Master's Thesis

Evaluation of Sensors and Mapping Approaches for Disasters in Tunnels

Max Leingartner, BSc

2015

Institute for Software Technology
Graz University of Technology

Supervisor and first reviewer: Univ.-Prof. Dipl.-Ing. Dr. techn. Franz Wotawa
Second reviewer: Ass. Prof. Dipl.-Ing. Dr. techn. Gerald Steinbauer

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

Date Signature

Abstract

In this thesis I present results of an evaluation of sensors and perception algorithms in a large-scale emergency response exercise. I deployed state-of-the art sensors like Lidars and publicly available mapping approaches in a simulated car accident in a tunnel. The main goal is to investigate how well existing technologies are accepted by first responders for such scenarios. A rich sensor data set was recorded during a reconnaissance mission with a robot and later analyzed off-line. I present results of the representations generated and discuss what techniques are already accepted by responders. Finally, I raise issues that have to be tackled in order to increase the acceptance.

Acknowledgement

Deploying a mobile robot in a large-scale emergency exercise is a lot of work and can not be done by a person alone. There were a bunch of people involved during the preparation, execution and post-processing phase which helped and supported me during the work for this master thesis.

First of all I want to thank Gerald Steinbauer and Franz Wotawa as my supervisors, who guided me through the whole thesis work. I also want to thank my colleagues Johannes Maurer and Clemens Mühlbacher for helping me during the preparation and the exercise itself.

I like to acknowledge the support of Andreas Trattler and ProgenoX GmbH for providing the sensor mounting box. Moreover, I like to thank Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE) for providing us the Odin robot for the exercise. Alexander Ferrein also needs to be thanked for as he came to Graz and provided us with the expensive Velodyne Scanner. I also like to thank Funkfeuer Graz who supported us with knowledge and equipment to setup a reliable WiFi network in the tunnel. Moreover, I like to thank Alexander Kleiner for the access to his implementation of the voting-based scan matching and Christof Hoppe and Christian Mostegel for access to their implementation of the Structure-from-Motion algorithm. Last but not least I like to thank Heimo Krajnz, Ernst Zechner and Mario Augustini from the professional fire brigade Graz for their support during the participation in the exciting tunnel exercise and the discussion of the results.

Personally I also want to thank my good friend Philipp and Maria who supported and encouraged me during the process of writing.

This work was supported by the European Fund for Regional Development (EFRE), the Land Steiermark and the Republic of Slovenia under the TEDUSAR grant.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

First responder risk their lives whenever they are in action. In cases like the search for injured people, detecting toxicities or to slack a fire, a special equipped robot could support first responder and take risk off them by using certain sensors for detecting people or by creating a map of the disaster area. With this help, first responder get a better knowledge about the area and are able to better estimate the risks of sending men into the place of action.

From that perspective, you can say that ground or aerial robots equipped with advanced sensing technologies such as 3D laser scanners and advanced mapping algorithms are deemed useful as a supporting technology for first responders. Lots of excellent research in the field exists, however, practical applications at real disaster sites are scarce. The reasons are manifold. One reason is that most systems and algorithms are neither suitable nor robust enough for the harsh and dynamic environments typically found in disaster relief. Another reason is that the systems are difficult to operate and a deployment in a real disaster need specially skilled and trained personnel. Such personnel is usually found at research institutions but can hardly be found in emergency response units. In order to gain acceptance for such technologies by first responders, the system has to be suitable, robust and it must be possible being operated by first responders.

A lot of research is conducted to equip robots with advanced capabilities such as autonomous exploration or object manipulation. In spite of this, realistic application areas for such robots are *teleoperated reconnaissance* or *search*. There is a good chance that new technologies in these particular fields will find their ways into regular operations in the next years as progress is made with ground and areal vehicles, sensors and algorithms. In this thesis I investigate how well state-of-the-art and off-the-shelf components and algorithms are suited for reconnaissance in disaster relief scenarios. The basic idea is to deploy the most common sensors and algorithms

in a disaster situation and evaluate how well the components work for these scenarios. For the initial evaluation we focused on sensors and mapping algorithms; Robot mapping has reached an advanced status and has a high potential for direct deployment.

Productive years of collaboration between the robotics group at IST and the Austrian fire brigade and gaining more acceptance year after year, led to an invitation to a large-scale emergency exercise and the underlying thesis.

Figure 1.1: First Responder entering the tunnel

1.1 Problem statement

As mentioned before, the main goal of this thesis is to get as many sensors as possible in the tunnel, record all the data and evaluate which sensors and algorithms perform best in disaster scenarios. The problem with these kind of scenarios is, that one has to cope with monotonic structures, fire, smoke, humidity and many people moving around which can affect sensor data or the performance of algorithms. Another goal for this thesis is to evaluate the usefulness of the resulting maps for first responders, get to know which information is important for them, and how such a system can support first responders in action. For the evaluation, we equipped a ground robot with typical sensors ranging from spectral, thermal and stereo cameras, 2D and 3D laser scanners to the Kinect. The robot can be teleoperated over long distances (up to 1 km) and sends back and records the sensor data. In order to test the robot in a most realistic scenario it was deployed as a scout in a large fire brigade emergency exercise in a street tunnel. The robot

was teleoperated towards an accident site inside the tunnel, see Figure 1.1. Raw and preliminary processed sensor data were immediately presented to the responders outside the tunnel. More complex mapping approaches were run off-line after deployment in the tunnel. All available data of the robot and the sensors have been recorded. This allowed us to evaluate different state-of-the-art sensors in a challenging realistic environment. Moreover, due to the active integration of the robot and its data into the mission, direct feedback from the responders to usefulness of the presented data representations was available.

1.2 Contribution

The main contributions of this thesis are (1) the evaluation of state-of-the-art and off-the-shelf technologies in a realistic emergency exercise, (2) a direct feedback of first responders on the acceptance of these technologies in their daily operation, (3) to provide a rich set of data from the emergency exercise that other researcher easily can reuse for their own work.

Furthermore a paper was accepted by the 11th IEEE International Symposium on Safety, Security, and Rescue Robotics [12].

The remainder of the thesis is organized as follows. In the next chapter, the robot and sensors setup is described followed by a description of the emergency exercise site and the data recordings. In Chapter 2, related research on mapping and public datasets is presented. In Chapter 6 we briefly discuss the outcome of different mapping algorithms on the dataset, before we evaluate the used algorithms in Chapter 7. Then, we give examples for improvements in Chapter 8 and an outlook on future work and conclude.

Chapter 2

Related Research

Related research was done by [13, 2, 1] exploring mines or other harsh outdoor environments[14, 15]. Research groups deploying different sensors for mapping tasks are, for instance, G. Brooker [16], D. Vivet [17] and R. Hahn [18] performing research on radar and thermal sensing technologies. But this area is not well researched and it could be a problem finding some commonly available software and hardware.

In the next sections the main points that are relevant for this thesis are discussed regarding related research.

2.1 Underground Scenarios

Within this section related underground scenarios are examinded. It will be interesting to see if any special precautions had to be done before running the experiments. It will help prepare the Loiblpass exercise and to circumvent mistakes that were done before.

Wong et al. [1] compared different kinds of range sensors (Time of Flight, Structured Light, Stereo Vision) by recovering a scene geometry in an underground area. The work they did is very closely related, but the environment from which the data for the geometry reconstruction was taken, was much smaller than our exercise site (see Figure 2.1 left). The corridors had a length in the range of 20 to 60 meters with nearly no features existing. To have a better mapping result, they placed blue fiducial cubes in the corridors. This is not possible in a real exercise scenario and therefore not applicable at the Loiblpass exercise. But in general their main goal was to evaluate sensors and not mapping algorithms and so the conclusion is that time-of-flight

sensors like the Sick LMS scanner provide the best results although the Microsoft Kinect camera outperforms cost greatly.

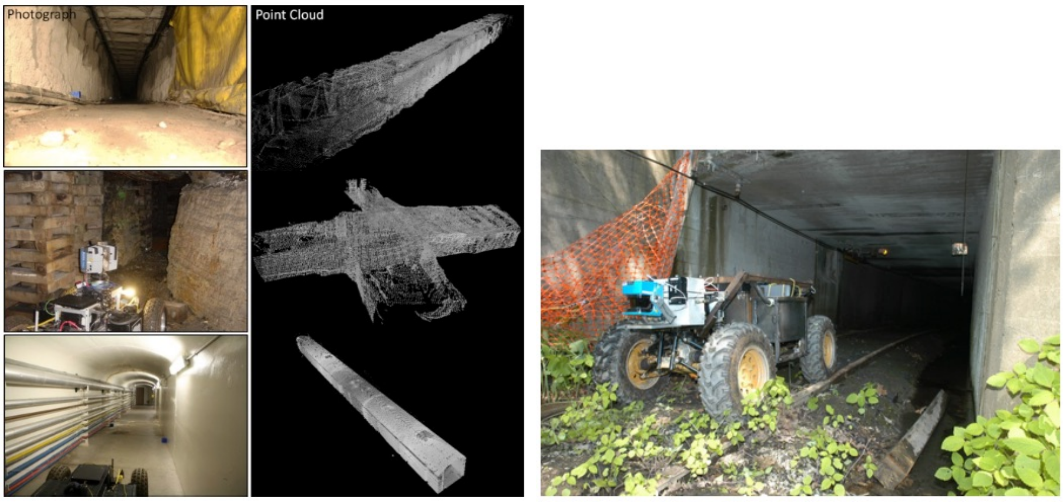


Figure 2.1: Monotonic structures in the scenarios of Wong et al. [1] (left) and A. Nüchter et al. [2] (right)

Another experiment was done by A. Nüchter et al. [2]. They explored abandoned mines with a tilting laser scanner deploying a full volumetric map. The mine consists of two 1.5 km long corridors which branches into numerous side corridors. Also this group had to cope with monotonic structures and rare features, as seen in Figure 2.1 right. Only the ground had some structure, railway tracks and other rumble commonly found in mines, which could have helped with the mapping process. For the robot itself, these obstacles were no problem because it had the necessary torque to overcome this barriers.

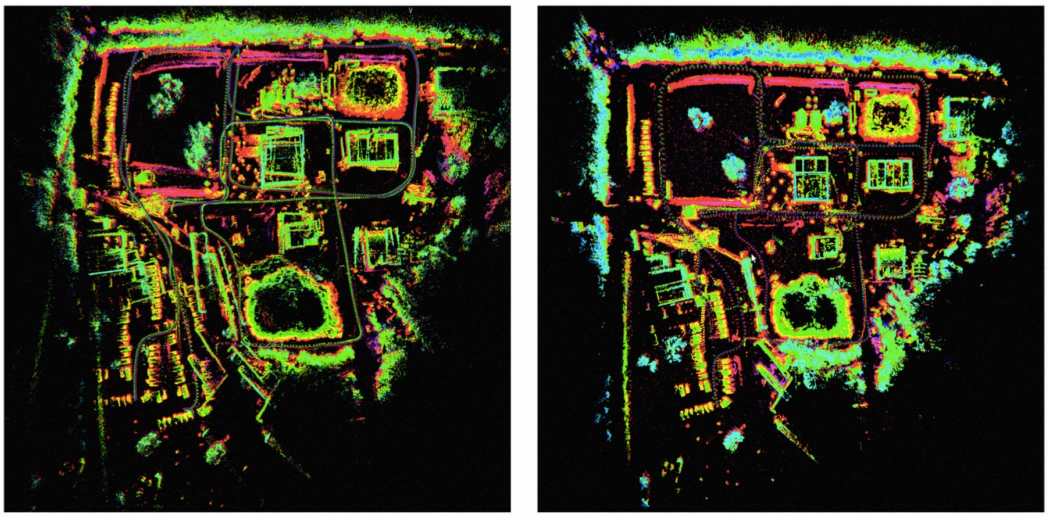
So both research groups have tunnel like structures and long flat walls in their scenarios which can make the mapping process hard because of lacking features. As this work has also some mapping processes included, their mapping approaches will be further examined in section 2.2.

2.2 Large-Scale Mapping Approaches

As mentioned before we will further discuss the mapping approaches of Wong et al. and A. Nüchter et al. and also some other related work regarding mapping in large areas. So Wong et al. [1] used blue fiducial cubes that were placed in surveyed locations to enable fast stitching of incremental models and the final alignments were tuned using iterative closest point (ICP), which is described in Section 5.7.6. The point clouds were taken from static locations but there is no further information on how they did their mapping or what software was used as their main goal was to evaluate different kinds of sensors.

A. Nüchter et al. explored abandoned mines with a tilting laser scanner deploying a 6D simultaneous localization and mapping (SLAM) algorithm. The data acquisition was done in the mine, the mapping process was done afterwards when the robot returned. They acquired 3D scans in a stop and go fashion and registered these scans with a variant of the iterative closest point algorithm which takes roughly 9 seconds per scan. Their method is called simultaneous matching where the first scan is the master scan and determines the coordinate system. Then a queue is initialized with the new scan. Afterwards the algorithm repeats three steps until the queue is empty. 1) If the current scan is the first scan of the queue, it is removed. 2) If the current scan is not the master scan, a set of all scans that overlap with the current scan is calculated and aligned with the ICP algorithms. 3) If the current scan changes its location then each single scan of the set of neighbors is added to the end of the queue. This method is completely automatic, no manual alignment has to be done.

As the algorithm looks promising, a more advanced version is being used for evaluation in this thesis and further information can be found in Section 6.2.2.



(a) Plain vanilla ICP: Whole map.

(b) 6D SLAM software: Whole map.

Figure 2.2: Comparison between 6DSlam approaches of J.Pellenz et al. (a) and A. Nüchter et al. (b)

A real-time 3D mapping algorithm was presented by J. Pellenz et al. in [15]. Unlike Nüchter et al., their approach does not make use of loop closing in their algorithm for performance reasons. The size of their point clouds was also reduced by using only areas with significant, feature-rich objects which a extracted by a terrain classification algorithm. A drawback of using this algorithm is that the data reduction introduces some aliasing, resulting in non optimal matching. Therefore, the algorithm can be applied in real-time using simple ICP at the cost of a lower map quality. See Figure 2.2 for comparison between the two algorithms. The left image shows for example multiple walls at regions that are visited more than once. This is due to the lack of loop closure techniques.

2.3 Harsh and Dynamic Environments

This section refers to some work that analyzes the effects of environmental influences, like smoke or dust, on different kinds of sensors.

In [19] the authors investigated how different densities of smoke effect measurements of laser scanners. In an experimental setup with changing smoke, density range measurements against

different surfaces were recorded. The smoke density was estimated by the change of the color saturation of a reference object. The basic observation was that above a certain density the range measurements suddenly drop to almost zero. The authors used the results to define a mathematical model and implemented it in a simulation environment for usage in the RoboCup Rescue League.

Figure 2.3: Static (left) and dynamic (right) test area for camera and laser scanners [3].

The authors of [3] present a similar dataset for sensors in challenging environments. They recorded data from cameras and laser scanners in outdoor scenarios with different environmental conditions (e.g., dust, smoke, or rain) as you can see in figure 2.3. Their results show that lasers are extremely affected by smoke, which leads to false detection of large obstacles. The same applies to camera images although infrared cameras return slightly better results as the penetration power is higher. Radar on the other hand performs much better. As radar operates at a bigger wavelength, it is much less affected by smoke or dust. This approach is similar to our attempt to provide rich datasets for the evaluation of perception algorithms. In contrast to their work, we focus on the integration of 3D sensors and the evaluation of sensors and algorithms (mainly SLAM) in emergency response scenarios. Also the duration of their tests was very short (about 2 minutes each) and the area very small in comparison to a long street tunnel like the Loiblpass.

