT* with 0% and 50% occlusion rate are shown in Fig. 5.13 and 5.14.

5.5 Conclusion

This chapter presented a self-localisation algorithm for mobile robots using panoramic images. The methods consist of existing state-of-the-art algorithms for creating local descriptors [80] and probabilistic state estimation [3] with an omni-directional imaging system on a mobile robot, and the experimental evaluation of the entire system in a real dynamic environment over an extended

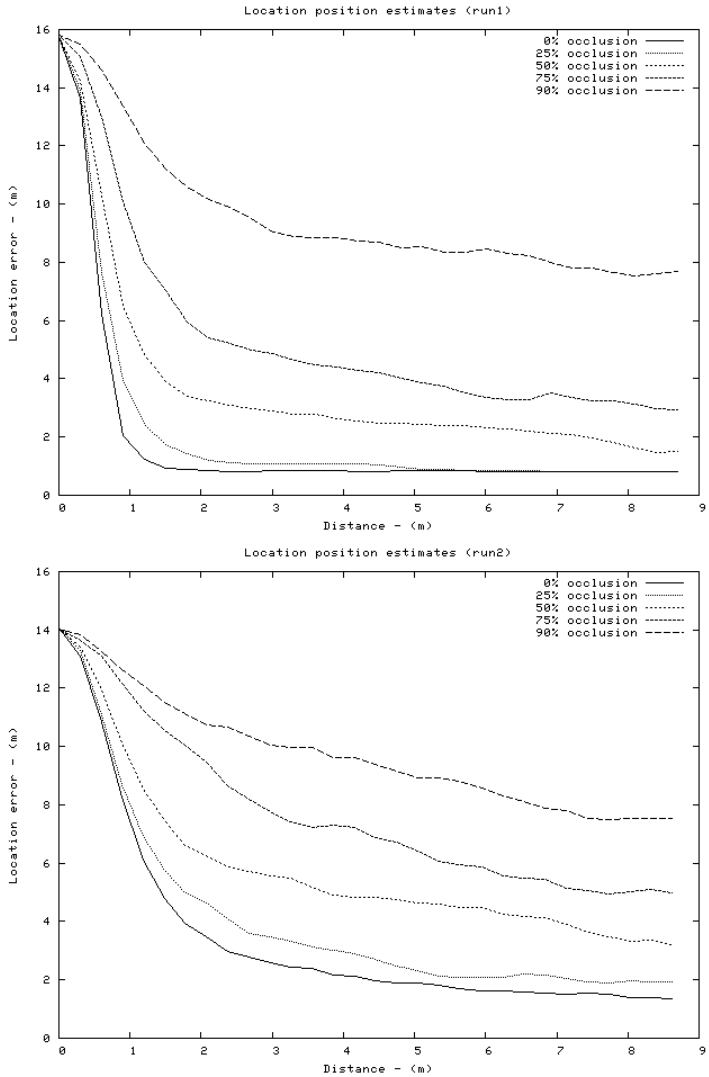

Figure 5.9: Localisation errors against distance travelled for global localisation experiments using MSIFT with different levels of occlusion. Top: results from Run1. Bottom: results from Run2.

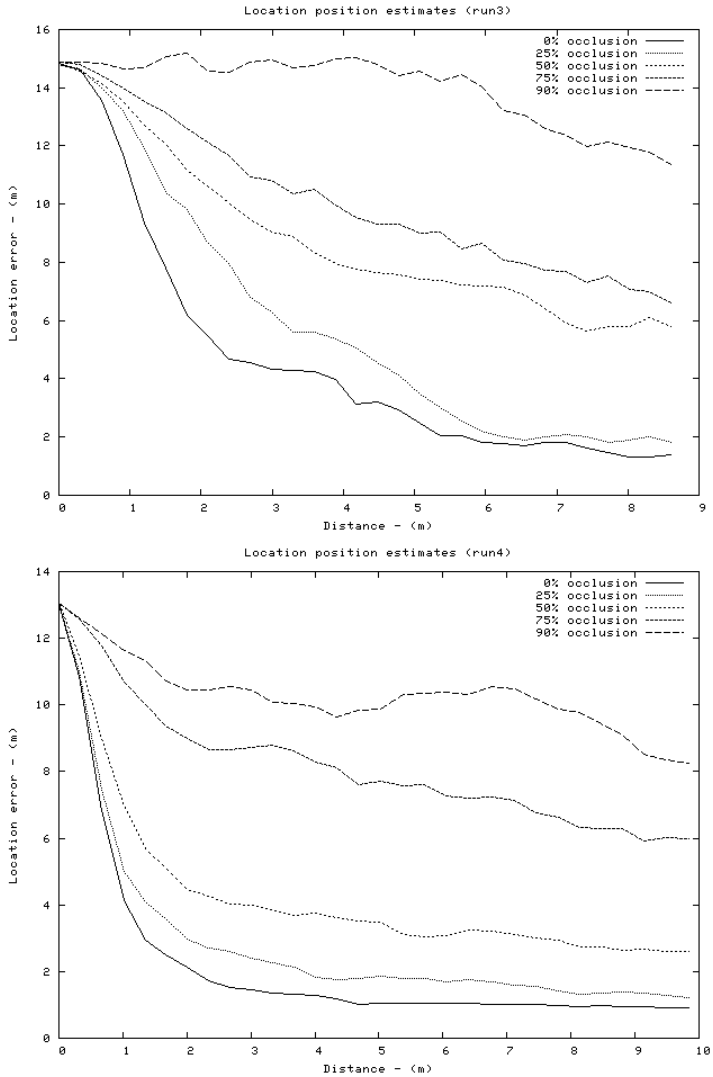

Figure 5.10: Localisation errors against distance travelled for global localisation experiments using MSIFT with different levels of occlusion. Top: results from Run3. Bottom: results from Run4.

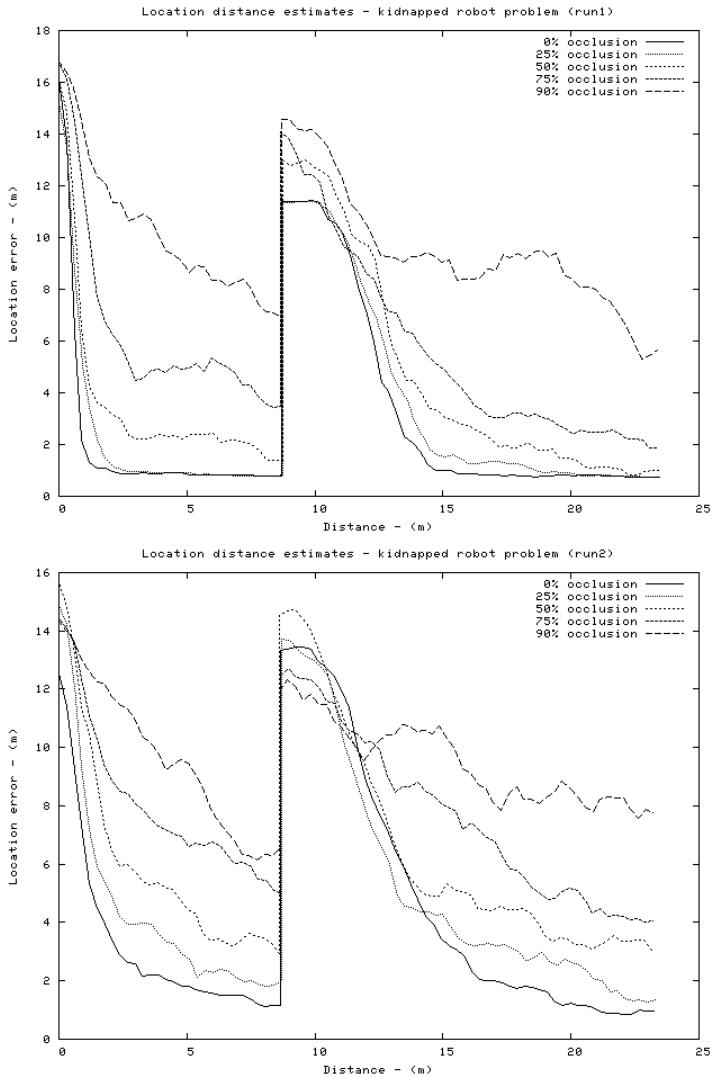

Figure 5.11: Localisation errors against distance travelled for the kidnapped robot scenario using MSIFT. Top: results from Run1. Button: results from Run2.

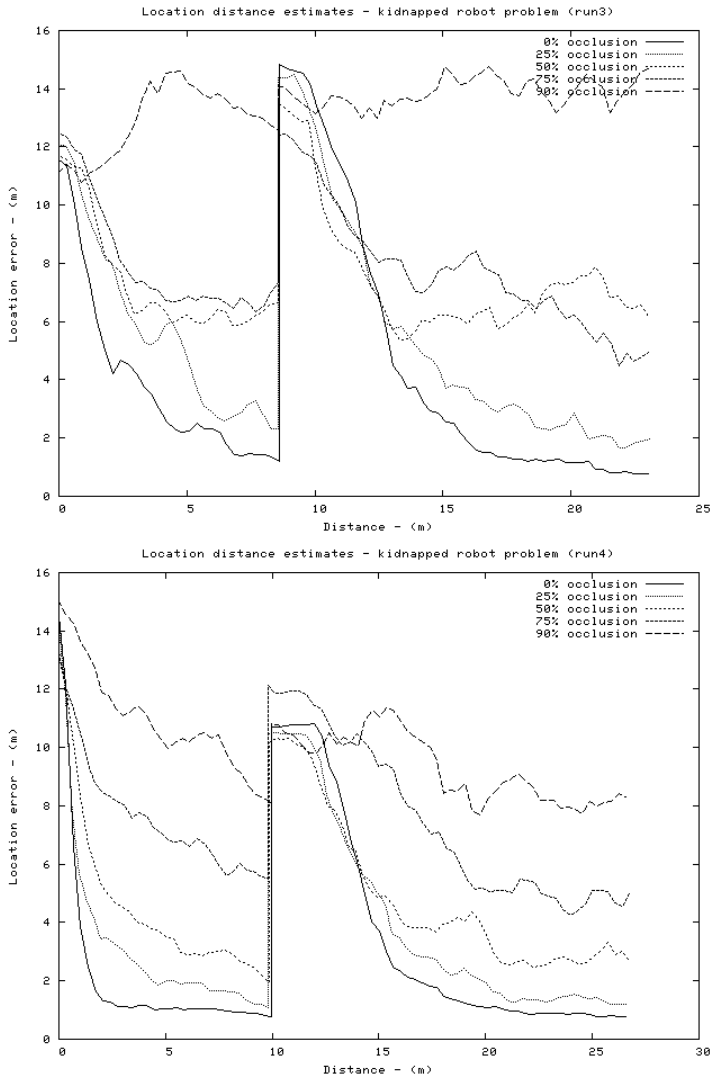

Figure 5.12: Localisation errors against distance travelled for the kidnapped robot scenario using MSIFT. Top: results from Run3. Bottom: results from Run4.

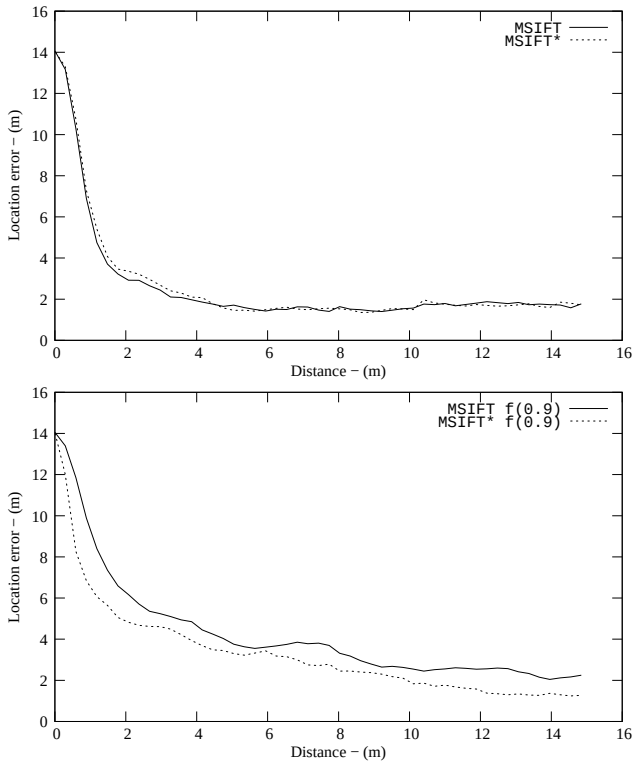

Figure 5.13: Localisation errors for MSIFT and MSIFT*. Top: results from Run2. Bottom: results from Run2 with 50% occlusion. With no occlusion, both MSIFT and MSIFT* performs similarly, however with more occlusion the improved matching shows better performance.

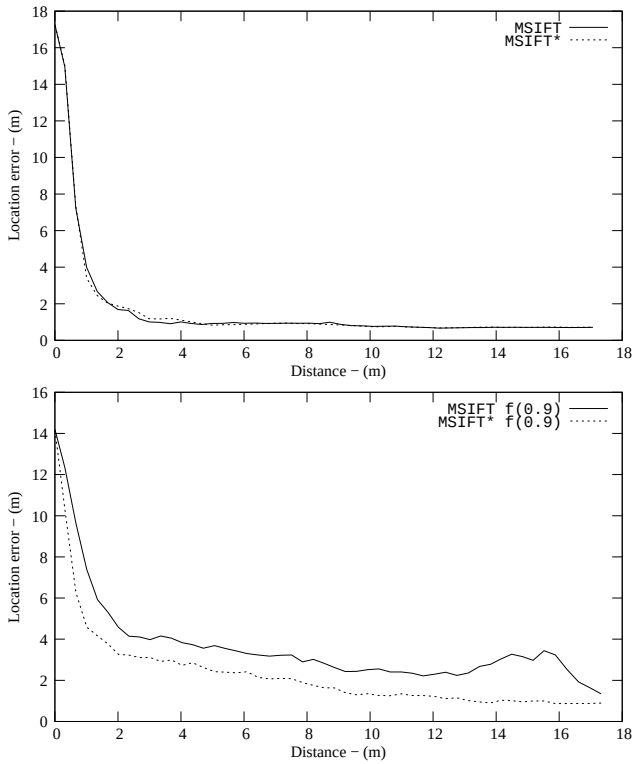

Figure 5.14: Localisation errors for MSIFT and MSIFT*. Top: results from Run4. Bottom: results from Run4 with 50% occlusion. With no occlusion, both MSIFT and MSIFT* performs similarly, however with more occlusion the improved matching shows better performance.

period of time. By using experiments with data collected on different days over a period of several months, it has been shown that even if the room has gone through some changes regarding location of furniture, objects and persons, or severe occlusion (virtual occlusions tested up to 90%), it is still possible to extract good position estimates.

Chapter 6
Omni-vision Based SLAM

This chapter presents a vision-based approach to SLAM in indoor / outdoor environments with minimalistic sensing and computational requirements. The approach is based on a graph representation of robot poses, using a relaxation algorithm to obtain a globally consistent map. Combined indoor and outdoor experiments demonstrate that the approach can handle qualitatively different environments (without modification of the parameters), that it can cope with violations of the "flat floor assumption" to some degree, and that it scales well with increasing size of the environment, producing topologically correct and geometrically accurate maps at low computational cost. Further experiments demonstrate that the approach is also suitable for combining multiple overlapping maps, e.g. for solving the multi-robot SLAM problem with unknown initial poses.

6.1 Introduction

This chapter presents a new vision-based approach to the problem of simultaneous localisation and mapping (SLAM). Especially compared to SLAM approaches using a 2D laser scanner, the rich information provided by a vision-based approach about a substantial part of the environment allows for dealing with high levels of occlusion [5] and enables solutions that do not rely strictly on a flat floor assumption. Cameras can also offer a longer range and are therefore advantageous in environments that contain large open spaces. It is further argued that vision can enable solutions in highly cluttered environments where laser range scanner based SLAM algorithms might fail [99].

The proposed method is named "Mini-SLAM" since it is minimalistic in several ways. On the hardware side, it relies solely on odometry and an omni-directional camera as the external source of information. This allows for less expensive systems compared to methods that use 2D or 3D laser scanners. Please note that the robot used for the experiments was also equipped with

a 2D laser scanner. This laser scanner, however, was not used in the SLAM algorithm but only to visualize the consistency of the created maps.

Apart from the frugal hardware requirements, the method is also minimalistic in its computational demands. Map estimation is performed on-line by a linear time SLAM algorithm on an efficient graph representation. The main difference to other vision-based SLAM approaches is that there is no estimate of the positions of a set of landmarks involved, enabling the algorithm to scale up better with the size of the environment. Instead, a measure of image similarity is used to estimate the relative pose between corresponding images ("visual relations") and the uncertainty of this estimate. Given these "visual relations" and relative pose estimates between consecutive images obtained from the odometry of the robot ("odometry relations"), the Multilevel Relaxation algorithm [47] is then used to determine the maximum likelihood estimate of all image poses. The relations are expressed as a relative pose estimate and the corresponding covariance. A key insight is that the estimate of the relative pose in the "visual relations" does not need to be very accurate as long as the corresponding covariance is modeled appropriately. This is because the relative pose is only used as an initial estimate that the Multilevel Relaxation algorithm can adjust according to the covariance of the relation. Therefore, even with fairly imprecise initial estimates of the relative poses it is possible to build geometrically accurate maps using the geometric information in the covariance of the relative pose estimates. Mini-SLAM was found to produce consistent maps in various environments, including, for example, a data set of an environment containing indoor and outdoor passages (path length of 1.4 km) and an indoor data set covering five floor levels of a department building.

Further, the Mini-SLAM approach is extended to address the multi-robot SLAM problem, demonstrating its ability to combine multiple overlapping maps with unknown initial poses. We also provide an evaluation of the robustness of the suggested approach with respect to poor odometry or a less reliable measure of visual similarity.

6.1.1 Related Work

Using a camera as the external source of information in SLAM has received increasing attention during the past years. Many approaches extract landmarks using local features in the images and track the positions of these landmarks. As the feature descriptor, Lowe's scale invariant feature transform (SIFT) [80] has been used widely [102, 10]. An initial estimate of the relative pose change is often obtained from odometry [10, 62, 66], or where multiple cameras are available, as in [36, 100], multiple view geometry can be applied to obtain depth estimates of the extracted features. To update and maintain the position of visual landmarks, Extended Kalman Filters (EKF) [25, 62], Rao-Blackwellised Particle Filters (RBPF) [36, 10] are among the most popular methods applied. The visual SLAM method in [25] uses a single camera. Particle filters were

utilised to estimate the path, while the landmark positions were updated with an EKF. In order to obtain metrically correct estimates, initial landmark positions had to be provided by the user. A similar approach described in [66] also uses a single camera but applies a converse methodology. The landmark positions were estimated with a Kalman filter and a particle filter was used to estimate the path.

Due to their suitability for addressing the correspondence problem, vision-based systems have been applied as an addition to laser scanning based SLAM approaches for detecting loop closure. The principle has been applied to SLAM systems based on a 2D laser scanner [57] and a 3D laser scanner [92].

Other mapping approaches have combined omni-directional vision for place recognition with odometry for obtaining geometric information in a graph. For example, Ranganathan and Dellaert [98] use odometry information to evaluate the likelihood of topological map hypotheses in a Markov-chain Monte Carlo (MCMC) framework. However, the emphasis of their work is on selecting the correct topology using very coarse visual features, and their approach is unlikely to scale to environments of the size presented here.

In the approach proposed in this chapter, the SLAM optimization problem is solved at the graph-level with the Multilevel Relaxation (MLR) method of Frese and Duckett [47]. This method could be replaced by alternative graph based SLAM methods, for example, the online method proposed by Grisetti et al. [52] based on the stochastic gradient descent method proposed by Olson et al. [94].

The rest of the chapter is structured as follows. Section 6.2 describes the proposed SLAM approach. It includes a overview of the SLAM optimisation technique (Sec. 6.2.1), a description of the way in which relations are computed from odometry (Sec. 6.2.2), and from visual similarity (Sec. 6.2.3). Then the experimental set-up is detailed and the results are presented in Section 6.3.

6.2 Mini-SLAM

The Mini-SLAM approach is based on two principles. First, odometry is fairly precise if the distance travelled is short. Second, by using visual matching, correspondence between robot poses can be detected reliably even though the covariance of the current pose estimate, i.e. the search region, is large.

We therefore have two different types of relations r; relations based on odometry r_o and relations based on visual similarities r_v.

6.2.1 Multi-Level Relaxation

The SLAM problem is solved at the graph-level, where the Multi-Level Relaxation (MLR) method of Frese and Duckett [47] is applied. A map is represented as a set of nodes connected in a graph structure. An example is shown in Fig. 6.1. Each node corresponds to the robot pose at a particular time and each

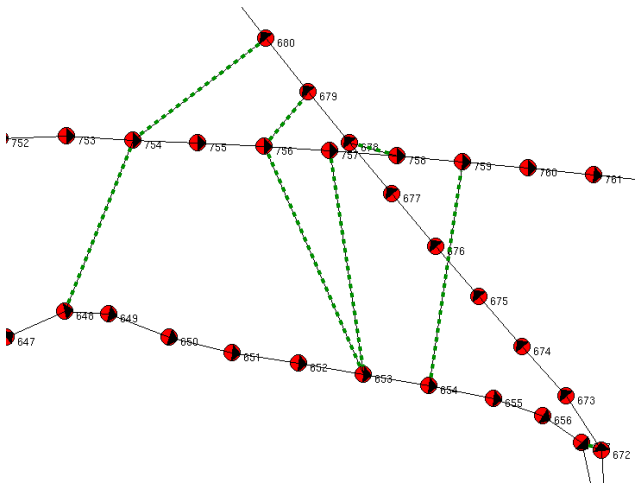

Figure 6.1: The graph representation used in multi-level relaxation (MLR). The figure shows the frames (nodes) and the relations (edges) both from odometry r_o and visual similarities r_v. Each frame a contains a reference to a set of features F_a extracted from the omni-directional image I_a, an odometry pose x_a^o, a covariance estimate of the odometry pose $C_{x_a^o}$, the estimated pose $\hat{x}_a$ and an estimate of its covariance $C_{\hat{x}_a}$. See also Fig. 6.2, which shows images from the region that is represented in the graph shown here.

link to a relative measurement of the spatial relation between the two nodes it connects. A node is created for each omni-image in this work and the terms node and frame are used interchangeably.

In this method, a map is represented as a set of nodes connected in a graph. Each node or frame corresponds to the robot pose at a particular time (in our case when an omni-image was taken), and each link corresponds to a relative measurement of the spatial relation between the two nodes it connects, see Fig. 6.1.

The function of the MLR algorithm can be explained as follows. The input to the algorithm is a set $\mathcal{R}$ of $m = |\mathcal{R}|$ relations on n planar frames (x, y, θ) (i.e. a two-dimensional representation is used). Each relation $r \in \mathcal{R}$ describes the likelihood distribution of the pose of frame a relative to frame b. It is modelled as a Gaussian distribution with mean μ^r and covariance C^r. The output is the maximum likelihood (ML) estimation vector $\hat{x}$ for the poses of all the frames. In other words, the purpose of the algorithm is to assign a globally consistent set of Cartesian coordinates to the nodes of the graph, based on local (relative), inconsistent (noisy) measurements, by maximising the total likelihood of all measurements.

6.2.2 Odometry Relations

By using odometry to add a relation r_o, the relative position change μ_{r_o} can be directly extracted from the odometry readings and the covariance C_{r_o} can be estimated by a motion model. In the implementation of Mini-SLAM the model suggested in [34] is used where the covariance is modelled as

$$C_{r_o} = \begin{bmatrix} d^2\delta^2_{X_d} + t^2\delta^2_{X_t} & 0 & 0 \\ 0 & d^2\delta^2_{Y_d} + t^2\delta^2_{Y_t} & 0 \\ 0 & 0 & d^2\delta^2_{\theta_d} + t^2\delta^2_{\theta_t} \end{bmatrix} \tag{6.1}$$

where d and t are the total distance travelled and total angle rotated by the mobile robot between the two frames. The δ_X parameters relate to the forward motion, the δ_Y parameters the side motion and the δ_θ parameters the rotation of the robot. The six parameters adjust the influence of the distance d and rotation t in the calculation of the covariance matrix. They were tuned manually once and kept constant throughout the experiments. Please note that an odometry relation r_o is only added between successive frames.

6.2.3 Visual Similarity Relations

Adding a relation r_v, which relies on visual similarities, requires to estimate the likelihood distribution between two frames using the method described in Chapter 4, see also Figs. 6.2 and 6.3. In addition, the following two steps are performed.

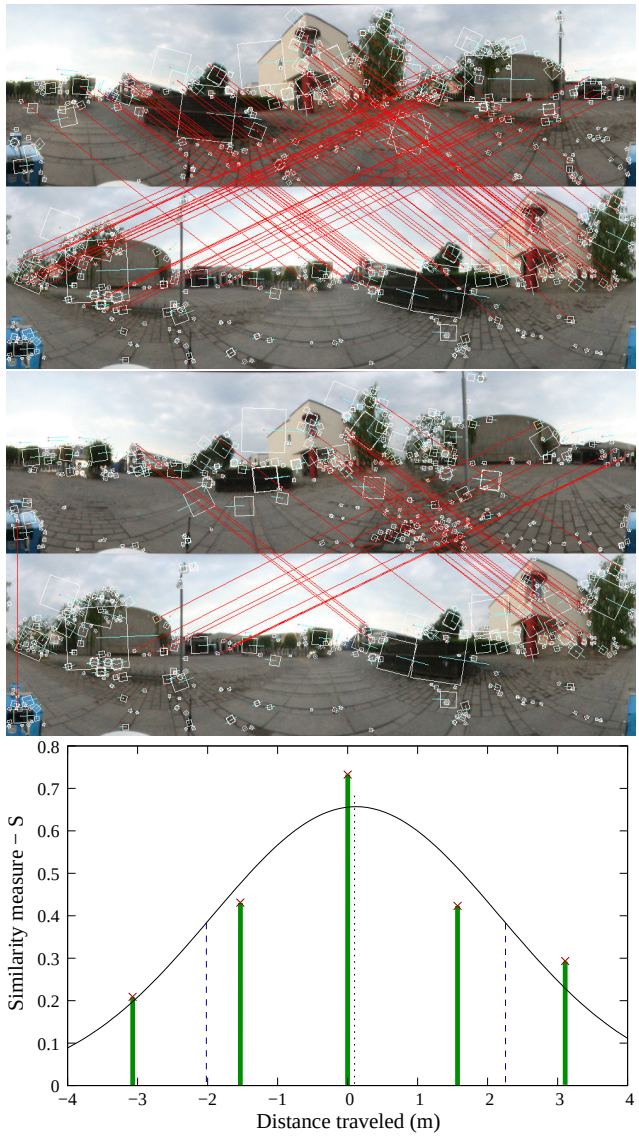

Figure 6.2: Example of loop closure detection outdoors. The distance to the extracted features is comparably large. The top figure shows feature matches at a peak of the similarity value $S_{678,758} = 0.728$, whereas the middle figure shows the matches two steps away $S_{680,758} = 0.286$ (~3 meters distance). The pose standard deviation $\sigma_{x^{rv}}$ and $\sigma_{y^{rv}}$ was estimated to be 2.059 m and the mean d_{μ} to 0.199 m. The lowest figures shows the similarity measures S with the distances obtained from odometry together with the fitted Gaussian curve.

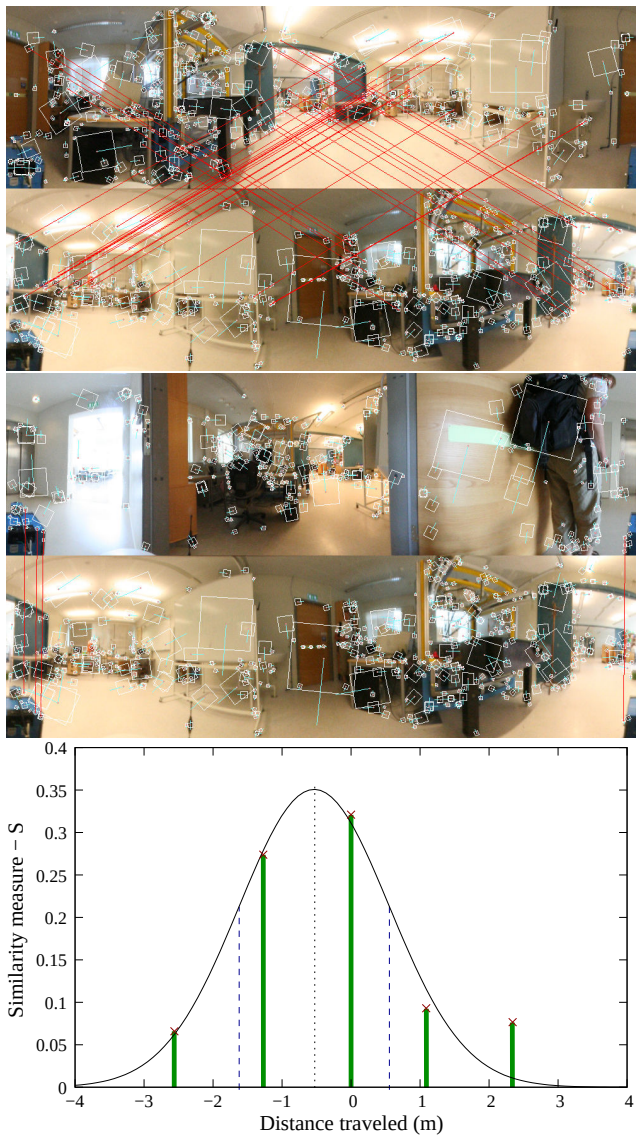

Figure 6.3: Example of loop closure detection indoors. Here the distance to the features is smaller compared to Fig. 6.2. The top figure shows matches at the local peak with a similarity value $S_{7,360} = 0.322$, whereas the middle figure shows the matches two steps away $S_{9,360} = 0.076$ (~3 meters distance). The pose standard deviation $\sigma_{x^{rv}}$ and $\sigma_{y^{rv}}$ was estimated to be 1.090 m and the mean d_μ to -0.534. The lowest figures shows the similarity measures S with the distances obtained from odometry together with the fitted Gaussian curve.