2.4 Registration Benchmarking

The acquired point clouds need to be compared against a ground truth, and to allow a qualitatively statement on the correctness of these point clouds. This chapter shows some related research on this topic.

Pomerleau et al. [4] present a dataset of point clouds for the evaluation of registration algorithms. The dataset comprises series of point clouds recorded in different indoor and outdoor environments. The point clouds were obtained using a tilting laser scanner. To generate a ground truth they used a theodolite (see Figure 2.4) to track specialized prisms fixed on a mobile platform to validate visual odometry performance. The system reduces infrastructure installation, in comparison to motion capture systems such as VICON, has a fixed precision (2 mm in position, $0.2°$ in attitude) and is independent of environmental location. GPS for example is only usable in free open outdoor scenarios where no obstacles are between satellites and the GPS module.

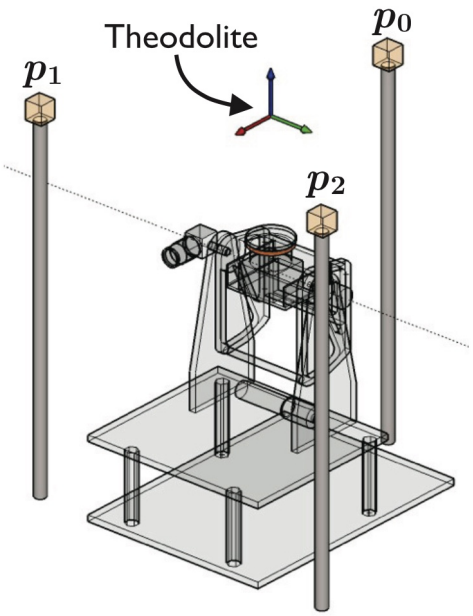


Figure 2.4: System of Pomerleau et al. [4]. Perspective view with the positions of the three prisms around the platform used to reconstruct the global pose.

In [20], the authors used the above dataset as a basis for a well-founded evaluation process for

ICP-based registration methods for point clouds. The authors provided an open-source library to customize individual registration approaches. Using the above dataset and a large set of test runs with varied parameters (e.g., initial pose) a statistical performance evaluation was made. Although, the process is designed for registration of scans and not for a complete map, it contains some valuable thought about reliable experimentation. Their conclusion after testing different variants of the ICP algorithm is, that there is the need for improved ICP methods for information-deprived and unstructured environments.

In [21], two evaluation methods for verifying the quality of obtained maps for RGB-D SLAM [11] systems is proposed. These methods are used for performance measurements of visual odometry and visual SLAM systems. The paper addresses only RGB-D sensors and it is not clear how the results can be extended to other sensors.

Another approach is to use a fiducial map metric for assessing the quality of the created map [5]. It makes use of a number of artificial objects (fiducials) which are placed in the environment at known positions, as u can see in Figure 2.5. These fiducial positions are well known and therefore you can evaluate obtained maps by looking at the fiducial positions and compute a score. The positive side of this method is, that you don't need a lot of ground truth information. Just the relative position of the fiducials to each other have to be known. This leads to the idea of using natural features, such as service niche in street tunnels, as such fiducials. The problem with natural features is, that ambiguities can occur if the map contains big localization errors. This can be avoided by specially placed fiducials.

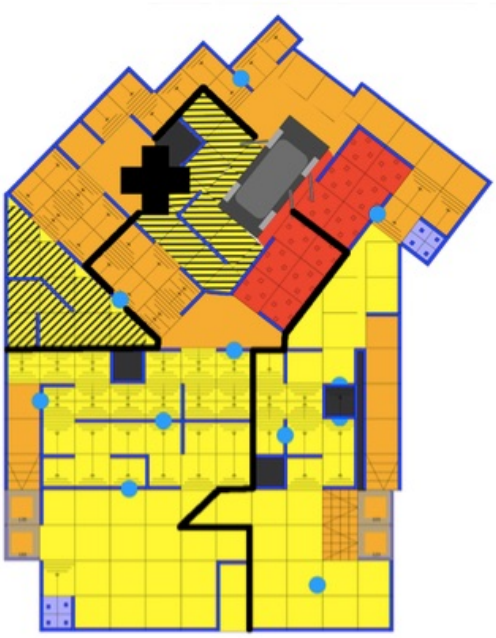


Figure 2.5: Map with fiducials as proposed by [5]

Also the SLAM evaluation toolkit developed at the department of computer science in Freiburg [22] looks promising. The framework allows the user an easy analysis and objective comparisons between different SLAM approaches. The metric for measuring the error of a SLAM system is based on the corrected trajectory, it does not rely on a global reference frame and uses only relative relations between poses. So the performance of a SLAM algorithm is not measured by comparing the map itself but by considering the poses of the robot itself. Furthermore the metric allows comparing SLAM approaches with different estimation techniques, the only requirement is, that the algorithm estimates the trajectory by a set of poses.

A number of further projects is concerned with collecting real-world data and providing ground truth for them.

For instance, the Radish (Robotic Data Set Repository) [23] repository comprises chiefly a number of range and annotated image datasets which are useful for localization tasks.
A similar project is Rawseeds (Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets) [24]. This project focuses also on range-based and vision-based

SLAM techniques, but also provides benchmark tools and evaluation tests. Rawseeds provides either benchmark problems which aim at testing algorithms. These datasets include a detailed description of the task, a multi-sensor dataset and evaluation methodology and tools. Or it provides complete benchmark solutions which extend benchmark problems with a description of the algorithm for solving the benchmark problem, algorithm output and an evaluation.

Chapter 3

System Design

Figure 3.1: Robot Odin

The aim of the research behind this thesis is to evaluate as many state-of-the-art sensors as possible simultaneously in an emergency response exercise. Therefore, we needed a reliable and robust robot platform with a high payload in order to carry a sensor suite with its various sensors (an overview is shown in Figure 3.2). In this section, we describe the robot and sensor suite in more details.

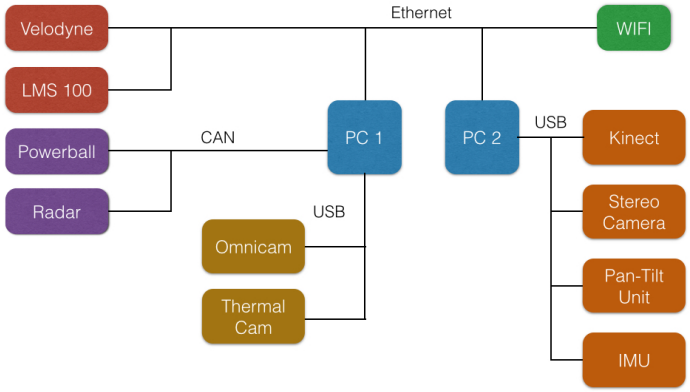


Figure 3.2: Overview of the connected sensors

3.1 Robot Platform

We decided to use the robot platform *Odin* because of its robustness and its load capacity. It is a military graded ground robot developed by the Communication, Information Processing and Ergonomic (FKIE) division of the German Fraunhofer Gesellschaft. Odin is a robust, highly versatile platform for effective support of first responders. The robot can be equipped with all kinds of payloads, just as needed by different application fields such as reconnaissance or fire extinguishing. It provides a modular sensor concept (quickly exchangeable instrument boxes) and is remotely controlled from a station situated in a control vehicle. The main focus is to have common interfaces for being able to switch payloads in a short period of time on-site. Odin can be adjusted to the needs of first responders during their mission and becomes a highly effective

supporting system. We use this concept of instrument boxes for our sensor suite.

Figure 3.3: Robot Platform Odin

Figure 3.3 shows the robot without the sensor suite. There is a number of parameters that makes Odin suitable for such an evaluation and, further, for deployment in real disaster response missions. It is able to operate for 4 hours and provides a maximum speed of 15 km/h. Because of its robust design it has a weight of 425 kg and is able to carry a payload of 150 kg. Moreover, using two 2D laser scanners, it is prepared for autonomous navigation. A summary of the robot's characteristics can be seen in Table 3.1.

Robot Characteristics	
Maximum Speed	15 km/h
Type of Drive	Chain Drive
Operating Time	4 h
Energy Source	Fuel cell
Output Voltage	24 V
Weight	425 kg
Payload	150 kg

Table 3.1: Summary of the most important features

3.2 Sensor Suite

The sensor suite consists of a metal payload box with the size of 1x0.6 meters. Inside this box some energy providing systems like voltage converters, the networking system (Ethernet and WIFI), electrical wiring, two industrial computers (Intel Core2Duo @ 2GHz) and an inertial measurement unit (IMU) are mounted. The box is powered from the robot via a standard 24 V connector. The sensor suite is shown in Figure 3.4.

We equipped the box with different kinds of sensors that were mounted on a rack above the box. In order to be able to evaluate the performance of different sensor technologies in a difficult environment with smoke and fire, we selected sensors with varying sensor modalities: laser scanners as they are common sensors in robotics and are very exact, spectral cameras for trying different mapping approaches, thermal cameras to detect high temperature areas and have a better sight under smoke, time of flight cameras in the form of a Microsoft Kinect as they are very cheap and return good results in practice and a radar as this sensor should not be affected by smoke or fire.

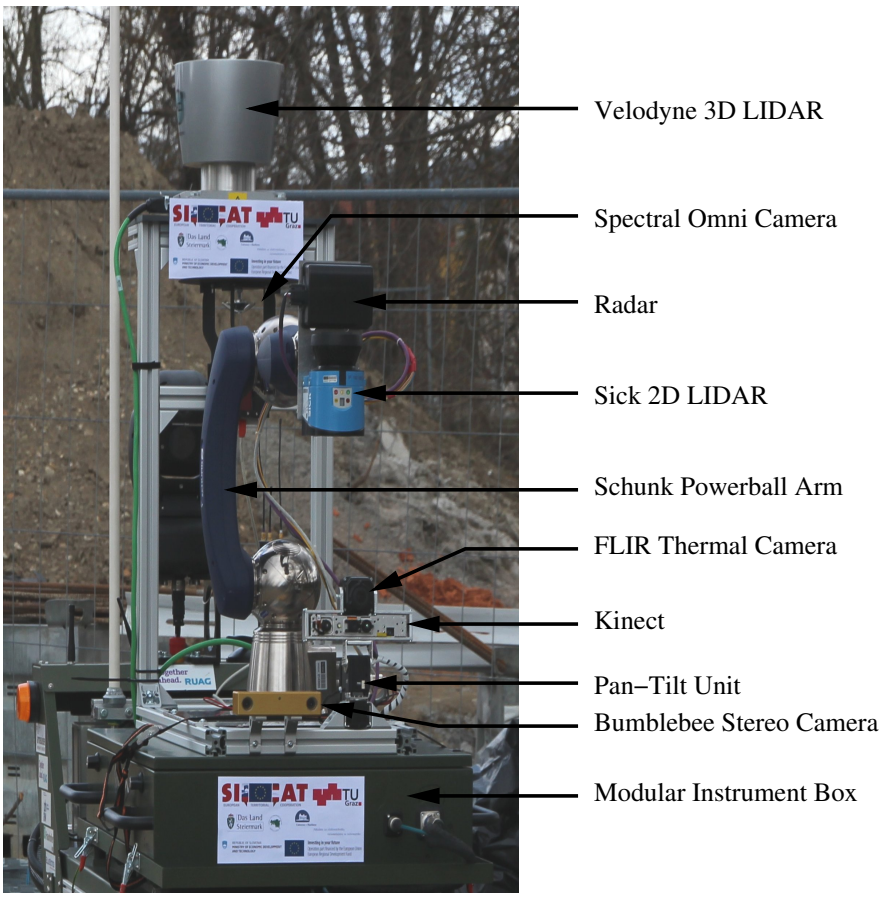


Figure 3.4: Front view on the robot and sensors.

As these sensors are very different to each other, the next sections describe them in more detail with all their features and applications.

3.2.1 Velodyne HDL-64E

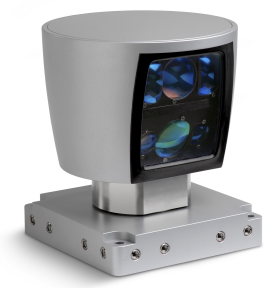


Figure 3.5: Velodyne HDL-64E 3D Laser Scanner.

The Velodyne HDL-64E 3D laser scanner is ideal for the most demanding perception and mapping applications because of it's 360° field of view, very high data rate and durability. The design uses 64 fixed-mounted lasers to measure the surrounding environment. Each laser is mechanically mounted to a specific vertical angle, with the entire unit spinning. This approach leads to large accurate 3D scans with up to 1.3 million points.[25]

This sensor was selected as a representative of premium-class sensors based on laser scanner technology. The scanner was mounted on top of the sensor suite and outputs it's data over ethernet.

Velodyne HDL-64E Specifications	
Field of View (azimuth)	360°
Angular Resolution (azimuth)	0.09°
Vertical Field of View (elevation)	26.8°
Frame Rate	5 Hz to 15 Hz
Revolutions per Minute	300 rpm to 900 rpm
Range	120 m
Power Supply	15 V @ 4 A
Output	UDP Ethernet Packets

Table 3.2: Summary of the most important features

3.2.2 Schunk Powerball Leight Weight Arm LWA 4P

Figure 3.6: Schunk Powerball Industrial Arm

The Schunk Powerball arm is a compact and flexible aid for stationary and mobile applications. Central elements of the design are three Powerball modules, which combine the movement of two axes. It features a weight/paylaod ratio of 2:1 with a weight of 12 kg. [26]

Schunk Powerball Specifications	
Max. Payload	6 kg
Number of Axis	6
Repeatability	0.06 mm
Drives	Brushless DC motors
Power Supply	24 V @ avg. 3 A
Single Axis Control	CANopen

Table 3.3: Schunk Powerball Specifications

These specifications make the arm a good choice for attaching different sensors. In this setup, a Sick LMS100 Laser Scanner and a Continental ARS308-T2 Radar was mounted on the arm (see Figure 3.7). As the Sick Scanner is only a 2D Laser Scanner, a connection with the Powerball and tilting axis movements, will lead to a 3D perception of the environment. Such a setup allows to record accurate 3D scans but for a cheaper price [27].

Figure 3.7: Setup of the Powerball with the Sick LMS100 and the ARS308-T2 Radar.

3.2.3 Pan-Tilt Unit

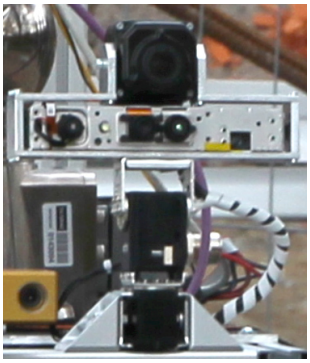


Figure 3.8: Pan-Tilt Unit with mounted Thermal Cam and Kinect

The Pan-Tilt unit is custom-made and consists of two Dynamixel Servo Motors to enable panning and tilting. It is controllable by software, has automatic moving modes, or is controllable

with a standard Joystick controller. The unit was equipped with a Microsoft Kinect and a FLIR PathfindIR Thermal camera. The advantage of this configuration is to being able to target the cameras directly on areas of interest.

3.2.4 Sick LMS100

Figure 3.9: Sick LMS100 2D Laser Scanner

The Sick LMS100 is a compact 2D Laser Scanner suited for indoor and outdoor applications. Publicly available software drivers makes it easy to work with the scanner and the combination with the Schunk Powerball arm allows to record accurate 3D scans but for a cheaper price [27]. Furthermore the LMS100 is a standard sensor in the field of robotics.

Sick LMS100 Specifications	
Field of View	270°
Angular Resolution	0.25°
Frame Rate	25 Hz to 50 Hz
Range	20 m
Power Supply	10 V to 30 V
Output	RS-232, TCP/IP

Table 3.4: Sick LMS100 features

3.2.5 Continental ARS308-T2

Figure 3.10: Continental ARS308-T2 Radar Sensor

In order to evaluate alternative technologies that can work in smoke and fire we add the radar sensor Continental ARS308-T2 to the robot arm. The sensor comes from the automotive industry and provides up to 64 range measurements based on a 77 GHz radar within a field of view of 56°. Radar waves are assumed to be less affected by smoke than laser beams.

Continental ARS308-T2 Specifications	
Field of View	17 ° to 56 °
Angular Resolution	0.1° far field, 1° close-up range
Vertical Field of View (elevation)	4.3°
Cycle Time	66 ms
Range	0.25 m to 200 m
Power Supply	12 V to 24 V
Output	CAN

Table 3.5: Continental ARS308-T2 features

3.2.6 Point Grey Bumblebee 2

Figure 3.11: Bumblebee 2 Stereo Vision Camera

The Point Grey Bumblebee 2 is a compact stereo vision camera. It provides a good balance between 3D data quality, processing speed, size and price. The camera is a good choice for testing mapping and tracking algorithms.

Point Grey Bumblebee 2 Specifications	
Field of View	2.5 mm with 97° HFOV
	3.8 mm with 66° HFOV
	6 mm with 43° HFOV
Resolution	1032 x 776 pixels
Frame Rate	20 FPS
Image Data Output	8, 12, 16 and 24-bit digital data
Power Supply	12 V
Data Connection	IEEE-1394 (FireWire)

Table 3.6: Point Grey Bumblebee 2 features

3.2.7 FLIR PathfindIR

Figure 3.12: FLIR PathfindIR Thermal Camera

The FLIR PathfindIR is a infrared camera for thermal imaging applications. The small size and the low weight make it ideal for mounting on the sensor suite. Primary the camera is used in the automotive sector but can also be integrated into military vehicle designs, or adapted for applications like this scenario. Thermal images can be used for object detection and visual mapping algorithms also in smoky environments. The camera was mounted on the pan-tilt unit.

FLIR PathfindIR Specifications	
Field of View	36° (H) x 27° (V) with 19 mm lens
Resolution	324 x 256 pixels
Frame Rate	25 FPS
Video Output	PAL composite video
Power Supply	12 V

Table 3.7: FLIR PathfindIR features

3.2.8 Microsoft Kinect V1

Figure 3.13: Microsoft Kinect V1

The Microsoft Kinect camera is a 3D camera at a much lower cost than traditional 3D-cameras (such as time-of-flight based cameras, e.g. SwissRanger 4000 [28]). The basic principle behind the Kinect depth sensor is emission of an IR pattern and the simultaneous image capture of the IR image with a CMOS camera, called structure from light [29]. The resulting point clouds are useful for 3D mapping approaches. [30]

As said before the camera was mounted on the pan-tilt unit. The interesting question is how well such a low-cost sensor performs in a challenging environment.

Microsoft Kinect Specifications	
Field of View	57° (H) x 43° (V)
Resolution	640 x 480 pixels
Frame Rate	30 FPS
Range	0.8 m to 3.5 m
Connection type	USB
Power Supply	5 V

Table 3.8: Microsoft Kinect features

3.2.9 Omnicam

Figure 3.14: Omnicam Setup

The Omnicam is a special camera, where a standard rgb camera is pointing upwards at a hyperbolic mirror. This setup leads to images with a 360° field of view.

3.2.10 XSens MTi-100

Figure 3.15: XSens MTi-100 Inertial Measurement Unit

The MTi-100 is a high performance IMU was mounted. It is useful for supporting the wheel odometry to get better results in robot movement.

XSens MTi-100 Specifications	
Standard full range gyro	450°/s
Standard full range acc	50 m/s^2
Output frequency	$\leq$ 2 kHz
IP-rating	IP 67
Connection type	RS232/RS485/422/UART/USB
Input voltage	4.5 V - 34 V

Table 3.9: XSens MTi-100 features

3.2.11 Computer

For the processing power two industrial computers (Intel Core2Duo @ 2GHz) were installed to record all the measurements. The sensors were connected to the computers in a way that guaranteed a balanced load (see Figure 3.2).

3.2.12 WIFI + Analog Radio Control

Finally, in order to be able to send information from the robot to the first responders outside the tunnel on-line, we had to use special WIFI Routers (G standard, 2.4 GHz) and directional

antennas. This was necessary because of the length of the tunnel (1570 m).

In order to control the robot from outside the tunnel, the robot was equipped with a special analog radio control unit which was capable of controlling the robot over long distances.

3.3 Software Platform

As the underlying software platform, the robot operating system (ROS) [31] was used. It is a flexible framework for writing robot software. The task of ROS is to simplify the creation of complex and robust robot behavior by supplying the user with a collection of tools, libraries and conventions. ROS is mainly developed for the use on the Linux distribution Ubuntu.

As a result ROS gained huge popularity over time at universities all around the world which lead to collaborations and software packages that are simple to integrate in your own projects.

For this reason the platform is well suited for the main task of this thesis, the evaluation of different mapping algorithms. As before mentioned ROS gives access to the latest research developments and regarding to this thesis, access to the latest mapping approaches. So the development of a software environment was accomplished in an acceptable amount of time.

3.4 Summary

So all these sensors had to be merged together in terms of power consumption (see Table 3.10) and data flow. Two computers with solid state disks were used, as the data flow of the cameras was so huge, it wouldn't be possible to record all the data on disk. As a result of using two computers, it was also necessary to synchronize the hardware clock between them, so the collected sensor data had the right timestamps. For this task the freely available software chrony[1] was used, which runs in background and keeps the time in sync with the master clock.

The electronic part regarding power consumption and voltage conversion for the different sensors was all set up and connected inside the payload box.

[1] See `http://chrony.tuxfamily.org`

Summarizing the payload box is built in such a way that it can operate on different kinds of robots that have enough space for mounting and the correct power source of 48 V. By plugging the power source in, the system will start automatically and can be accessed over a remote connection.

Sensor	Technology	Range	FoV (H)	Frame Rate	Power
Schunk Powerball	Industrial Arm	-	-	-	480 W
Sick LMS100	Lidar	20 m	270°	50 Hz	19.2 W
Continental AR-308	Radar	200 m	17 ° to 56 °	15 Hz	35 W
Velodyne	Lidar	120 m	360°	15 Hz	48 W
2 PC's	Computer	-	-	-	84 W
Microsoft Kinect	Camera	3.5 m	57°	30 FPS	2.5 W
Xsens IMU	Motion	-	-	$\leq$ 2 kHz	0.91 W
Bumblebee	Camera	-	97°	20 FPS	4 W
Thermal Cam	Camera	-	36°	25 FPS	5 W
Omnicam	Camera	-	360°	25 FPS	2 W
WIFI	Network	2 km	-	-	~10 W
Total Power Consumption					~700 W

Table 3.10: Summary of the used equipment with most important features.

Chapter 4

Test Scenario and Data Recording

This chapter will give a description on the scenario where the emergency exercise was taking place, what environmental conditions had to be coped with and how the gathered information was made persistent.

4.1 Scenario

The selected scenario for the emergency exercise was an accident of a minivan transporting hazardous materials and a coach in a tunnel. The exercise was conducted in the street tunnel Loiblpass located on the border between Slovenia and Austria. The tunnel is located at a height of $1068\,\mathrm{m}$ above sea level and has a length of $1570\,\mathrm{m}$.
First announcements indicated that the robot will be exposed to fire and smoke in the tunnel, but on exercise day the conclusion was that it is too dangerous for the people involved and would also take too much time for cleaning the tunnel afterwards.

Figure 4.1: Exercise scenario location at Loiblpass.

The main task of the exercise was to secure the hazard materials (see Figure 4.2 middle) and to rescue about 30 injured and trapped people (see Figure 4.2 bottom). Overall 400 first responders from police, ambulance, fire brigade, alpine rescue service, and K–9 units were involved.

The task of the robot was to enter the tunnel in a teleoperated fashion, to pursuit towards the accident location, and send back reconnaissance data including regular and thermal images and 2D and 3D maps. The received data were immediately presented to the incident commander and the staff.

Figure 4.2: Exercise scenario - collision of a coach with a minivan transporting hazard materials in the tunnel. On top robot *Odin* and the sensor suite entering the tunnel. In the middle responders secure the hazard materials. On the bottom ambulances take care of victims in the coach.

4.2 Data Recording

The software packages used to operate the sensors and record the data are based on the Robot Operating System (ROS) [31]. Most of the used packages are standard packages and freely available. We mainly made use of sensor driver packages.

The industrial computers were equipped with a solid state disk in order to be able to handle the high data rate of the sensors.

The data format used for storing the measurements was the ROS bag format[1], which is a standard format for storing ROS messages in files. Individual sensor measurements were saved in individual files to ease post-processing (see Table 4.1). Using ROS bag files also facilitates the dissemination of the recorded data to other interested research groups. All recorded data is available at our wiki site[2] .

Online visualization of data was done with the ROS tool rviz in a multi-host configuration.

ROS Bag Files	
Transformations, Odometry and IMU data	10 MB
Kinect data	1.6 GB
Thermal camera data	289 MB
Omnidirectional camera data	104 MB
Velodyne data	16 GB
Radar data	461 kB

Table 4.1: Bag files and file sizes

[1] See `http://www.ros.org/wiki/Bags` for details.
[2] ROS Bag Files `http://www.tedusar.eu/cms/de/research/loibl`

Chapter 5

Background On Mapping

In this chapter some basic information on the mapping process and vocabulary used, is presented to the reader.

5.1 What is ”Mapping”?

In the scope of this thesis the term ”mapping” means to capture a bounded area cartographically. It's the graphical representation of measuring points. These points can be obtained from different kinds of sources like laser scans, camera images or point clouds. Most of the time, one exposure site is not sufficient. Therefore measurements on different positions have to be made, to get a map rich of information. And these specific measurements have to be stitched together in a way the real world is represented best. Stitching, or scan matching, will be one of the main tasks in the following mapping algorithms.

5.2 Laserscan

A laserscan is a crowd of data measured by a laser scanner like the Sick LMS100 described in Section 3.2.4, has an infrared laser and a rotating mirror inside its enclosure. With the laser, the distance to the object hit can be measured. In combination with a horizontally rotating mirror, many different measurement points can be acquired. The resulting set of points expresses the contours of the area the laserscan was made. See Figure 5.1 for an example.

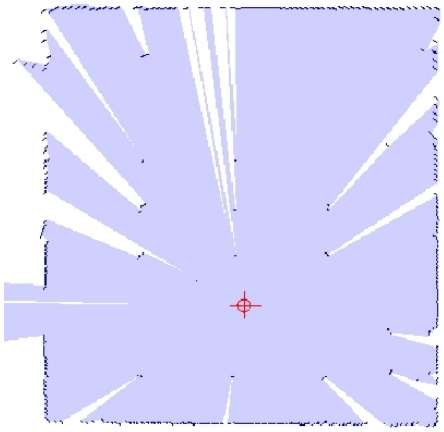


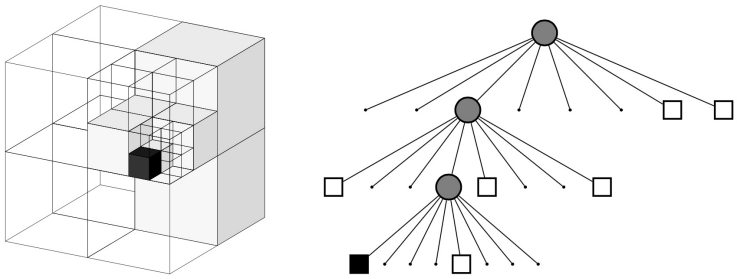
Figure 5.1: Example of a 360° laserscan.

5.3 Odometry

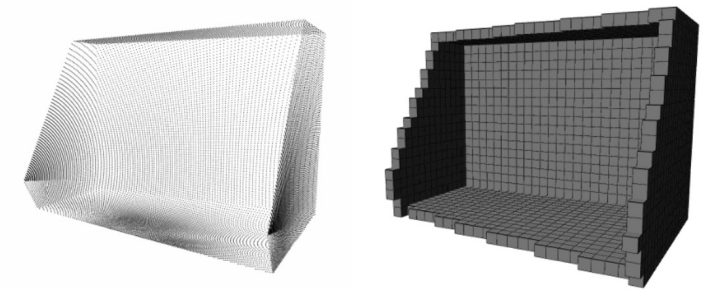
Odometry is a method for estimating the position and orientation of a vehicle. For the estimation the revolutions and direction of the wheels are measured and used for computing the travelled distance. In robotics odometry is one of the easiest methods for localizing a robot. But as this method can have a huge error because of slipping wheels or low-frequency measurements is mostly used in combination with other localization techniques such as scan matching or IMU measurements.

5.4 Octomap

An octomap, presented by Hornung et al. [6], is based on octrees(see Figure 5.2), a tree-based data structure for visualizing 2D, 2.5D and 3D data. The advantages to simple point clouds are lossless compression, memory efficient, compact map files and multi-resolution grid maps. This approach is able to represent volumetric 3D models that include free and unknown space by using probabilistic occupancy estimation.



(a) Octree



(b) 3D map structure

Figure 5.2: Octomap representation by A. Hornung et al. [6]. a) Example of an octree storing free (shaded white) and occupied (black) cells. b) A simulated laser scan (left) is integrated in the octomap 3D structure (right).

5.5 Inertial Measurement Unit

An inertial measurement unit is an electronic device that measures and reports a craft's velocity, orientation, and gravitational forces.
Typically inertial measurement units (IMU) consist in each case of three orthogonally arranged accelerometers and gyroscopes. Furthermore a magnetic field sensor can be used to gain more precision and to correct drift errors of the above mentioned sensors.

In robotics an IMU is mostly used for estimating movements but only in interaction with another localization method as the drift error becomes otherwise too big, so they are normally only one component of a navigation system. Another scope of function is keeping sensors in balance when

for example the underground is not flat and the laser scanner has to be horizontally aligned.

5.6 Sensor Fusion

In general sensor fusion is the combination of outputs from multiple sensors to gain information of better quality than using the sensors individually. In robotics this term is for instance used to combine odometry and IMU data for an overall better pose estimation. Sensor fusion is a term that covers a number of algorithms and methods, including the Kalman filter, which is described later in Section 5.7.2.

5.7 Mapping Fundamentals

The following methods are the basis for the evaluated algorithms and are important for their understanding. Hence these methods are described in the following sections in more detail.

5.7.1 RANSAC

RANSAC is an abbreviation for ”RANdom SAmple Consensus” [32]. It is an robust iterative method to estimate model parameters from a set of observed data evxen if a significant number of outliers are present.

For example on the left picture in Figure 5.3 is a data set with many outliers and a line has to be fitted. On the right side you can see the fitted line with RANSAC where outliers were rejected.