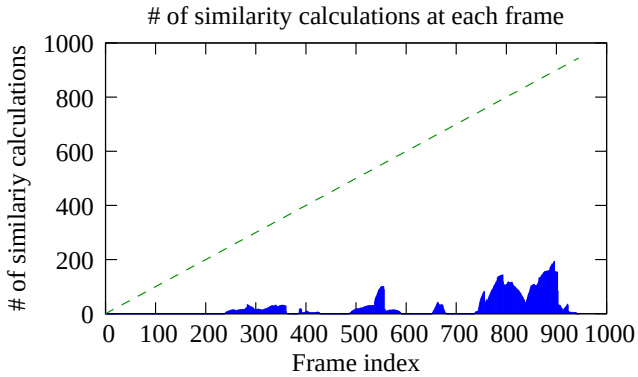

Figure 6.4: Number of similarity calculations performed at each frame in the outdoor/indoor data set. The first frames were compared around frame 240, since up to then none of the previous frames were within the search area around the current pose estimate defined by the estimated pose covariance. The diagonal line indicates the linear increase for the case that the frames to match are not pre-selected.

Selecting Frames to Match

In order to speed up the algorithm and make it more robust to perceptual aliasing (the problem that different regions have similar appearance), only those frames are selected for matching that are likely to be located close to each other.

Consider the current frame b and a previously recorded frame a. If the similarity measure was to be calculated between b and all previously added frames, the number of frames to be compared would increase linearly, see Fig. 6.4. Instead, frames are only compared if the current frame b is within a search area around the pose estimate of frame a. The size of this search area is computed from the estimated pose covariance.

From the MLR algorithm (see Sec. 6.2.1) we obtain the maximum likelihood estimate $\hat{x}_b$ for frame b. There is, however, no estimate of the corresponding covariance $C_{\hat{x}}$ that could be used to distinguish whether frame a is likely to be close enough to frame b so that it can be considered a candidate for a match, i.e. a frame for which the similarity measure $S_{a,b}$ should be calculated. So far, we have defined two types of covariances: the odometry covariance C_{r_o} and the visual relation covariance C_{r_v}. To obtain an overall estimate of the relative covariance between frame a and b we first consider the covariances of the odometry relations r_o between a and b and compute relative covariance $C_{x^o_{a,b}}$ as

$$C_{x^o_{a,b}} = \sum_{j \in (a,b-1)} \mathbf{R}_j C_{r_{o_j}} \mathbf{R}_j^T. \tag{6.2}$$

$\mathbf{R}_j$ is a rotation matrix, which is defined as

$$\mathbf{R}_j = \begin{pmatrix} \cos(\hat{x}^\theta_{j+1} - \hat{x}^\theta_j) & -\sin(\hat{x}^\theta_{j+1} - \hat{x}^\theta_j) & 0 \\ \sin(\hat{x}^\theta_{j+1} - \hat{x}^\theta_j) & \cos(\hat{x}^\theta_{j+1} - \hat{x}^\theta_j) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{6.3}$$

where $\hat{x}^\theta_j$ is the estimated orientation of frame j.

As long as no visual relation r_v has been added, either between a and b or any of the frames between a and b, the relative covariance $C_{\hat{x}_{a,b}}$ can be determined directly from the odometry covariance $C_{x^o_a}$ and $C_{x^o_b}$ as described above. However, when a visual relation $r_v^{a,b}$ between a and b is added, the covariance of the estimate $C_{\hat{x}_b}$ decreases. Using the covariance intersection method [116], the covariance for frame b is therefore updated as

$$C_{\hat{x}_b} = C_{\hat{x}_b} \oplus (C_{\hat{x}_a} + C_{r_v^{a,b}}), \tag{6.4}$$

where $\oplus$ is the covariance intersection operator. The covariance intersection method weights the influence of both covariances C_a and C_b as

$$C_A \oplus C_B = [\omega C_A^{-1} + (1-\omega) C_B^{-1}]^{-1}, \tag{6.5}$$

The parameter $\omega \in [0, 1]$ is chosen so that the determinant of the resulting covariance is minimized [65].

The new covariance estimate is also used to update the frames between a and b by adding the odometry covariances $C_{x^o_{a..b}}$ in opposite order (i.e. simulate that the robot is moving backwards from frame b to a). The new covariance estimate for frame $j \in (a, b)$ is calculated as

$$C_{\hat{x}_j} = C_{\hat{x}_j} \oplus (C_{\hat{x}_b} + C_{x^o_{b,j}}). \tag{6.6}$$

Visual Relation Filtering

To avoid adding visual relations with low similarity, visual similarity relations $r_v^{a,b}$ between frame a and frame b are only added if the similarity measure exceeds a threshold $t_{vs} : S_{a,b} > t_{vs}$. In addition, similarity relations are only added if the similarity value $S_{a,b}$ has its peak at frame a (compared to the neighbouring frames $N(a)$), see also Section 4.2.2. There is no limitation on the number of visual relations that can be added for each frame.

6.2.4 Fusing Multiple Data Sets

Fusion of multiple data sets recorded at different times is related to the problem of multi-robot mapping [67], where each of the data sets is collected concurrently with a different robot. The motivation for multi-robot mapping is not only to reduce the required time to explore an environment but also to

merge the different sensor readings in order to obtain a more accurate map. The problem addressed is equivalent to "multi-robot SLAM with unknown initial poses" [58] because the relative poses between the data sets are not given. The exploration problem is not considered here.

Only a minor modification of the standard method described above is necessary to address the problem of fusing multiple data sets. The absence of relative pose estimates between the data sets is compensated for by not limiting the search region for which similarity calculations S are performed. This is implemented by incrementally adding data sets and setting the relative pose between consecutively added data sets initially to (0,0,0) with an infinite pose covariance. Such odometry relations appear as long, diagonal lines in Fig. 6.16 representing the transition between *lab* to *studarea* and *studarea* to *lab – studarea*.

6.3 Experimental Results

In this section, we present results from five different data sets.

- Outdoor / indoor data set - a 1.4 kilometre run with both indoor and outdoor images
- Multiple floors - contains 5 different floors using 3 elevators, contains loops in the "level dimension"
- Partly overlapping data - 3 different data sets, indoor environment where one data set overlaps the other two

An overview of all data sets is presented in Table 6.1. All data sets were collected with our mobile robot Tjorven, see Fig. 2.2. The platform uses "skid-steering", which is prone to bad odometry. In the different data sets different wheel types (indoor / outdoor) were used. The robot's odometry was calibrated (for each wheel type) by first driving forward 5 meters to obtain a distance per encoder tick value, and second by completing one full revolution to determine the number of differential encoder ticks per angular rotation. Finally the drift parameter was adjusted so that the robot would drive forward in a straight line, i.e. to compensate for the slightly different size of the wheel pairs.

The omni-directional images were first converted to panoramic images with a resolution of 1000 x 289 pixels. When extracting SIFT features the initial doubling of the images was not performed, i.e. SIFT features from the first octave were ignored, simply to lower the amount of extracted features.

The results are presented both visually with maps obtained by superimposing laser range data using the poses estimated with Mini-SLAM and quantitatively by the mean squared error (MSE) from ground truth data. Since the corresponding pose pairs $< \hat{x}_i, x_i^{GT} >$ between the estimated pose $\hat{x}_i$ and the

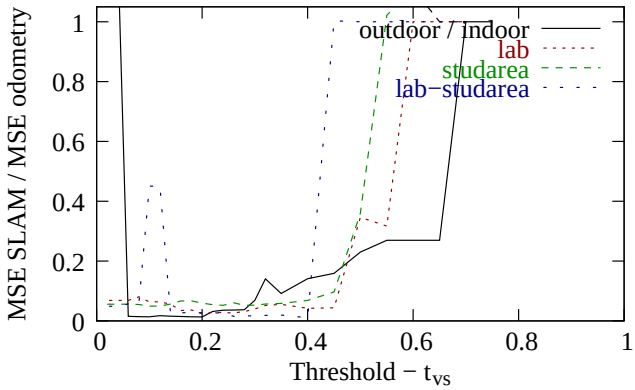

Figure 6.5: The influence of threshold parameter t_{vs}.

ground truth pose x_i^{GT} are known, rigid transformation can be applied directly; we apply the method suggested by Arun et al. [8].

To investigate the influence of the threshold t_{vs}, described in Section 6.2.3, the MSE was calculated for all data sets for which ground truth data were available. The result in Fig. 6.5 shows that the value of the threshold t_{vs} can be selected so that it is nearly optimal for all data sets and that there is a region in which minor changes of the t_{vs} do not strongly influence the accuracy of the map. Throughout the remainder of this section a constant threshold $t_{vs} = 0.2$ is used .

In order to give a better idea of the function of the Mini-SLAM algorithm, the number of visual relations per node depending on the threshold t_{vs} is shown in Fig. 6.6. The overview of all data sets presented in Table 6.1 also contains the number of similarity calculations performed and the evaluation run time on a Pentium 4 (2GHz) processor with 512 MB of RAM memory. This time

Table 6.1: For each data set: number of nodes $\#\hat{x}$, visual relations $\#r_v$, performed similarity calculations #S, average number of extracted visual features μ_F per node with variance σ_F, evaluation run time T (excluding the similarity computation).

	$\#\hat{x}$	$\#r_v$	#S	μ_F	σ_F	T (s)
outdoor / indoor	945	113	24784	497.5	170.0	66.4
multiple floor levels	409	198	13764	337.9	146.7	21.0
lab	86	60	443	571.5	39.6	3.6
studarea	134	31	827	426.6	51.1	9.4
lab – studarea	86	10	101	459.8	125.8	3.8

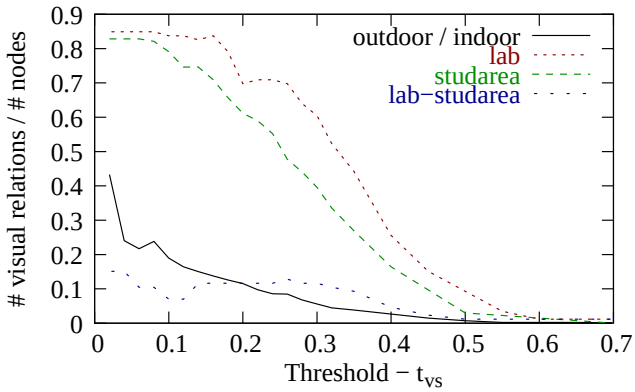

Figure 6.6: The amount of visual nodes added to the graph depending on the threshold t_{vs}.

does not include the time required for the similarity computation. Each similarity calculation (including relative rotation and variance estimation) took 0.30 seconds using a data set with an average of 522.3 features with standard deviation of 21.4. Please note, however, that the implementation used for feature matching was not optimised for computational efficiency.

6.3.1 Outdoor / indoor data set

A large set of 945 omni-directional images was collected over a total distance of 1.4 kilometers with height differences of up to 3 meters. The robot was driven manually and the data were collected in both indoor and outdoor areas over a period of 2 days (due to the limited capacity of the camera battery).

Visualised results

To visualise the maximum likelihood (ML) estimate $\hat{x}$ of the robot poses, laser scans acquired at the same time (and pose) as the omni-images were used to render an occupancy map. See Fig. 6.7 for the whole map using grid cells of size 25x25 cm^2 grid size. In Fig. 6.8 only the centre part is shown with a grid size of 10x10 cm^2.

Comparison to ground truth obtained from DGPS

To evaluate the accuracy of the created map, the robot position was measured with differential GPS (DGPS) while collecting the omni-directional images. Thus, for every SLAM pose estimate there is a corresponding DGPS position $< \hat{x}_i, x_i^{DGPS} >$.

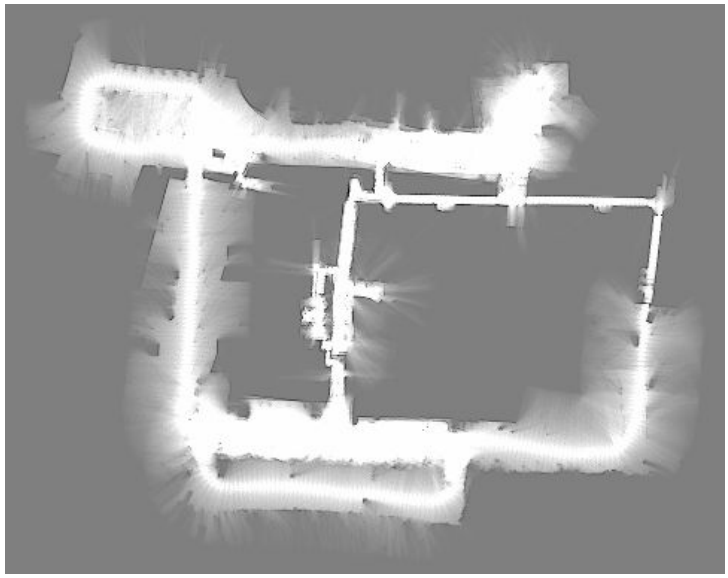

Figure 6.7: Visualised map using laser range data for each image node. Note that the laser data are used only for visualisation. The grid size is 25x25 cm^2.

DGPS gives a smaller position error than GPS. However since only the signal noise is corrected, the problem with multipath reflection still remains. DGPS is also only available if the radio link between the robot and the stationary GPS is functional. Thus, only a subset of pose pairs $< \hat{x}_i, x_i^{DGPS} >_{i=1..N}$ can be used for ground truth evaluation. DGPS measurements were considered only when at least five satellites were visible and the radio link to the stationary GPS was functional. The valid DGPS readings are indicated as light (blue) dots in Fig. 6.9. The total number of pairs used to calculate the MSE for the whole map was 377 compared to the total number of frames which was 945. To measure the difference between the poses estimated with Mini-SLAM $\hat{x}$ and the DGPS positions x^{DGPS} (using UTM WGS84, which provides a metric coordinate system), the two data sets have to be aligned. Since the correspondence of the filtered pose pairs is known, $< \hat{x}_i, x_i^{DGPS} >$, an optimal rigid alignment can be determined directly with the method by Arun et al. [8] as described above.

The mean square error (MSE) between x^{DGPS} and $\hat{x}$ for the data set shown in Fig. 6.9 is 4.89 meters. To see how the MSE evolves over time when creating the map, MSE was calculated from the new estimates $\hat{x}$ after each new frame was added. The result is shown in Fig. 6.10 and compared to the MSE obtained using only odometry to estimate the robot's position. Please note that the MSE was evaluated for each frame added. Therefore, when DGPS data are

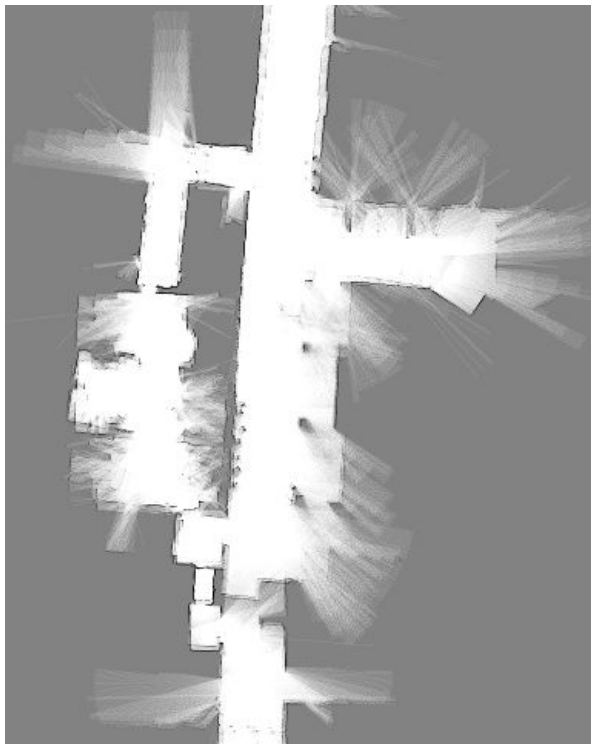

Figure 6.8: Visualised map using laser range data for the centre parts of the map. The grid size is 10x10 cm^2.

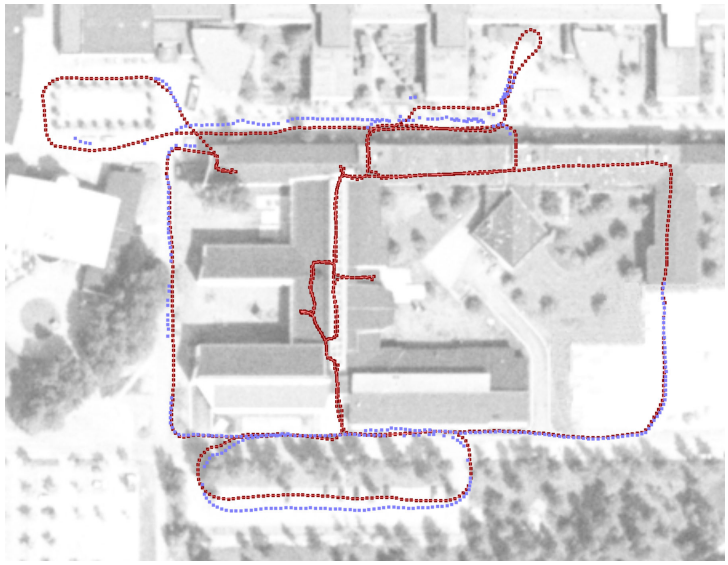

Figure 6.9: DGPS data x^{DGPS} with aligned SLAM estimates $\hat{x}$ displayed on an aerial image. The darker (red) squares show the Mini-SLAM poses and the lighter (blue) the DGPS poses for which the number of satellites was considered acceptable. The deviation seen at the bottom (the car park) is mainly caused by the fact that the car park is elevated compared to the rest of the environment.

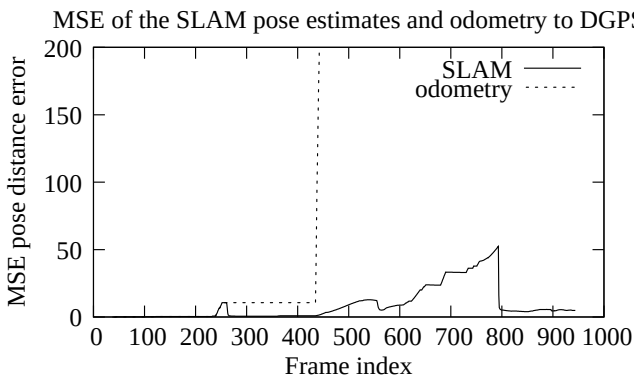

Figure 6.10: Evolution of the MSE between the ground truth position obtained from DGPS readings x^{DGPS} and the Mini-SLAM estimate of the robot pose $\hat{x}$ as frames are added to the map. Drops in the MSE indicate that the consistency of the map has been increased. The final MSE of the raw odometry was 377.5 m^2.

Figure 6.11: Environment images from floor levels 1-5.

not available, the odometry MSE x^o will stay constant for these frames. This can be seen, for example, between frames $250 - 440$ in Fig. 6.10. For the same frames, the MSE of the SLAM estimate $\hat{x}$ is not constant since new estimates are computed for each frame added and loop closing also occurs indoors or generally when no DGPS is available. The first visual relation r_v was added around frame 260. Until then, the error of the Mini-SLAM estimate $\hat{x}$ and the odometry MSE x^o were the same.

6.3.2 Multiple floor levels

This data set was collected inside the technology department building at Örebro University. It includes all (five) floor levels and connections between the floor levels by three elevators, see Fig. 6.11. The data contain loops in 2-d coordinates and also loops involving different floor levels. This data set contains 419 panoramic images and covers a path with a length of 618 meters. The geometrical layout differs for the different floors, see Fig. 6.12. No information about the floor level is used as an input to the system, hence the robot pose is still described using (x, y, θ).

Visualised results

There are no ground truth data available for this data set. It is possible, however, to get a visual impression of the accuracy of the results from Fig. 6.13. The figure shows occupancy grid maps obtained from laser scanner readings and raw odometry poses (left), or the Mini-SLAM pose estimates (right), respectively. All floors are drawn on top of each other without any alignment. To further illustrate the Mini-SLAM results, an occupancy map was also created separately for each floor from the laser scanner readings and Mini-SLAM pose estimates, see Fig. 6.12. Here, each pose was assigned to the corresponding floor level manually.

This experiment mainly illustrates the robustness of data association that is achieved using omni-directional vision data. The similarity matrix and a

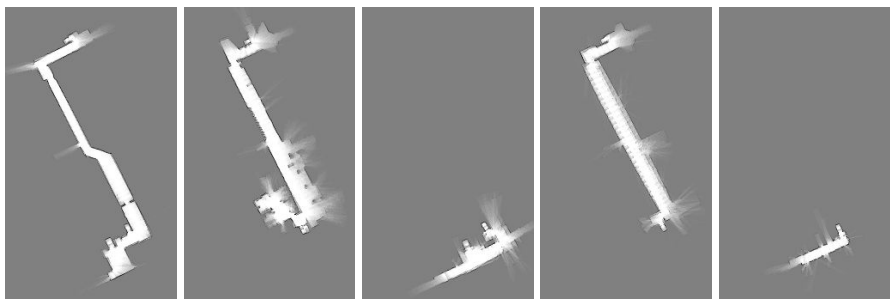

Figure 6.12: Occupancy maps for floor levels 1-5 drawn using the laser data at each estimated pose. The assignment of initial poses to floor levels was done manually and is only used to visualise these maps.

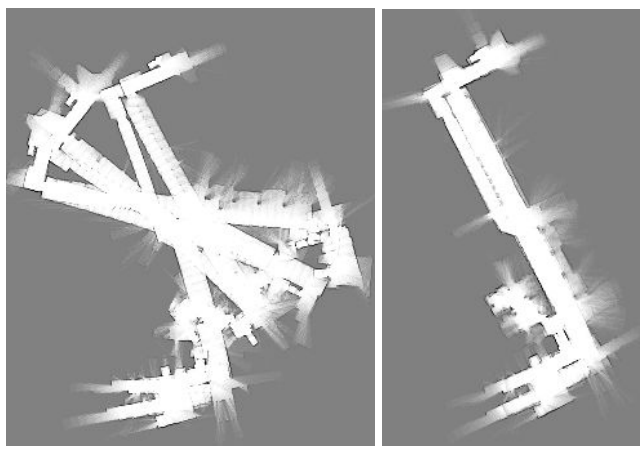

Figure 6.13: Occupancy grid map of all five floors drawn on top of each other. Left: Gridmap created using pose information from raw odometry. Right: Using the estimated robot poses from Mini-SLAM.

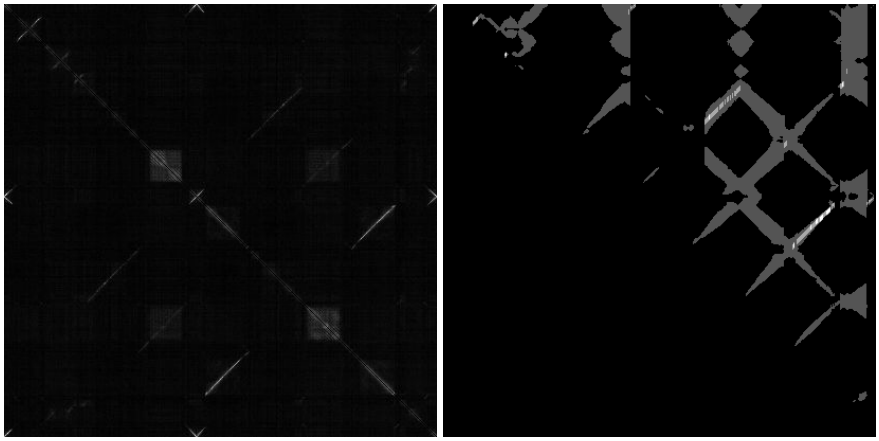

Figure 6.14: Left: Pose similarity matrix for the "Multiple floor levels" data set. Right: Similarity access matrix showing which similarity measures were used in the Mini-SLAM computation. Brighter pixels were used more often where as black indicates that the similarity measure is not used.

similarity access matrix for the "Multiple floor levels" data set are shown in Fig. 6.14.

6.3.3 Partly overlapping data

This data set consists of three separate indoor sets: lab (*lab*), student area (*studarea*) and a combination of both (*lab – studarea*), see Fig. 6.15. Similar to the data set described in Section 6.3.2, omni-directional images, 2D laser range data and odometry were recorded. The ground truth x^{GT} is determined by using laser scanner and odometry together with the MLR approach as in [47].

Visualised results

Fig. 6.16 shows parts of the final graph where different colours represent the different data sets. In Fig. 6.17 laser readings generated from raw odometry and the estimated poses from the proposed method are drawn together with the estimated path.

Fig. 6.18 shows the similarity matrix and the similarity access matrix for the *lab – studarea* data set.

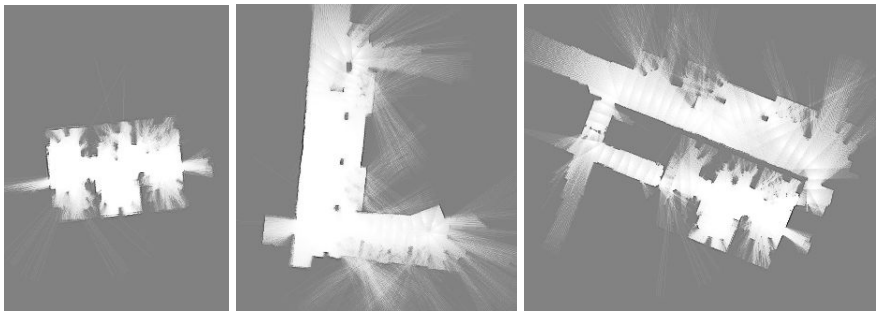

Figure 6.15: Sub-maps for the partly overlapping data. Left: lab. Middle: studarea. Right: lab – studarea, overlapping both lab and studarea.

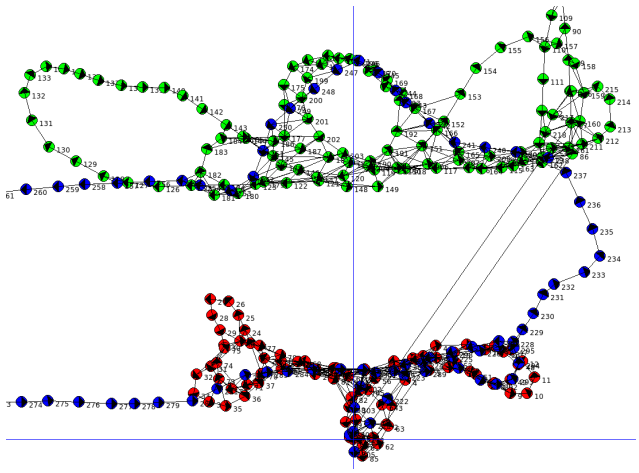

Figure 6.16: Parts of the final MLR graph containing the three different data set coloured with red (lab), green (studarea) and blue (lab – studarea).

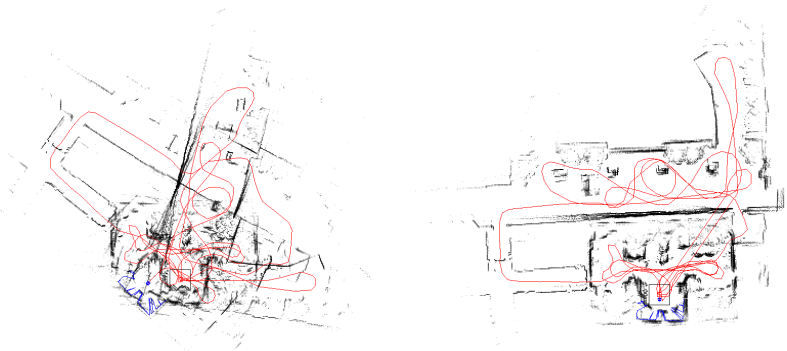

Figure 6.17: Results using the partly overlapping data set. Left: Laser based map using the raw odometry and Right: Laser map using the poses from the proposed method.

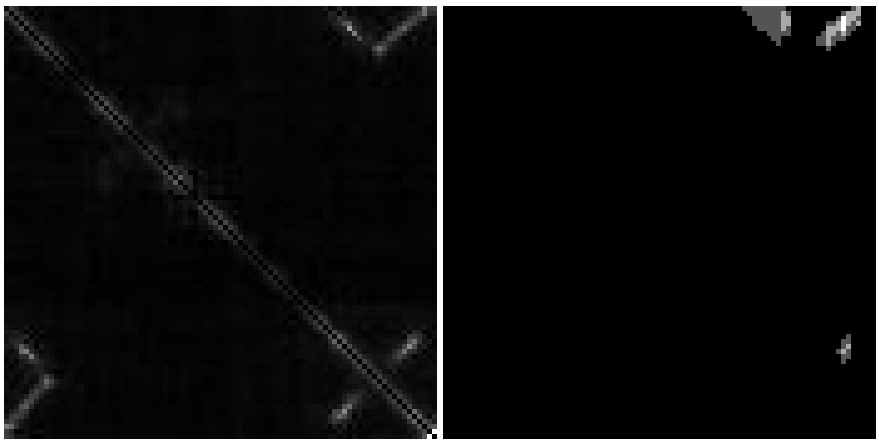

Figure 6.18: Left: Pose similarity matrix for the lab−studarea data set. Right: Similarity access matrix showing which similarity measures are used in the proposed method. Brighter pixels were used more often.

Table 6.2: MSE results before and after merging of the data sets and using odometry only.

	lab	studarea	lab – studarea
before fusing	0.002	0.029	0.036
after fusing	0.002	0.029	0.013
raw odometry	0.065	0.481	1.296

Table 6.3: MSE results (mean and stddev) after adding a random variable drawn from $N(0, \sigma)$ to each similarity measure $S_{a,b}$.

σ	mean	stddev
0.02	0.03	0.004
0.05	0.03	0.011
0.10	0.11	0.074
0.20	0.94	0.992
0.40	1.35	1.304
0.80	1.49	1.240

Comparison to ground truth obtained from laser based SLAM

As described in Sec. 6.2.4 fusion of multiple maps was motivated both by the multi-robot mapping aspect and the increased accuracy of the resulting maps. Instead of simply adding the different maps onto each other, the fused maps also use the additional information to improve the accuracy of the sub-maps. This is illustrated in Table 6.2 which shows the MSE (again obtained by determining the rigid alignment between $\hat{x}$ and x^{GT}) before and after the fusion was performed. While the data sets lab and studarea shows a negligible change in accuracy, lab – studarea clearly demonstrate a large improvement.

Robustness evaluation

The suggested method relies on incremental pose estimates (odometry) and a visual similarity measure S. The robustness of the method is evaluated by corrupting these two inputs and evaluating the performance. The studarea data set is used and each evaluation was repeated 10 times.

In the first test, the similarity measures S were modified by adding a random variable drawn from a Gaussian distribution $\mathcal{N}(0, \sigma)$ with varying standard deviation σ, see Table 6.3. The amount of added noise has to be compared to the range of $[0, 1]$ in which the similarity measure S lies, see Eq. 3.13.