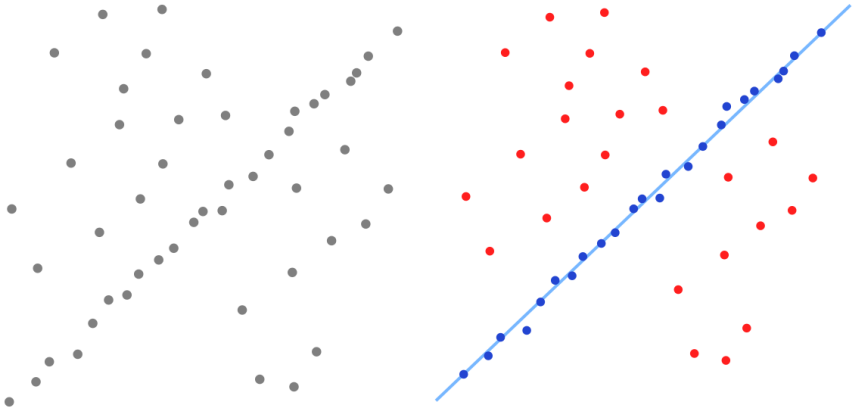


Figure 5.3: Example of the RANSAC algorithm fitting a line in a data set with outliers.

A problem of the algorithm is when the number of iterations is limited, then the probability of a reasonable result is lower and may not fit the data in a good way.

The algorithm is often used in machine learning, computer vision and robotics. RGBD-SLAM will be discussed in more detail in 6.2.1.

5.7.2 Kalman Filter

The Kalman filter, described in Thrun et al. [7], is a technique for implementing Bayes filters for prediction and filtering in linear systems. The filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state.

In Figure 5.4 the behavior of the algorithm is illustrated. The Kalman filter is alternating two steps. A measurement update step in which sensor data is integrated into the present belief. This step decreases uncertainty in the robot's belief. And a prediction step in which the belief is modified in accordance to an action, for example a movement. Here the uncertainty is increased.

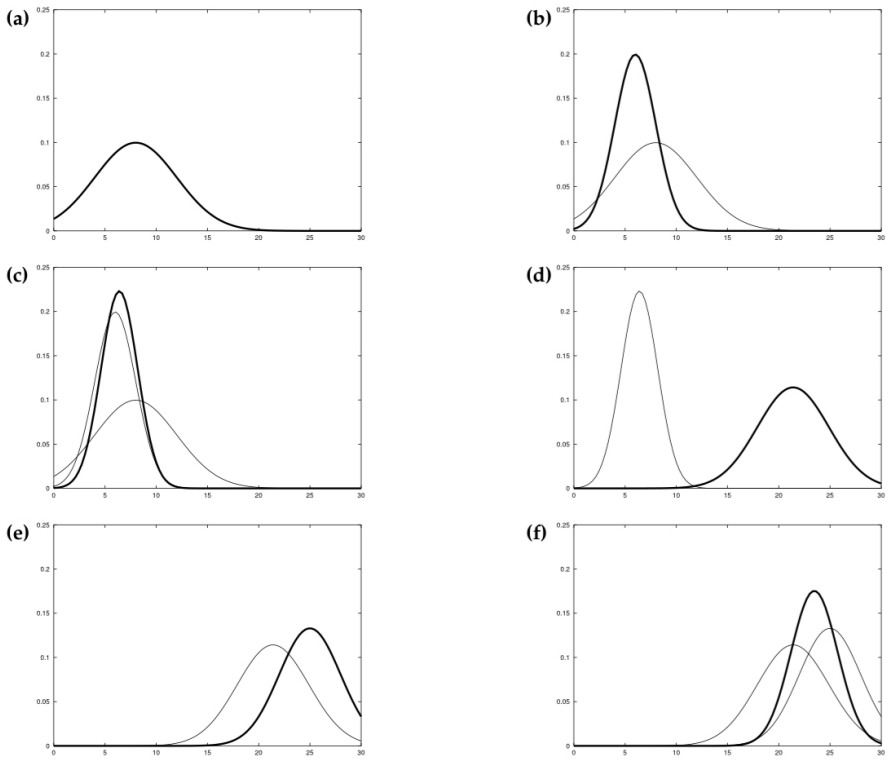


Figure 5.4: Illustration of Kalman filters by Thrun et al. [7]: (a) initial belief, (b) a measurement (in bold) with the associated uncertainty, (c) belief after integrating the measurement into the belief using the Kalman filter algorithm, (d) belief after motion to the right (which introduces uncertainty), (e) a new measurement with associated uncertainty, and (f) the resulting belief.

An application in robotics would be sensor fusion, as described before in section 5.6, to estimate the robots position. The problem here is, that robot movement cannot be described by linear state transitions. For such cases the extended Kalman filter (EKF) is used, where the assumption is, that the next state probability and the measurement probabilities are governed by nonlinear functions.

The EKF has become just about the most popular tool for state estimation in robotics. Its strength lies in its simplicity and in its computational efficiency.

5.7.3 Particle Filter

Particle filters or Sequential Monte Carlo (SMC) methods, described in Thrun et al. [7], are a set of on-line posterior density estimation algorithms that estimate the posterior density of the state-space by directly implementing the Bayesian recursion equations. SMC methods use a sampling approach, with a set of particles which represent the samples of a posterior distribution. Each particle has a importance weight assigned to it that represents the probability of that particle being sampled from the probability density function. A resampling step can be included when the weights become too uneven. Particles with low weights will then be replaced by new particles in the proximity of particles with higher weights.

There are some advantages in comparison to the use of Kalman filters. First, instead of representing the distribution by a parametric form, particle filters represent a distribution by a set of samples drawn from this distribution. Such a representation is approximate, but it is non-parametric, and therefore can represent a much broader space of distributions than, for example, Gaussians. Second, a particle filter can process raw measurements and third, it can solve global localization problems. Table 5.1 shows the differences.

In robotics this filter is used in Monte Carlo localization (MCL), described in Thrun et al. [33], which is an algorithm for robot localization and the most popular approach to date. Given a map of the environment, the algorithm estimates the position and orientation of a robot as it moves and senses the environment. The algorithm uses a particle filter to represent the distribution of likely states, with each particle representing a possible state, i.e. a hypothesis of where the robot is. The algorithm typically starts with a uniform random distribution of particles over the configuration space, meaning the robot has no information about where it is and assumes it is equally likely to be at any point in space. Whenever the robot moves, it shifts the particles to predict its new state after the movement. Whenever the robot senses something, the particles are resampled based on recursive Bayesian estimation, i.e. how well the actual sensed data correlate with the predicted state. Ultimately, the particles should converge towards the actual position of the robot.

This procedure is illustrated in Figure 5.5. At first a set of pose particles are drawn at random and uniformly distributed positions over the entire pose space, as seen in Figure 5.5a. As the robot senses the door, importance factors are assigned to each particle which is shown in Figure 5.5b. The height of each particle shows its importance weight. Beside of this change, the set of particles is identical to the one in Figure 5.5a.

The particle set after resampling is shown in Figure 5.5c. There is an increased number of

particles near the three likely places. In Figure 5.5d the new measurement assigns non-uniform importance weights to the particle set. On the second door most of the aggregated probability mass is centered, which is also the most likely location. Another resampling step is shown in Figure 5.5e, as the robot moved further. This example shows the correct approximation of the posterior.

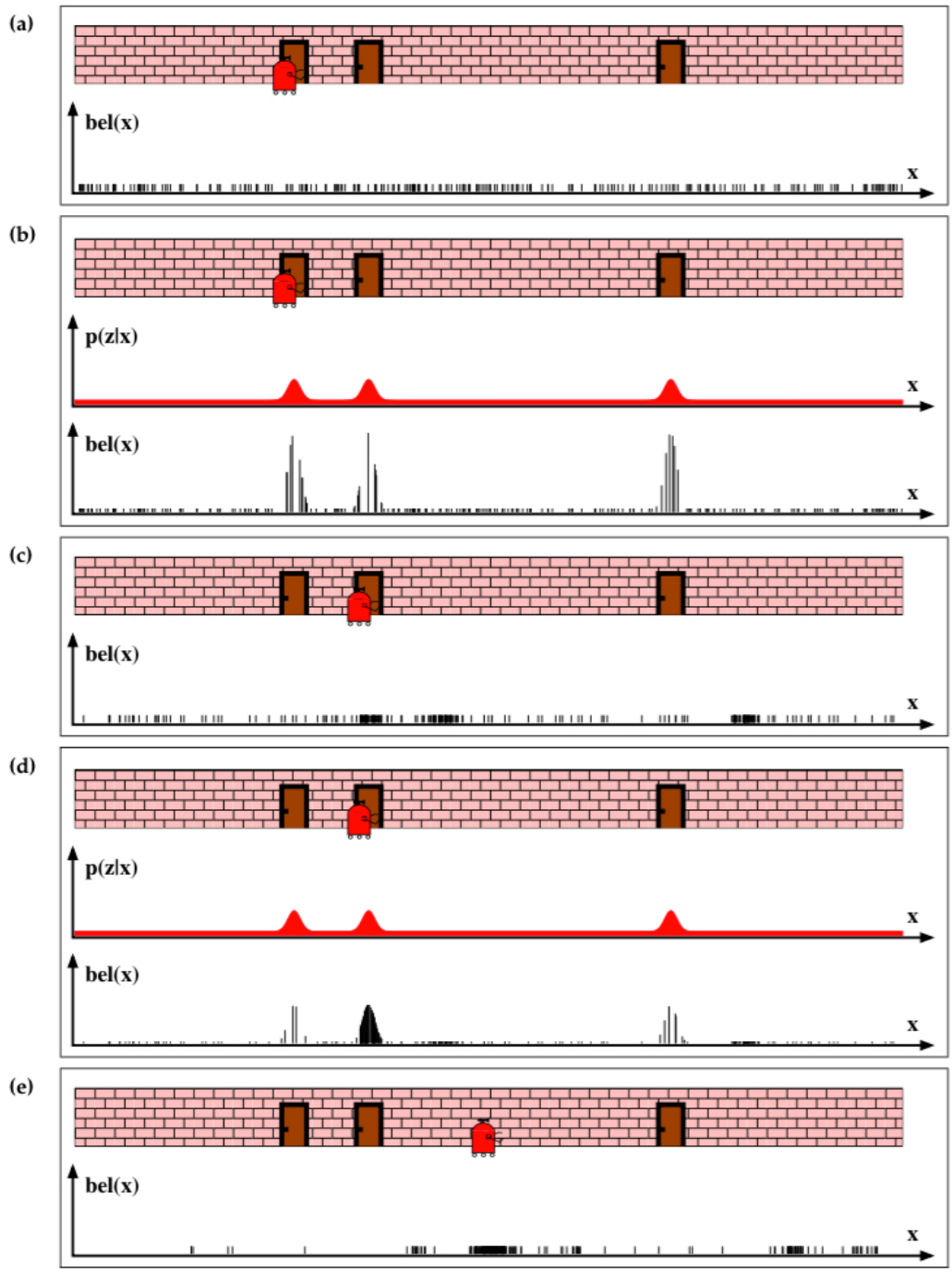


Figure 5.5: Visual explanation of the Monte Carlo localization (MCL)

	EKF	MCL
Measurements	landmarks	raw measurements
Measurement noise	Gaussian	any
Posterior	Gaussian	particles
Efficiency (memory)	++	+
Efficiency (time)	++	+
Ease of implementation	+	++
Resolution	++	+
Robustness	-	++
Global localization	no	yes

Table 5.1: Comparison between extended Kalman filter and Monte Carlo localization.

Particle filters are also used in Simultaneous Localization and Mapping (SLAM), which is described later in Section 5.7.5. As SLAM has a lot of different approaches, the ones using particle filters are known as FastSlam and grid-based approaches with Rao-Blackwellized Particle filters. There a single particle has to consist of a pose and a map, where the map is represented by a 2D Gaussian.

5.7.4 Scan Matching

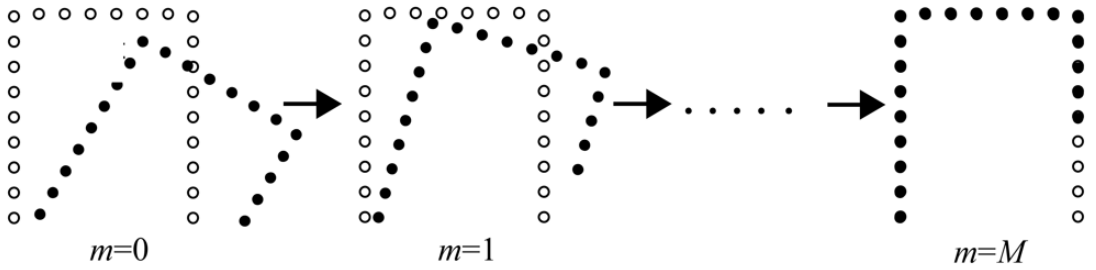


Figure 5.6: Simple visualization of the working principle of scan matching [8]

.

As stated before odometry and sensors like GPS have errors in their measurements that lead to faulty locations. Therefore scan matching is used to find the rigid-body transformation, given

a scan and a map, or a scan and a scan, or a map and a map, that aligns them best. In Figure 5.6 two scans are presented where one of the scans is slightly translated and rotated. The goal is to find a transformation where both scans overlap the most. Two properties, robustness and performance, are very important for the realtime capabilities of this principle. Consequently different methods with it's advantages and disadvantages are available.

Iterative closest points is the one mostly used, as well as in this thesis and will be described in section 5.7.6. Other methods would be the "Correlative Method"[34] and using the Hough Domain[35].

5.7.5 SLAM

The SLAM problem, short for Simultaneous Localization and Mapping (described in Thrun et al. [7]), is a method for building a map and estimate a pose at the same time. The problem is, that neither the position nor the map is known apriori. So it appears to be a chicken-and-egg problem and both tasks have to be estimated.

The general approach to solve the SLAM problem is addressed using probabilities. SLAM is usually explained by the conditional probability:

$$p(x_t, m|z_{1:t}, u_{1:t})$$

x_t = State of the robot at time t
m = Map of the environment
$z_{1:t}$ = Sensor inputs from time 1 to t
$u_{1:t}$ = Control inputs from time 1 to t

There are different approaches for solving SLAM. The most trivial one is to define the current position of the robot as the origin. So the position is known and the first measurement can be registered to the yet empty map. Then the robot will move and takes another measurement. With the overlap between the two measurements, a new absolute position can be computed and the new unknown measurement data will be added to the map. In consequence the map will be built incrementally until the whole area is measured, as the robot normally can't see the whole environment with just one measurement.

To implement such an approach, different SLAM techniques are available. The most known ones are EKF SLAM, FastSLAM (described in Section 6.1.1) or Grid-based approaches with Rao-Blackwellized Particle Filters (Section 6.1.2).

EKF SLAM applies an extended kalman filter which has some limiting assumptions like feature-based maps which require significant engineering of feature detectors. The Gaussian noise assumption limits the amount of uncertainty in the posterior to be relatively small as otherwise intolerable errors will be introduced. Also the approach can only process positive measurements of landmarks which is a direct consequence of the Gaussian belief representation. EKF SLAM has been applied with considerable success in a number of robotic mapping problems. Its main drawbacks are the need for sufficiently distinct landmarks, and the computational complexity required for updating the filter.

SLAM is important for missions where no map nor an absolute position is available but for all where the robot should be able to autonomously explore the area and build a map for later navigation.