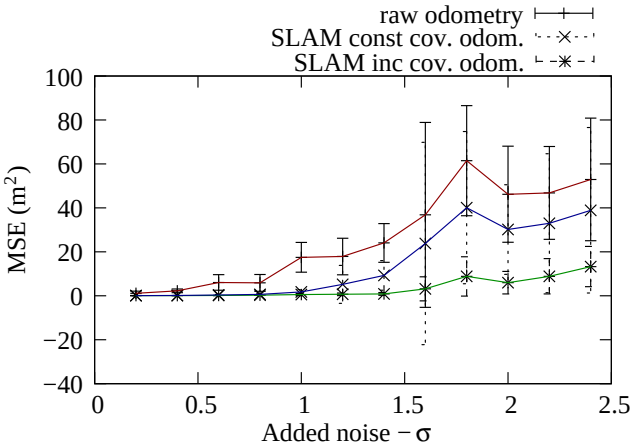

Figure 6.19: MSE results ($mean$ and $stddev$) for x (odometry) and $\hat{x}$ (estimated poses) after corrupting the odometry using random variables drawn from $N(0, \sigma)$. The plot also shows the MSE when the odometry covariance is increased with the added noise.

The robustness evaluation with respect to the similarity measure S shows that the system can handle additional noise to some extent, but incorrect visual relations will affect the accuracy of the final map. This illustrates that the proposed method, as many others, would have difficulties in perceptually similar locations in case the uncertainty of the pose estimates $C_{\hat{x}}$ is high.

In the second test, the odometry values were corrupted by adding additional noise to the incremental distance d and the orientation θ. The corrupted incremental distance d' is calculated as

$$d' = d + 0.1d\mathcal{N}(0, \sigma) + 0.2\theta\mathcal{N}(0, \sigma), \tag{6.7}$$

and the orientation θ' as

$$\theta' = \theta + 0.2d\mathcal{N}(0, \sigma) + \theta\mathcal{N}(0, \sigma). \tag{6.8}$$

Since the odometry pose estimates are computed incrementally the whole later trajectory is affected when adding noise at a particular time step.

The results of the robustness evaluation with the corrupted odometry are shown in Fig. 6.19 together with the MSE of the corrupted odometry. These results show that the system is robust to substantial odometry errors. A failure case is shown in Fig. 6.20.

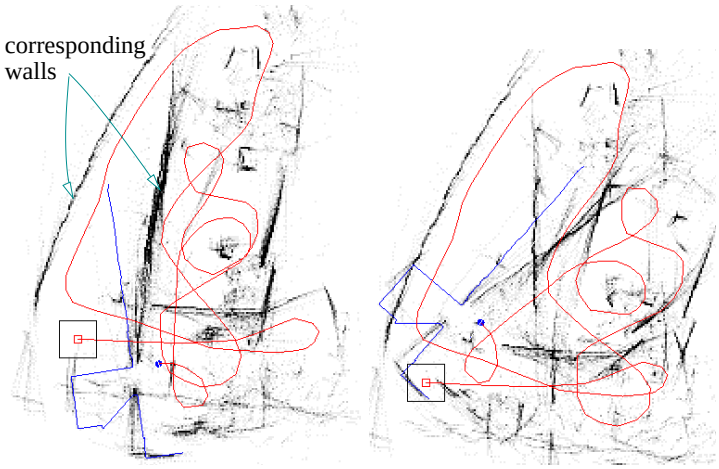

Figure 6.20: A failure case where the corrupted odometry error became too large resulting in a corrupted map. Left: SLAM map. Right: raw odometry.

6.4 Conclusions

Mini-SLAM combines the principle of using similarity of panoramic images to close loops at the topological level with a graph relaxation method to obtain a metrically accurate map representation and with a novel method to determine the covariance for visual relations based on visual similarity of neighbouring poses. The proposed method uses visual similarity to compensate for the lack of range information about local image features, avoiding computationally expensive and less general methods such as tracking of individual image features.

Experimentally, the method scales well to the investigated environments. The experimental results are presented by visual means (as occupancy maps rendered from laser scans and poses determined by the Mini-SLAM algorithm) and by comparison with ground truth (obtained from DGPS outdoors or laser-based SLAM indoors). The results demonstrate that the Mini-SLAM method is able to produce topologically correct and geometrically accurate maps at low computational cost. A simple extension of the method was used to fuse multiple data sets so as to obtain improved accuracy. The method has also been used without any modifications to successfully map a building consisting of 5 floor levels.

Mini-SLAM generates a 2-d map based on 2-d input from odometry. It is worth noting that the "outdoor / indoor" data set includes variations of up to 3 meters in height. This indicates that the Mini-SLAM can cope with violations of the flat floor assumption to a certain extent. We expect a graceful degradation in map accuracy as the roughness of the terrain increases. The

representation should still be useful for self-localization using 2-d odometry and image similarity, e.g. using the global localization method in [5], which in addition could be used to improve the robustness towards perceptual aliasing when fusing multiple data sets. In extreme cases, of course, it is possible that the method would create inconsistent maps, and a 3-d representation should be considered.

Part III

3D Vision Sensor Configuration

Chapter 7
Vision and 3D Laser Scanner Interpolation

This chapter addresses fusion of the input from a (planar) camera and a 3D laser scanner by means of vision-based interpolation of the low resolution, depth information obtained from the 3D range sensor. 3D range sensors, particularly 3D laser range scanners, enjoy a rising popularity and are used nowadays for many different applications. The resolution provided by 3D range sensors in the image plane is typically much lower than the resolution of a modern colour camera. In this chapter the focus lies on methods to derive a high-resolution depth image from a low-resolution 3D range sensor and a colour image. The main idea is to use colour similarity as an indication of depth similarity, based on the observation that depth discontinuities in the scene often correspond to colour or brightness changes in the camera image. Five interpolation methods are presented and compared with an independently proposed method based on Markov Random Fields. The proposed algorithms are non-iterative and include a parameter-free vision-based interpolation method. In contrast to previous work, ground truth evaluation with real world data and analysis of both indoor and outdoor data are presented. We further suggest and evaluate four methods to determine a confidence measure for the accuracy of interpolated range values.

7.1 Introduction

3D range sensors are getting more and more common and are found in many different areas. A large research area deals with acquiring accurate and very dense 3D models, where potential application domains include, for example, documenting cultural heritage [75], excavation sites and mapping of underground mines [111]. A lot of work has been done in which textural information obtained from a camera is added to the 3D data. For example, Sequeira et al. [104] present a system that creates textured 3D models of indoor envi-

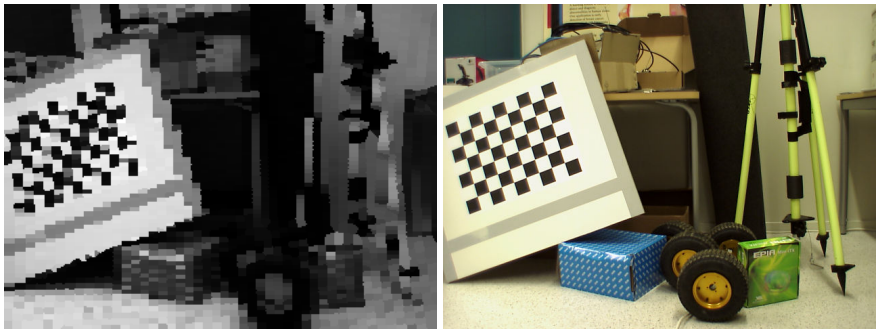

Figure 7.1: Left: Image intensities plotted with the resolution of the 3D scanner. The laser range readings were projected onto the right image and the closest pixel regions using Euclidean distance were set to the intensity of the projected pixel for better visualisation. Right: Calibration board used for finding the external parameters of the camera, with a chess board texture and reflective tape (grey border) used to locate the board in 3D using the remission / intensity values from the laser scanner.

ronments using a 3D laser range sensor and a camera. Früh and Zakhor [48] generate photo-realistic 3D reconstructions from urban scenes by combining aerial images with textured 3D data acquired with a laser range scanner and a camera mounted on a vehicle.

In most of the approaches that combine a range scanner and a camera, the vision sensor is not actively used during the creation of the model and is instead only used to add texture to the extracted model. An exception is the work by Haala and Alshawabkeh [55], in which the camera is used to add line features detected in the images into the created model.

To add a feature obtained with a camera to the point cloud obtained with a laser range scanner, it is required to find the mapping of the 3D laser points onto pixel coordinates in the image. If the focus instead lies on using the camera as an active source of information, which is the problem considered in this chapter, the fusion step also addresses the question of how to estimate a 3D position for each pixel or sub-pixel in the image. The resolution that the range sensor can provide is much lower than that sobtained with a modern colour camera. This can be seen by comparing Fig. 7.1, left, created by assigning the intensity value of the projected laser point to its closest neighbours, with the corresponding colour image in Fig. 7.1, right. See also Fig. 7.2 containing an image with projected laser range readings.

The only approach that uses colour information from a camera image to obtain a high-resolution 3D point model from a low-resolution 3D range scan seems to be the algorithm by Diebel et al. [27], where both colour information and the raw depth information are used. Their method is also compared with the methods proposed in this chapter and is further described in Section 7.3.

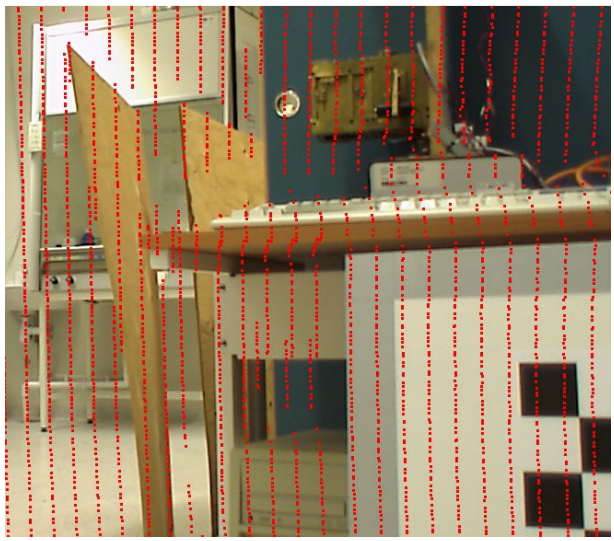

Figure 7.2: An image with projected laser range data. Note that the projected data are displayed with a 3×3 pixels rectangle to be more clearly visible. The 'gaps' in the scan are due to the physical displacement of approx. 0.2 meters between the origin of the laser and the origin of the camera. Basically the difference in origin means that the laser and the camera have slightly different views, which makes the angular resolution of the projected range data (shown in this figure) dependent on the actual range. For example, the camera origin is located forwards (and upwards) relative to the laser. This can especially be seen at the left/right side in the image when the projected range data changes horizontally with changing range values. Since the camera is mounted above the laser, gaps will also occur vertically. In addition, there are a few vertical 'gaps' due to missing range measurements.

7.2 Proposed Vision-based Interpolation Approaches

The main idea is to interpolate low-resolution range data provided by a 3D laser range scanner under the assumption that depth discontinuities in the scene often correspond to colour or brightness changes in the camera image of the scene.

For the problem under consideration, a set of N laser range measurements $\mathbf{r}_1..\mathbf{r}_N$ is given where each measurement $\mathbf{r}_i = (\theta_i, \pi_i, r_i)$ contains a tilt angle θ_i, a pan angle π_i and a range reading r_i corresponding to 3D Euclidean coordinates (x_i, y_i, z_i).

The image data consists of a set of image pixels $\mathbf{P}_j = (X_j, Y_j, C_j)$, where X_j, Y_j are the pixel coordinates and $C_j = (C_j^1, C_j^2, C_j^3)$ is a three-channel colour value. By projecting a laser range measurement $\mathbf{r}_i$ onto the image plane, a projected laser range reading $\mathbf{R}_i = (X_i, Y_i, r_i, (C_i^1, C_i^2, C_i^3))$ is obtained, which associates a range reading $\mathbf{r}_i$ with the coordinates and the colour of an image pixel.

The interpolation problem can now be stated, for a given pixel $\mathbf{P}_j$ and a set of projected laser range readings $\mathbf{R}$, as to estimate the interpolated range reading r_j^* as accurately as possible. Hence we denote a query point $\mathbf{R}_j^* = (X_j, Y_j, r_j^*, C_j^1, C_j^2, C_j^3)$.

Five different interpolation techniques are described in this section and compared with the MRF approach described in Section 7.3.

7.2.1 Nearest Range Reading (NR)

Given a pixel $\mathbf{P}_j$, the interpolated range reading r_j^* is assigned to the laser range reading r_i corresponding to the projected laser range reading $\mathbf{R}_i$ which has the highest likelihood p obtained as

$$p(\mathbf{P}_j, \mathbf{R}_i) \propto e^{-\frac{(X_j - X_i)^2 + (Y_j - Y_i)^2}{\sigma^2}}, \tag{7.1}$$

where σ is the point distribution variance. Hence, the range reading of the closest point (regarding Euclidean pixel distance) will be selected.

7.2.2 Nearest Range Reading Considering Colour (NRC)

This method is an extension of the NR method using colour information in addition. Given a pixel $\mathbf{P}_j$, the interpolated range reading r_j^* is assigned to the range value r_i of the projected laser range reading $\mathbf{R}_i$ which has the highest likelihood p obtained as

$$p(\mathbf{P}_j, \mathbf{R}_i) \propto e^{-\frac{(X_j - X_i)^2 + (Y_j - Y_i)^2}{\sigma_d^2} - \frac{||C_j - C_i||^2}{\sigma_c^2}}, \tag{7.2}$$

where σ_d and σ_c is the variance for the pixel point and the colour respectively.

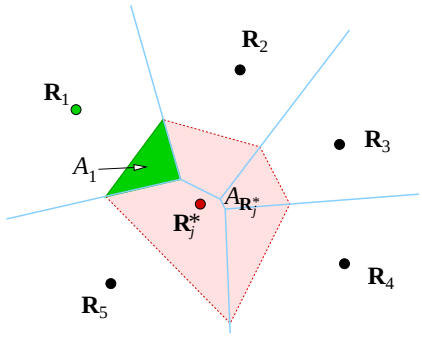

Figure 7.3: Natural neighbours $\mathbf{R}_1 ... \mathbf{R}_5$ of $\mathbf{R}_i^*$. The interpolated weight of each natural neighbour $\mathbf{R}_i$ is proportional to the size of the area which contains the point's Voronoi cell and the cell generated by $\mathbf{R}_j^*$. For example, the nearest neighbour $\mathbf{R}_1$ will have influence based upon the area of A_1.

7.2.3 Multi-Linear Interpolation (MLI)

Given a set of projected laser range readings $\mathbf{R}_1 ... \mathbf{R}_N$, a Voronoi diagram V is created by using their corresponding pixel coordinates $[X, Y]_{1...N}$. The natural neighbours NN to an interpolated point $\mathbf{R}_j^*$ are the points in V, which Voronoi cell would be affected if $\mathbf{R}_j^*$ is added to the Voronoi diagram, see Fig. 7.3. By inserting $\mathbf{R}_j^*$ we can obtain the areas $A_{1...n}$ of the intersection between the Voronoi cell due to $\mathbf{R}_j^*$ and the Voronoi cell of $\mathbf{R}_i$ before inserting $\mathbf{R}_j^*$ and the area $A_{\mathbf{R}_j^*}$. The areas $A_{1...n}$ are used to compute a normalisation factor. The weight of the natural neighbour $\mathbf{R}_i$ is calculated as

$$w_i(\mathbf{R}_j^*) = \frac{A_i}{A_{\mathbf{R}_j^*}}. \tag{7.3}$$

The interpolated range reading r_j^* is then calculated as

$$r_j^* = \sum_{i \in NN(\mathbf{R}_j^*)} w_i r_i. \tag{7.4}$$

This interpolation approach is linear [106]. One disadvantage is that nearest neighbourhood can only be calculated within the convex hull of the scan-points projected onto the image. However, this is not considered as a problem since the convex hull encloses almost the whole image, see Fig. 7.4, bottom left.

7.2.4 Multi-Linear Interpolation Considering Colour (LIC)

To fuse colour information with the MLI approach introduced in the previous subsection, the areas $A_{\mathbf{R}_i}$ and $A_{\mathbf{R}_j^*}$ are combined with colour weights $w^c_{1...n}$ for each natural neighbour based on spatial distance in colour space.

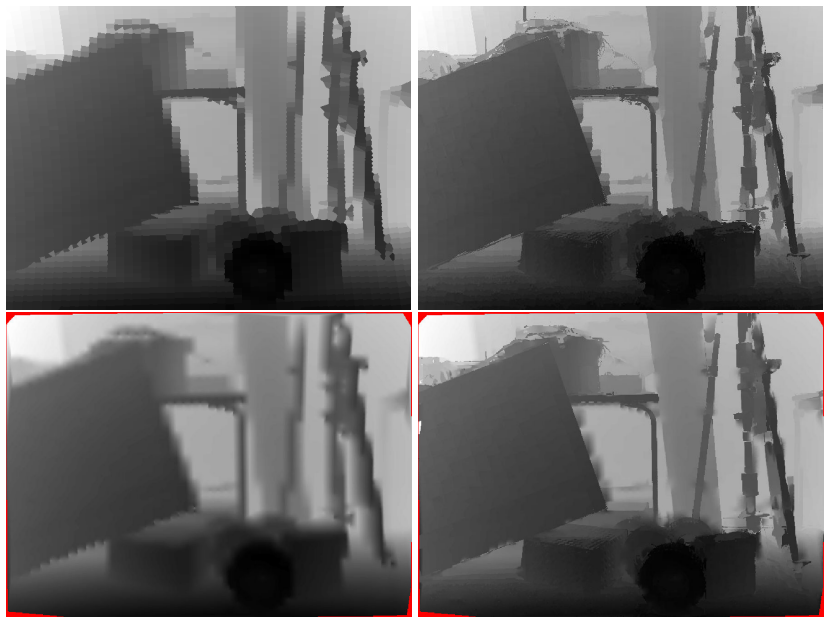

Figure 7.4: Top left: Depth image generated with the NR method. Top right: Depth image generated with the NRC method, small details are now visible. Note that a depth image generated from a similar viewpoint as the laser range scanner makes it very difficult to see flaws of the interpolation algorithm. Bottom left: MLI. Bottom right: LIC.

Similar as in Section 7.2.2, a colour variance σ_c is used to compute a colour weight w_i^c as

$$w_i^c(\mathbf{R}_j^*) = e^{-\frac{||C_i - C_j||^2}{\sigma_c^2}}. \tag{7.5}$$

The colour based interpolated range reading estimation is then done with

$$r_j^* = \sum_{i \in NN(\mathbf{R}_j)} \frac{w_i w_i^c}{W^c} \tag{7.6}$$

where $W^c = \sum_{i=1}^{n} w_i^c$ is used as a normalisation factor.

7.2.5 Parameter-Free Multi-Linear Interpolation Considering Colour (PLIC)

One major drawback of the methods presented so far and the approach presented in Section 7.3 is that they depend on parameters such as σ_c. To avoid the need to specify colour variances, the intersection area $A_{\mathbf{R}_i}$ defined in Section 7.2.3 is used to compute a colour variance estimate for each nearest neighbour point $\mathbf{R}_i$ as

$$\sigma_{c_i} = \frac{1}{n_i - 1} \sum_{j \in A_i} \|\mu_i - C_j\|^2 \tag{7.7}$$

where $\mu_i = \frac{1}{n_i} \sum_{j \in A_i} C_j$ and n_i is the number of pixel points within the region A_i. σ_{c_i} is then used in Eq. 7.5.

This results in an adaptive adjustment of the weight of each point. In the case of a large variance of the local surface texture, colour similarity will have less impact on the weight w_i^c.

7.3 Related Work

To our knowledge, the only work using vision for interpolation of 3D laser data is [27] where a Markov Random Field (MRF) framework is used. The method works by iteratively minimising two cost functions: ψ stating that the raw laser data and the surrounding estimated depths should be similar and ϕ stating that the depth estimates close to each other with a similar colour should also have similar depths. The first constraint is obtained as

$$\psi = \sum_{i \in N} k(r_i^* - r_i)^2 \tag{7.8}$$

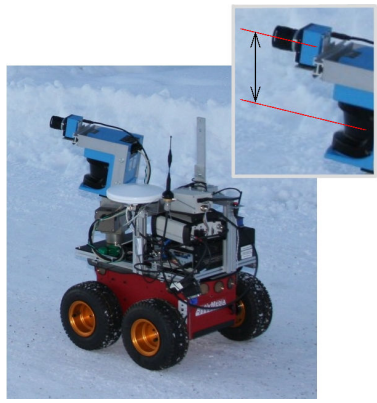

Figure 7.5: The robot Tjorven with a close up part which shows the displacement between the camera and the laser, which causes parallax errors.

where k is a constant and the sum runs over the set of N positions which contain a laser range reading r_i and r_i^* is the interpolated range reading for position i. The second constraint is obtained as

$$\phi = \sum_i \sum_{j \in \mathcal{N}(i)} e^{(-c||C_i - C_j||^2)} (r_i^* - r_j^*)^2, \tag{7.9}$$

where c is a constant, C is the pixel colour and $\mathcal{N}(i)$ are the neighbourhood pixels around position i.

The function to be minimised is the sum $\psi + \phi$.

7.4 Evaluation

The experimental evaluation was performed using both simulated and real data. All data sets D were divided into two equally sized parts D_1 and D_2. One dataset, D_1, is used for interpolation and D_2 is used as the ground truth where each laser range measurement is projected onto image coordinates. Hence for each ground truth point $\mathbf{R}_i$ we have the pixel positions $[X, Y]_i$ and the range r_i. The pixel position $[X, Y]_i$ is used as input to the interpolation algorithm and the range r_i is used as the ground truth. The performance of the interpolation algorithms is analysed based on the difference between the interpolated range r_i^* and the range r_i from the ground truth.

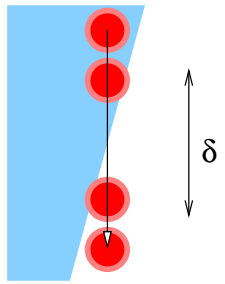

Figure 7.6: When the laser range finder spot covers an area which contains different depths (blue and white areas), the range reading returned might be unreliable and vary anywhere between the closest to the furthest range (shown as the region δ).

7.5 Experimental Setup

7.5.1 Hardware

The scanner used is a 2D SICK LMS-200 mounted together with a 1 megapixel (1280x960) colour CCD camera on a pan-tilt unit from Amtec. The sensors are more thoroughly described in Section 2.1.2. The displacement between the camera's optical axis and the laser is approx. 0.2 m, see Fig. 7.5. The SICK scanner has a larger spot size compared to many other laser scanners and often gives wrong range estimates close to edges where the laser spot covers multiple objects at different distances, see Fig. 7.6. Of course, this flaw of the sensor will be reflected in the ground truth data as well. The angular resolution of the laser scanner is 0.5 degrees. Half of the readings were used as ground truth, so the resolution of the points used for interpolation is 1 degree. The vertical resolution depends on the wrist movements but iss rescaled to approx. 0.5 degrees.

The camera displacement was determined using the calibration procedure found in Appendix B.

7.6 Results - Interpolation

7.6.1 Colour spaces investigated

The most common colour spaces were compared to evaluate whether better illuminance/shading invariance could be useful. The colour spaces compared were standard RGB, Normalised RGB $(r/(r+g+b), g(r+g+b), b/(r+g+b))$, HSV and YUV. In HSV the V component and for YUV the Y component is set to a constant, see Fig. 7.7. Since a consistent improvement could not be

Figure 7.7: Left : HSV colour space, where the V component is set to a constant. Right : YUV, where the Y component is set to a constant, colour-space. The RGB colour space can be found in Fig. 7.1.

observed for any of the colour spaces tested, only results based on standard RGB normalised to [0,1] are presented in this chapter.

In all experiments the colour variance $\sigma_c = 0.05$ and the pixel distance variance $\sigma_d = 10mm$ were used, which were found empirically. The parameters used within the MRF approach described in Section 7.3 were obtained by extensive empirical testing and were set to $k = 2$ and $C = 10$. The optimisation method used for the MRF method was the conjugate gradient method described in [97] and the initial depths were estimated with the NR method. In all experiments the full resolution (1280x960) of the camera image was used.

7.6.2 Simulated Data

The simulated data are shown in Fig. 7.8, which were created from a model based on a set of coloured planes. Each point was obtained by finding the intersection between the simulated scanner ray and the model. The simulated scans show the benefits of using the distance and colour for interpolation, see Table 7.1. By using colour information the selection of interpolation points is improved. In the results the LIC method gives the lowest mean error. However the NRC gives the lowest maximum error which is likely to have been caused by an overestimation of the colour variance in LIC meaning that scan points with differing colour too have to much influence on the interpolation.

This would be avoided if the colour variance estimation were to be used (PLIC). Note that both PLIC and MRF requires a full image and are therefore not considered. Since the simulated environment consists only of planes, the multi-linear approaches work fairly well.

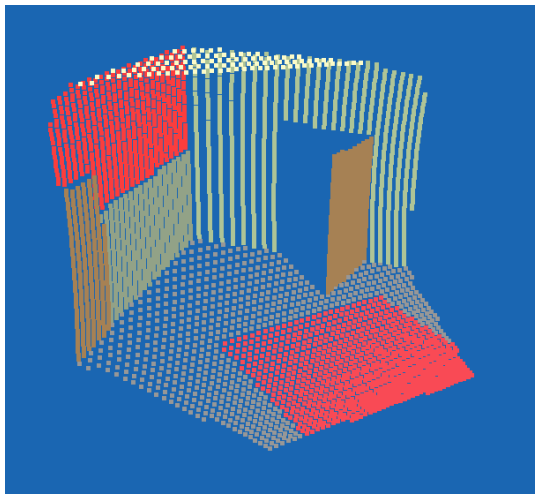

Figure 7.8: The simulated 3D scan, which provided 4165 scan points as ground truth data.

Table 7.1: Distance error using the simulation data.

	NR	NRC	LI	LIC
mean	0.039	0.028	0.017	***0.007***
max	1.685	***0.198***	1.112	1.095
σ^2	0.120	0.020	0.096	0.028

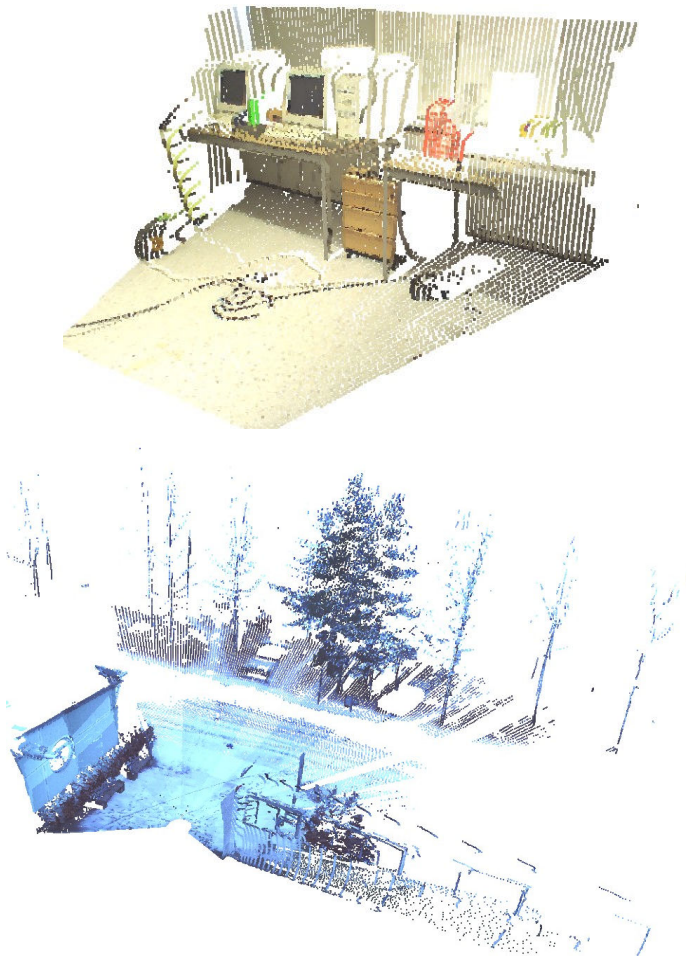

Figure 7.9: Left: The third indoor evaluation scan, Indoor_3. Right: Scans taken in winter time with some snow, $\mathrm{Outdoor}_1 - \mathrm{Outdoor}_3$.

Table 7.2: Results from Indoor_1, Indoor_2 and Indoor_3 data sets.

	NR	NRC	MLI	LIC	PLIC	MRF
$\bar{e}$	0.065	0.054	0.052	***0.048***	0.049	***0.048***
$o_{0.1}$	0.161	***0.117***	0.149	0.118	0.123	0.136
$o_{0.2}$	0.112	0.083	0.069	0.076	0.077	***0.063***
$o_{0.5}$	0.034	0.029	0.016	0.022	0.023	***0.012***
$o_{1.0}$	0.000	0.000	0.000	0.000	0.000	0.000
$o_{3.0}$	0.000	0.000	0.000	0.000	0.000	0.000
$\bar{e}$	0.123	0.134	0.109	0.107	0.109	***0.106***
$o_{0.1}$	0.148	0.143	0.172	***0.140***	0.149	0.154
$o_{0.2}$	0.095	0.097	0.108	***0.090***	0.092	0.094
$o_{0.5}$	0.056	0.068	0.050	0.051	0.053	***0.047***
$o_{1.0}$	***0.013***	0.034	0.026	0.028	0.027	0.025
$o_{3.0}$	0.006	0.006	***0.004***	***0.004***	***0.004***	***0.004***
$\bar{e}$	0.088	0.072	0.067	***0.060***	***0.060***	0.067
$o_{0.1}$	0.109	***0.096***	0.143	0.110	0.107	0.132
$o_{0.2}$	0.080	***0.071***	0.097	0.072	***0.071***	0.093
$o_{0.5}$	0.061	0.048	***0.021***	0.036	0.034	0.031
$o_{1.0}$	0.011	0.010	0.008	***0.007***	0.009	0.009
$o_{3.0}$	0.004	0.002	0.002	0.002	***0.001***	0.003

7.6.3 Experimental Data

All the interpolation algorithms described in this section were tested on real data consisting of three indoor scans and three outdoor scans, see Fig. 7.9. The outdoor scans were taken in winter time with snow, which presents the additional challenge that most of the points in the scene have very similar colours.

The results are summarised in Tables 7.2 and table 7.3, which show the mean error with respect to the ground truth $\bar{e}$, and the percentage of outliers o_t for different thresholds t. The percentage of outliers is the percentage of points for which the interpolated range value deviates from the ground truth value by more than a threshold t (specified in meters in Tables 7.2 and 7.3).

For the indoor data sets, which comprise many planar structures, the lowest mean error was found with the multi-linear interpolation methods, particularly LIC and PLIC, and MRF interpolation. LIC and PLIC produced fewers (but larger) outliers.