5.7.6 Iterative Closest Point

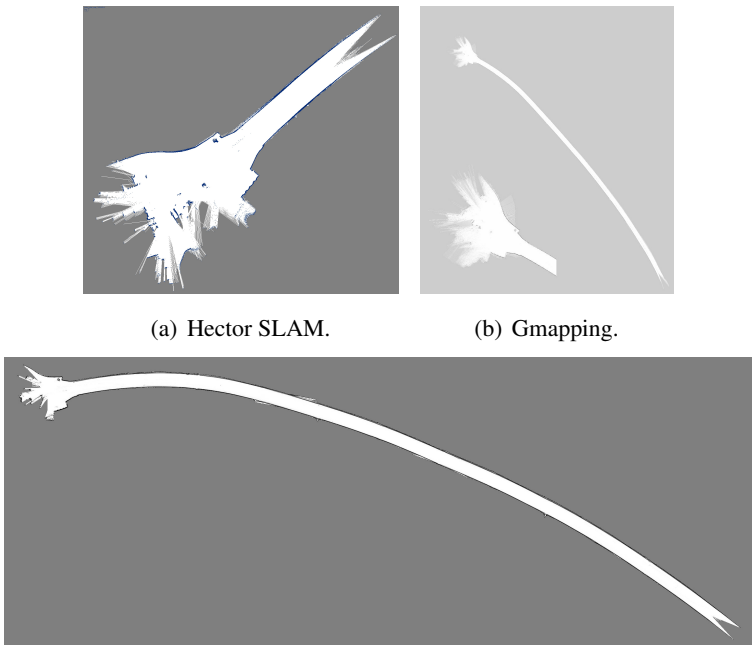
ICP, short for Iterative Closest Point is a powerful algorithm for registering point clouds or laserscans together. (See Figure 5.6) For the data sets to be fitted, transformations are calculated so the distances between the data sets are minimized. For this purpose to every point of one data set, the closest point of the other data set is determined. The sum of squares of the distances is minimized through adjustment of the transformations. This is an iterative task and is executed as long as the optimum is found.[36, 37]

As mentioned above this algorithm is used for scan matching but also for localization.

The popularity of the algorithm for matching 3D point clouds has produced many different variants of the ICP algorithm. As this would go too far, further information on efficient variants of the ICP algorithm can be found in [38].

Chapter 6

Mapping Algorithms



(a) Hector SLAM. (b) Gmapping.

(c) Voting-Based Scan Matching.

Figure 6.1: 2D Maps generated from the tunnel dataset.

In order to evaluate the usefulness of sensors and mapping algorithms for the use in the emergency response scenario, popular 2D and 3D mapping approaches were selected with ready-to-run implementations available. Beside algorithms that work with 2D and 3D laserscans, also

included were two approaches that work with the Microsoft Kinect. These sensors are cheap and light-weight options compared to laser scanners.

6.1 2D Mapping

6.1.1 Hector SLAM

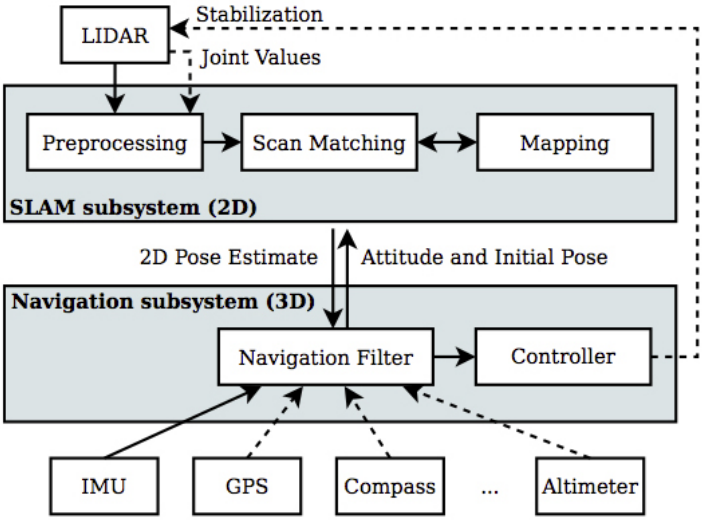


Figure 6.2: Overview of the Hector SLAM system.[9]

Input Data
Laserscan, IMU

Type of Map
2D Grid Map

Hector SLAM [9] combines a robust 2D scan matching approach for laser scans with a full 3D navigation system (position, orientation and linear velocity) based on data from an IMU. Their scan matching approach is based on optimization of the alignment of beam endpoints with the

map learnt so far. A multi-resolution map and an approach that matches scans to the entire map completes the approach. Although the robot is able to follow a path of full 6D poses, the algorithm provides only a 2D occupancy grid map. One attribute that differs from other 2D Mapping approaches is, that it leverages the high update rate provided by modern LIDAR systems. This is possible as the algorithm is keeping computational requirements low and is not using loop closure methods.
The approach is interesting for two reasons. First, it is available as a free ROS package and works quite fine for structured environments. Second, the approach is today used as standard mapping approach in the RoboCup Rescue Robot competition [39].

6.1.2 GMapping

Input Data
Laserscan, Odometry

Type of Map
2D Grid Map

GMapping [10] is a highly efficient Rao-Blackwellized particle filter for SLAM to learn grid maps from laser range data. This kind of particle filter makes use of the following factorization to estimate the joint posterior:

$$p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = p(m|x_{1:t}, z_{1:t})p(x_{1:t}|z_{1:t}, u_{1:t-1})$$

with $x_{1:t}$ as the trajectory, m as the map, the observations $z_{1:t}$ and the odometry measurements $u_{1:t-1}$.

Now the trajectory can be estimated at first and then the map is computed given that trajectory. This technique is called Rao-Blackwellization. In consequence "mapping with known poses" [40] is used to estimate the posterior over maps $p(m|x_{1:t}, z_{1:t})$. The particle filter is then used to estimate the posterior $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ over the potential trajectories, where each particle represents a potential trajectory of the robot. Also each sample carries an individual map of the environment. Taking into account not only the movement of the robot but also the most recent observation, drastically decreases the uncertainty about the robots pose in the prediction step of the filter (see Figure 6.3).

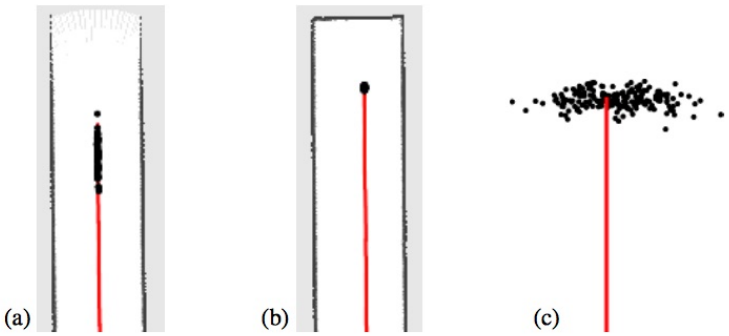


Figure 6.3: Typical particle distributions during a mapping process with GMapping. Open corridor (a). Dead end corridor (b). Raw odometry motion model (c).[10]

The particle filter is used in GMapping as a sampling importance resampling (SIR) filter. The process can be summarized by the steps sampling, importance weighting, resampling and map estimation.

In this approach the proposals were improved and adaptive resampling was introduced which increased the performance of the algorithm significantly. The improved proposal distribution leads to the generation of samples with an high likelihood, see Figure 6.3. This then again reduces the number of required samples. The proposal distribution is based on the observation likelihood of the most recent sensor information, the odometry, and a scan-matching process.
The second improvement, the adaptive resampling is a criterion that decides when to perform the resampling step. Unnecessary resampling actions are reduced and this consequently reduces the risk of particle depletion.

6.1.3 Voting-Based Scan Matching

Input Data
Laserscan, Odometry, IMU

Type of Map
2D Grid Map

To support teleoperated navigation with real-time mapping, voting-based scan matching was introduced by Kleiner and Dornhege [41]. The algorithm is based on incremental scan matching in combination with gyro measurements which support the estimation of the rotation between successive scans. The resulting robot transformations and the scans are then fed into a grid-based algorithm to optimize the overall trajectory.

To make the mapping process work on embedded systems with lower computational power, the detection of loop-closures, as seen in GMapping 6.1.2, is missing.
Also because of the voting-based approach, scan matching is stopped early if enough evidence for a transformation between two scans is found. This trick makes the algorithm quite efficient in practice. The approach was successfully tested in several emergency response exercises like in Disaster City and results show that the quality of generated maps is close to that generated by computational costive algorithms.

6.2 3D Mapping

6.2.1 RGB-D SLAM

Input Data
Monochrome and depth image, colored point cloud

Type of Map
Pose graph with colored point clouds

RGB-D SLAM [11] is an approach for visual SLAM using RGB-D sensors such as the Microsoft Kinect camera (see 3.2.8).

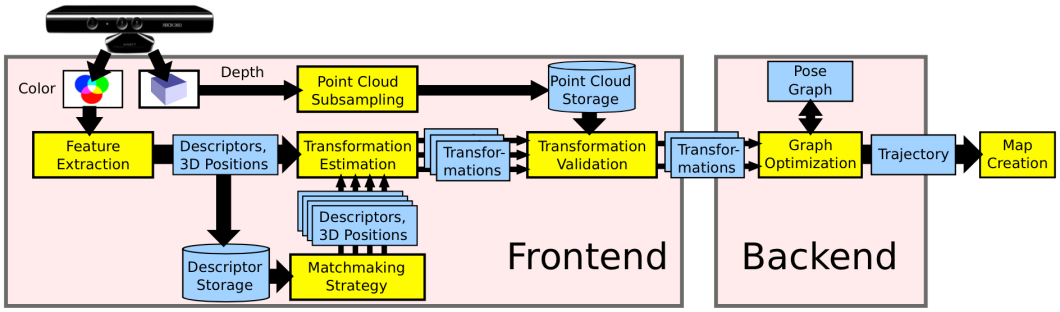


Figure 6.4: Schematic overview of the RGB-D SLAM approach [11]

.

Visual key points (SURF or SIFT) are extracted from the color images and depth images are used to localize them in 3D. The current image is only matched against a subset of previously acquired images to achieve online processing. RANSAC (described in section 5.7.1) is used to robustly estimate the transformations between RGB-D frames and optimize the pose graph using non-linear optimization. The graph's nodes correspond to camera views and the edges to the estimated 3D transformations. Finally, a volumetric 3D map of the environment is generated that can be used for robot localization, navigation and path planning.
So with the use of a hand-held camera, RGB-D SLAM allows to quickly acquire 3D models of objects and indoor scenes.

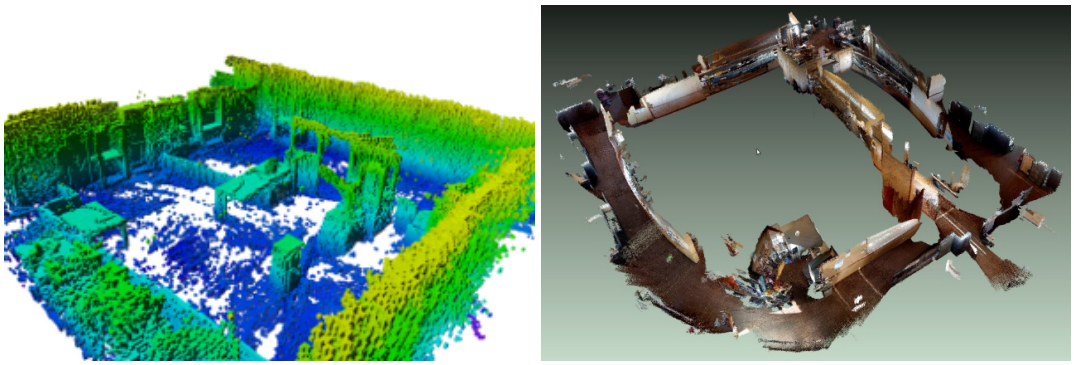


Figure 6.5: Resulting maps of an indoor scenario. A 3D point cloud[1]. (left) Point cloud with overlayed color information[2]. (right)

A beam-based environment measurement model is introduced to improve the reliability of the transformation estimates by evaluating the quality of a frame-to-frame estimate. With this quality measure in use, this approach can deal significantly better with highly challenging scenarios. Detailed experiments were performed to assess the quality of the resulting point clouds and pose graphs.

All the source code required to run RGB-D SLAM is released as open-source.

6.2.2 The 3D Toolkit

Input Data
3D Poses, Point Clouds

Type of Map
Point Cloud

The 3D Toolkit provides algorithms and methods to process 3D point clouds. It includes automatic precise registration of 3D scans (6D simultaneous localization and mapping, 6D SLAM) and other tools, e.g., a fast 3D viewer, plane extraction software, etc. The toolbox follows the approach presented in [42] which is a combination of a 6D pose prediction, a local scan matching based on ICP together with loop detection and a global map optimization.

[2]http://ais.informatik.uni-freiburg.de/projects/datasets/octomap
[2]http://www.hizook.com/blog/2010/03/28/low-cost-depth-cameras-aka-ranging-cameras-or-rgb-d-cameras-emerge-2010

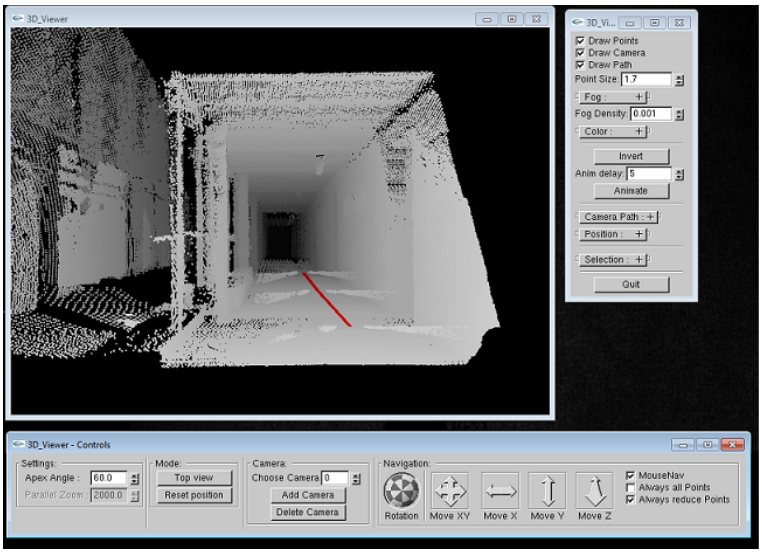


Figure 6.6: 3DTK Toolkit - Graphical User Interface[3].

There are two categories for 6D matching approaches of 3D surfaces:

Scan matching as optimization problem uses a cost function for the quality of the alignment of the scans. The range images are registered by determining the rigid transformation (rotation and translation) which minimizes the cost function. This is also known as the iterative closest point (ICP) algorithm (see Section 5.7.6).

Feature based scan matching extracts distinguishing features of the range images and uses corresponding features for calculating the alignment of the scans. Even though this approach is more intuitive, it cannot be applied to scan matching in mines, since the surface structure of the mine is too simple. In consequence there are not many features and an algorithm based on feature matching will fail as only geometric or characteristic information is extracted from the scans by this method [2]. ICP uses raw measurement data and is therefore more suitable.

As a first step before scan matching, the point clouds have to be reduced to a fixed reduction percentage of points because on one side the precision gain is very low for a larger number of points and on the other side the computational time can be reduced dramatically [4].

[3] http://slam6d.sourceforge.net/html/doc/show.html

Beside the standard point-to-point variant of ICP there is also a point-to-plane variant that can be stable in structured environments, when the overlap is kept high and the state estimation remains within 10 cm and 10°. These types of conditions are usual for laboratory experiments but are unlikely to happen in real applications. Also the extraction of surface normal vectors adds more time to the registration process than the time that can be saved by saving on the number of iterations [20].