With the outdoor data the results obtained were more diverse. For the data set $Outdoor_1$, which contains some planar structures, a similar result as in the case of the indoor data was observed. For data sets with a very small portion of planar structures, such as $Outdoor_2$ and $Outdoor_3$, the mean error was generally much higher and the MRF method performed slightly better compared to the multi-linear interpolation methods. This is likely to be due to the absence of planar surfaces and the strong similarity of the colours in the image recorded at winter time. It is noteworthy that in this case, the nearest neighbour interpola-

Table 7.3: Results from Outdoor_1, Outdoor_2 and Outdoor_3 data sets.

	NR	NRC	MLI	LIC	PLIC	MRF
$\bar{e}$	0.067	0.068	0.056	0.059	***0.054***	***0.054***
$o_{0.1}$	0.147	0.160	0.156	0.146	***0.138***	0.150
$o_{0.2}$	0.076	0.080	0.078	0.073	***0.068***	0.076
$o_{0.5}$	0.032	0.032	0.016	0.020	***0.015***	0.016
$o_{1.0}$	0.005	0.002	0.001	***0.001***	0.002	***0.001***
$o_{3.0}$	***0.000***	***0.000***	0.001	0.001	***0.000***	***0.000***
$\bar{e}$	0.219	0.294	0.235	0.322	0.275	***0.218***
$o_{0.1}$	0.196	0.240	0.242	0.269	0.264	***0.187***
$o_{0.2}$	***0.096***	0.152	0.140	0.168	0.160	0.098
$o_{0.5}$	***0.047***	0.088	0.077	0.094	0.083	0.051
$o_{1.0}$	0.036	0.057	0.043	0.059	0.049	***0.030***
$o_{3.0}$	***0.016***	0.023	0.019	0.028	0.022	0.017
$\bar{e}$	0.526	0.584	0.522	0.574	0.500	***0.498***
$o_{0.1}$	***0.222***	0.232	0.296	0.258	0.268	0.242
$o_{0.2}$	***0.157***	0.170	0.224	0.193	0.205	0.181
$o_{0.5}$	***0.102***	0.115	0.156	0.125	0.130	0.106
$o_{1.0}$	0.078	0.085	0.091	0.083	0.086	***0.067***
$o_{3.0}$	0.027	0.029	0.029	0.030	***0.026***	***0.026***

tion method *without* considering colour (NR) performed as well as MRF. The interpolation accuracy of the parameter-free PLIC method was always better or comparable to the parameterised method LIC.

7.7 Confidence Measure

The interpolated range reading r_j^* may be a good estimate of the actual range or it might deviate substantially from the true value. Therefore a confidence measure for the correctness of the interpolated range reading estimate is desirable, allowing to detect and handle erroneous measures appropriately.

In this section four different confidence measures were proposed and evaluated.

7.7.1 Proximity to the Nearest Laser Range Reading (NLR)

This confidence measure is based on the distance between the pixel position of the interpolated point $\mathbf{R}_j^*$ and the nearest projected laser range reading $\mathbf{R}_i$. The idea is that if the interpolated pixel point is close to a point where a range measurement is available the interpolation is considered more trustworthy.

$$\text{NLR}(\mathbf{R}_j^*, \mathbf{R}_i) = e^{-\sqrt{(X_j^* - X_i)^2 + (Y_j^* - Y_i)^2}} \tag{7.10}$$

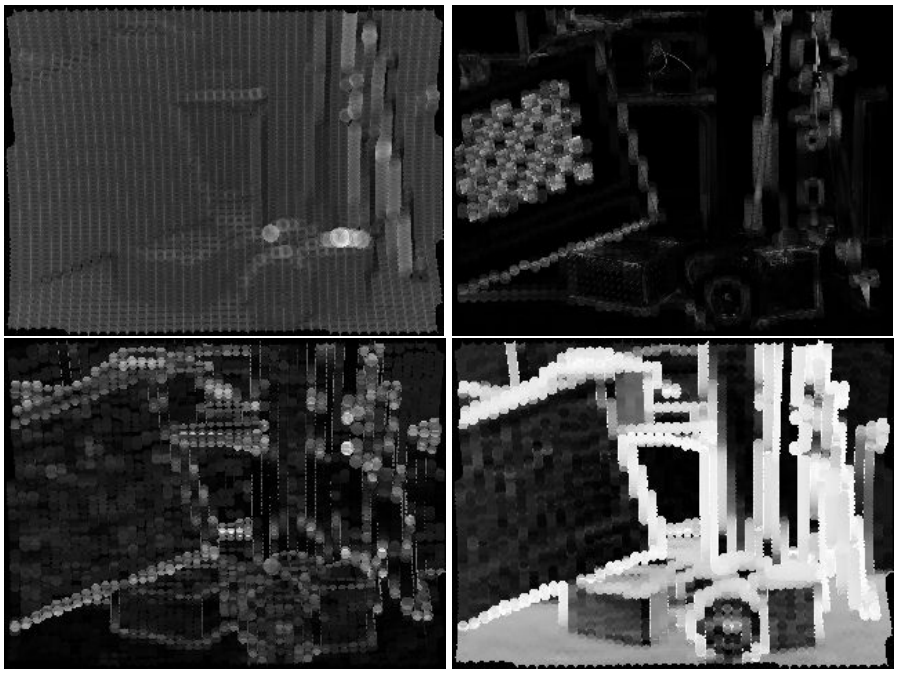

Figure 7.10: Visualisation of the confidence measures proposed. From left to right: NLR indicating the distance to the closest point, NLRC indicating colour distance, PS indicating the plane factor of the neighbourhood of the interpolated point and AON indicating the angle difference between the normal of the extracted local plane and the camera axis. The parameter free method (PLIC) was used for interpolation.

7.7.2 Proximity to the Nearest Laser Range Reading Considering Colour (NLRC)

This confidence measure is based on the distance between the colour of the pixel of the nearest projected laser range reading $\mathbf{R}_i$ and the colour of $\mathbf{R}_i^*$. The NLRC confidence measure is based on the principle that the confidence value should decrease if the two points have different colour.

$$\mathrm{NLRC}(\mathbf{R}_j^*, \mathbf{R}_i) = e^{-||C_j - C_i||} \tag{7.11}$$

7.7.3 Degree of Planar Structure (PS)

Our confidence in the range interpolation also depends on how well a planar surface can be fitted to the local neighbours $\mathrm{NN}(\mathbf{R}_j^*)$ of the interpolation point $\mathbf{R}_j^*$ since planar surfaces support a linear interpolation technique very well. The neighbours are either determined from the grid defined by the projected laser range readings or the nearest neighbours found in the Voronoi tessellation. The parameters of the planar surface are obtained from the 3D covariance matrix of $\mathrm{NN}(j)$ where the two main eigenvectors are extracted, which span a planar surface S_j with the normal vector $\mathbf{n}_j$. The confidence measure is then calculated from the average distance of the local neighbours to the fitted plane as

$$\mathrm{PS}(\mathbf{R}_j^*) = e^{-\frac{1}{NN}\sum_{i \in NN(\mathbf{R}_j^*)} ||\mathbf{r}_i \cdot \mathbf{n}_j - d_j||}, \tag{7.12}$$

where d_j is the distance of the plane S_j to the origin and $\mathbf{r}_i = (x_i, y_i, z_i)$ is the 3D position of point i.

7.7.4 Angle Between the Optical Axis and the Fitted Plane Normal (AON)

This confidence measure considers the orientation of the planar surface S_j described in the previous section relative to the optical axis of the camera $\mathbf{z}_{cam}$. If the angle between the normal vector $\mathbf{n}_j$ and the optical axis is small, the confidence should be high since we expect only one reflection from the laser scanner and the displacement between the laser and the camera will have a negligible impact.

$$\mathrm{AON}(\mathbf{R}_j^*) = \mathbf{z}_{cam} \cdot \mathbf{n}_j \tag{7.13}$$

7.8 Result - Confidence Measure

With the exception of the NLR method, a distinct negative correlation was found for all the confidence measures proposed in this section, see Fig. 7.11. Due to the experimental setup where the evaluation points were taken from

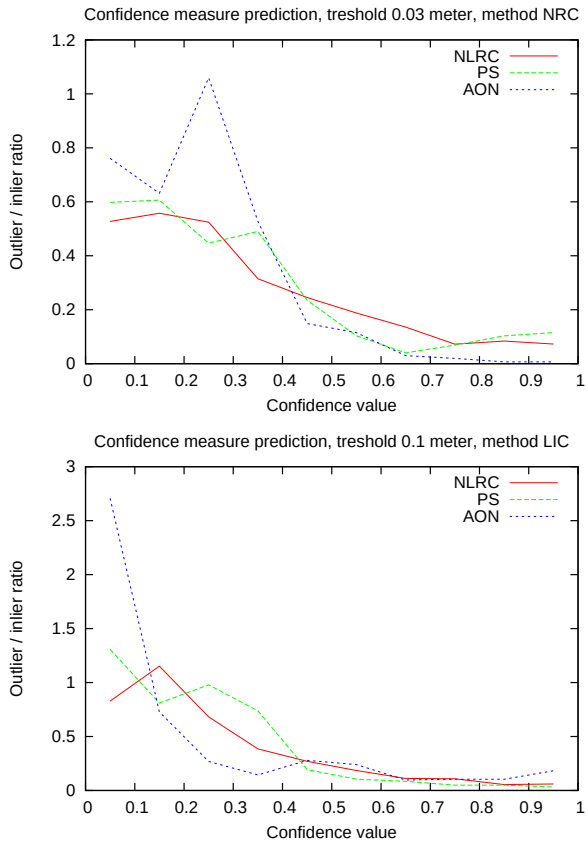

Figure 7.11: Behaviour of the confidence measures introduced in this chapter. The graphs show the ratio of the number of outliers / number of inliers, depending on the confidence in the interpolated points. All points with a depth error > 0.03 meter are considered outliers in the upper image and in the lower graph the threshold was 0.1 meter. Top: Indoor_1 data set with method NRC. Bottom: $\mathrm{Outdoor}_2$ with method LIC.

the laser scanner in an evenly spaced grid, meaning that the distance to the closest neighbour for all evaluation points in the grid should be the same and in fact each evaluation point should have at least two neighbours with the exact same distance. However, the parallax errors caused by the displacement, see Fig. 7.5, will move the evaluation point and therefore result in a lower distance between $\mathbf{R}_j^*$ and $\mathbf{R}_i$, which made the proposed NLR method give high confidence correlated with large parallax errors. A large parallax error indicates a large depth difference compared to its neighbour's depth values, which instead should indicate a low confidence. However, in a non-evenly spaced evaluation grid, this method seems likely to give good confidence measures.

Figure 7.11 shows the inlier/outlier ratio depending on the confidence calculated with the NLRC, PS, and AON methods. Interpolated range values were classified as outliers if the deviation from the ground truth value was larger than a third of the mean error obtained with the particular interpolation method. The same general trend of a clear negative correlation, however, was observed with all interpolation methods and for all data sets.

7.9 Conclusions

This chapter is concerned with methods for deriving a high-resolution depth image from a low-resolution 3D range sensor and a colour image. We suggest five interpolation methods and compare them with an alternative method proposed by Diebel and Thrun [27]. In contrast to previous work, we present ground truth evaluation with simulated and real world data and analyse both indoor and outdoor data. The results of this evaluation do not allow us to single out one particular interpolation method that provides a distinctly superior interpolation accuracy, indicating that the best interpolation method depends on the content of the scene. Altogether, the MRF method proposed in [27] and the PLIC method proposed in this chapter provided the best interpolation performance. While providing basically the same level of interpolation accuracy as the MRF approach, the PLIC method has the advantage that it is a parameter-free and non-iterative method, i.e. that a certain processing time can be guaranteed. Another advantage of the proposed methods compared to the MRF method is that depth estimates can be obtained without calculating a full depth image. For example, if interpolation points are extracted in the image using a vision-based method (i.e. feature extraction), we can directly obtain a depth estimate for each feature, which is used in the registration method in Chapter 8.

In addition four methods to determine a confidence measure for the accuracy of interpolated range values are proposed and evaluated. Three of the proposed confidence values showed a distinct negative correlation with the occurrence of outliers. This was observed independent of the scene content and the interpolation method applied.

Chapter 8
Vision-aided 3D Laser Scanner Registration

This chapter describes a novel registration approach that is based on a combination of visual and 3D range information. To identify correspondences, local visual features are obtained from the images of a standard color camera and the depth of these features is determined from the range measurements of a 3D laser scanner. The range measurements are also used to estimate the position covariance of the visual features. To exploit these covariance estimates, the registration constraint is based on the Mahalanobis distance between the corresponding visual features. Experimental results are presented in both outdoor and indoor environments.

8.1 Introduction

Registration or scan-matching is a popular approach in robotics to obtain relative pose estimates, and as such a core component of many SLAM algorithms. Most work published in the past considers 2D-motion in an indoor environment, however, nowadays more attention is directed towards complete 6DOF methods.

Since vision is particularly suited to solve the correspondence problem (data association), vision-based systems have been applied as an addition to laser scanning based SLAM approaches for detecting loop closing. This principle has been applied to SLAM systems based on a 2D laser scanner [57] and a 3D laser scanner [92]. When using registration methods which rely on a weaker criterion for correspondence, i.e. point to point distance as in [12], a good initial estimate is very important for the robustness of the system. By instead using the strong correspondences visual features can provide, a good initial estimate is not necessary [92]. It is further argued that vision can enable solutions in highly cluttered environments where pure laser range scanner based methods fail [99].

This chapter presents a registration method which relies on the sensory configuration described in Section 2.1.2, which utilises a 2D SICK LMS-200 laser range finder mounted together with a 1 megapixel (1280x960) colour CCD camera on a pan-tilt unit from Amtec. The key aspect is to utilise the strong visual correspondences with the depth accuracy obtained from the laser scanner. The benefits of using vision come at almost no extra cost since a camera is much less expensive than a 3D laser scanner.

8.2 Related Work

To combine the discriminant property of local visual features with a 3D laser scanner has be utilised in a related approach by Newman et al. [92], where SIFT features were used to detect loop closure events in a 3D SLAM approach. In contrast to their method where SIFT features are used to obtain an initial pose estimate (by determining the essential matrix between two images) and the full point cloud is considered afterwards, registration in our approach is carried out using only 3D points that are associated with matching visual features.

Since this work incorporates both visual and 3D laser information there is some overlap in the proposed method compared to approaches using only one of the sensor modalities. Methods that utilise 3D laser data are commonly based on the ICP algorithm [93, 115]. Another 3D laser based approach is the 3D-NDT method by Magnusson et al. [84]. Common to all 3D laser based registration methods is that the output consists of a relative position between two scan poses, and in addition, that initial estimates are available. In 'pure' vision based solutions, based on multiple-view geometry, the problem of determining the relative pose also includes determining the scale of the motion (in translation) [56]. One approach to determine the scale is to use a predefined pattern with known geometrical properties as part of the first image [25]. However, unless an object with known geometrical properties is shown again, this gives problems with scale drift. Another commonly used approach is to have multiple cameras and uses triangulation to get a depth estimate for each visual feature as, for example, in [102]. To obtain a relative pose, estimation using stereo images has, for example, been done using Colour ICP [64], an ICP version which also incorporates the colour as part of the distance function. In [90], the standard ICP method is combined with a constraint based on the optical flow. What seems to be the standard nowadays, compared to generatings dense stereo images that are matched using ICP based methods, is to use visual features that are tracked over several images. By using multiple images (i.e. a sequence of images), and not only images from two poses, the depth estimate of each visual feature will improve [56]. To use multiple images taken at different poses to improve the position of each visual feature and at the same time determine the pose of each image is directly related to the SLAM problem or, if the number of frames to search for corresponding features is limited, related to visual odometry, for example, used in the Mars rovers [85]. In addition, many approaches

that use a sequence of images relys on initial pose estimates from, for example, odometry [85, 10, 62, 66].

However, to directly apply the registration using the visual features together with an estimated position and position covariance obtained from a 3D laser scanner without any requirement of initial pose estimates has to our knowledge not yet been exploited.

8.3 Method

The proposed approach is based on local visual features for which the depth and position covariance are estimated. The visual feature used here is the SIFT feature described in Section 3.2.1. The position covariance for visual features is obtained from the laser range measurements surrounding the visual feature location. For example, if the detected feature is located on a poster, a planar surface, the feature's position covariance will be smaller (especially perpendicular to the surface) compared to a feature extracted from a branch.

As stated in the previous section, most current approaches to scan registration depend on reasonably accurate initial pose estimates. In the proposed method, the correspondences are solely determined by the visual features and not from spatial distance. As a result, no initial pose estimates is required. Therefore, no initial pose estimate is required.

The registration procedure can be described as follows: first, SIFT features are computed in the planar images recorded within the current scan data $\mathcal{S}_c$ and compared to the SIFT features found in the images belonging to previous scan $\mathcal{S}_p$ using the feature matching scheme (Sec. 3.3). Next, depth values r^* are estimated for all matching feature pairs P in $\mathcal{S}_p$ and $\mathcal{S}_c$, using the Nearest Range Reading (NR) method (Sec. 7.2.1), i.e. using the closest projected 3D laser point. Pairs of 3D points corresponding to matching feature pairs P are then used together with the feature position covariance to obtain the relative pose estimate between $\mathcal{S}_p$ and $\mathcal{S}_c$. The relative pose is expressed as $(\mathbb{R}, \mathbf{t})$ where $\mathbb{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector (see Sec. 8.3.5).

8.3.1 Estimating the Visual Feature Depth

To obtain a 3D position estimate of an extracted feature $F = [X, Y], H$, where $[X, Y]$ is the pixel position of the feature, an interpolation step is performed using the NR method, described in Section 7.2.1, to obtain an estimated range reading r^*. By back-projecting the estimated range r^* using the pixel coordinates $[X, Y]$ the 3D position $\mu_F = (x, y, z)$ of the feature F is obtained.

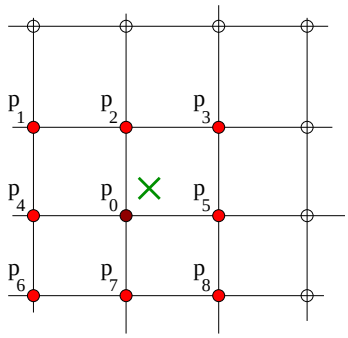

Figure 8.1: Laser points used to estimate the covariance. The $\times$ represents a visual feature. Circles represent range readings, where the filled dots $p_{0..M}$ represent range readings used to obtain the covariance estimate. The central filled dot p_0 represents the laser point from which the depth of the visual feature was determined. The horizontal lines indicate the 2D laser reading scanning plane and the vertical lines the tilt movement of the wrist.

8.3.2 Estimating the Visual Feature Covariance

To obtain a covariance of each visual feature point C_F, the closest projected laser point p_0 relative to the visual feature F in the image plane is used together with M surrounding laser points $p_{1..M}$. The covariance C_F is then calculated as

$$C_F = \frac{1}{M-1} \sum_{i=0}^{M} (p_i - \mu)^2, \tag{8.1}$$

where $\mu = \frac{1}{M} \sum_{i=0}^{M} p_i$. In our experimental evaluation we used $M = 8$, see Fig. 8.1.

8.3.3 Rigid Iterative Closest Point

The iterative closest points (ICP) algorithm [12, 23], finds a rigid body transformation $(\mathbb{R}, \mathbf{t})$ between two scan poses $\mathcal{S}_p$ and $\mathcal{S}_c$ by minimising the following constraint

$$J(\mathbb{R}, \mathbf{t}) = \sum_{i=1}^{N} \|p_i^c - \mathbb{R}p_i^p - \mathbf{t}\|^2, \tag{8.2}$$

where p_i^p and p_i^c are the corresponding (closest) points from scan poses $\mathcal{S}_p$ and $\mathcal{S}_c$. The selection of the corresponding pairs in the standard version of ICP is done by using a distance metric to search for the closest point. This search is the most time consuming part of the algorithm, and to decrease the search time a common approach is to use a Kd-tree [44]. ICP as well as other least squares

methods assumes that the measurements contain an identical and independent Gaussian noise.

To obtain the rigid transformation that minimises the above equation, there exist various closed-form solutions. In our approach we have adopted the singular value decomposition method proposed by Arun et al. [8].

In our approach, the correspondence is detected using visual features, i.e. an exhaustive search is not required in the spatial domain. In addition, since the proposed method relies on a vision based approach, the assumption of identical and independent noise of the feature point is a problematic approximation as discussed below.

8.3.4 Rigid Generalised Total Least Squares ICP

Generalised Total Least Square ICP (GTLS-ICP) has been proposed by San-Jose et al. [101] as an extension of ICP. This method is similar to standard ICP but also incorporates a covariance matrix for each point. Instead of minimising Eq. 8.2, GTLS-ICP utilises the following function:

$$J(\mathbb{R},\mathbf{t}) = \sum_{i=1}^{N} (q_i - p_i^c)^T C_{q_i}^{-1} (q_i - p_i^c) + \sum_{i=1}^{N} (p_i^c - q_i)^T C_{p_i^c}^{-1} (p_i^c - q_i), \quad (8.3)$$

where $q_i = \mathbb{R}p_i^p + \mathbf{t}$. The covariance matrix C_{q_i} is obtained by rotating the eigenvectors of the covariance matrix $C_{p_i^p}$, obtained from Eq. 8.1, with the rotation matrix $\mathbb{R}$. However, there is no closed-form solution to minimise this function and the method instead iteratively estimates the rigid body transformation $\mathbb{R}$ and $\mathbf{t}$. In our implementation we first use the standard ICP method (Sec. 8.3.3) and after convergence then apply a conjugate gradient method to minimise Eq. 8.3.

8.3.5 Rigid Trimmed Extension

Since visual features are used to establish corresponding scan points, no further means of data association, (such as searching for closest data points in ICP) is necessary. Although the SIFT features were found to be very discriminative (see for example [87]), there is of course still a risk that some of the correspondences are not correct. To further decrease the possibility of erroneous point associations, only a set fraction of the correspondences with the smallest spatial distance between corresponding points is used for registration. In the experiments presented in this chapter the fraction was set to 70%. Because the fraction of data points that is used to estimate the relative pose $[\mathbb{R},\mathbf{t}]_t$ between two scans depends on the previous estimate $[\mathbb{R},\mathbf{t}]_{t-1}$ (since the relative pose estimate affects the spatial distance between corresponding points), the minimisation needs to be applied in an iterative manner. Thus relative pose

updates are calculated repeatedly with the minimisation using the previous estimate $[\mathbb{R}, \mathbf{t}]_{t-1}$ as input to the next iteration step until a stopping criterion is met. Any initial pose estimate can be used (in the experiment presented in this work, the method always start with the assumption of an identical pose). The stopping criterion used is the experiments in this chapter is the change of the mean squared error (MSE) of the spatial distance between the corresponding points compared to the previous iteration. The optimisation was stopped if the difference was less than 10^{-6} m^2.

Note that the spatial distance between the corresponding points is used also together with the covariance based ICP method to select the 70% fraction of the corresponding points. Otherwise points with a large covariance tend to be selected as members of the 70% fraction with the smallest Mahalanobis distance, which was found to decrease registration accuracy.

8.4 Setup and Experimental Results

8.4.1 Data Collection

The scanner used is a 2D SICK LMS-200 mounted together with a 1 megapixel (1280x960) colour CCD camera on a pan-tilt unit from Amtec. The sensors are mounted on the mobile robot Tjorven (Fig. 2.2) and the complete system are more thoroughly described in Section 2.1.2. The robot is manually driven around and was stopped for data collection. For each robot pose, 3D range and image data were collected as follows. First, three sweeps are carried out with the laser scanner at -60, 0 and 60 degrees relative to the robot orientation (horizontally). During each of these sweeps, the tilt of the laser scanner is continuously shifted from -40 degrees (looking up) to 30 degrees (looking down). After the three range scan sweeps, seven camera images were recorded at -90, -60, -30, 0, 30, 60, and 90 degrees relative to the robot orientation (horizontally) and at a fixed tilt angle of -5 degrees (looking up). The full data set acquired at a single scan pose is visualised in Fig. 8.2. The angular resolution of the laser scanner was set to 0.25 degrees with a 100 degrees field of view.

8.4.2 Indoor Experiment

To evaluate the registration performance, a data set "$registration - indoor$" consisting of 22 scan poses, i.e. from 66 laser scanner sweeps and 154 camera images (as described in Section 8.4.1) was collected in an indoor lab environment. The first scan pose and the last scan pose were collected at a similar position. An example of the registration result can be seen in Fig. 8.3.

The performance metric chosen to evaluate the registration method is the translation and angular distance between the final pose estimated from the registration and the ground truth final pose. Since the first and the last scan pose were taken at a similar position, the ground truth was determined by matching

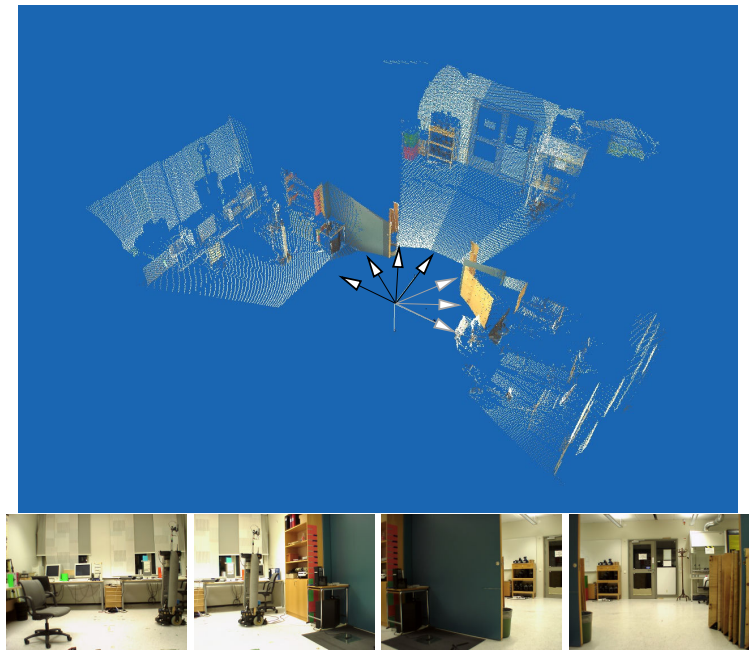

Figure 8.2: Top: Full data set acquired for a single scan pose comprising three sweeps with the laser scanner fused with colour information from seven camera images, where the first four images (marked with dark arrows in the top figure) are shown at the bottom.

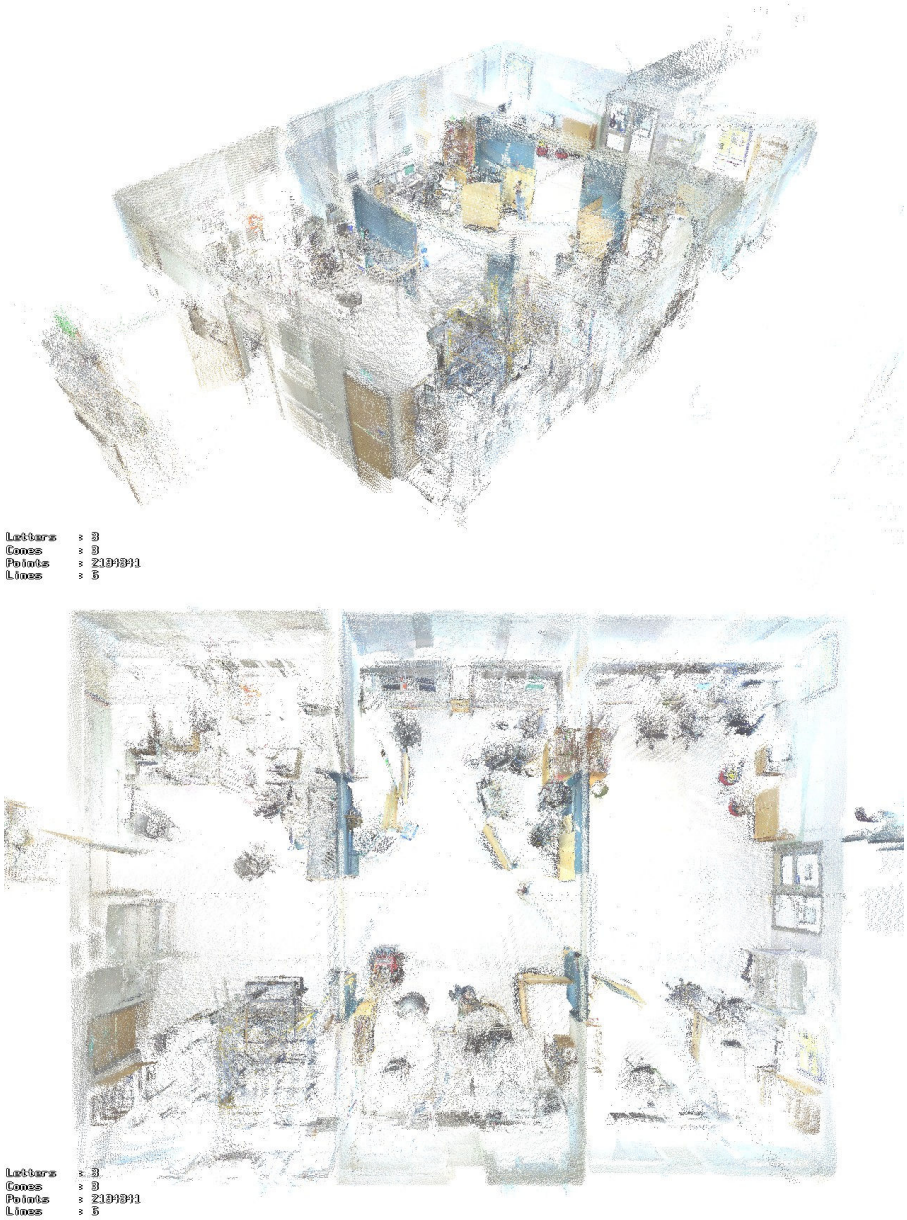

Figure 8.3: Result of sequential registration of 22 scan poses. The visualised data comprise of 3 $\times$ 22 registered scans and the corresponding colours from 7 $\times$ 22 camera images.

the first scan pose with the last scan pose using the trimmed ICP version. The estimated position was calculated by sequentially registering all 22 scan poses, which means that only one small failure in one of the registrations can heavily influence the estimate of the final pose.

For further evaluation, the number of corresponding matches $\mathcal{N}$ that were used in the registration was also investigated.

Table 8.1 show the euclidean pose error d (in meters) together with the rotational error α (in radians). Since corresponding matches were selected randomly, each sequential registration was repeated 5 times. These results show that the performance of Tr.GTLS – ICP is better compared to Tr.ICP when there are fewer corresponding matches and $\mathcal{N}$ is low. When the number of available matches increases the two methods show more similar results. The increased error with a higher number of corresponding points $\mathcal{N}$ is likely to depend on the random selection of points.