Also, the optimal set of parameters for scan matching vary for different scenarios. In summary, the use of 6D SLAM software for mapping on an autonomous robot is right now not feasible, because of the lack of real-time capability and the need for parameter optimization depending on the environment.

6.2.3 Parallel Tracking and Mapping - PTAM

Input Data
Monochrome or colored image

Type of Map
Pose graph with colored point clouds

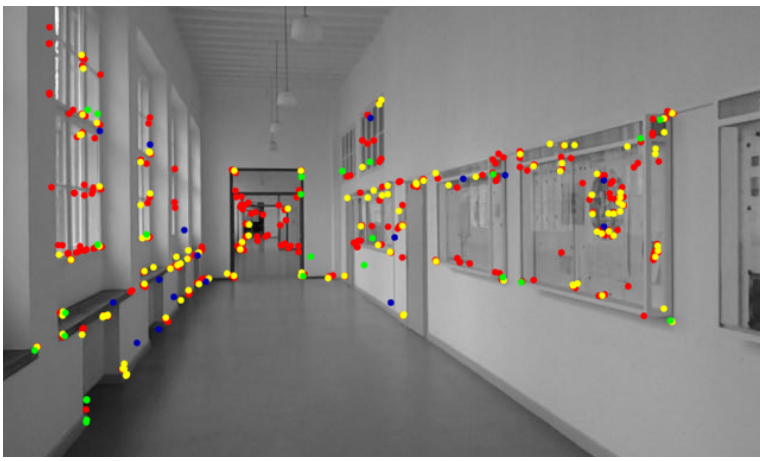


Figure 6.7: A frame with feature points that are tracked by PTAM. The color of a point indicates the size of the feature in the image[4]. These features appear in the reconstructed map.

PTAM [43] is a method that can track the 3D position of a moving camera in an unknown scene in real time. It is a SLAM system that does not need any prior information about the world such as markers or known natural feature targets. Instead it works out the structure of the world as it evolves. Beside the tracking ability, PTAM can also build a 3D map of point features, which are depicted in Figure 6.7. The mapping process is based on keyframes, which are processed using bundle adjustment.

A thing that needs to be kept in mind is, that this method needs a correct initialization (see Figure 6.8). That means between the first two key-frames, the camera has to be translated to estimate a ground plane to work on. Also the authors state that the area to be tracked should be mostly static and small. These limitations could lead to problems in the tunnel exercise.

[4] http://www.navvis.lmt.ei.tum.de/2012/10/scale-preserving-long-term-visual-odometry-for-indoor-navigation-2/ptam_cam

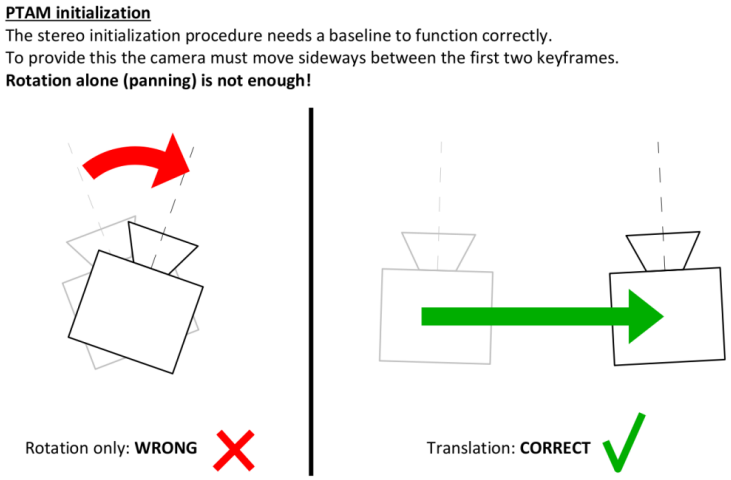


Figure 6.8: Correct initialization of PTAM[5]. The camera must be translated, not just rotated, between the first two key-frames to initialize the system correctly.

6.2.4 Structure-from-Motion (SfM)

Input Data
Monochrome or colored image

Type of Map
Pose graph with colored point clouds

Hoppe et. al. [44] proposed an online SfM approach, that gives feedback about the reconstruction quality of 3D models during the image acquisition process. So this system supports users by telling them if the last acquired image can be used for reconstruction or also if there are some parts of the scene missing. The underlying framework is principally the same as used in PTAM (See Section 6.2.3), as the results in Section 7.2.1 will show. The reason this algorithm was used, is that this method was developed by the Institute for Computer Graphics and Vision (ICG) at Graz University of Technology. So it was more easy to get feedback from the developer about

[5]http://www.robots.ox.ac.uk/ gk/PTAM/

the resulting maps. The downside is, that their approach is only commercially available, so we couldn't try it out by ourselves.

6.2.5 2D Localization and 3D Octomapping

Input Data
Laserscan, Point Cloud

Type of Map
2D Grid Map, Point Cloud, Octomap

This method is not an algorithm by itself. It is a combination of different approaches. The idea is to have a fast 2D localization algorithm like GMapping or Hector Mapping and place the raw 3D data, in the form of an octomap, at 2D poses obtained by the 2D localization.
The positive side when using this approach is that you avoid the computational costly 3D registration techniques. On the other side the resulting 3D map has not the quality of a map generated by a 3D mapping process. But as 3D mapping approaches are very rare, it could be a good option to try out. It is also the only online 3D mapping approach in this thesis.

6.3 Summary

In this chapter all the selected mapping algorithms were described.
To sum it up, Table 6.1 gives a short information about the main aspects of the algorithms:

Algorithm	Input Data	Type of Map	Real-Time
GMapping	Laserscan, Odometry	2D Grid Map	Yes
Hector SLAM	Laserscan, IMU	2D Grid Map	Yes
VBMapping	Laserscan, Odometry, IMU	2D Grid Map	Yes
RGB-D SLAM	Colored Point Cloud, Depth Image	Point Cloud	Yes
3D Toolkit	3D Poses, Point Cloud	Point Cloud	No
PTAM	Greyscale / Color Image	Point Cloud	Yes
SfM	Greyscale / Color Image	Point Cloud	Yes
2D Loc with Octomap	Laserscan, Point Cloud, Odometry	3D Grid Map	Yes

Table 6.1: Summary of the different mapping algorithms

Chapter 7

Evaluation

The goal of this thesis is to evaluate which sensors and algorithms perform best in disaster scenarios and how the support for first responders can be improved. This chapter gives an insight to the methodology and the results of the evaluation task. Furthermore the acceptance and usefulness for first responders is discussed.

7.1 Methodology

In the evaluation of the sensors and mapping approaches we followed the following procedure. We first recorded all available data during the reconnaissance mission in the tunnel. The responders allocated a slot of 20 minutes during the emergency response drill to complete the data recording. The robot was teleoperated from a parking lot near the tunnel portal along the road in the tunnel until the accident site for approximately 800 m from the northern Austrian portal. We recorded data only for the way in. This took about 13 minutes. There was a lot of activity in front and in the tunnel, e.g., responders and vehicles moved into and out of the tunnel. For safety reasons we had to stop the robot near the curbs if a vehicle wanted to pass the robot. Obviously, this led to quite noisy and cluttered data. We had to perform the mission in a one-shot style without a training run like in a real emergency response. This fact led to some technical difficulties like missing data of the Kinect (see Section 9 for a detailed discussion).

Figure 7.1: Robot Odin on his way in the tunnel.

Compressed raw data like camera images were immediately sent back to the control station for quick inspection by the responders. It was not possible to transmit complex data like 3D laser scans or map due to the limited bandwidth of the wireless connection. These complex data were processed and analyzed offline after the exercise.

For the evaluation of the sensors and the mapping approach we used well known and publicly available methods as we are interested in how well available techniques work for such a scenario. For the 3D mapping approaches we used the data of the Velodyne Lidar without any pre-processing. In the 2D mapping approaches we used a simulated 2D laser scan extracted from the Velodyne data. In the visual SLAM approaches we use the images of the thermal camera and the RGB camera of the Kinect.

The original idea of the responders to set the tunnel on fire and to generate smoke was discarded because of the continuous slope of the tunnel towards south. Therefore, all smoke would have been sucked towards the northern portal like in a chimney. As a result, we were not able to test the sensors in different environmental conditions.

Due to the fact that the Loibl tunnel is almost 70 years old there is no ground truth data like 2D or 3D maps available. For that reason, we were only able to qualitatively evaluate the results. The qualitative evaluation was done in discussions with experienced first responders.

7.2 Results

7.2.1 Mapping Quality

The map generated with Hector SLAM is depicted in Figure 7.2. Although, the approach is indented to fuse 2D scan matching with a 6D pose estimation the available implementation[1] does not support the 6D pose estimation (i.e., based on odometry or IMU). Due to this fact the tunnel portal is mapped well because it provides enough diverse structures for scan matching. Inside the tunnel the scans become too similar because of the two almost straight walls. Although, there are some service niches these features are not distinctive enough. Therefore, inside the tunnel there is no valid estimation for the movement of the robot because odometry and IMU are not used. As a consequence the robot estimates it is standing still and registers all remaining scans at the same pose. This leads to a corrupted map with only a correct part of the outer, more structured area.

Figure 7.2: Generated map by Hector SLAM.

[1] See `http://www.ros.org/wiki/hector_slam`.

GMapping overcomes the problems of similar laser scans by also integrating odometry information into its computation. This is a great benefit in this specific environment and leads to a well-defined map from the tunnel portal to the accident site. The resulting entire map is depicted in Figure 7.3. We used the standard ROS porting of GMapping[2].

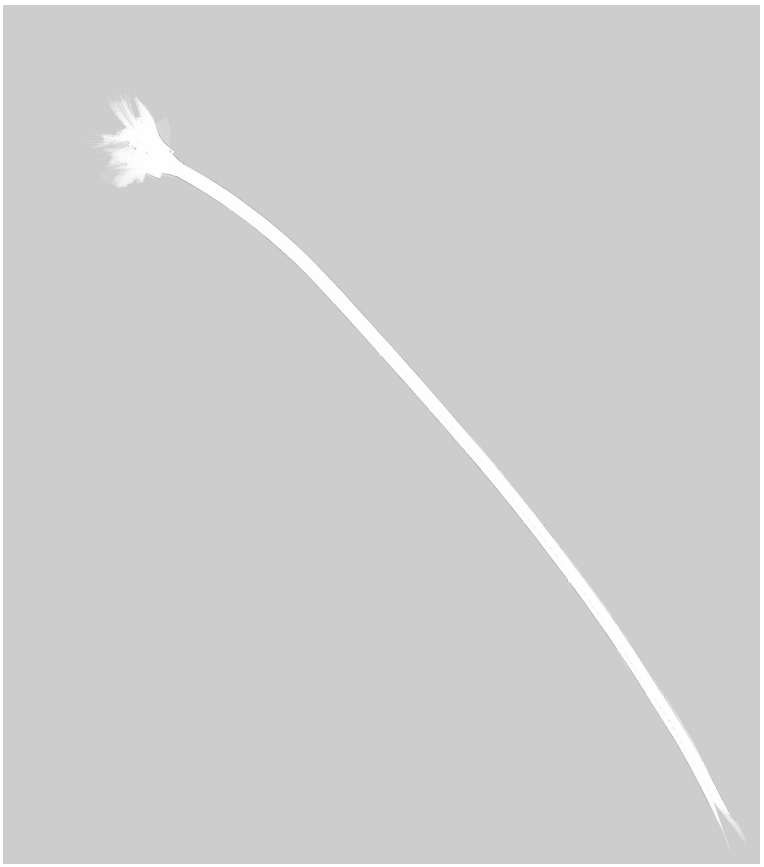


Figure 7.3: Generated map by GMapping.

The resulting map using Voting-Based Scan Matching is depicted in 7.4. It incorporates also odometry data and has therefore no problem mapping the monotonic tunnel. Although, it provides a map with more clear details the curvature of the tunnel is depicted larger in the map than it is in in reality. We suppose that this fact comes from the low frequency of the odometry

[2]See `http://www.ros.org/wiki/gmapping`.

(please refer to Section 9 for details) because the approach heavily uses it for the local scan arrangement.

Figure 7.4: Map generated by DCMapping.

2D Localization with 3D Octomapping is the only 3D mapping approach that works online. As stated in Section 6.2.5, there is no 3D point cloud registration. So if the timestamp of the 2D position of the robot differs slightly to the timestamp of the 3D point cloud data, the 3D data is not aligned with the underlying 2D map, as depicted in Figure 7.5. The result shows some rotational errors at distinct locations.

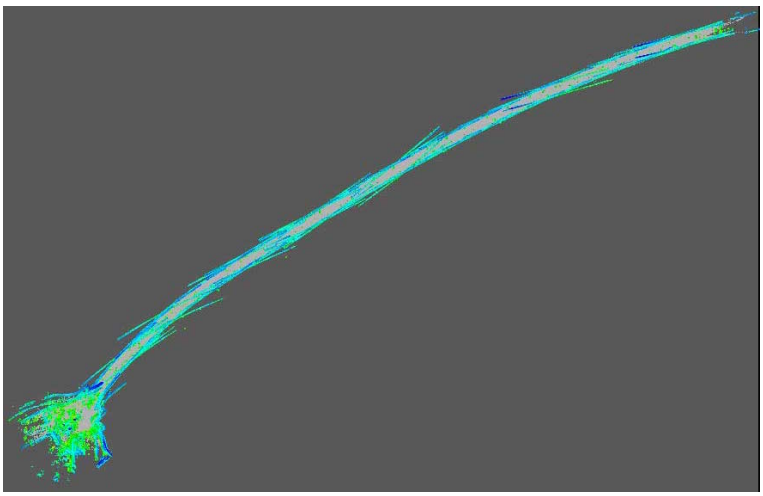
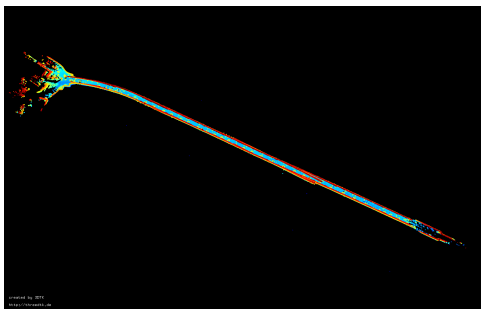


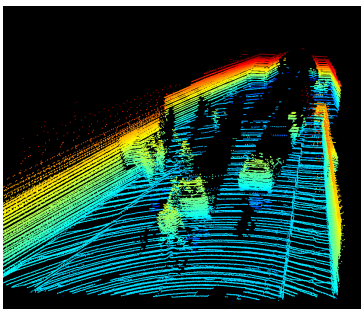
Figure 7.5: Result of 2D Localization with 3D Octomapping.

The results of the 6D SLAM ICP matching algorithm used in the 3D Toolkit[3] are depicted in Figure 7.6. The approach is able to generate a globally consistent 3D map of the entire tunnel (see Figure 7.6(a)). Anyhow, there are some registration errors at the end of the tunnel near the accident site. We suppose this is caused by evasive maneuvers the robot had to execute due to people and cars moving in the robot's way. A major drawback of this toolkit is that it not integrated in the ROS framework. It reads the input from specific scan data and pose estimation files. The toolkit in its current form is not prepared for real-time mapping. Therefore, it is currently of limited use of first responders because they need their maps with in a couple of minutes. An advantage of this approach is the richness of map details. Figure 7.6(c) shows a detailed view of the portal. There are first responders and their vehicles clearly visible. Details of the accident site are shown in Figure 7.6(b). Here the service niche and vehicles blocking the tunnel are nicely recognizable. In general this approach has the disadvantage that due to the high accuracy of the 3D Lidar and the fact that it only registers point clouds, artifacts like persons walking by remain in the map.