Table 8.1: Registration results given in meters and radians using the trimmed registration versions

	Tr. ICP		Tr. GTLS – ICP	
$\mathcal{N}$	$d \pm \sigma_d$	$\alpha \pm \sigma_\alpha$	$d \pm \sigma_d$	$\alpha \pm \sigma_\alpha$
10	1.14±0.54	0.30±0.18	0.84±0.33	0.25±0.11
15	0.76±0.83	0.17±0.24	0.70±0.85	0.18±0.22
20	0.30±0.11	0.05±0.02	0.24±0.14	0.06±0.04
30	0.09±0.05	0.04±0.02	0.11±0.08	0.04±0.01
40	0.14±0.07	0.03±0.01	0.19±0.10	0.04±0.02
60	0.13±0.03	0.03±0.01	0.15±0.06	0.03±0.02

8.4.3 Outdoor Experiment

An outdoor data set "registration – outdoor" consisting of 32 scan poses was collected outdoors close to a building (see Fig. 8.4). However, in this data set there is no ground truth available and therefore the conclusions are drawn from visual inspection alone. Looking at Fig. 8.5, for example, the left wall appears much clearer when using Tr. GTLS – ICP indicating that the registration results are better. Also the lamp post appears to be duplicated in Tr. ICP but not when using Tr.GTLS – ICP. This can be explained by that the number of outdoor features which has a high covariance is difficult to predict, for example features lying on thin objects such as branches. If the depth variance is high, the Tr. GTLS – ICP method uses the bearing to the feature rather than the actual estimated feature position. Of course other problems occur with larger feature distances, for example, calibration errors will have a larger impact.

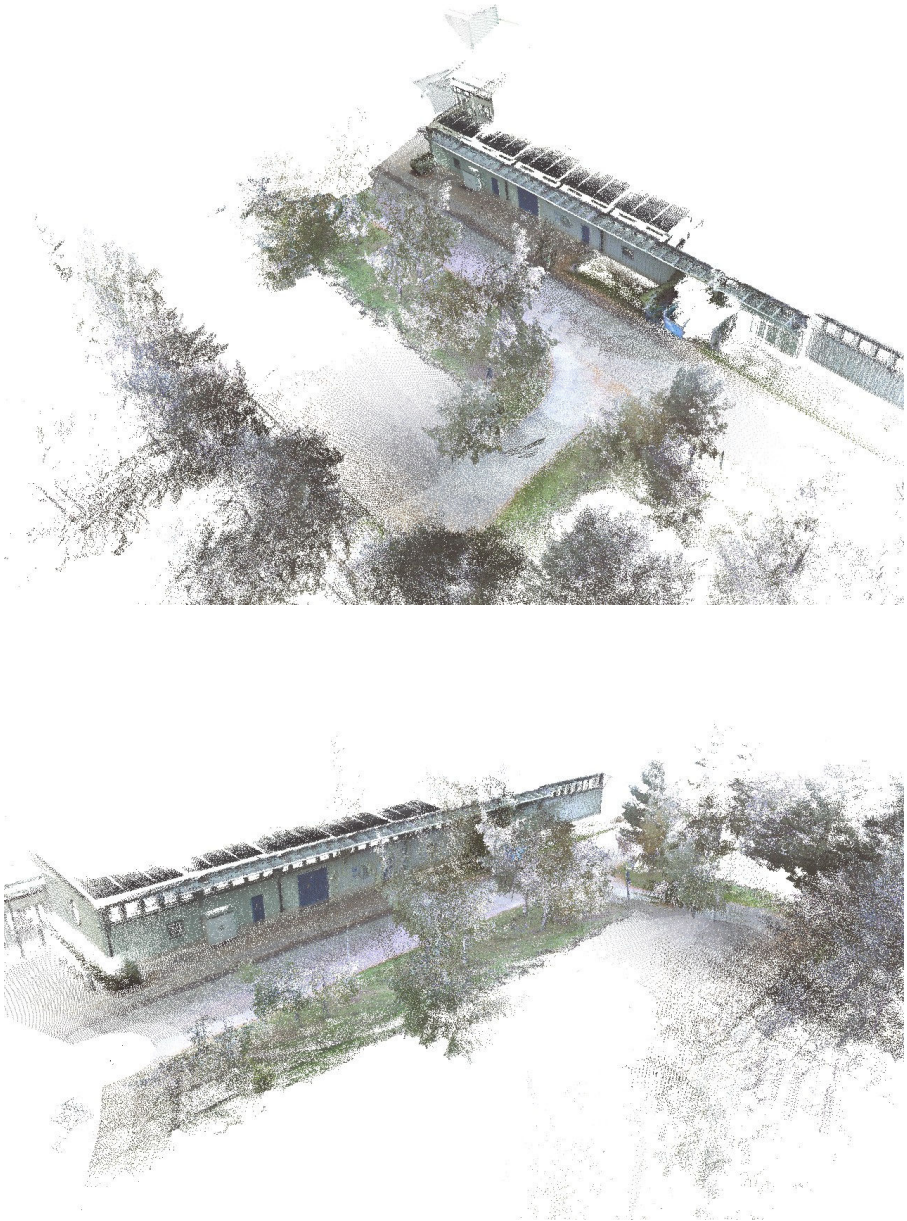

Figure 8.4: An outdoor registration result using Tr. GTLS – ICP visualised with 1.5 millions of coloured laser range readings.

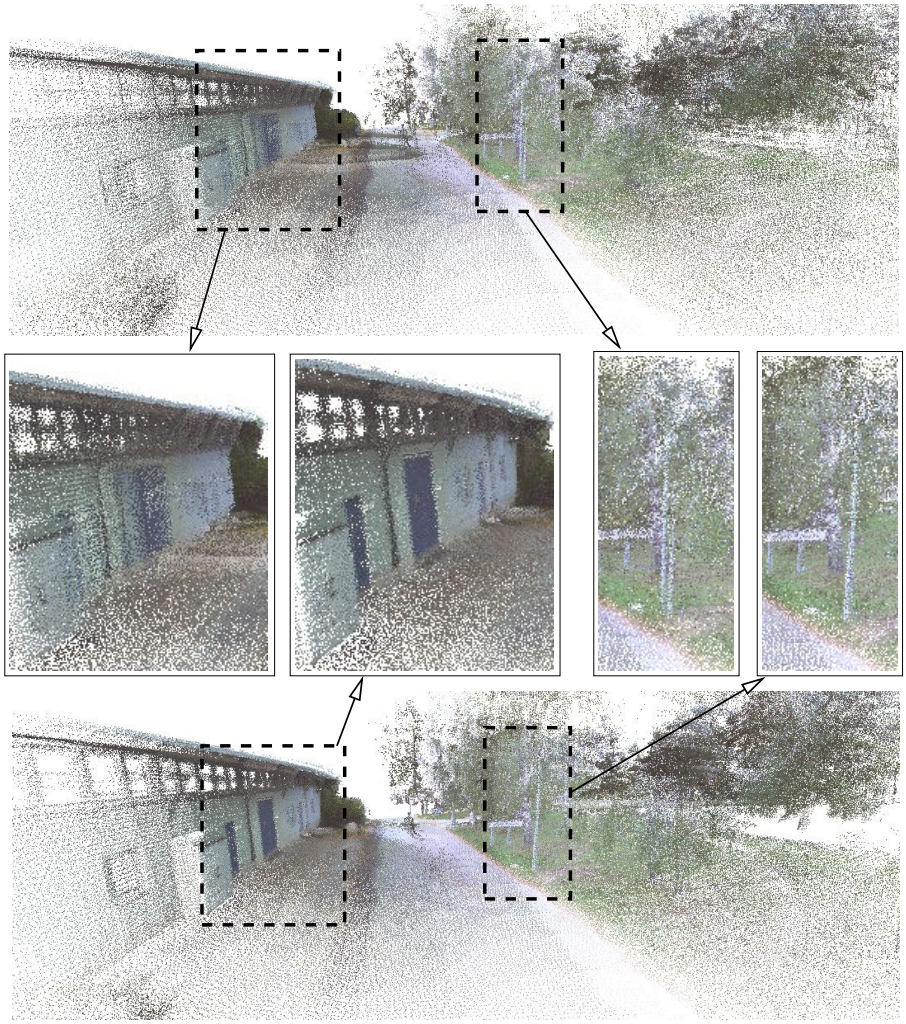

Figure 8.5: Outdoor registration results using Tr. ICP (top) and Tr. GTLS – ICP (bottom) without using any limits of corresponding points $\mathcal{N}$. It can be seen that the building wall is more accurately constructed in the Tr. GTLS – ICP method, also the lamp post to the right is duplicated in the Tr. ICP method.

8.5 Conclusions

In this chapter we have proposed a registration method that uses visual features to handle the correspondence problem. The method integrates both vision and a 3D laser to obtain depth estimates and does not rely on any initial estimate for registration. The 3D laser scanner is in addition used to obtain a covariance estimate for the extracted visual features, which is incorporated into the registration algorithm.

Chapter 9
Mapping and Localisation with Vision and Range Sensing

This chapter addresses the mapping and the localisation problem in 3D using vision and a 3D laser scanner. Two appearance based approaches are proposed which utilise visual features to determine correspondences between two scans consisting of both 3D laser data and camera images. A major difference to most other mapping and localisation algorithms is that no initial estimate of the robot pose is required.

9.1 Vision and 3D Laser Scanner based 3D-SLAM

Depending on how sensor data or rather the map are represented, typical SLAM methods can be divided into:

- landmark tracking based methods,
- pose-relation based methods (or graph-based), and
- grid-map based methods.

In landmark tracking based approaches, the position of both the landmarks and the robot are determined from multiple observations (of the landmarks). Given the position of each landmark, the robot pose can be estimated through its observations of the landmarks and in a similar way, the landmark positions can be updated using a new observation from the robot. Extended Kalman Filters (EKF) are commonly applied to update the position of each feature [124, 45].

Pose relation based (commonly called "graph-based" or "network-based") methods instead focus on relative pose estimates. The input consists of a graph where each node represents a robot pose and each vertex represents a relative pose estimate between two nodes (relation), see Fig. 9.1. For example, the

method for obtaining the Maximum Likelihood (ML) estimate in Mini-SLAM (Ch. 6) uses this approach.

Grid-map based methods have been widely used in 2D whereas it is a very uncommon approach in 3D. One large group of methods are particle-filter based approaches, for example Fast-SLAM [88]. Each particle represent a trajectory and the particle filter evolves by sub-sampling the particle set (set of trajectories) which best fits the current measurements. A major difficulty for grid-map based methods in 3D environments is the increased complexity when dealing with 6 degrees of freedom (DOF), instead of 3 DOF for the 2D case, which requires an increased number of particles to handle the larger state-space. In addition, the time to evaluate each trajectory and to update and maintain a grid-map is substantially more demanding in 3D compared to the 2D case. However, in 2D environments, particle-filter based grid-map approaches are known to have produced large and consistent maps [53]. Fairfield et al. [39] recently developed a 3D particle filter based grid approach for underwater mapping of tunnels. In their work they utilised depth sensors and an inertial measurement unit (IMU) which "gives excellent measurements for all but x and y". To cope with the computational demands of the update and storage requirement of each particle a special data structure is created called a Deferred Reference Counting Octree (DRCO).

9.1.1 Landmark/Feature Tracking Based Methods

A key part of many vision based SLAM methods is to determine positions of landmarks and these methods would therefore be sorted into the landmark based approach bin. Determining the positions of landmarks is commonly done by tracking each feature over multiple frames, integrating the measurements to decrease the position variance. To obtain initial relative pose estimates, odometry is either used directly from the robot or derived directly from the camera images. As stated in Section 8.2, stereo-cameras are often used to avoid scale ambiguities and scale drift. In [25], a SLAM method is suggested using a single camera. In this case the scale ambiguity is addressed by first presenting an object with known geometric properties. Popular methods to update poses of visual features are Extended Kalman Filters (EKF) [25, 62], Rao-Blackwellised Particle Filters (RBPF) [36, 10].

Not only vision methods rely on landmarks. Several laser scan based SLAM approaches have been proposed which typically use landmarks or features such as corners and walls.

One complete framework which can handle and update landmark positions in a computationally very efficient way is the Tree-Map algorithm [46]. The key aspect of this method is the tree structure which allows multiple 'blocks' of landmarks to be updated simultaneously.

For a more extensive overview, please refer to [45, 109].

9.1.2 Pose Relation Based Approaches

In contrast to most vision based methods, the "Mini-SLAM" method proposed in Chapter 6 and the method proposed in this chapter work without the need to track features over different poses. Both proposed SLAM methods consider only similarities without the need to track and update each feature position, which is typically computationally expensive [45].

Compared to the few vision based methods relying on relation based methods there exist a variety of 2D and 3D laser range scanner based approaches. One of the first examples using 2D data is the work by Lu and Milios [82]. A relaxation based method, using Gauss-Seidel iteration, was originally proposed by Duckett et at. [30] which uses gradient descent (GD). This approach was later extended to Multi-Level Relaxation (MLR) by Frese et al. [47], using a multigrid approach to improve performance in loop closing and to handle linearization errors due to robot rotation (which was addressed using a compass in [30]). MLR however is currently only implemented in 2D. Wulf et al. [122] compare the most common relaxation based methods for 3D data.

Olson et al. [94] use an approach based on stochastic gradient descent (SGD) to optimise the global poses. Note that the main difference compared to the work in [30] is that a constraint (vertex) is selected to move a set of nodes instead of selecting a single node and move it due to its connected nodes. In the work by Olson, only 2D data were evaluated, however due to their assumption of linear angular subspaces, which does not hold in the case of 3D data, their approach is not directly applicable in 3D [112]. The problem of linear subspaces has recently been addressed by Grisetti et al. [51], who present a pose relation based method that works in 3D. In their approach a variant of the gradient descent method is applied together with incremental spherical linear interpolation (SLERP), to address the non-commutativity property of rotation in 3D. In Triebel et al. [113], global constraints are used based on extracted planar surfaces from 3D data together with local constraints from the correspondent point in the interest closest point (ICP) based registration. In their approach the robot poses are estimated with a conjugate gradient method. In another approach by Triebel et al. [115], the constraints in a global pose network were optimised using LU (Lower Upper triangular) decomposition. An approach similar to our proposed method is the work by Newman et al. [92], in which vision techniques and 3D range data are combined. Loop closure is detected by using appearance based similarity measures. The successive registration is done using 3D laser data and odometry.

9.1.3 Comparison Between Vision and 3D Laser Scanner Based Methods

An important observation regarding visual SLAM methods is that relation based methods hardly exist. The reason is most likely that, for typical vision

systems, the measurement noise of the feature pose estimates is simply too high to use the features directly. Therefore each feature has to be tracked over several frames to increase the certainty in the feature pose. By contrast, pose relation based methods are most common for laser scan based systems. This is likely due to the accuracy that the laser provides, which makes it unnecessary to extract and track features (handle correspondences) in 3D laser data. Also many registration methods such as the standard ICP [12, 23] and NDT [15] do not rely on extracted features.

In general, vision based systems are well suited to address the data association problem, possibly better compared to 3D laser based systems, as suggested in [92, 57, 71]. One exception is the data association results shown in the work by Bosse et al. [18], who detect correspondences between 2D local maps generated with a 2D laser range scanner. The method converts each local map into histograms, and the histogram are then matched using correlation. Worth noting is that the data set used to validate the approach contains a very small amount of self similar local maps that could cause perceptual aliasing.

Table 9.1: Comparison of vision and laser range finder (LRF) based methods, $\surd$ - good, $\times$ -weak.

	Vision	LRF
pose relation methods	$\times$	$\surd$
handle correspondences	$\surd$	$\times$

9.1.4 A Method Combining Vision and 3D Laser Scanner

In the same manner as the registration approach in Chapter 8, the method proposed here uses local visual features with estimated 3D positions. The proposed method, called 3DVF-SLAM has several similarities compared to the Mini-SLAM method in Chapter 6. Both Mini-SLAM and 3DVF-SLAM rely on visual similarity measures between images to determining correspondences, e.g. to detect loop closing. The difference is the usage of an accurate range sensor (3D laser range scanner) to obtain depth estimates. This allows to determine a good position estimate μ_F and covariance C_F for each feature F (see Section 8.3.1). In comparison, stereo-vision based approaches do not have the same accuracy and typically need to track the feature over several frames to improve the position estimate. Basically this adds the possibility to use appearance based approaches using local visual features but with much higher geometrical accuracy.

The proposed method can be described as follows: Given a set P of n robot poses $\mathbf{x}_{1..n}$ together with a set of extracted features $F_{\mathbf{x}_{1..n}}$ at each robot pose, an initial estimate of all poses can be calculated by performing successive registration of each scan pair, i.e. registering the last scan $\mathbf{x}_i$ with the previous one

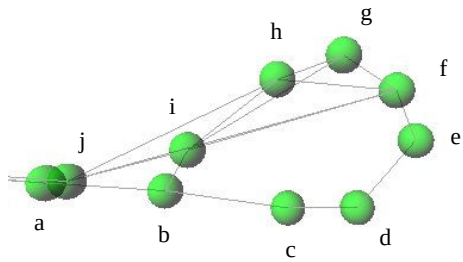

Figure 9.1: An example of a pose graph in 3D, seen from above, consisting of data used also in Fig. 9.3. Each sphere represents a robot pose $\mathbf{x}$ and each line represents a relation r.

$\mathbf{x}_{i-1}$ as done in the evaluation experiments in Chapter 8. With this approach, however, the errors will accumulate. One solution is to introduce additional registration results as constraints (relations). If the current pose $\mathbf{x}_i$ is registered to a previous pose $\mathbf{x}_j$ but not its direct predecessor (i.e. $j < i - 1$), commonly called "loop closure", the uncertainty of the pose estimates decreases. Hence, after the first loop closing occurs, an additional relation is added and for the $\mathbf{x}_{1..n}$ robot poses there exist $r_{1..m}$ relations where $m = n + 1$. Basically what happens is that an overestimated equation system is obtained ($m > n$). By adding more relations or constraints the pose of each node will be determined more accurately since more measurements are incorporated. A pose graph containing both robot poses $\mathbf{x} = [a, b, ..., i]$ and relations r can be seen in Fig. 9.1. Different relations will have different impact on pose errors since the error grows with the number of poses (assuming that the distance between successive poses is similar). A relation which connects two nodes which previously were separated by many relations generally provides more information in terms of pose error reduction than a relation which connects two nodes which are separated with few relations. For example, the relation $r_{i,b}$ in Fig. 9.1 reduces the number of relations between a to i to two, compared to nine if only successive relations were used.

To detect whether or not a relation r should be added, the similarity measure S is used. Hence the proposed approach 3DVF-SLAM is in this manner similar to the Mini-SLAM approach (Ch. 6) with the difference that there is only one type of relation r, which is based on the optimisation constraint in the registration method described in Chapter 8. Given two robot poses $\mathbf{x}_i, \mathbf{x}_j$ and a set of matched feature pairs $< F^k_{\mathbf{x}_i}, F^k_{\mathbf{x}_j} >_{k=1..N}$ the constraint caused by

the relation $r_{\mathbf{x}_i,\mathbf{x}_j}$ is obtained by minimising the following function (see also Sec. 8.3.4):

$$\begin{aligned} J(\mathbb{R}_{\mathbf{x}_i,\mathbf{x}_j}, \mathbf{t}_{\mathbf{x}_i,\mathbf{x}_j}) &= \sum_{k=1}^{N} (\mu_{q_k} - \mu_{F^k_{\mathbf{x}_i}})^T C^{-1}_{q_k} (\mu_{q_k} - \mu_{F^k_{\mathbf{x}_i}}) + \\ & \sum_{k=1}^{N} (\mu_{F^k_{\mathbf{x}_i}} - \mu_{q_k})^T C^{-1}_{F^k_{\mathbf{x}_i}} (\mu_{F^k_{\mathbf{x}_i}} - \mu_{q_k}), \end{aligned} \tag{9.1}$$

where $\mu_{q_k} = \mathbb{R}_{\mathbf{x}_i,\mathbf{x}_j} \mu_{F^k_{\mathbf{x}_j}} + \mathbf{t}_{\mathbf{x}_i,\mathbf{x}_j}$ and where $\mathbb{R}_{\mathbf{x}_i,\mathbf{x}_j}$ and $\mathbf{t}_{\mathbf{x}_i,\mathbf{x}_j}$ are the relative rotation and translation from pose $\mathbf{x}_i$ to $\mathbf{x}_j$. The covariance matrix C_{q_k} is obtained by rotating the eigenvectors of the covariance matrix $C_{F^k_{\mathbf{x}_j}}$.

By rewriting the optimisation constraint as

$$K(\mathbf{x}_i, \mathbf{x}_j) = J(\mathbb{R}_{\mathbf{x}_i,\mathbf{x}_j}, \mathbf{t}_{\mathbf{x}_i,\mathbf{x}_j}) \tag{9.2}$$

the problem of determining the robot poses given the relation constraints can now be defined as minimising

$$L(\mathbf{x}_{1..n}) = \sum_{i=0}^{n} \sum_{j=i+1}^{n} V(i,j) K(\mathbf{x}_i, \mathbf{x}_j), \tag{9.3}$$

where $V(i,j)$ is a binary variable that decides whether the similarity measure $S_{i,j}$ is above a preselected threshold $\mathcal{T}$. The total number of summations needed in Eq. 9.3 is the number of relations m. For the optimisation of Eq. 9.3 the Fletcher-Reves conjugate gradient optimisation method [40] is applied.

9.2 Experimental Results

To evaluate the proposed 3DVF-SLAM method, a data set (indoor 3D−SLAM) was obtained by manually driving the robot around in our lab collecting 23 different scans at 23 locations. At each location 3 laser scanner sweeps and 7 images were recorded, i.e. 69 laser scanner sweeps and 161 camera images for the whole data set, see also Section 8.4.1. A generated "bird's eye view" figure of the data set can be seen in Fig. 9.2, where coloured 3D laser data is drawn at the estimated poses $\mathbf{x}_{1..23}$. In Fig. 9.3 and Fig. 9.4 the difference between using successive registration to obtain poses and the 3DVF-SLAM method is visualised.

Qualitative results are obtained by calculating the planeness of the estimated poses $\mathbf{x}_{1..23}$, since the data was collected indoors the ground truth assumption is that all poses should be lying in a plane, see also Fig. 9.4. The plane P is obtained by spanning the two largest eigenvectors $\lambda_{1,2}$ calculated from the covariance matrix C of all poses. The mean squared error (MSE) is calculated using the distance of each pose $\mathbf{x}$ to the plane P, see Table 9.2.

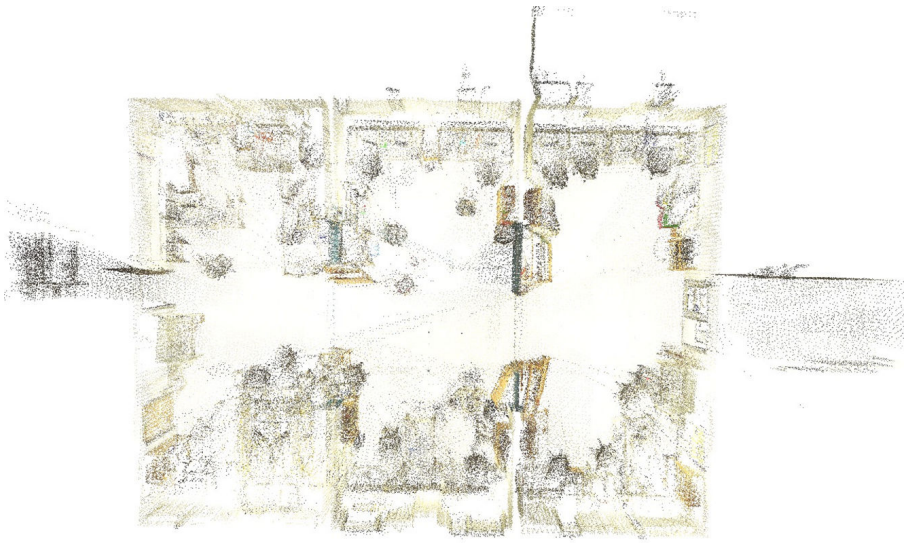

Figure 9.2: The result using 3DVF-SLAM with the test data set consisting of 3×23 registered scans and 7×23 camera images. This "bird's eye view" figure shows the point cloud created by using the laser range measurements at the estimated poses $\mathbf{x}_{1..23}$. The colour is obtained from the camera images by projecting the range measurement onto the image plane.

Table 9.2: MSE comparison between successive registration and the proposed 3DVF-SLAM method.

	successive registration	SLAM
MSE	$1.172 \cdot 10^{-3}$	$0.187 \cdot 10^{-3}$

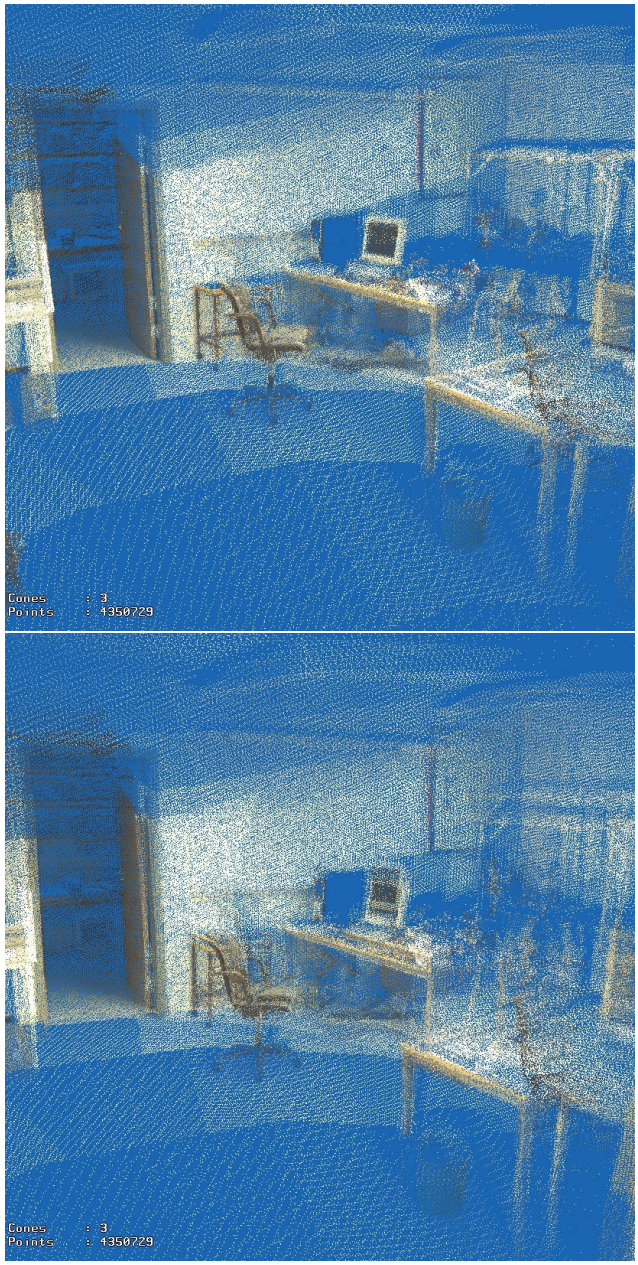

Figure 9.3: Comparison between using the 3DVF-SLAM technique (top) and using the (odometry-like) successive registration (bottom). The difference can, for example, be seen at the screen and the office chair.

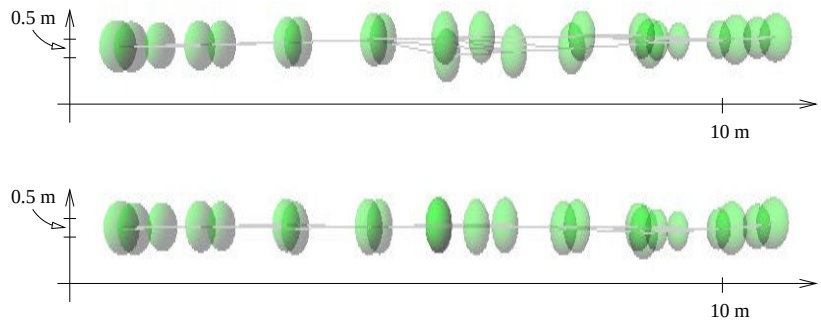

Figure 9.4: Comparison of the pose relation graphs seen from the side. Since the robot was driven on an indoor flat surface the nodes should appear on a straight line in a side view, please note that the difference in scale is created for clarity proposes. See also Table 9.2. Top: successive registration. Bottom: 3DVF-SLAM.

9.3 Vision and 3D Laser Scanner based Localisation

Global localisation is to determine the pose estimate of a robot with respect to a previously learnt or given map without any initial pose estimate, essentially answering the question "where am I?". Since the goal here is to determine the position in 3D, geometrical and not topological localisation is discussed.

The most apparent difference between localisation in 2D and localisation in 3D is the much larger state space. In 2D localisation, a planar world, particle filters are a very common approach which were also utilised in Chapter 5.

9.3.1 Vision-based Methods

A 3D stereo vision approach using a particle filter has been suggested by Elinas et al. [35]. The motion of the robot is derived from vision and uses multiple view geometry, the Essential matrix and reprojection to obtain the visual odometry readings. Motion estimation from cameras or visual odometry has been well-studied in computer vision. One reason for using stereo-vision compared to monocular (single camera) systems is the scale ambiguity and scale drift which occurs while using a single camera [56].

9.3.2 3D Laser Scanner-based Methods

An approach suggested by Kümmerle et al. [70] uses a 3D laser scanner where the environment is represented with a Multi-Level Surface (MLS) 3D-map [115]. In these experiments the laser was only pointing horizontally, i.e. using only 2D-laser data. Feature based methods that rely on laser data are also suggested

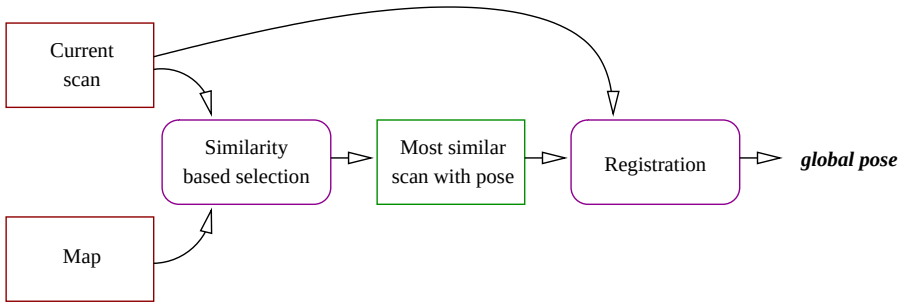

Figure 9.5: Overview of similarity based global localisation. The global pose is obtained by first determining the scan pose with the most similar appearance using the similarity measure obtained from the images and then adding the registration result to the selected scan pose from the map. The map consists of a set of scan poses which consist of a set of extracted visual features in 3D. The pose of each scan is determined by the 3DVF-SLAM method.

in the literature, for example, Adams et al. [4] rely on detecting tree trunks or similar cylindrical shapes as landmarks. Another feature based approach was proposed by Lingemann et al. [78] who utilise a fast scan-matching approach based on filters and a polar representation to obtain distinguishable landmarks. Please note that their localisation only handles 2D range data as input, then the localisation results are then used as initial pose estimates to a 6D SLAM algorithm. A completely different approach for outdoor localisation in 3D, regarding sensory equipment, is the work by Shmitz et al. [91] who use an Inertial Measurement Unit (IMU), odometry and DGPS receivers to update a Kalman filter. The system gives rough estimates of the position, without using either vision or laser scanners.