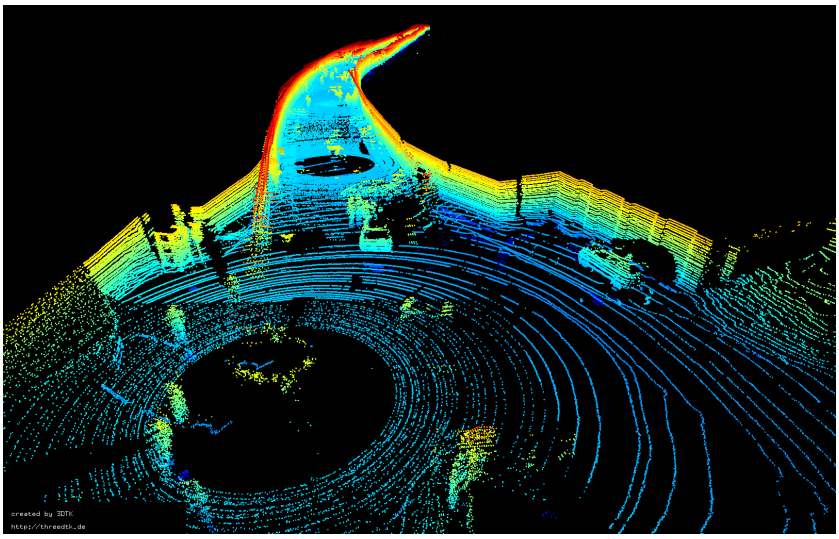
[3] See `http://slam6d.sourceforge.net`.



(a) Tunnel top view.



(b) Accident site.



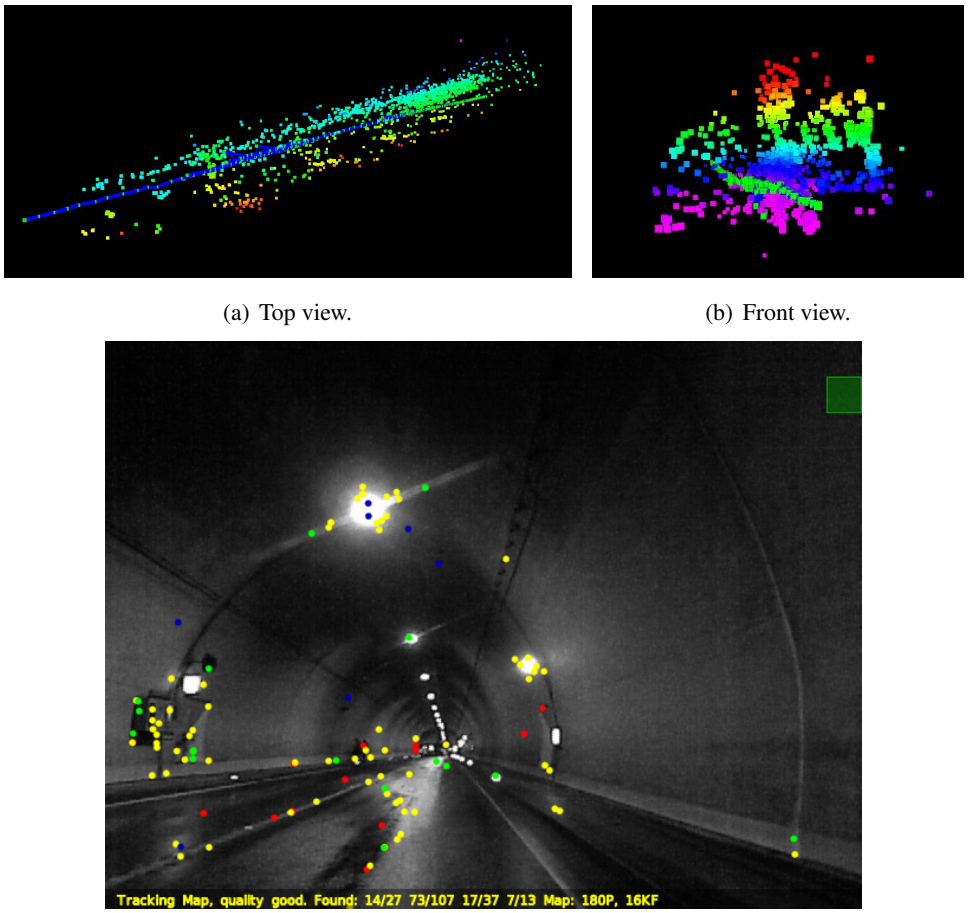
(c) Tunnel portal.

Figure 7.6: 3D Maps generated from the tunnel dataset with 3D toolkit.

Figure 7.7 depicts the results using the PTAM approach[4]. The approach uses key points (see Figure 7.7(c)) in 3D to track the path of the camera. Figure 7.7(a) depicts the path and the key points for a smaller tunnel segment. While the path (shown in blue) is accurately estimated the 3D key points have some uncertainty (see Figure 7.7(b)). Although the points are along a tube-like structure the tunnel is hardly recognizable. Please note this visual SLAM approach was done with the RGB image of the Kinect. Data from the thermal camera did not provide any good results. The original idea was that the thermal camera is able to deliver images even the tunnel is dark or full of smoke. Unfortunately, the thermal images did not contain enough

[4]See `http://www.robots.ox.ac.uk/~gk/PTAM`.

structure.



(a) Top view.

(b) Front view.

(c) Camera view with key points.

Figure 7.7: 3D Maps generated from the tunnel dataset with PTAM.

The Structure-from-Motion approach from the Institute for Computer Graphics and Vision (ICG) at Graz University of Technology, led to similar results as PTAM as depicted in Figure 7.8. The alignment of the camera was not optimal and the structure of the tunnel too monotonic. Consequently the algorithm was only able to compute about 400 correct 3D poses.

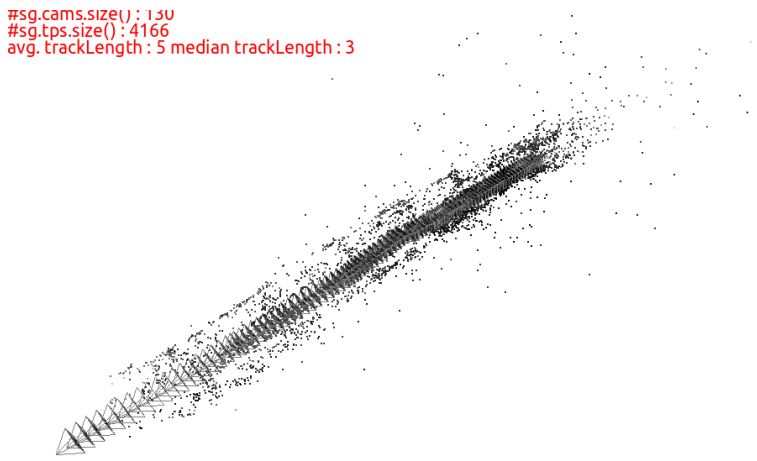


Figure 7.8: Structure-from-Motion approach by ICG.

Because of a problem in the recording of the depth image of the Kinect unfortunately the planned evaluation of RGBD-SLAM was not possible.

7.2.2 Comparison of maps

To evaluate the quality of the obtained maps, a reference map was needed. As a ground truth, an exactly measured map of the tunnel, was not available, other sources for street maps were investigated. The obvious choice was to look at the street mapping service applications available by Google, Apple and Microsoft, depicted in Figure 7.9.

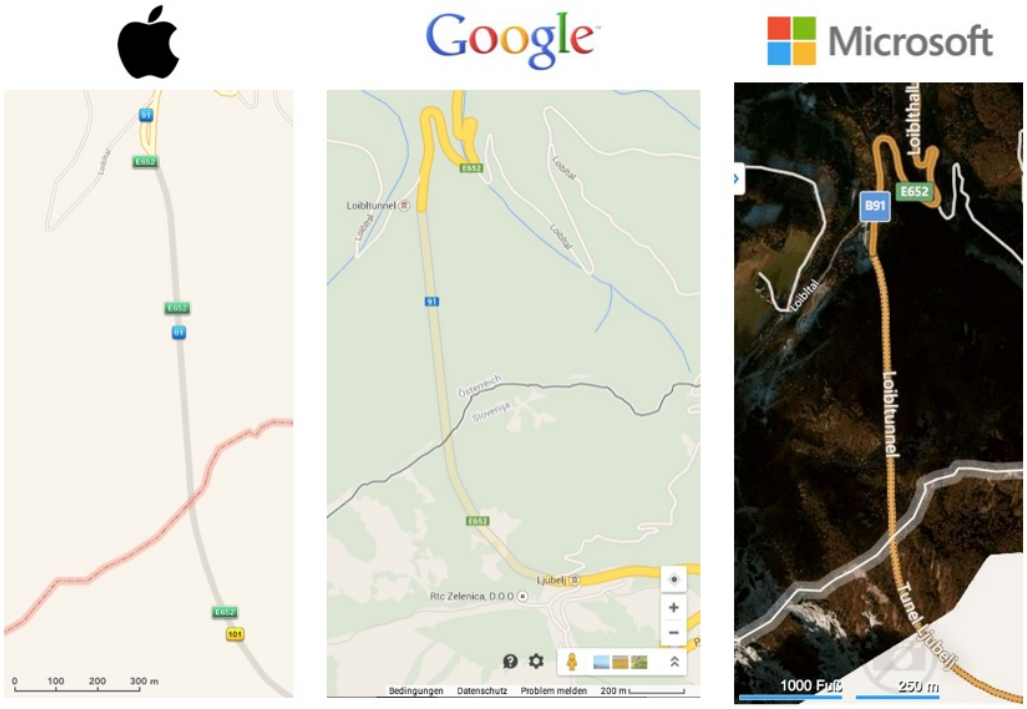
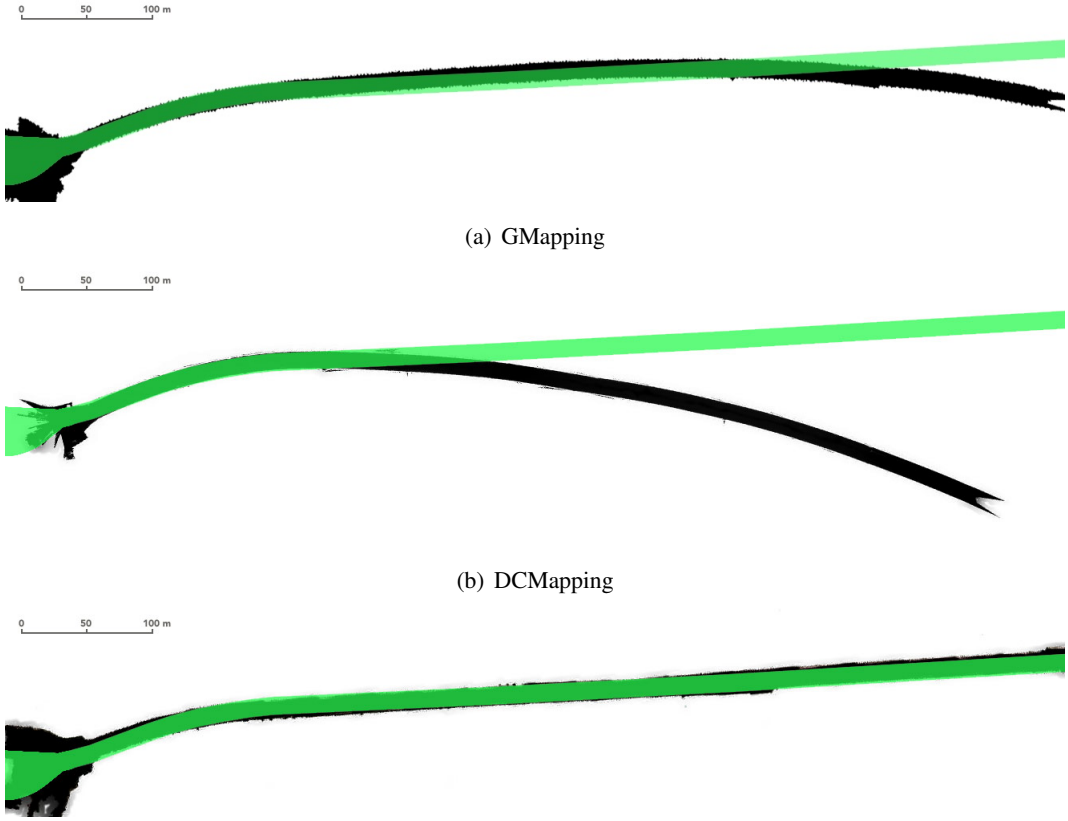


Figure 7.9: Reference maps used for evaluating Odin's results[5].

One can see from the images in Figure 7.9, that the street maps of Apple and Microsoft have a high similarity to each other and also to the maps generated in this thesis. As the tunnel mapped by Google looks like a straight line, it is regarded as not correct and not further used for evaluation.

In Figure 7.10 the reference map is layed over the acquired maps to better detect errors during the mapping process. As one can see, only three mapping approaches were used for comparison here. The other approaches were not used because the mapping failed or did not run till the end and therefore a more precise analysis is pointless.

[5] All trademarks and logos are the property of their respective owners



(a) GMapping

(b) DCMapping

(c) 3D Toolkit

Figure 7.10: Comparison of obtained maps to a reference map (depicted in green).

On the top picture, Figure 7.10(a), GMapping is aligned pretty well to the reference map. After the first bend, about 100 meters in the tunnel, there is a small rotational error due to the bad odometry data. Near to the end there is an even bigger rotational error. The reason is the same as before but additionally at this point the robot had to do some evasive maneuvers which lead to wrong alignments of scans.

DCMapping, depicted in Figure 7.10(b), has also a good alignment with the reference map for the first 100 meters. But then the bend shows a too big curvature which in sequence leads to a wrong heading of the tunnel. The cause for this misalignment seems to be lying in the voting-based approach, where the scan matching is stopped early when the algorithm detects the scans

are matching well enough (see Section 6.1.3). The rest of the map looks nearly the same as the one of GMapping, with the same little bend near the end of the tunnel when the robot had to evade. This leads to the assumption that the problem here lies not in the mapping approaches itself but in the odometry data.

In the last picture, Figure 7.10(c), the map obtained by the 3D Toolkit nearly perfectly fits the reference map. The reason for the good result here is, that the algorithm has 3D scans, where you have more features to match against. Furthermore, as the algorithm is working offline, only certain poses with their related laserscans are taken for computation which seems to have a positive effect on the mapping process. The poses have about 10 meters in distance between each other whereas in the previous approaches every scan and position was taken into account, which in the case of problematic areas like at the evasive maneuver, leads to problems during the mapping process.

7.2.3 Acceptance by First Responders

In order to asses the acceptance of the technology and the provided maps we discussed the results with first responders of a professional fire brigade. Basically, the first responders judged the 2D maps of marginal usefulness. The reason is that usually the responders have access to street maps of tunnels (metric, topological, equipment). 2D maps can be useful if the tunnel in question is new to the unit and no maps are available at all. Where for instance the length of the tunnel is less interesting than local details.

Figure 7.11: Discussion about results with first responders.

The first responders gave a totally different judgment for the 3D maps. Again, the overall map of the tunnel with information about the length or curvature of the tunnel is less important due to the same reasons mentioned already above. In contrary the responders pointed out that the detailed information about entities (vehicles, victims, debris) and their posture inside the tunnel are extremely useful. This is true in particular if the visibility in the tunnel due to electrical blackout, fire or smoke is limited. This statement suggests to intensify research not only on traditional mapping but focus also on semantic mapping approaches. If approaches are able to automatically label objects or annotate a map with high-level information this, will give a real benefit to the responders. Also the fusion of mapping with other information sources (e.g., spectral or thermal imaging) as already proposed in [45, 46] are promising. Finally, the responders judged an interactive presentation (e.g. zoom in, fly trough) of the 3D maps much more useful than a static presentation.

The responders in charge already invited us to deploy an improved system at the next emergency exercise.

7.3 Public Dataset

All data recorded during the exercise are publicly available on the web[6]. The website provides the recorded data of all sensors such as Lidar, cameras, radar data and all important static and dynamic transformations. The data is provided in several ROS bag files as mainly standard ROS sensor messages respectively standard ROS transformations. For the radar we defined a proprietary ROS message which definition is also available on the web. Moreover, we provide a simulated 2D laser scan extracted from the Velodyne data. A detailed description of the recorded data and an instruction for using them is available. We are convinced that the public data allows and motivates other researchers to further investigate the data.

7.4 Summary

In this chapter the methodology was described and the results discussed with feedback from first responders. In Table 7.1 the mapping results are summarized.

Algorithm	Used Odometry	Type of Map	Real-Time	Map Quality	Acceptance
GMapping	Yes	2D Grid Map	Yes	Very Good	Partially
Hector SLAM	No	2D Grid Map	Yes	Failed	Failed
VBMapping	Yes	2D Grid Map	Yes	Good	Partially
RGB-D SLAM	-	-	-	-	-
3D Toolkit	Yes	3D Point Cloud	No	Good	Good
PTAM	No	3D Point Cloud	Yes	Failed	Failed
SfM	No	3D Point Cloud	Yes	Failed	Failed
2D Loc with Octomap	Yes	3D Grid Map	Yes	Acceptable	Good

Table 7.1: Summary of the evaluation results.

[6] See `http://www.tedusar.eu/cms/en/research/loibl`.

Chapter 8

Lessons Learned

The preparation for and the participation at the large-scale emergency response exercise taught us valuable lessons that can be used to increase the acceptance of modern technologies for the daily use across first responders.

Preparation The time slot allocated for the reconnaissance task with a robot is very limited. There is one run where everything has to work from the beginning. There is no time for testing the setup, like brightness levels of the camera or even if all sensors are working and logging their data. Test cases have to be created and executed with the whole system in a time frame before the mission, so emerging bugs can be fixed in time. As this is nothing new in a common development cycle, it is often hard to fulfill every development step because time and resources are scarce.

The same applied to this mission. Much time was needed for setting up all the sensors and integrating them in a modular payload box.
The other problem was, that some parts were available at first the day before the exercise. As a consequence, the first test with the complete setup could first be made a few hours before the actual mission. This was a huge gamble and ultimately it didn't work out for all of the sensors.

Also, the deployed systems should have to follow a push-button setup. What that means is, that you have a switch on the robot and when you activate it, the whole system is starting up automatically without any complicated start scripts, on-site software reconfiguration or restart

of crashed components. The deployed system mainly based on ROS and standard sensors is far away from that requirement. For instance the depth information of the Kinect were lost due to a wrong compression flag and we lately found out that the robot's odometry was only available at a frame rate of 2 Hz.

Cheap is sometimes not enough Although, there is tremendous progress in computer vision, visual SLAM and cheap 3D sensors the results still are not able to compete with expensive Lidar solutions and point cloud registration algorithms. Even if there is hope that in near future such inexpensive solutions will become mature, for now in order to reach acceptance by responders, the techniques available need to to give more visually appealing and detailed information.

Odometry As stated before the robot's odometry was only available at a frame rate of 2 Hz, so some simple interpolation had to be done, so the algorithms didn't complain about a too low frequency. Furthermore to improve odometry, we integrated IMU measurements with wheel revolutions. This was done with the help of a publicly available ROS package, Robot Pose EKF, which is using an extended Kalman filter for sensor fusion. Some adjustments had to be done, as the package is outputting 3D Poses and the system needed an odometry message type.

2D Localization + Octomap Since there was no working online 3D mapping approach available at the time, Stefan Kohlbrecher of the university of Darmstadt gave us some ideas on how to use 2D localization in combination with 3D data in the form of octomaps. With this solution it was possible to generate 3D maps in real time. For a more detailed description, see Section 6.2.5.

Sensor alignment Placing the camera at the correct position is also something that has to be taken care of when using methods like structure from motion and PTAM respectively. Therefore it is not sufficient to place the camera in driving direction, although the walls of the tunnel were visible. Christian Mostegel of Graz University of Technology gave us some feedback on the correct positioning of the camera. So from his point of view the camera should be placed facing the tunnel wall, slightly upwards or downwards, to get more features from the camera image.

Chapter 9

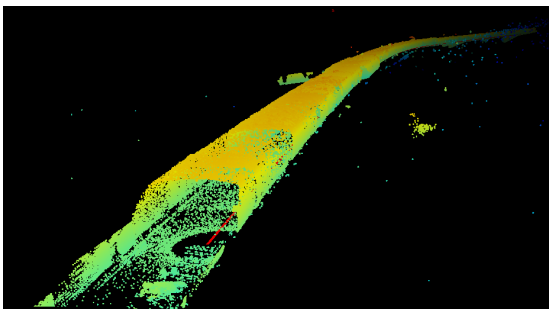
Conclusion and Future Work

In this thesis the results of deploying state-of-the-art and off-the-shelf sensors and mapping approaches in a large-scale emergency scenario were presented. In order to evaluate these techniques and to judge the acceptance of them by first responders, data were recorded during a reconnaissance mission and offline processed with state of the art mapping algorithms. High sophisticated sensors like a Velodyne 3D Lidar in combination with point cloud registration produced results that were accepted by the responders. Cheaper sensors like cameras or 2D laser in combination with related mapping approaches yet not produce useful results.

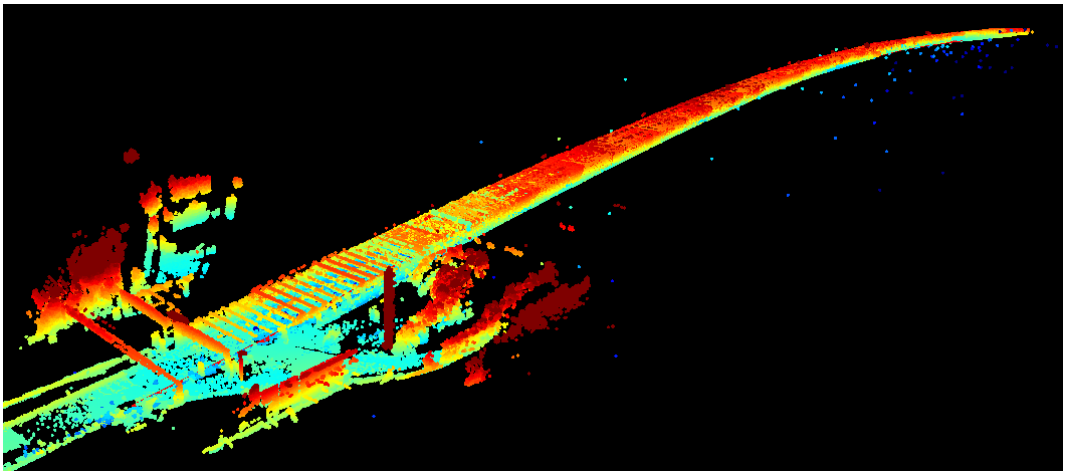
Although, we were not able to record all the desired data due to time, technical and tactical constraints, we provided sensor data and maps that can help to improve the acceptance of novel technologies by responders. Primarily, the quick and fail-safe deployment of the ROS-based robot system was a major issue. While most of the packages used are open source, it took a lot of time to install and configure each package for our needs. All data recorded in the mission is publicly available and we hope that other researchers will build up on our results. As a consequence a paper [12] regarding this work was presented at the international symposium on Safety, Security, and Rescue Robotics (SSRR) 2013 in Sweden.

In November 2014, the tunnel mapping experiment was repeated. The aim was to improve the evaluation of sensors and algorithms w.r.t. the quality, soundness, and usefulness based on experiences made during the work of this thesis. As the scenario was not an actual emergency response exercise but maintenance activities in the Plabutsch highway tunnel[1] (depicted in Figure 9.1), the time available for setup and exercise was not as critical as during the thesis work. The focus was on a subset of promising sensors based on results presented in this thesis.

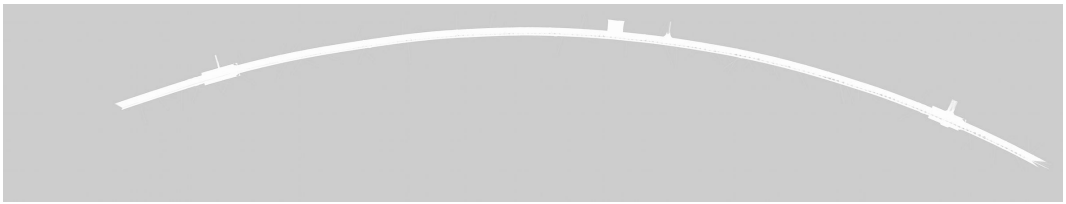
[1] See `http://www.tedusar.eu/cms/de/research/plabutsch`



(a) 3D Toolkit inside Plabutsch tunnel.



(b) Entrance view of the Plabutsch tunnel acquired with the 3D Toolkit.



(c) Plabutsch inside generated with GMapping.

Figure 9.1: Maps generated from recordings in the Plabutsch highway tunnel.

For future work we will investigate how environmental conditions (e.g. smoke, fire) affects the sensors and mapping approaches. Moreover, we have to work on a more stable robot system that is quickly and easy to deploy. Finally, the extraction of objects and dangers from sensor data is an important issue raised by responders.

Bibliography

[1] U. Wong, A. Morris, C. Lea, J. Lee, C. Whittaker, B. Garney, and R. Whittaker, "Comparative evaluation of range sensing technologies for underground void modeling," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3816–3823, 2011.

[2] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, "6d slam with an application in autonomous mine mapping," in *In Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1998–2003, 2004.

[3] T. Peynot, S. Scheding, and S. Terho, "The Marulan Data Sets: Multi-sensor Perception in a Natural Environment with Challenging Conditions," *International Journal of Robotics Research*, vol. 29, pp. 1602–1607, Nov. 2010.

[4] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *International Journal of Robotics Research*, vol. 31, pp. 1705–1711, Dec. 2012.

[5] S. Schwertfeger, A. Jacoff, J. Pellenz, and A. Birk, "Using a fiducial map metric for assessing map quality in the context of robocup rescue," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pp. 208–214, 2011.

[6] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," 2012.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[8] M. Ozaki, "Laser-based pedestrian tracking in outdoor environments by multiple mobile robots @ONLINE," Dec. 2014.

[9] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A Flexible and Scalable SLAM System with Full 3D Motion Estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, IEEE, November 2011.

[10] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 34–46, 2007.

[11] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, pp. 177–187, Feb 2014.

[12] S. G. Leingartner M., Maurer J. and F. A., "Evaluation of sensors and mapping approaches for disasters in tunnels," pp. 1–7, 2013.

[13] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, pp. 803–827, 2007.

[14] A. D. Chambers, S. Achar, S. T. Nuske, J. Rehder, B. M. Kitt, L. J. Chamberlain, J. Haines, S. Scherer, and S. Singh, "Perception for a river mapping robot," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, September 2011.

[15] J. Pellenz, D. Lang, F. Neuhaus, and D. Paulus, "Real-time 3d mapping of rough terrain: A field report from disaster city," in *Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on*, pp. 1–6, 2010.

[16] G. Brooker, J. Martinez, and R. Hennessey, "Millimetre wave radar imaging of mining vehicles," in *Radar Conference (EuRAD), 2010 European*, pp. 284–287, 2010.

[17] D. Vivet, P. Checchin, and R. Chapuis, "Radar-only localization and mapping for ground vehicle at high speed and for riverside boat," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2618–2624, 2012.

[18] R. Hahn, D. Lang, M. Haselich, and D. Paulus, "Heat mapping for improved victim detection," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pp. 116–121, 2011.

[19] O. Formsma, N. Dijkshoorn, S. van Noort, and A. Visser, "RoboCup 2010," ch. Realistic simulation of laser range finder behavior in a smoky environment, pp. 336–349, Berlin, Heidelberg: Springer-Verlag, 2011.

[20] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, pp. 133–148, Apr. 2013.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *IEEE International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[22] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Autonomous Robots*, vol. 27, pp. 387–407, Nov. 2009.

[23] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003.

[24] D. Marzorati, M. Matteucci, D. Migliore, and D. Sorrenti, "On the collection of robot- pose ground-truth for indoor scenarios in the rawseeds project," in *Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems at IEEE/RSJ IROS*, 2008.

[25] V. Lidar, "Velodyne hdl-64e product description @ONLINE," July 2014.

[26] Schunk, "Powerball lightweight arm lwa 4.6 product description @ONLINE," July 2014.

[27] H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 34, pp. 181 – 198, 2003.

[28] Mesa, "Swissranger 4000 description @ONLINE," Jan. 2015.

[29] Mesa, "Principles of kinect @ONLINE," Jan. 2015.

[30] P. L. A. M. M. H. T. G. M.R. Andersen, T. Jensen and P. Ahrendt, "Kinect depth sensor evaluation for computer vision applications @ONLINE," July 2014.

[31] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software in Robotics*, 2009.

[32] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[33] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[34] E. Olson, *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*. 2009.

[35] L. I. Andrea Censi and G. Grisetti, *Scan matching in the hough domain*. 2005.

[36] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.

[37] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, Feb. 1992.

[38] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm.," in *3DIM*, pp. 145–152, IEEE Computer Society, 2001.

[39] S. Kohlbrecher, K. Petersen, G. Steinbauer, J. Maurer, P. Lepej, S. Uran, R. Ventura, C. Dornhege, A. Hertle, R. Sheh, and J. Pellenz, "Community-driven development of standard software modules for search and rescue robots," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–2, 2012.

[40] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, pp. 61–74, July 1988.

[41] A. Kleiner and C. Dornhege, "Operator-assistive mapping in harsh environments," in *IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR)*, pp. 1–6, IEEE, 2009.

[42] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3D mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130 – 142, 2008.

[43] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, (Nara, Japan), November 2007.

[44] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr, "Online feedback for structure-from-motion image acquisition," in *Proceedings of the British Machine Vision Conference*, pp. 70.1–70.12, BMVA Press, 2012.

[45] D. Hurych, K. Zimmermann, and T. Svoboda, "Fast Learnable Object Tracking and Detection in High-resolution omnidirectional images," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2011.

[46] Borrmann, Dorit and Elseberg, Jan and Nüchter, Andreas, "Thermal 3D Mapping of Building Facades," in *Intelligent Autonomous Systems 12* (S. Lee, H. Cho, K.-J. Yoon, and J. Lee, eds.), vol. 193 of *Advances in Intelligent Systems and Computing*, pp. 173–182, Springer Berlin Heidelberg, 2013.