9.3.3 Similarity-based 3D Global Localisation

The main difference with previous approaches and the proposed method is to use a laser range scanner, which typically produces much higher accuracy than, for example, a stereo-camera, to extract the depth estimates of the visual features. The method, called 3DVF-localisation, uses the correspondences between the visual features and their 3D position to accurately determine the relative displacement (registration) between two robot poses in 3D. In addition to registration, the visual features are also used to find the most visual similar scan pose in the map, hence to determine which of the scan poses in the map the current scan $\mathcal{S}_c$ is registered against, called appearance based localisation. Hence, the proposed global localisation method relies on the following two parts:

- appearance based localisation,
- registration without initial pose estimate.

For a brief overview of the method, see Fig. 9.5.

The similarity measure S is used in the appearance based localisation to determine the most similar scan pose $\mathcal{S}_s$ in the map. In Fig. 9.6 a similarity matrix is shown for two data sets where the most similar scan for each column is marked with a $\times$. No accumulation of evidence to handle multiple hypothesis was done in the expereiments although the method could easily be extended with the particle filter based approach suggested in Chapter 5, where the odometry reading is replaced with successive registration as described in Chapter 8.

The appearance based localisation returns a scan pose without any relative pose estimate. Hence, a required key property of the registration method is to register without any initial pose estimate. In the proposed method we apply the registration method described in Chapter 8, where a relative pose estimate can be obtained by using the extracted visual features in 3D.

The global localisation pose is finally calculated by adding the global pose of the most similar scan $\mathcal{S}_s$ with the relative pose from $\mathcal{S}_s$ to the current scan $\mathcal{S}_c$ obtained from the registration.

9.3.4 Experimental Results

Two data sets were used in the experimental evaluation. First, the map was created using the 3DVF-SLAM method described above using the $\mathtt{indoor\ 3D-SLAM}$ data set. For each scan pose in the $\mathtt{indoor\ 3D-localisation}$ data set the pose was determined by the proposed method. Visual results can be seen in Fig. 9.7, showing a coloured point cloud created from sensor data from each estimated pose obtained from the 3DVF-localisation method. In Fig. 9.8 the global position obtained from the localisation is drawn upon the map, represented as the grey point cloud.

Please note that both the 3DVF-SLAM and 3DVF-localisation methods are used in the difference detection (Ch. 10).

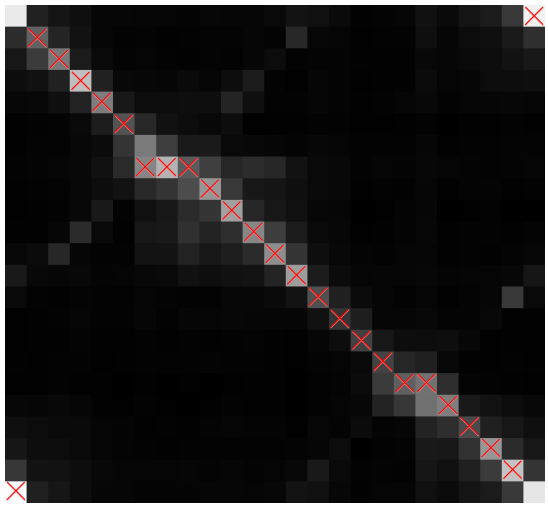

Figure 9.6: A similarity matrix of two different data sets: indoor 3D – SLAM, the map (rows), and indoor 3D – localisation, localisation test data (columns), collected at different times in the same area. Each cell shows the visual similarity of two scan poses, where brighter regions corresponds to a higher similarity. The × illustrates the appearance based localisation results, hence which map scan pose each scan in the indoor localisation data set should be registered against.

Figure 9.7: Localisation result created by using coloured laser scan data, where the pose of each scan is obtained from the proposed global localisation method.

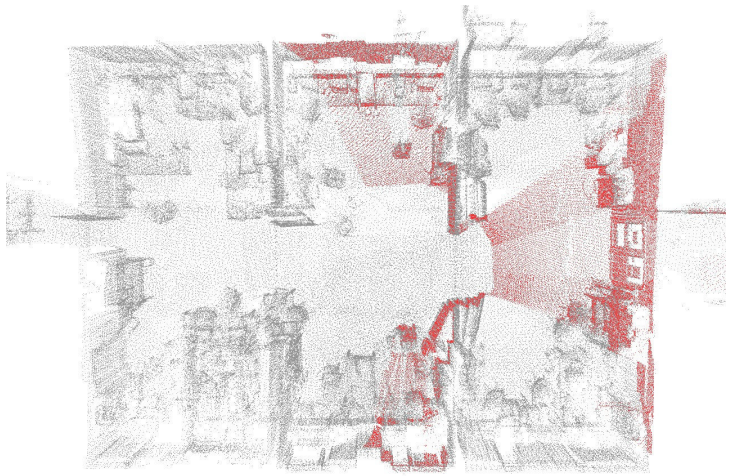

Figure 9.8: Localisation results showing only one scan pose. The laser scanner data for the scan pose that is localised is drawn in dark (red) together with the grey point cloud representing the map.

Chapter 10
Difference Detection using Vision and 3D Range Sensing

This chapter presents a system for autonomous change detection with a security patrol robot. Difference detection or change detection can be described in a more popular way as an automated "find five errors", see Fig. 10.1, but instead of comparing two pictures, the task is to detect changes in a real 3D world environment over time. The main difficulties for humans to detect changes probably lies in the fact that the two states of the environment cannot co-exists and therefore the difference has to be spotted using our memory alone. The autonomous application shown in this chapter utilises methods previously introduced in earlier chapters. In an initial step a reference model of the environment is created. After a certain time period, when changes have occurred, data is collected and compared to the reference model. The difference detection is based on coloured 3D point clouds obtained from a 3D laser range scanner and a colour CCD camera. The proposed approach introduces several novel aspects, including a registration method that utilises local visual features to determine point correspondences (thus essentially working without an initial pose estimate) and the 3D-NDT representation with adaptive cell size to efficiently represent both the spatial and colour aspects of the reference model. A qualitative experimental evaluation in an indoor lab environment is presented, which demonstrates that the proposed system is able to register and detect changes in spatial 3D data and also to detect changes that occur in colour space and are not observable using range values only.

10.1 Introduction

An important task for robotic security systems is surveillance of a specified area. Typical security patrol missions require detection of changes in the environment and description of detected differences with respect to a previously determined reference state. A human watchman is first shown around the premises to learn

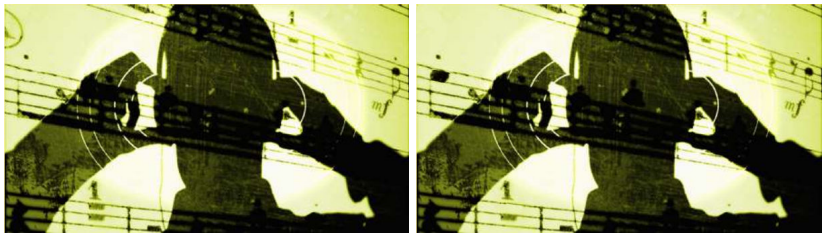

Figure 10.1: "Find five errors" example. The task is to detect five areas that are different in the two images.

the reference state of the environment. The mission of the security patroller is then to check for changes in the environment, e.g. looking for open doors, open windows, water leaks, blocking of fire hoses, etc. [2]. Due to their dull and potentially dangerous character it is desirable to delegate security patrols to mobile robots. Current approaches often require teleoperation [79]. Accordingly, research has focused on adjustable autonomy or semi-autonomy to decrease the amount of "cognitive burden" to the operator (Seeman et al. [103], Goodrich et al. [50], for example).

This chapter presents a system for autonomous change detection with a security patrol robot. As for its human counterpart the robot watchman is expected to determine the reference state of the environment (the reference model) in an initial phase. The actual mission requires discovering changes with respect to the reference model and describing the differences. An alarm may be triggered whenever changes are detected or detected differences can be reported to a human operator for further analysis.

An effective system for autonomous difference detection needs to fulfil a couple of demands. First, creation of the reference model should be simple and require minimal effort by a user. Ideally the robot would acquire and update the reference model of the environment autonomously. The problem of updating the reference model is related to the dynamic mapping problem that investigates continuous adaptation of maps over time [14]. Here we assume that the mobile security robot is first guided by a human operator in order to learn a model of the original, unmodified environment. A second major requirement especially with regards to large environments is that the reference model is represented efficiently but nevertheless allows determination of small changes in the environment. Representing the environment at a high resolution so as to avoid the need to inspect all parts of the environment from a small distance can be seen generally as the third major requirement of an autonomous difference detection system.

Difference detection has been studied in the context of recognising parked vehicles by Ishikawa et al. [60]. In their work, Ishikawa et al. use an omnidirectional camera together with GPS and INS (Inertia Navigation System) to

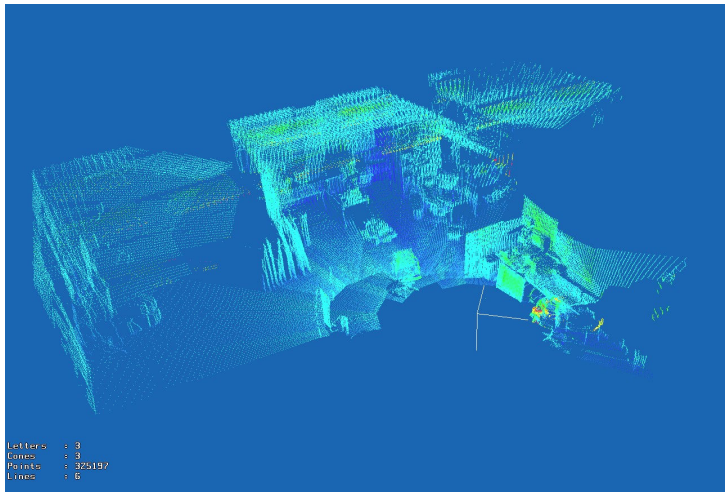

Figure 10.2: An example of a spatial distribution assigned with colour from a thermal camera (instead of a standard CCD camera). By utilising thermal information one related approach to difference detection could be to detect changes in temperature. To detect changes in temperature could be useful, for example, in monitoring machinery or improving energy consumption in buildings.

obtain depth estimates, which were subsequently compared with a pre-built model.

By adding additional sensor modalities to the system such as a thermal camera (see Fig. 10.2), additional monitoring applications emerge from the system. For example, one application where thermal information could be used is to monitor machinery such as turbines (to monitor bearings, etc.). A mobile combination of a thermal camera with a 3D range scanner could cover large regions and therefore be a more cost efficient method compared to covering the area with multiple thermal cameras. One other promising approach is to create 3D heat models of buildings to observe heat dissipation with the goal to obtain less energy consuming houses.

10.2 Overview of the Difference Detection System

An overview of the proposed difference detection system is shown in Fig. 10.3. Corresponding to the two columns in the figure, the approach decomposes into two parts: acquisition of the reference model (shown to the left of the dashed line) and pre-processing of new data (shown to the right of the dashed line) for the actual detection of differences between new data and the reference model (indicated by the box below the dashed line).

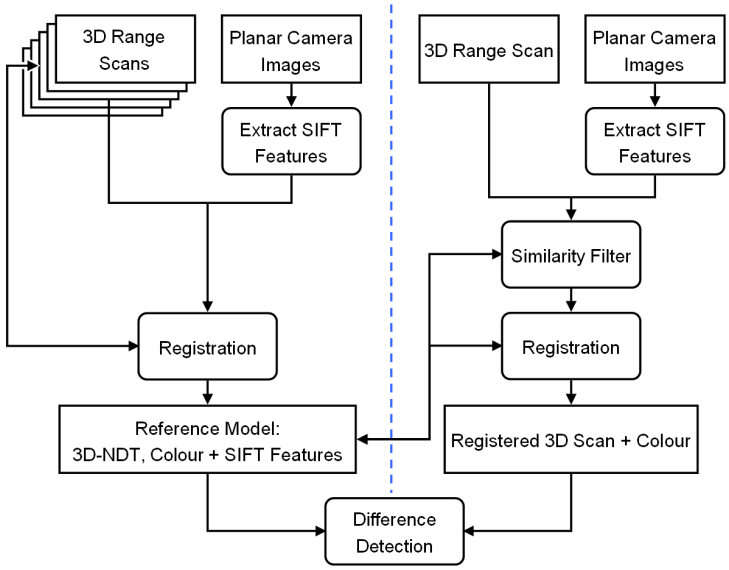

Figure 10.3: Overview of the difference detection system. Acquisition of the reference model is shown on the left side of the dashed line and pre-processing of new data for the actual difference detection to the right of the dashed line.

The proposed difference detection approach comprises three main components described in Sections 10.3 – 10.5 below. A very important aspect is accurate registration of the data, see Section 10.3. The requirement of the registrations method is to produce accurate results even under the condition that a reasonable relative pose estimate is not available (when there is essentially no estimate of the relative pose between two data sets other than that they were recorded in the same environment). This is achieved by using local visual features extracted from the camera images to solve the data association problem in the registration process, see Chapter 8.

In order to represent the environment efficiently, we apply the Normal Distribution Transformation (NDT) with adaptive cell splitting (Sec. 10.4.1) to the spatial point distribution and the colour distribution (Sec. 10.4.3 and Sec. 10.4.3). Finally, based on the 3D-NDT representation, and respectively the Colour 3D-NDT representation, difference probabilities are calculated as described in Sections 10.5.1 and 10.5.2.

A visualisation of the difference probabilities in a fairly trivial case where two 3D laser scan sweeps are compared can be seen Fig. 10.4. The corresponding 3D-NDT representation is shown in Fig. 10.5 using ellipsoids to indicate the eigenvalues of the respective covariance matrices.

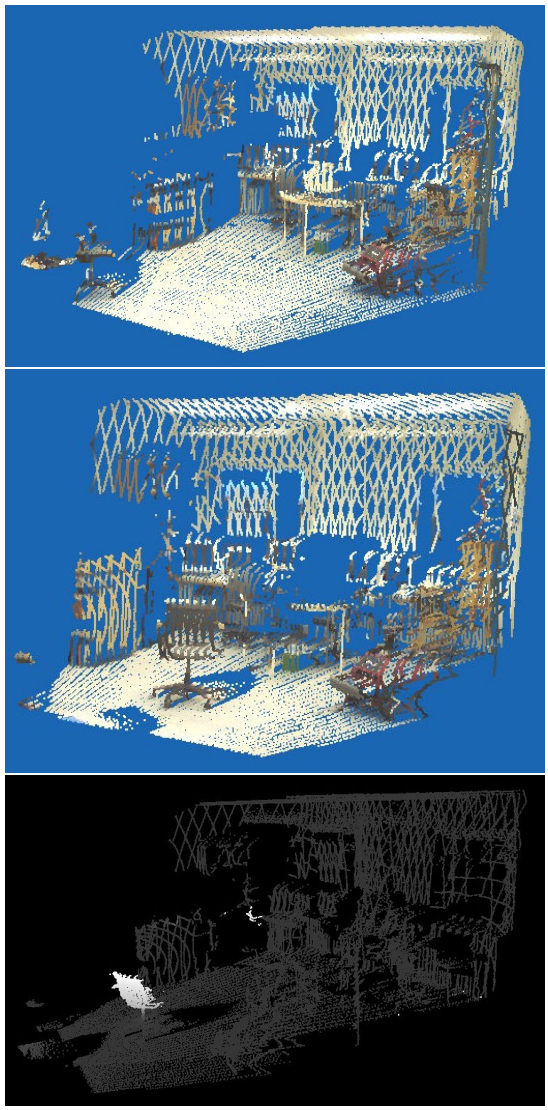

Figure 10.4: Top, middle: 3D range scans recorded at a relatively small displacement of approx. 1 m. Bottom: Difference probability corresponding to a chair that was placed in the scene before the second scan was recorded. Brighter regions indicate a higher difference probability. The displacement between the two scans was obtained using the registration method described in Section 8.3.3.

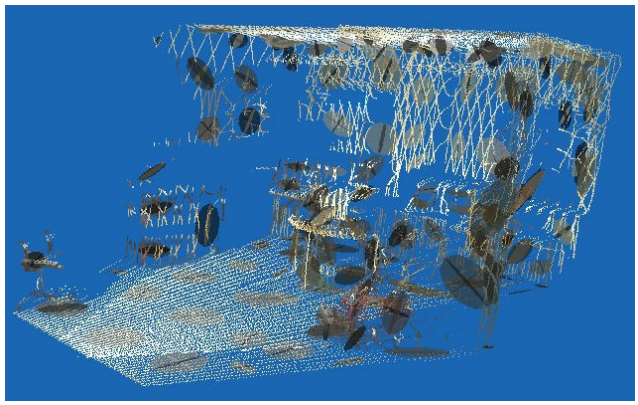

Figure 10.5: Visualisation of the 3D-NDT representation corresponding to the scan in Fig. 10.4, top. Ellipsoids indicate the eigenvalues of the respective covariance matrices.

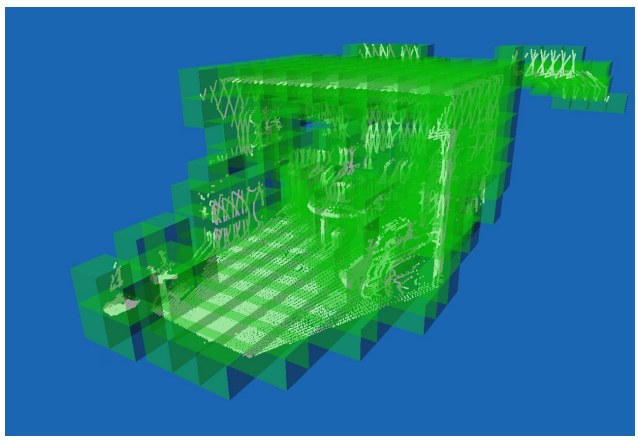

Figure 10.6: The scan shown in Fig. 10.5 divided into a set of cells **C** represented as transparent green boxes used in the 3D-NDT representation.

10.3 Registration

A very important aspect of the difference detection system is that the relative position $[\mathbb{R}, \mathbf{t}]$ between the current $\mathcal{S}_c$ and previous scans $\mathcal{S}_p$ is precisely known. Accurate registration is therefore a fundamental requirement. Since the measurements (3D range scans and colour images) to be compared will be obtained not only at different poses but also at substantially different times, registration should be robust to a certain level of changes in the environment. In addition, it cannot be expected in general that the security patrol robot maintains a consistent coordinate system in-between acquisition of the reference model and a difference detection request. As opposed to most current approaches to scan registration we therefore need a registration method that does not depend on reasonably accurate initial pose estimates.

In order to cope with the condition that initial pose estimates may not be available, local visual features are used to establish correspondences between data points (data association) within the entire set of measurements as described in Section 9.3.3. This localisation approach (Sec. 9.3.3) to determine the global pose is used throughout this chapter.

10.4 Normal Distribution Transform (3D-NDT)

One of the major requirements for an autonomous difference detection system is that is scales well with the size of the environment. This demands an efficient representation of the reference model that compresses the data (to be able to store and maintain representations of large environments) and is yet able to represent small details allowing for detection of small changes. To address this issue, the normal distribution transform (NDT) is used to represent the environment. The NDT was introduced by Biber et al. [15] and first used to register 2D laser scans. The method has been extended to 3D scan registration by Magnusson et al. [83]. However, in this work we use a different approach to registration since visual information is used and no initial pose estimates are available, as discussed in Section 10.3.

The basic principle of NDT and 3D-NDT is to represent the environment by using a set of Gaussian distributions. First, the considered space is divided into cells, see Fig. 10.6. Each cell that contains a certain minimum number of points is represented by a Gaussian with the mean value and covariance matrix computed from the points in the respective cell (Fig. 10.5). To make sure that the covariance is representative of the spatial content of each cell, the minimum number of points was set to 5 in the experiments presented in this chapter. More formally 3D-NDT can be described as follows. We consider a point cloud $\mathbf{P} = p_1, p_2, ..., p_n$ with points $p = (x, y, z)$ given in 3D Cartesian coordinates. The environment is divided into a set of cells $\mathbf{C} = c_1, c_2, ..., c_k$ and for each cell c_i the number N_{c_i} of points p_{c_i} which lie within the cell's

boundaries are used to calculated the spatial distribution. For each cell c, the mean μ_c and covariance C_c are calculated as

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} p_{c_i} \tag{10.1}$$

$$C_c = \frac{1}{N_c - 1} \sum_{i=1}^{N_c} (p_{c_i} - \mu_c)^2. \tag{10.2}$$

Note that the NDT representation does not require evenly spaced data and hence can be calculated without further sub-sampling. In order to keep the storage requirements within the limits of the computer with which the computations were made (i.e. 512 MB of RAM memory), however, all scans were sub-sampled with a 3D-grid resolution of $0.1 \times 0.1 \times 0.1$ m^3.

10.4.1 Adaptive Cell Splitting

The level of detail that is maintained by the NDT depends on the chosen cell size. In order to select the resolution of the NDT representation according to the local content of a scene, we use an adaptive cell size approach that determines whether or not to split a cell c according to its covariance matrix C_c.

From the picture of the covariance matrix as an ellipsoid (Fig. 10.5), it is clear that a single Gaussian, i.e. one covariance matrix, can efficiently describe planes (one of the ellipsoids axis is small) and lines (two of the ellipsoid axis are small). Therefore, a large volume v_c of this ellipsoid, indicating that none of the principal axis is small, was chosen as the criterion for cell splitting:

$$v_c = |\lambda_1||\lambda_2||\lambda_3|, \tag{10.3}$$

where λ_i are the different eigenvalues of the covariance matrix. A higher value indicates a higher need to divide the cell. In the experiments presented in this work, the splitting threshold was set to 0.001 m^3.

The decision about where to split a cell is made using the mean value μ_c and the direction of the eigenvector e_c^{max} with the highest eigenvalue λ_c^{max}. The points p_{c_i} in the cell c_i that is to be split are assigned to new cells c_i^a and c_i^b according to the sign of

$$p_{c_i}(e_c^{max} - \mu_c). \tag{10.4}$$

Consequently, the cell is split at the plane through the centre μ_c and orthogonal to the largest eigenvector e_c^{max}.

10.4.2 Colour 3D-NDT

In order to be able to detect changes that do not reveal themselves through the range readings but which are observable in the planar images recorded along with the range readings, colour information is also incorporated into the NDT representation. This allows detection of changes caused by thin objects, for example, a poster that has been removed from the wall, as long as the colour differs sufficiently from the background.

In addition to points $p = [x, y, z]$ in 3D space, we now also consider corresponding colour values $\breve{p} = [C^1, C^2, C^3]$. Accordingly, the cells in the colour 3D-NDT representation are described by a mean value $\breve{\mu}$ and covariance $\breve{C}$ in addition to the spatial mean μ and covariance C introduced above. The colour mean $\breve{\mu}$ and covariance $\breve{C}$ are calculated using Eq. 10.1 and Eq. 10.2 replacing the points p_{c_i} with their associated colour values $\breve{p}_{c_i} = [C^1, C^2, C^3]$.

The RGB colour space is used in the proposed difference detection system. To obtain some degree of invariance against changing illumination, RGB values are converted to the YUV colour space, the intensity Y is set to a constant value of 0.5, and then the YUV values are converted back to RGB.

10.4.3 Adaptive Cell Splitting with Colour

Colour cell splits are generally performed in the same way as the cell splits in the spatial domain as described above. However, in the colour space the criterion for a split is only dependent on the highest eigenvalue $\breve{\lambda}_c^{max}$ of the covariance matrix $\breve{C}_c$. A colour split is carried out if $\breve{\lambda}_c^{max}$ is larger than a predefined threshold. This threshold was set to 10 in this chapter referring to RGB values between 0 and 255. Compared to the volume measure in Eq. 10.3, a modified criterion is used since lines and planes in colour space do not generally correspond to consistent structures in the environment.

10.5 Difference Probability Computation

10.5.1 Spatial Difference Probability

A probabilistic value of the point p being different from the reference model is computed using the 3D-NDT representation of the reference model. First, the cell c is determined that contains the point p. Using the mean μ_c and covariance C_c of this cell, the spatial difference probability is calculated as

$$p_{diff}(p) \propto 1 - e^{-(p-\mu_c)^T C_c^{-1} (p-\mu_c)}. \tag{10.5}$$

If adaptive cell splitting is used and therefore each grid cell can contain multiple mean and covariance values, the difference probability p_{diff} is calculated for all sub-cells using Equation 10.5 and the lowest probability is assigned to point p.

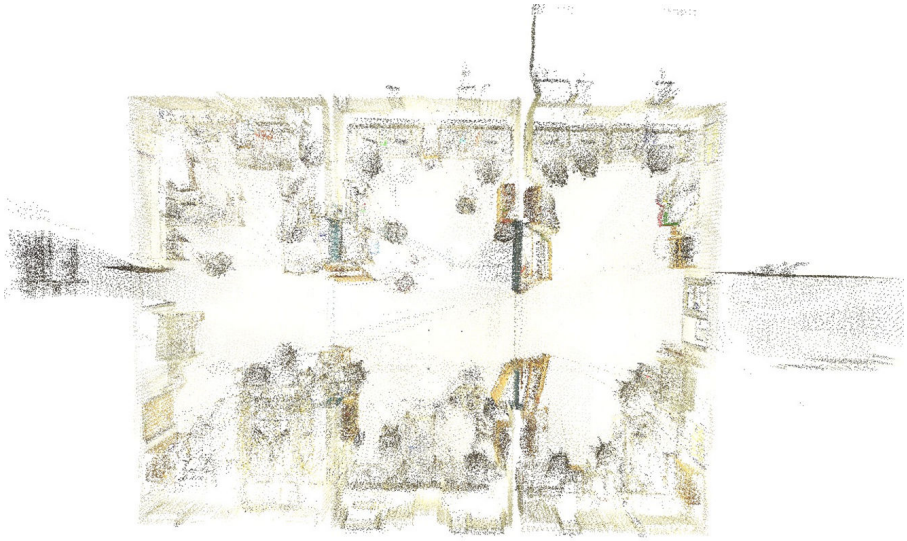

Figure 10.7: The reference model created in our evaluation experiment consisting of 3 × 22 registered scans and the corresponding colours from 7 × 22 camera images.

Note that the difference probability will depend on the structure of the point cloud used to generate the 3D-NDT representation of the reference model. In the case of planar regions, for example, the covariance matrix in Equation 10.5 will emphasize differences orthogonal to the planar structure so that even very small deviations can be detected there.

10.5.2 Colour Difference Probability

In the same way as the spatial difference probability, the colour difference probability is calculated as

$$p_{\mathrm{colourdiff}}(\check{\mathbf{p}}) \propto 1 - e^{-(\check{\mathbf{p}}-\check{\boldsymbol{\mu}}_c)^T \check{\mathbf{C}}_c^{-1}(\check{\mathbf{p}}-\check{\boldsymbol{\mu}}_c)}. \tag{10.6}$$

As described in the previous section, the difference probability will vary depending on the colour distribution of the selected cell, giving higher probabilities to colour changes in regions of very uniform colour distribution.

10.6 Validation Experiment

The data were collected as described in Section 8.4.1.

To evaluate the proposed difference detection system, a reference model of an indoor lab environment was created from the data set recorded at 22 robot

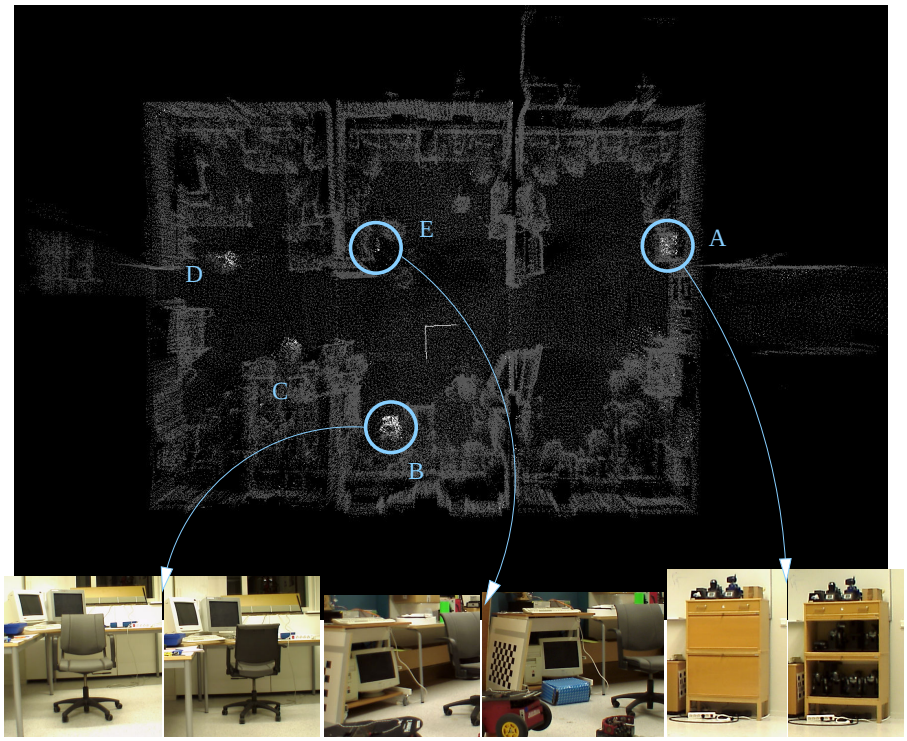

Figure 10.8: Difference probability as found in 24 scans compared to the reference model shown in Fig. 10.7. Brighter regions indicate a higher difference probability. Difference probability was computed from range values only using Eq. 10.5. The pictures in the lower part of the figure show the changes that occurred at the marked region. The left picture in the pairs is the previous state (reference) and the right one contains the change. A - a drawer has been opened. B - a chair is moved. C - a chair is moved. D - a chair is moved. E - a blue box is added.

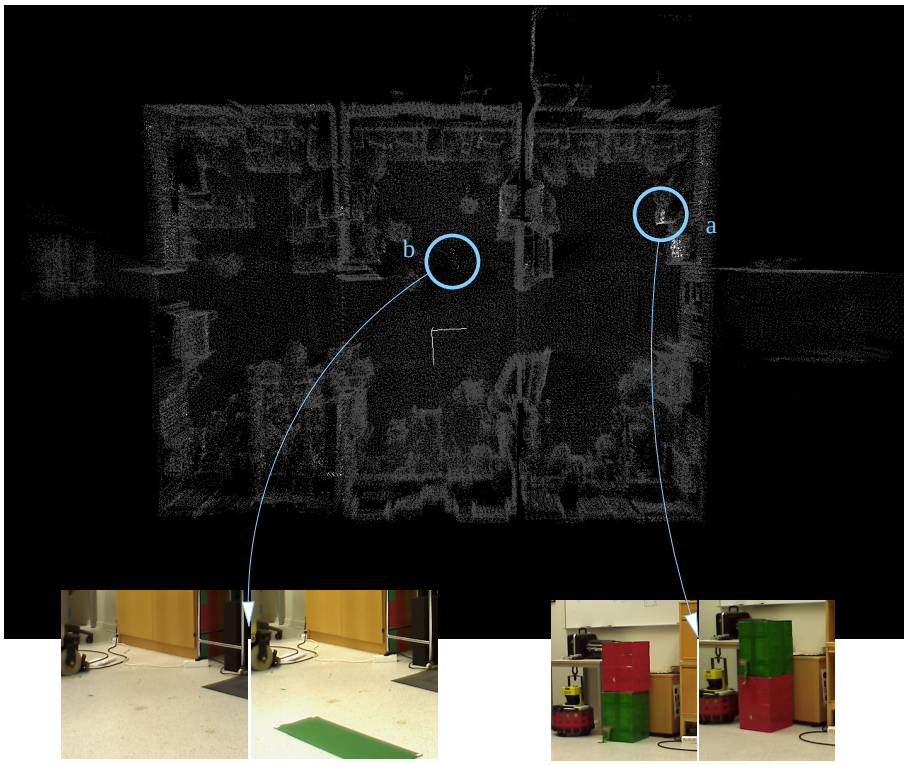

Figure 10.9: Difference probability as found in 24 scans compared to the reference model shown in Fig. 10.7. Brighter regions indicate a higher difference probability. Difference probability was scomputed from colour information using Eq. 10.6. The pictures in the lower part of the figure show the changes that occurred at the marked region. The left picture in the pairs is the previous state (reference) and the right one contains the change. a - two similar boxes with different colour were swapped. b - a coloured paper stripe has been placed on the floor. There are a few parts of the images which indicate non-existing changes (false alarms), which is probably due to different camera viewpoints that caused different illumination and different reflections from ceiling lamps.

poses (reference data set), i.e. from 66 laser scanner sweeps and 154 camera images. The reference model is shown in Fig. 10.7. Then, controlled changes (described in the Results section 10.6.1) were introduced and a new data set was recorded at 24 different poses (difference detection data set). The two data sets overlap with each other and the difference detection data set were treated independently from each other, i.e. an a priori unknown position was assumed for the 24 poses in the difference detection data set.

10.6.1 Results

Fig. 10.8 shows the result of the evaluation experiment: the point cloud of the combined difference detection scans (registered for better visualisation) are shaded according to the computed difference probability. Brighter regions indicate a higher probability of changes in the environment. Fig. 10.8 shows the difference probability obtained from spatial data only, i.e. applying Eq. 10.5. Fig. 10.9 shows the difference probability obtained from colour data only, i.e. using Eq. 10.6.

All the changes to the reference model can be found in the difference probability point clouds. The changes are indicated with an alphabetic character in Fig. 10.8: a sliding door that was opened (A), three chairs that were moved (B,C,D), and a relatively small box of approx. $0.15 \times 0.25 \times 0.4$ m^3 that was repositioned (E). Further changes are not detectable using range data only but can be found when using colour data, see Fig. 10.9. These changes are two equally sized boxes (approx. $0.4 \times 0.4 \times 0.5$ m^3) that were swapped (a) and a coloured paper stripe (approx. 0.1×0.7 m^2) that was fixed to the floor (b).

10.7 Conclusion

In this chapter a system for autonomous change detection suitable for a security patrol robot was presented. The method uses vision and 3D range data to build a reference model of the environment and detect changes with respect to this model. The approach was verified in a real world experiment in an indoor lab environment, demonstrating that the proposed system is able to register 3D scans and to detect changes in spatial data and also to detect changes that occur in colour space and are not observable using range values only. Apart from the description of this system for autonomous difference detection, the particular contribution is the introduction of novel methods including the 3D-NDT representation with adaptive cell size to efficiently represent both the spatial and colour aspects of the reference model.

Part IV
Conclusions

Chapter 11
Conclusions and Future Work

This chapter summarises the scientific contributions presented in this thesis and suggests directions for future work.

11.1 Summary

The basic skills required by mobile robots for navigation in a real world environment include the capability to create and maintain a useful representation of the environment, typically in the form of map, and to determine the robot's pose within the map. These fundamental skills, or fundamental problems are therefore a central topic in mobile robotics research. How these problems can be addressed depends to a large extent on the sensor modalities used.

This thesis addresses the fundamental problems of registration, localisation and simultaneous localisation and mapping (SLAM) using cameras as the primary sensor modality in combination with odometry or a 3D laser range scanner. A major principle used is to consider local features obtained from camera images. These features describe small parts of an image instead of the whole image at once, which has several benefits especially regarding robustness to environmental changes. A second key attribute of the methods proposed in this thesis is that the local features do *not* have to be tracked over a sequence of images. The majority of alternative methods that use local features for mobile robot navigation track a large number of local features over successive frames in order to improve the accuracy of the position estimate and this tends to be computationally expensive. To avoid tracking local visual features, the work presented in this thesis partly relies on appearance based approaches, which compute how similar two images are without extracting geometrical properties of the environment. Here the key idea is to relate image similarity with the hypothesis that the images were taken at a similar position. The measure of similarity used in this thesis is based on the number of corresponding local visual features between two images.

Two different sensory setups are considered in this thesis. The first, referred to as Omni-directional sensor configuration, consists of an omni-directional camera and the robot's odometry. The second sensory setup, referred as 3D Vision sensor configuration, consists of a standard planar CCD camera and a 2D laser range scanner, both mounted on a pan/tilt unit attached to a mobile robot. These two sensory setups cover together a wide application area in mobile robotics. The first configuration is very minimalistic both in terms of size and weight, which makes it suitable for cheap and small robots. The second sensory setup is more costly and sophisticated and covers another segment of the mobile robot application domain with high demands on accuracy but not so hard financial constraints.

11.1.1 Omni-directional sensor configuration

This sensor setup consists of odometry and an omni-directional camera, realised by a special mirror attached to a standard camera. Despite the inexpensive hardware, the SLAM method proposed in Chapter 6 "Mini-SLAM" was found to create consistent maps in medium- to large-scale indoor and outdoor environments (robot path lengths up to 1.4 km) in a computational efficient manner. Odometry is used to obtain relative pose estimates between successive frames, meaning that it is only used over small distances where it is sufficiently accurate. Apart from the accuracy of odometry over short distances, Mini-SLAM relies on the discriminative power of local visual features that allow robust data association and thus large-scale map correction through closed loops. To avoid unnecessary image comparisons and to improve robustness with respect to perceptual aliasing, a covariance estimate of the mobile robot's pose is used to limit the area in which potential matches are expected. The robustness of the method was evaluated by corrupting the input consisting of odometry readings and appearance based similarity measures. Mini-SLAM was shown to be especially stable against odometric errors showing a graceful degradation of the performance. To the best knowledge of the author, the combination of a purely appearance based approach with odometry is completely new in the context of visual-SLAM.

The Mini-SLAM method was used to address the multi-robot mapping problem, that is to fuse data collected by multiple robots into a single map without knowing the relative poses between the data sets. It was shown to not only consistently fuse the data sets but also to improve the consistency of the joint map.

The same appearance based similarity measure computed from the number of matching local visual features was used for global localisation with respect to an existing map, i.e. for localisation without any initial pose estimate. The map consists of local visual features extracted from a set of omni-directional images with known positions (the positions could be the output of the Mini-SLAM method, for example). A particle filter approach was used that accumulates

sensor evidence possibly supporting multiple pose hypotheses over time. Apart from the methodological achievement, the main contribution of this work is the long term investigation in a dynamic populated environment over several months, and the robustness evaluation with up to 90% virtual occlusions. The results show that even if the environment has gone through changes after the map was created, for example, by moved objects or simulated by virtual occlusions, it is still possible to obtain good position estimates.

11.1.2 3D Vision sensor configuration

The second sensory setup considered in this thesis consists of a standard 1 megapixel CCD camera mounted together with a 2D laser range finder onto a pan/tilt unit (to obtain 3D range data) fitted to a mobile robot. Odometry was available but not used in this sensor configuration. By not relying on pose sensors such as odometry, which incrementally integrate measurements over time, it is demonstrated that the methods can process sensor data from two poses for which no relative position is available. They would therefore be suitable to handle sensory readings from multiple robots, for example.

Local image features are central to many of the approaches proposed in this thesis. An important task is therefore to combine high-resolution camera images with the low-resolution 3D laser range data to obtain depth estimates for each pixel (or sub-pixel) by an image dependent interpolation of the 3D laser range values. The main idea is to use colour similarity as an indication of depth similarity, based on the observation that depth discontinuities in the scene often correspond to colour or brightness changes in the camera image. A set of interpolation methods using colour information is proposed and evaluated using real world indoor and outdoor data. The results showed that the accuracy is dependent on the scene content. This is particularly striking comparing indoor and outdoor environments. In addition to the interpolation methods, four measures of the confidence of the interpolation accuracy are proposed. The evaluation shows a clear correlation between the interpolation accuracy and the confidence measure. By using the interpolation method for each pixel in the camera image it is possible to obtain range estimates with a resolution that is an order of magnitude higher than provided from the laser scanner data alone. To obtain "Super Resolution" in this way constitutes an application area in itself. With respect to the application of local visual features for robot navigation, non-iterative interpolation methods are proposed, which allow to obtain depth estimates in predictable, constant time for a set of sub-pixel coordinates representing the image position of the extracted local visual features. It was found that non-iterative methods gave a similar performance as the iterative method.

Using the discriminative property of local visual features together with their 3D position estimated with the proposed interpolation method creates a novel possibility to register 3D point clouds and thus to obtain the relative position between two robot poses. The key aspects of the proposed registration method

is that it uses local visual features to handle correspondences, which makes the registration work without initial relative pose estimates assuming that there is some overlap between the data and no perceptual aliasing. Most registration methods, and especially laser based approaches, rely on initial pose estimates that in many cases are not available at all or not with the required accuracy. The performance usually degrades quickly with the quality of the initial pose estimates. The proposed method was shown to accurately register successive data collected both in indoor and outdoor environments. By considering the covariance estimate of each local visual feature in addition to the position estimate produced higher registration accuracy outdoors and in cases where only a few corresponding features were found, for example, due to a small overlap. A large depth covariance means that the position is likely to be inaccurate. Even so, the accuracy of the bearing angle is not related to the accuracy of the depth estimate and contains useful information.

The proposed 3D registration method, which combines an appearance based approach with 3D metric data, has also been used for 3D metric mapping and 3D metric global localisation. Both the mapping and localisation methods exploit the property that no initial pose estimates are required to obtain accurate relative poses from registration. To select candidate robot poses for registration an appearance based measure of image similarity is used. The mapping problem is then treated as an optimisation problem where the distance between corresponding local visual feature pairs, weighted with a covariance estimate (a large covariance corresponds to a lower weight), is minimised. The mapping method was shown to improve the consistency of an indoor map by an order of magnitude compared to using incremental registration alone.

The corresponding localisation method uses a combination of global appearance based localisation, which determines the map pose with the most similar appearance, followed by a local visual feature based registration. This method was able to accurately localise the robot globally without any initial pose estimates in an indoor environment.

Finally, an approach to difference detection is proposed that combines all the methods developed for the "3D Vision sensor configuration" (interpolation, registration, mapping and localisation). The purpose is to determine changes in real 3D environments. This task is very difficult to perform for a human and very useful for security robots. An additional contribution developed for the proposed difference detection approach is the combination of spatial and colour coordinates. This makes it possible to detect spatial changes but also changes that only occur in colour space. The method was shown to both detect rather small spatial changes but also to detect colour changes alone. In order to detect the changes correctly, it is essential that the registration method is very accurate. Thus, the results also demonstrate the accuracy of both the generated map and the localisation methods. Another property, which is important for change detection, is that the registration method does not rely on good initial pose estimates. Such pose estimates will often not be available because this

would require to maintain a consistent coordinate system over the potentially long duration between building the reference model of the environment and the detection of changes.

11.2 Conclusions

A central observation applied and confirmed in this thesis is that local visual features are very discriminative and therefore particularly suitable to address the problem of data association. An important contribution of this work is to demonstrate that tracking of these local visual features over multiple frames as it is used in standard approaches is not necessary neither to obtain metric maps, nor to perform metric localisation. This is achieved by a combination of visual appearance based similarity measures with rough relative pose estimates to obtain accurate metric maps. Because the local features are not tracked over multiple frames, the SLAM problem can be addressed using a graph based structure, which can be optimised in an efficient manner. A graph structure where each node is a robot pose is commonly used in laser based approaches but rarely in vision based approaches. The novel aspect lies in the appearance based visual registration method that is used to incorporate visual relations into the graph. Two different cameras with different resolution and with different curved mirror lenses have been utilised. This demonstrates that the proposed visual appearance based methods works directly without any additional modifications such as tedious calibration.

Another contribution of this work is the combination of a vision sensor with a 3D laser scanner to obtain accurate position (and covariance) estimates at the sub-pixel level, for example of the position of local visual features. Several non-iterative methods were proposed to obtain an interpolated depth value for an arbitrary (sub)-pixel position in the camera image and to determine a confidence value of the performed interpolation. Outdoor environments typically involve longer range measurements, which means that the resolution is decreased while natural outdoor objects, such as trees, have a comparably thin and complex structure. Therefore to properly exploit the principle that colour and intensity changes correlate with changes in depth, it is important that the resolution of the laser is high enough to be able to handle the structure of natural objects. This implies that to fully utilise the interpolation methods outdoors (with natural objects) would typically require a higher angular resolution of the laser range scanner compared to indoor environments.

In the proposed 6DOF registration method, only a small degradation in performance was observed at the turning points of the robot, which means that the overlap between the successive scans can be small. In case of a pure laser based systems, a small overlap between scans rises problems especially related to data association. However, if only small, the observed degradation demonstrates that even with the comparatively reliable data association obtained from the local visual features, the amount of overlap will influence the registration accu-

racy. In our case, the size of the overlap will influence the number of matched features, but also the angular distribution of the matched features, which especially affects the angular accuracy. A small overlap means that the feature matches are located within a narrow "cone" and that the number of feature matches is comparably low.

11.3 Future Work

One important question that needs further investigation is how to combine the two different sensory setups and associated approaches so as to exploit their complementary strengths. The first sensory setup (Omni-directional sensor configuration) corresponds to methods that rely on visual appearance and odometry information for mapping and localisation in an efficient way rather than with high accuracy. The second sensory setup (3D Vision sensor configuration) instead gives high accuracy in a comparatively complex, costly and time consuming manner. Many robotic applications do not require an accurate global pose estimate. Instead the accuracy of the relative pose estimate between the robot and an object to be manipulated is of higher interest. For example, a high relative pose accuracy is needed for an autonomous forklift truck to pick up pallets and to unload them in a pallet rack. In the area between the docking locations, the accuracy of the localisation estimate is less crucial. A sensible approach for this scenario would be to utilise the methods proposed in Part II (using the Omni-directional sensor configuration) for global localisation whereas relative poses would be estimated using the methods from Part III (using the 3D Vision sensor configuration). In general, the methods from Part II are suitable for large-scale navigation, whereas the methods from Part III are more suitable for accurate small-scale, high-resolution navigation.

Though the proposed approaches are different in terms of accuracy and efficiency they share many fundamental aspects and therefore the 3D Vision sensor configuration can be utilised together with the methods from Part II (not the other way around since the methods in Part III also rely on metric information about the local visual features). For example, the global localisation approach, using the 3D Vision sensor configuration can easily be extended to use the particle filter based approach proposed using the Omni-directional sensor setup in Part II, Section 5.2. This can be done without changing the representation of the map (the map consists of a set of visual features for each node, which are used to determine the image similarity and, together with the position and covariance estimate, to determine the relative pose between two frames). Similarly, the "Mini-SLAM" approach proposed in Part II can be used to obtain a map based on visual appearance and odometry only and this map could then be used to initialise the mapping method proposed in Part III.

The current bottleneck in all methods is the time to calculate the similarity measure, that is, to match the local visual features. This has been addressed by using the likelihood of the robot pose to decrease the amount of required

comparisons. Other approaches to speed up the comparisons need to be integrated. This includes faster search methods, for example, approximate nearest neighbour or to utilise dedicated hardware, for example, FPGAs or GPUs. Also alternative visual features could be considered with a smaller size of the descriptor, for example, the SURF feature. A further possibility to be investigated is to utilise global features as a filtering step before the more computationally demanding local visual feature comparisons are carried out.

Currently, the similarity measure used is based on the number of matched features which means that local features occurring from repetitive structures, for example, a brick wall are treated in the same way as features that occur very rarely. Hence, by weighting the visual features based on how often they occur will increase the robustness towards problem related to perceptual aliasing.

A problem not addressed in this thesis is how to create an optimal map. This includes the problem to determine when a new omni-directional image, or new data from the 3D laser range scanner and the planar camera, should be added into the map. This should be done adaptively to keep the map small while assuring that it contains a sufficient amount of data in areas where changes occur rapidly, for example. A related topic for future work is to investigate how to determine where the robot should move to (exploration), to increase the information content (resolution and coverage) and the consistency of the created map.

A further related topic, is the issue of handling dynamic changes, such as moved objects but also variations due to illumination and seasons, in an efficient manner. This can be addressed by determining for each potential new node if it should directly be added to the map, if an old node should be replaced or if the candidate node should only be used for localisation purposes.

11.3.1 Omni-directional sensor configuration

The methods proposed in Part II were all tested with omni-directional cameras, therefore it would be interesting to extend the proposed methods to work with other types of sensors, for example, to use a planar camera equipped with a wide angle lens or even to solely use a 3D-laser range scanner in an appearance based manner.

Another line of research would be to aim at a pure vision-based system, by replacing odometry by estimating relative pose estimates directly from the camera (visual-odometry).

Even though the tested environment contained modest elevation changes, the proposed methods have so far only been using a 2D representation of the world. A natural extension is therefore to implement and test the proposed methods in full 6DOF environments, for example, using a helicopter or an underwater platform.

11.3.2 3D Vision sensor configuration

Local visual features are very robust to changes in the environment. What would be interesting is to evaluate whether the proposed appearance based registration method using local visual features, is more robust to environmental changes than, for example, the commonly used ICP method using range data.

The proposed difference detection method currently uses the full image resolution only to obtain the local visual features, whereas the actual difference detection is performed at the resolution of the range data. It would be possible to either use the interpolation method so as to obtain a high resolution (coloured) 3D point cloud or to extend the difference detection to do difference detection in the image. Also, the difference detection should be evaluated in an outdoor environment. Difference detection is closely related to the problem of dynamic mapping. For example, one could present incremental differences over a time frame, which could be suitable for monitoring the progress of a construction site. To use and evaluate other sensor modalities such as thermal cameras for difference detection could, for example, be used to monitor machinery and heat dissipation in houses over time.

Part V

Appendices

Appendix A
Notations and Symbols

Part I

p	pixel	35
x, y	pixel coordinates	35
σ	scale	35
I	image	37
L	scale space	37
$*$	convolution operator	37
G	Gaussian function	37
s	scales per octave	37
D	Difference of Gaussian (DoG) function	38
$\mathbf{x}$	interest point position (x, y, σ)	38
$\hat{\mathbf{x}}$	scale space extremum	38
$\mathbf{H}$	Hessian matrix	39
m	gradient magnitude	40
θ	gradient orientation	40
H	descriptor histogram vector	41
$\mathcal{N}(F)$	sub-window of feature F	42
F	image feature	43
S	similarity measure	43
P	image feature pairs	43
$M_{a,b}$	number of matched feature pairs	43
n_{F_a}	number of features in F_a	44
$\mathbf{M}$	similarity matrix	44

Part II

μ	relative pose	50
$\mu_{x,y}$	relative position	50
μ_θ	relative orientation	50

C	relative pose covariance 50
$C_{x,y}$	relative position covariance 50
σ^2_θ	relative orientation variance 50
$N(a)$	neighbouring frames of a 56
(x, y, θ)	robots pose 63
$p(X_t\|Z_{1:t})$	probability of being in state X_t given measurements $Z_{1:t}$ 62
X_t	system state 62
S_t	particle set at time t 62
$Z_{1:t}$	history of measurements up to time t 62
z_t	measurement at time t 62
N	number of particles 62
$x_t^{(i)}$	particle (containing the robots 2D pose) 62
$\pi_t^{(i)}$	particle weight 62
u_t	odometry reading 63
$\hat{X}_t$	system state estimate at time t 63
(x, y)	robots position 63
θ	robots orientation 63
$S^{(i)}$	similarity measure for particle $x_t^{(i)}$ 64
f_w	weighting function 64
f_{forget}	factor in a forgetting inertia model 65
f_{keep}	factor in a keep random inertial model 65
T	minimum distance travelled between database poses 70
r	relation 83
r_o	relation based on odometry 83
r_v	relation based on visual similarities 83
$\mathcal{R}$	set of relations 85
m	number of relations 85
n	number of frames 85
μ_r	relative pose for relation r 85
C_r	relative pose covariance for relation r 85
$\hat{x}$	vector of estimated poses 85
d	total distance travelled by the robot 85
t	total angular rotation by the robot 85
X_d, X_t	odometry parameters forward motion 85
Y_d, Y_t	odometry parameters side motion 85
δ_d, δ_t	odometry parameters rotation 85
$\hat{x}_b$	estimated pose for frame b 88
$C_{\hat{x}_b}$	estimated pose covariance for frame b 88
$C_{x_a^o}$	covariance obtained from odometry at frame a 89
t_{vs}	similarity threshold 89

x_i^{DGPS}	Differential GPS position for frame i	92

Part III

N	number of laser range measurements	110
$\mathbf{r}$	laser range scanner measurement	110
(θ, π)	pan and tilt angles of laser beam	110
r	laser distance measurement	110
(x, y, z)	3D point in Euclidean coordinates	110
$\mathbf{P}$	image pixel	110
(X, Y)	pixel coordinates	110
C	three-channel colour value	110
$\mathbf{R}$	projected laser range reading	110
r^*	estimated range value	110
$\mathbf{R}^*$	interpolated point	110
p	likelihood	110
σ	point distribution variance	110
σ_d	pixel point distribution variance	110
σ_c	pixel colour distribution variance	110
V	Voronoi diagram	111
NN	natural neighbour	111
A	intersecting Voronoi cell area	111
w	weight of natural neighbour	111
w_c	colour based weight	111
W^c	normalization factor	113
(ψ, ϕ)	MRF framework minimization constraints	113
(k, c)	MRF framework parameters	114
$\overline{e}$	mean range error r relative to the ground truth	119
o_t	percentage of outliers using a threshold t (in meters)	119
S	extracted planar surface	122
$\mathbf{n}$	normal vector	122
$\mathbf{z}_{cam}$	optical axis of the camera	122
$\mathcal{S}_c$	current/latest 3D scan	127
$\mathcal{S}_p$	previous 3D scan	127
$\mathbb{R}$	rotation matrix	127
$\mathbf{t}$	translation vector	127
μ_F	visual feature position estimate in 3D	127
C_F	visual feature covariance matrix in 3D	128
p_0	closest projected laser range reading	128
M	number of surrounding laser points	128
J	optimization constraint	128
(p_i^p, p_i^c)	corresponding point pair from scan pose $\mathcal{S}_p$ and $\mathcal{S}_c$	128

C_p	covariance matrix of point p	129
$\mathcal{N}$	number of corresponding points	133
d	Euclidean pose error (in meters)	133
α	sum of rotational errors (in radians)	133
$\mathbf{x}$	robot pose in 3D	141
K	optimization constraint for two robot poses	142
L	optimization constraint for a pose graph	142
V	binary variable based on the similarity measure S	142
P	plane fitted to the estimated poses $\mathbf{x}_{1..n}$	142
$\mathcal{S}_s$	the most similar scan of $\mathcal{S}_c$	147
$\mathbf{P}$	point cloud	157
p	3D point	157
$\mathbf{C}$	set of cells used in 3D-NDT representation	157
c	a single cell in 3D-NDT representation	157
N_c	number of point in a cell c	157
p_c	points located in cell c	157
μ_c	mean position of points p_c in a cell c	158
C_c	covariance of points p_c in a cell c	158
v_c	volume of covariance ellipsoid	158
λ_i	eigenvalue	158
e_c^{max}	eigenvector with the highest eigenvalue	158
λ_c^{max}	highest eigenvalue of covariance C_c matrix	158
(c_i^a, c_i^b)	new cells created by splitting	158
$\breve{p}$	corresponding 3 channel colour values to point p	159
$\breve{\mu}_c$	colour mean value of cell c	159
$\breve{C}_c$	colour covariance value of cell c	159
$\breve{\lambda}_c^{max}$	highest eigenvalue of covariance $\breve{C}_c$ matrix	159
$p_{diff}(p)$	difference probability of point p	159
$p_{colourdiff}(\breve{p})$	colour difference probability of colour $\breve{p}$	160

Appendix B
External and Internal Camera Calibration

This appendix describes the calibration method to automatically estimate both the internal and external parameters of the camera. The internal parameters describe how a 3D point in the camera coordinate system is projected to 2D pixel coordinates. The external parameters contains the relative rotation and translation of another coordinate system, in this case the wrist joint. Hence, the external parameters is used to describe how a 3D point given in the in wrist joint coordinates is projected to a 3D point in the camera coordinate system.

B.1 Introduction

The methods described in Part III are all using projection of 3D coordinates into 2D pixel coordinates and vice verse. Therefore, the accuracy in calibration results directly affect the results in those methods. The calibration routine proposed in this appendix is done using a special calibration board, see Fig.B.1. The calibration board contains a chessboard pattern and has reflective stripes on the side. This combination makes it easy to both detect the board (using the corners of the chessboard squares) in the camera image and (using the remission values, i.e. the intensity value of the returned laser beam) in the laser range data. The external parameters for the camera is obtained by optimising the SSD between the chessboard corners location obtained from the camera images and the laser data. In the laser range data the location of the chessboard corners are inferred from the location of the reflective stripes.

B.2 Internal Calibration

The main principle of determining the internal parameters is to combine multiple images containing the chessboard pattern, Fig. B.2, taken from various

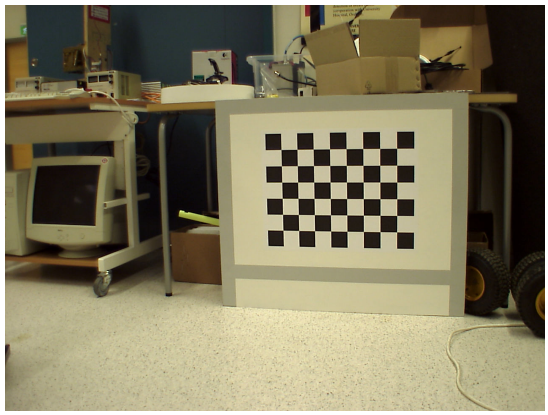

Figure B.1: Calibration board, consisting of a standard chessboard pattern and reflective stripes (grey).

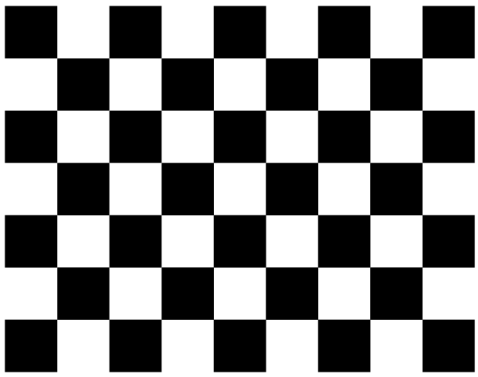

Figure B.2: Calibration pattern.

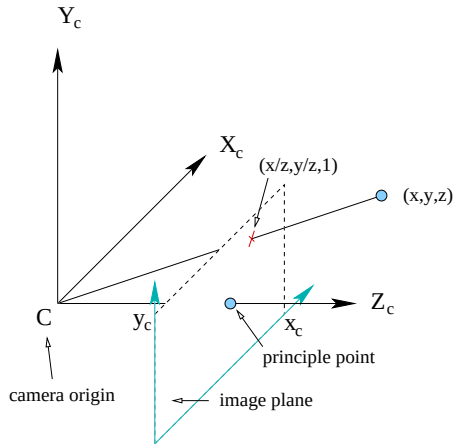

Figure B.3: Overview of a camera coordinate system.

viewpoints and for each image to determine the pixel coordinates of the intersecting squares. The distance d_{square} between each chessboard corner point (the length of the square) is known and is in this case 54mm.

The set of internal parameters to be obtained in the camera calibration [56] consists of a calibration matrix K and a distortion vector d.

The calibration matrix K consists of 4 parameters

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{B.1}$$

where α_x and α_y is the focal length of the camera in pixel coordinates. Two variable are used to express the focal length due to the fact that cameras sensor array elements are typically not formed as perfect squares. The principle point (x_0, y_0) is given by the pixel coordinates of the point where the principle axes passes through the image plane, see Fig. B.3.

To handle lens distortion, a distortion vector d is used. The distortion vector d consists of 4 parameters. $d = [k_1, k_2, p_1, p_2]$ where k_1 and k_2 are radial distortion coefficients and p_1 and p_2 are tangential distortion coefficients in polynomials (see Eq. B.4-B.5).

The chessboard pattern provides geometrical constraints d_{square}, the internal parameters, calibration matrix K and the distortion vector d can therefore be found by minimising the distance between the chessboard corners located in the image and the projected ones as recorded by the camera, see Fig. B.4. Note that the K matrix is relative to the cameras own coordinate system therefore

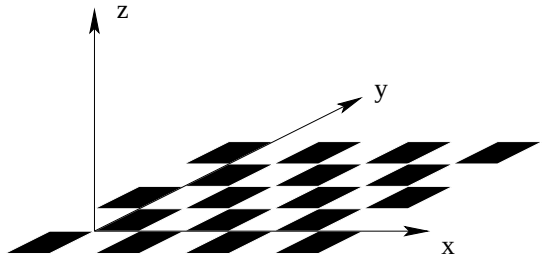

Figure B.4: Location of the chessboard corner points in 3D.

the rotation $\mathbf{R}_n$ and translation $\mathbf{t}_n$ of the chessboard pattern in each calibration image n has to be determined.

Given a 3D point $X = (x, y, z)$ in the camera coordinate system (X_c, Y_c, Z_c), see Fig. B.3, the projected point $\tilde{X} = (\tilde{x}, \tilde{y}, 1)$ is defined as

$$\tilde{x} = x/z \tag{B.2}$$

$$\tilde{y} = y/z \tag{B.3}$$

To get the point in the distorted image $\tilde{X}' = (\tilde{x}', \tilde{y}', 1)$ (in camera 3D coordinates) the distortion coefficients are used

$$\tilde{x}' = \tilde{x}(1 + k_1 r^2 + k_2 r^4) + 2p_1\tilde{x}\tilde{y} + p_2(r^2 + 2\tilde{x}^2) \tag{B.4}$$

$$\tilde{y}' = \tilde{y}(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2\tilde{y}^2) + 2p_2\tilde{x}\tilde{y} \tag{B.5}$$

where $r^2 = \tilde{x}^2 + \tilde{y}^2$.

Finally $\tilde{X}'$ is converted into the pixel coordinate system X_p as

$$X_p = K\tilde{X}'. \tag{B.6}$$

This method uses the principal point (x_0, y_0) as the centre of the radial distortion calculations.

B.3 External Calibration

Up to now, the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ has be used to denote the pose of the calibration board with respect to the camera, which from now on is denoted $\mathbf{R}_{image}$ and $\mathbf{t}_{image}$.

The external parameters typically contains information about the orientation (rotation matrix $\mathbf{R}_{cam}$) and location (translation vector $\mathbf{t}_{cam}$) of a camera in a robot coordinate frame. One issue here is that the camera is not fixed in the

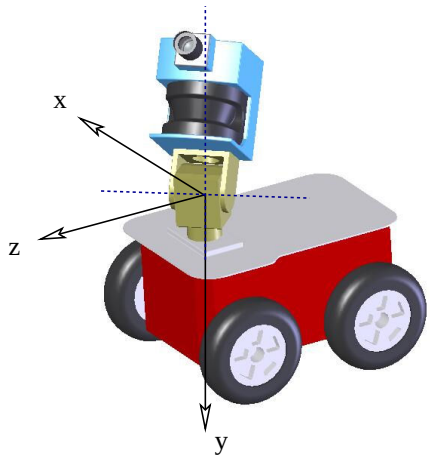

Figure B.5: Coordinate system of the robot. The dotted lines indicate the two rotation axis (pan/tilt) of the wrist at its current position.

robot coordinate frame since the camera location is dependent on angles of the wrist joint. The calibration parameters are therefore selected to be the relative to the wrist joint. The robot coordinate system is then defined to have the same origin as the wrist joint, see Fig. B.5.

The robot coordinate system is defined in Fig. B.5. It was selected following the convention commonly used to describe image coordinates that is, the top left (0,0) and the depth are denoted z.

Because the laser scanner is also mounted on the wrist, the external parameters for the laser is also relative to the wrist joint. Hence, four different sets of parameters has to be found; two for the laser scanner ($\mathbf{R}_{laser}$, $\mathbf{t}_{laser}$) and two for the camera ($\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$).

B.3.1 Laser Calibration

The laser offset with respect to the wrist joint is currently not calibrated, instead it has been estimated based on the cad drawings obtained from the manufacturers. The external parameters are set to: $\mathbf{R}_{laser} = \mathrm{I}$ where I is the 3x3 unit matrix and $\mathbf{t}_{laser} = [0, -0.1378, 0.0605]$.

B.3.2 Camera Calibration

To find the $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$ we exploit the fact that the laser calibration parameters ($\mathbf{R}_{laser}, \mathbf{t}_{laser}$) are known. It is therefore possible to use the laser data transformed into the robot coordinate system. One key problem is to find the

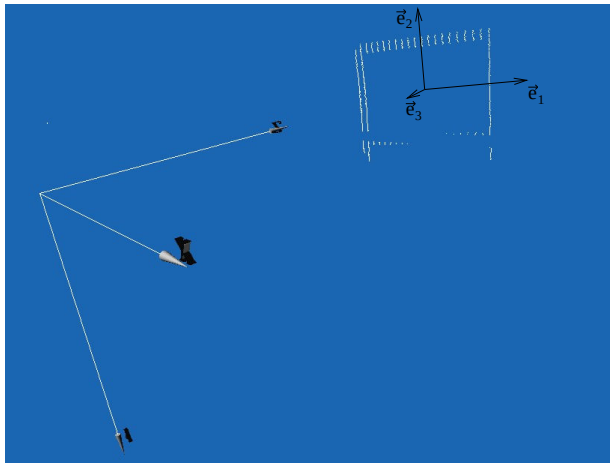

Figure B.6: Segmented scan data based on remission values. Note that all the points from the reflective area are present (and only a few points remains from the rest of the scan data). The data are clustered to remove any extra outliers.

pose of the calibration board $\mathbf{R}_{board}$ and $\mathbf{t}_{board}$ within the laser data with respect to the the robot coordinate system. The pose of the calibration board is obtained by using the remission values from the laser scanner since the calibration board has reflective stripes on the edges (see Fig. B.1), it is possible to find these edges by thresholding the remission values, see Fig. B.6.

To assure that the thresholded scan data only contain data points from the calibration board, the scan data are clustered and only the cluster with the largest amount of laser points is kept. Defining this subset of highly reflective points as S_{calib} and $n = |S_{calib}|$, the plane which contains the calibration board can be extracted by calculating the covariance matrix C_{calib} for the scan S_{calib} as

$$C_{calib} = \frac{1}{n-1} \sum_{i=1}^{n} (p_i - \mu_{calib})^2 \tag{B.7}$$

where

$$\mu_{calib} = \frac{1}{n} \sum_{i=1}^{n} p_i. \tag{B.8}$$

The eigenvectors $\vec{e}_{1,2,3}$ and the corresponding eigenvalues $\lambda_{1,2,3}$ for C_{calib} are calculated and sorted based on the eigenvalues so that $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$. The calibration board is estimated to be located in a plane P with normal vector $\vec{n} = \vec{e}_3$ containing the point μ_{calib}.

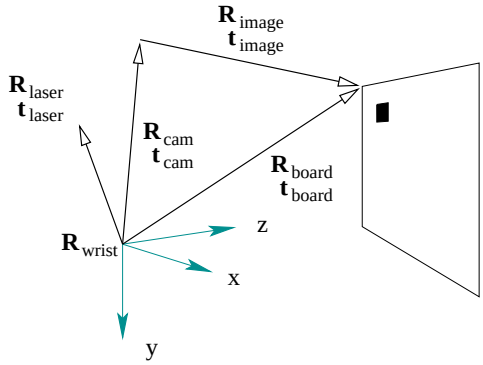

Figure B.7: The external parameters to be found are $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$. All other rotation matrices and translation vectors have been determined.

The size of the board is found by projecting all points S_{calib} on to the vectors $\vec{e}_{1..3}$ and finding the $min_{1..3}$ and $max_{1..3}$ value for each vector. The translation $\mathbf{t}_{board}$ (to the top left corner), see Fig. B.7, is then calculated as

$$\mathbf{t}_{board} = \mu_{calib} + min_1\vec{e}_1 + max_2\vec{e}_2 + 0.5(min_3 + max_3)\vec{e}_3. \quad (B.9)$$

The goal is to exact the chessboard points from the laser scanner, we need to get the correct orientation of the board. Because the board is not symmetrical, see Fig. B.8 the orientation of the board can be found by comparing the centre of the board with the centre μ_{calib} calculated from S_{calib}. To be sure that the calibration board was not accidentally positioned upside down, we look at the min_2 and max_2 value (corresponds to the $\vec{e}_2$ vector). If $|min_2 > max_2|$ the translation is instead calculated (compared to Eq. B.9 and Fig. B.8) as

$$\mathbf{t}_{board} = c + min_1\vec{e}_1 + max_2\vec{e}_2 + 0.5(min_3 + max_3)\vec{e}_3. \quad (B.10)$$

The rotation matrix $\mathbf{R}_{board}$ for the calibration board is calculated as

$$\mathbf{R}_{board} = \begin{bmatrix} \vec{e}_{1_x} & \vec{e}_{2_x} & \vec{e}_{3_x} \\ \vec{e}_{1_y} & \vec{e}_{2_y} & \vec{e}_{3_y} \\ \vec{e}_{1_z} & \vec{e}_{2_z} & \vec{e}_{3_z} \\ \cdot & & \end{bmatrix} \quad (B.11)$$

To find the pose of the calibration board the orientation of the plane created by $\vec{e}_1$ and $\vec{e}_2$ has to be estimated, see Fig. B.9. The parameter θ is therefore determined by minimising the area A of the rectangle containing all scan points S_{calib} calculated as

$$A(\theta) = |(max_1 - min_1)R_\theta\vec{e}_1| \cdot |(max_2 - min_2)R_\theta\vec{e}_2|, \quad (B.12)$$

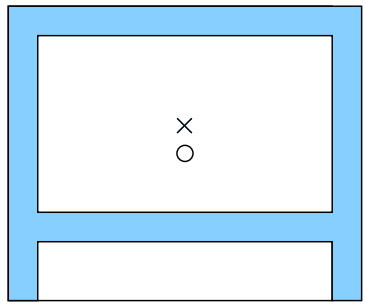

Figure B.8: The reflective areas on the calibration board, shown in grey, and the mean value of the chessboard pattern $\circ$ and the reflective material $\times$. Since the reflective area is not symmetrical the mean value of the scan points located on the reflective material will not be located in the centre of the calibration board.

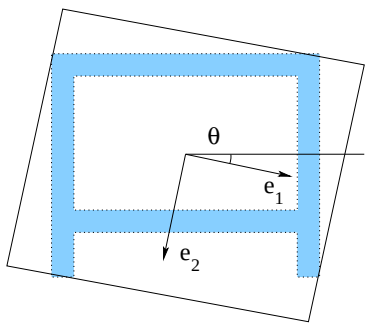

Figure B.9: The angle θ of the orientation of the board used to obtain an orientation estimate of the calibration board. In this case the optimal solution is when $\theta = 0$.

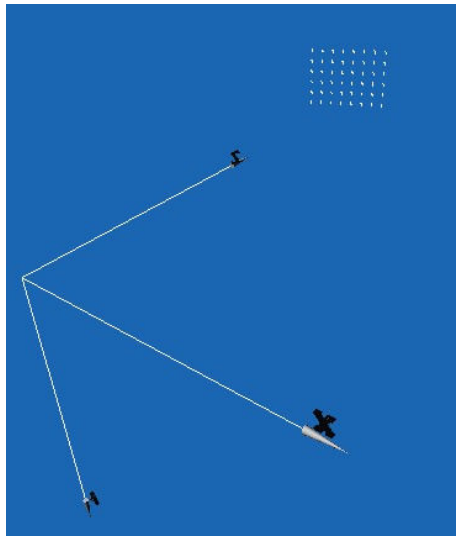

Figure B.10: A result of calibration. The points represent estimated positions of the chessboard corners extracted from camera images shown in white (using the obtained parameters $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$) and laser scan data shown in red.

where the rotation matrix $\mathbf{R}_\theta$ describes a rotation by the angle angle θ around the $\vec{e}_3$ axis.

We have now two different measurements of the location of the board, one from the camera image $\mathbf{R}_{image}$ and $\mathbf{t}_{image}$ (see Section B.2) the one extracted from the laser scanner data, $\mathbf{R}_{board}$ and $\mathbf{t}_{board}$.

What is left is to find the $\mathbf{t}_{cam}$ and $\mathbf{R}_{cam}$ parameters which is now fairly strait forward (we now have the parameters for the $image$, the $board$ and the $wrist$ parameters, see Fig. B.7).

The parameters $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$ are found by minimising the SSD of the chessboard corners found in the images with the calculated position of the chessboard corners obtained from the laser scan data. The chessboard corners in the laser scan data are easily obtained from the board position (the location of the chessboard pattern offset is known as well as the size of each square). The distance d is the distance from one point to the corresponding point. The optimisation method used is the Fletcher-Revies gradient method. A calibration result is presented in Fig. B.10.

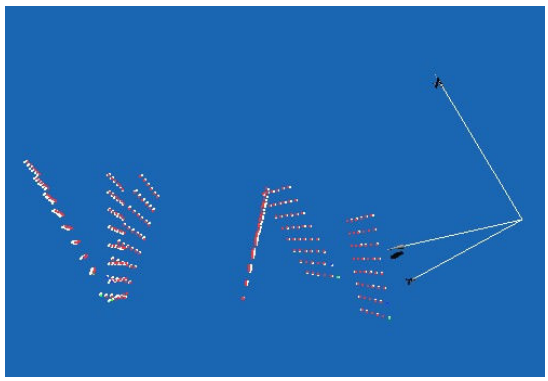

Figure B.11: Calibration result using multiple calibration scans (seen from top / down perspective). The points shows estimated positions of the chessboard corners extracted from camera images shown in white (using the obtained parameters $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$) and laser scan data, shown in red, for all calibration board poses. In this case all data are used simultaneously to find $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$.

Bibliography

[1] MESA Imaging AG. http://www.mesa-imaging.ch.

[2] Securitas Inc. http://www.securitas.com.

[3] A. Doucet, N. de Freitas and N. Gordon, editor. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Berlin, Heidelberg, New York NY, 2001.

[4] M. Adams, S. Zhang, and L. Xie. Particle filter based outdoor robot localization using natural features extracted from laser scanners. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, volume 2, pages 1493–1498, 2004.

[5] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3348–3353, 2005.

[6] M. Artac, M. Jogan, and A. Leonardis. Mobile robot localization using an incremental Eigenspace model. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1025–1030. IEEE, 2002.

[7] M. Artač and A. Leonardis. Outdoor mobile robot localisation using global and local features. In Danijel Skočaj, editor, *Computer vision - CVWW '04 : proceedings of the 9 th Computer Vision Winter Workshop*, pages 175–184. Slovenian Pattern Recognition Society, 2004.

[8] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[9] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[10] T.D. Barfoot. Online visual motion estimation using FastSLAM with SIFT features. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pages 579–585, 2005.

[11] J. Beis. Indexing without invariants in model-based object recognition, 1997.

[12] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[13] P. Biber, H. Andreasson, T. Duckett, and A. Schilling. 3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*. IEEE/RSJ, 2004.

[14] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *Proc. of Robotics: Science and Systems (RSS)*, June 8-11 2005.

[15] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2003.

[16] P. Biber and W. Straßer. nScan-matching: Simultaneous matching of multiple scans and application to SLAM. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2270 – 2276, 2006.

[17] P. Blaer and P. Allen. Topological mobile robot localization using fast vision techniques. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1031–1036. IEEE, 2002.

[18] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only slam in large unstructured environments. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4820–4826, 2007.

[19] G.R. Bradski. *Open Source Computer Vision Library*. Intel Corporation, 2001.

[20] W. Burgard, D. Fox, G. Lakemeyer, D. Haehnel, D. Schulz, W. Steiner, S. Thrun, and A. Cremers. Real robots for the real world — the rhino museum tour-guide project. In *Proc. of the 1998 AAAI Spring Symposium*, 1998.

[21] P. Buschka. *An Investigation of Hybrid Maps for Mobile Robots*. PhD thesis, Dept. of Technology, Örebro University, SE-701 82 Örebro, Sweden, 2005.

[22] A. Cassandra, L. P. Kaelbling, and J. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pages 963–972. IEEE/RSJ, 1996.

[23] Y. Chen and G. Medioni. Object Modelling by Registration of Multiple Range Images. *Image and Vision Computing*, 10(3):145–155, 1992.

[24] I. J. Cox. Blanche: position estimation for an autonomous robot vehicle. *Autonomous robot vehicles*, pages 221–228, 1990.

[25] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1403–1410, 2003.

[26] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. IEEE, 1999.

[27] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 291–298. MIT Press, Cambridge, Massachusetts, 2006.

[28] G. Dissanayake, H. F. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1009–1014, 2000.

[29] T. D'Orazio, F.P. Lovergine, M. Ianigro, E. Stella, and A. Distante. Mobile robot position determination using visual landmarks. *IEEE Transaction on Industrial Electronics*, 41(6):654–662, 1994.

[30] T. Duckett, S. Marsland, and J. Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, 2002.

[31] T. Duckett and U. Nehmzow. Mobile robot self-localisation and measurement of performance in middle scale environments. *Robotics and Autonomous Systems*, 24(1–2):57–69, 1998.

[32] H. F. Durrant-Whyte, D. Rye, and E. Nebot. Localization of autonomous guided vehicles. In *Proc. of the 8th Int. Symposium on Robotics Research*, pages 613–625, 1995.

[33] E. Pagello E. Menegatti, M. Zoccarato and H. Ishiguro. Image-based Monte-Carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[34] A. I. Eliazar and R. Parr. Learning probabilistic motion models for mobile robots. In *Proc. of the twenty-first Int. Conf. on Machine learning (ICML)*, page 32, 2004.

[35] P. Elinas and J. Little. σMCL: Monte-Carlo localization for mobile robots with stereo vision. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[36] P. Elinas, R. Sim, and J. Little. σSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1564–1570, 2006.

[37] S. P. Engelson. *Passive map learning and visual place recognition*. PhD thesis, Yale University, New Haven, CT, USA, 1994.

[38] V. Étienne and R. Laganière. An empirical study of some feature matching strategies. In *Proc. 15th Int. Conf. on Vision Interface*, pages 139–145, 2002.

[39] N. Fairfield, G. A. Kantor, and D. Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 2007.

[40] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7:149–153, 1964.

[41] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*., July 1999.

[42] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

[43] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[44] J. H. Freidman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software*, 3(3):209–226, 1977.

[45] U. Frese. *An* $O(\log n)$ *Algorithm for Simultaneous Localization and Mapping of Mobile Robots in Indoor Environments*. PhD thesis, University of Erlangen-Nürnberg, 2004.

[46] U. Frese. Treemap: An $O(\log n)$ Algorithm for Indoor Simultaneous Localization and Mapping. *Autonomus Robots*, 21(2):103–122, 2006.

[47] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, 2005.

[48] C. Früh and A. Zakhor. 3D model generation for cities using aerial photographs and ground level laser scans. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 31–38, 2001.

[49] J. Gonzalez-Barbosa and S. Lacroix. Rover localization in natural environments by indexing panoramic images. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1365–1370. IEEE, 2002.

[50] M. A. Goodrich, D. R. Olsan Jr., J. W. Crandall, and T. J. Palmer. Experiments in adjustable autonomy. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2001.

[51] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3d. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2007.

[52] G. Grisetti, D. Lordi Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online constraint network optimization for efficient maximum likelihood map learning. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Pasadena, CA, USA, 2008.

[53] G. Grisetti, C. Stachniss, and W Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 32:16–25, 2006.

[54] C. Schroeter H.-M. Gross, A. Koening and H.-J. Boehme. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pages 1031–1036. IEEE/RSJ, 2003.

[55] N. Haala and Y. Alshawabkeh. Application of photogrammetric techniques for heritage documentation. In *2nd Int. Conf. on Science & Technology in Archaeology & Conservation*, 2003.

[56] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[57] K. L. Ho and P. Newman. Loop closure detection in SLAM by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

[58] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. In *Proc. of Robotics: Science and Systems (RSS)*, June 2005.

[59] R. Powers I. Nourbakhsh and S. Birchfield. Dervish: An office-navigation robot. *AI Magazine*, 16(2):53–60, 1995.

[60] K. Ishikawa, J. Takiguchi, and M. Hatayama. Parking-vehicles recognition using spatial temporal data (a study of mobile robot surveillance system using spatial temporal gis part 2). In *IEEE Int. Workshop on Safety, Security and Rescue Robotics (SSRR)*, 2005.

[61] C. Jennings and D. Murray. Stereo vision based mapping and navigation for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1694–1699, 1997.

[62] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1944–1950, 2006.

[63] M. Jogan and A. Leonardis. Parametric eigenspace representations of panoramic images, workshop on omnidirectional vision applied to robotic orientation and nondestructive testing (ndt). In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, pages 31–36. IEEE, 2001.

[64] A. Johnson and S. B. Kang. Registration and integration of textured 3-d data. In *InternationalConference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, pages 234 – 241, May 1997.

[65] S. J. Julier and J. K. Uhlmann. Using covariance intersection for slam. *Robotics and Autonomous Systems*, 55(1):3–20, 2007.

[66] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM algorithm for robust localization and mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 24–29, 2005.

[67] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pages 3232–3238, 2003.

[68] J. Kosecka and F. Li. Vision based topological Markov localization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1481–1486. IEEE, 2004.

[69] B.J.A. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, 2001.

[70] R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard. Monte carlo localization in outdoor terrains using multi-level surface maps. In *Proc. of the International Conference on Field and Service Robotics (FSR)*, Chamonix, France, 2007.

[71] L. Kunze, K. Lingemann, A. Nüchter, and J. Hertzberg. Salient visual features to help close the loop in 6d slam. In *ICVS Workshop on Computational Attention & Applications - WCAA 2007*, 2007.

[72] J. J. Leonard and H. F Durrant-Whyte. *Directed sonar sensing for mobile robot navigation*. Kluwer Academic Publishers, Boston, 1992.

[73] J. J. Leonard and H. F. Durrrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics*, 7(3):376–382, 1991.

[74] T. S. Levitt and D. T. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3):305–360, 1990.

[75] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 131–144. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[76] A. Lilienthal, F. Streichert, and A. Zell. Model-based shape analysis of gas concentration gridmaps for improved gas source localisation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3575 – 3580, 2005.

[77] T. Lindeberg. *Scale-Space Theory in Computer Vision (The International Series in Engineering and Computer Science)*. Springer-Verlag, Berlin, Heidelberg, New York NY, December 1993.

[78] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann. High-speed laser localization for mobile robots. 51(4):275–296, 2005.

[79] J. N. K. Liu, M. Wang, and B. Feng. ibotguard: an internet-based intelligent robot security system using invariant face recognition against intruder. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 35(1):97–105, 2005.

[80] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[81] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[82] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[83] M. Magnusson and T. Duckett. A comparison of 3d registration algorithms for autonomous underground mining vehicles. In *Proc. of the European Conference on Mobile Robots*, 2005.

[84] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[85] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers: Field reports. *J. Field Robot.*, 24(3):169–186, 2007.

[86] O. Martínez Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1742–1747, 2005.

[87] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[88] M. Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

[89] H.P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 116 – 121, 1985.

[90] L.-P. Morency and T. Darrell. Stereo tracking using icp and normal flow constraint. In *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, volume 4, pages 367–372, 2002.

[91] M. Proetzsch N. Schmitz, J. Koch and K. Berns. Fault-tolerant 3d localization for outdoor vehicles. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pages 941–946, 2006.

[92] P. M. Newman, D. M. Cole, and K. L. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1180–1187, 2006.

[93] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-based laser scan matching for outdoor 6d slam. In *Proc. KI: Advances in Artificial Intelligence. 28th Annual German Conference on AI*, pages 304–319, 2005.

[94] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269, 2006.

[95] L. Paletta, S. Frintrop, and J. Hertzberg. Robust localization using context in omnidirectional imaging. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2072–2077. IEEE, 2001.

[96] J. Porta and B. Kröse. Appearance-based concurrent map building and localization. *Robotics and Autonomous Systems*, 54(2):159–164, 2006.

[97] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[98] A. Ranganathan, E. Menegatti, and F. Dellaert. Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics*, 22:92– 107, 2005.

[99] D. C. Asmar S. M. Abdallah and J. S. Zelek. Towards benchmarks for vision SLAM algorithms. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1542–1547, 2006.

[100] J.M. Sáez and F. Escolano. 6dof entropy minimization slam. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1548–1555, 2006.

[101] R. San-Jose, A. Brun, and C.-F. Westin. Robust generalized total least squares iterative closest point registration. In *Seventh Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, 2004.

[102] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[103] M. Seeman, M. Broxvall, A. Saffiotti, and P. Wide. An autonomous spherical robot for security tasks. In *IEEE Int. Conf. on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS)*, 2006.

[104] V. Sequeira, J. Goncalves, and M.I. Ribeiro. 3d reconstruction of indoor environments. In *Proc. of Int. Conf. on Image Processing (ICIP)*, pages 405–408, 1996.

[105] J. Shi and C. Tomasi. Good features to track. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[106] R. Sibson. *A brief description of natural neighbour interpolation*, pages 21–36. John Wiley & Sons, Chichester, 1981.

[107] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1080–1087, 1995.

[108] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[109] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[110] S. Thrun, W. Bugard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, Massachusetts, 2005.

[111] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4270–4275, 2003.

[112] R. Triebel. *Three-dimensional Perception for Mobile Robots*. PhD thesis, Inteligent Autonoumous Group, Albert-Ludwigs Universität, Freiburg im Breisgau, Germany, 2007.

[113] R. Triebel and W. Burgard. Improving simultaneous mapping and localization in 3d using global constraints. In "*Proc. of the National Conference on Artificial Intelligence (AAAI)*", 2005.

[114] R. Triebel, B. Frank, J. Meyer, and W. Burgard. First steps towards a robotic system for flexible volumetric mapping of indoor environments. In "*Proc. of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV)*", 2004.

[115] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2006.

[116] J.K. Uhlmann. *Dynamic map building and localization for autonomous vehicles*. PhD thesis, University of Oxford, 1995.

[117] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1023–1029. IEEE, 2000.

[118] C. Valgren, T. Duckett, and A. J. Lilienthal. Incremental spectral clustering and its application to topological mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4283–4288, 2007.

[119] N. Vlassis, B. Terwijn, and B. Kröse. Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 7–12, 2002.

[120] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omni-directional vision for robot navigation. In *Proc. of the IEEE Workshop on Omnidirectional Vision (OMNIVIS)*, page 21. IEEE Computer Society, 2000.

[121] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.

[122] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. Ground truth evaluation of large urban 6d slam. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pages 650–657, 2007.

[123] B. Yamauchi. A frontier based approach for autonomous exploration. In *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, page p. 146, 1997.

[124] D. C. K. Yuen and B. A. MacDonald. A comparison between extended kalman filtering and sequential monte carlo techniques for simultaneous localisation and map-building. In *In Proc. of the Australasian Conference on Robotics and Automation (ARAA)*, pages 111–116, 2002.

[125] Z. Zivkovic, B. Bakker, and B. Kröse. Hierarchical map building and planning based on graph partitioning. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 803–809, 2006.

Index

Publikationer *i serien* Örebro Studies in Technology

1. Bergsten, Pontus (2001) *Observers and Controllers for Takagi – Sugeno Fuzzy Systems*. Doctoral Dissertation.
2. Iliev, Boyko (2002) *Minimum-time Sliding Mode Control of Robot Manipulators*. Licentiate Thesis.
3. Spännar, Jan (2002) *Grey box modelling for temperature estimation*. Licentiate Thesis.
4. Persson, Martin (2002) *A simulation environment for visual servoing*. Licentiate Thesis.
5. Boustedt, Katarina (2002) *Flip Chip for High Volume and Low Cost – Materials and Production Technology*. Licentiate Thesis.
6. Biel, Lena (2002) *Modeling of Perceptual Systems – A Sensor Fusion Model with Active Perception*. Licentiate Thesis.
7. Otterskog, Magnus (2002) *Produktionstest av mobiltelefonantenner i mod-växlande kammare*. Licentiate Thesis.
8. Tolt, Gustav (2003) *Fuzzy-Similarity-Based Low-level Image Processing*. Licentiate Thesis.
9. Loutfi, Amy (2003) *Communicating Perceptions: Grounding Symbols to Artificial Olfactory Signals*. Licentiate Thesis.
10. Iliev, Boyko (2004) *Minimum-time Sliding Mode Control of Robot Manipulators*. Doctoral Dissertation.
11. Pettersson, Ola (2004) *Model-Free Execution Monitoring in Behavior-Based Mobile Robotics*. Doctoral Dissertation.
12. Överstam, Henrik (2004) *The Interdependence of Plastic Behaviour and Final Properties of Steel Wire, Analysed by the Finite Element Metod*. Doctoral Dissertation.
13. Jennergren, Lars (2004) *Flexible Assembly of Ready-to-eat Meals*. Licentiate Thesis.
14. Jun, Li (2004) *Towards Online Learning of Reactive Behaviors in Mobile Robotics*. Licentiate Thesis.
15. Lindquist, Malin (2004) *Electronic Tongue for Water Quality Assessment*. Licentiate Thesis.
16. Wasik, Zbigniew (2005) *A Behavior-Based Control System for Mobile Manipulation*. Doctoral Dissertation.

17. Berntsson, Tomas (2005) *Replacement of Lead Baths with Environment Friendly Alternative Heat Treatment Processes in Steel Wire Production*. Licentiate Thesis.

18. Tolt, Gustav (2005) *Fuzzy Similarity-based Image Processing*. Doctoral Dissertation.

19. Munkevik, Per (2005) *"Artificial sensory evaluation – appearance-based analysis of ready meals"*. Licentiate Thesis.

20. Buschka, Pär (2005) *An Investigation of Hybrid Maps for Mobile Robots*. Doctoral Dissertation.

21. Loutfi, Amy (2006) *Odour Recognition using Electronic Noses in Robotic and Intelligent Systems*. Doctoral Dissertation.

22. Gillström, Peter (2006) *Alternatives to Pickling; Preparation of Carbon and Low Alloyed Steel Wire Rod*. Doctoral Dissertation.

23. Li, Jun (2006) *Learning Reactive Behaviors with Constructive Neural Networks in Mobile Robotics*. Doctoral Dissertation.

24. Otterskog, Magnus (2006) *Propagation Environment Modeling Using Scattered Field Chamber*. Doctoral Dissertation.

25. Lindquist, Malin (2007) *Electronic Tongue for Water Quality Assessment*. Doctoral Dissertation.

26. Cielniak, Grzegorz (2007) *People Tracking by Mobile Robots using Thermal and Colour Vision*. Doctoral Dissertation.

27. Boustedt, Katarina (2007) *Flip Chip for High Frequency Applications – Materials Aspects*. Doctoral Dissertation.

28. Soron, Mikael (2007) *Robot System for Flexible 3D Friction Stir Welding*. Doctoral Dissertation.

29. Larsson, Sören (2008) *An industrial robot as carrier of a laser profile scanner. – Motion control, data capturing and path planning*. Doctoral Dissertation.

30. Persson, Martin (2008) *Semantic Mapping Using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle*. Doctoral Dissertation.

31. Andreasson, Henrik (2008) *Local Visual Feature based Localisation and Mapping by Mobile Robots*. Doctoral Dissertation.