



Licentiate Thesis

3D Scan Matching for Mobile Robots with Application to Mine Mapping

MARTIN MAGNUSSON
Technology

3D Scan Matching for Mobile Robots with Application to Mine Mapping

*Studies from the Department of Technology
at Örebro University 17*



Martin Magnusson

3D Scan Matching for Mobile Robots with Application to Mine Mapping

© Martin Magnusson, 2006

Title: 3D Scan Matching for Mobile Robots with Application to Mine
Mapping

ISSN: 1404-7225

Abstract

This thesis is concerned with three-dimensional scan registration, in particular of underground mine tunnels. Registration of partial range scans is an essential part of building 3D maps, and autonomous creation of reliable maps is a first step towards autonomous mining vehicles.

The thesis presents a survey of relevant sensor technology and discusses the advantages and disadvantages of different sensors for use in mine environments. A survey of the state of the art in scan registration algorithms is also presented, as well as a number of relevant applications.

A new algorithm for registration of 3D data is presented, which is a generalisation of the normal distributions transform (NDT) for 2D data developed by Biber and Straßer. A detailed quantitative and qualitative comparison of the new algorithm with existing registration algorithms is shown. Results with actual mine data, some of which were collected with a new prototype 3D laser scanner, show that the presented algorithm is faster and in many cases more accurate, compared with the current standard in 3D registration.

Acknowledgements

Firstly, I would like to thank my main supervisor, Tom Duckett, for all of his guidance and friendship over the past two years. I would also like to thank my co-supervisor Achim Lilienthal for the extensive feedback, valuable ideas, and support during the writing of this thesis.

I would like to extend my gratitude to Peter Biber for the collaboration and company during his stay at Örebro University during the fall of 2004.

Thanks also to the people at Atlas Copco Rock Drills — mainly Rolf Elsrud, the manager for the 3D profiling project, and Roland Pettersson — for their participation in the work, and for access to the test mine and equipment.

Furthermore, I want to thank the people at Optab Optronikinnovation AB for giving us a scanner to work with. It was also very nice to be able to work there while writing the initial part of the thesis. Thanks to Henrik Gustafsson for good company during my time at Optab, and for all the help setting up and using the laser scanner. Thanks to Lars-Erik Skagerlund, Torgny Atterfalk, and the others for their work and their assistance when collecting data.

Many thanks, also, to all my colleagues at AASS at Örebro University, for very good company, support, and many fun times; in the coffee room, at conferences, on the go-kart track, and in the Alps. Special thanks to Henrik Andreasson for his work setting up the robot platform Tjorven.

Finally, I would like to thank Atlas Copco Rock Drills and Robotdalen for their financial support.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Application to mine mapping	3
1.4	Main contributions	5
1.5	Structure of the thesis	5
2	Sensors	7
2.1	Radar	8
2.2	Lidar	9
2.2.1	The Optab lidar	13
2.3	Sonar	14
2.4	Stereo vision	15
2.5	Projected-light triangulation	16
2.6	Discussion	17
3	Basic techniques	19
3.1	Dealing with noise	19
3.2	Registration	20
3.2.1	ICP	22
3.2.2	IDC	28
3.2.3	pIC	28
3.2.4	Point-based probabilistic registration	29
3.2.5	NDT	29
3.2.6	Gaussian fields	30
3.2.7	Branch-and-bound registration	30
3.3	Surface description	31
3.4	Localisation and mapping	34
3.4.1	Multi-view surface matching	35
3.4.2	3D occupancy grids	36
3.4.3	SLAM algorithms	36
3.5	Surface reconstruction and visualisation	38

4	Applications	41
4.1	Indoor mapping	41
4.1.1	Kurt3D	41
4.1.2	Ariadne	42
4.2	Outdoor mapping	42
4.2.1	Kurt3D	42
4.2.2	Berkeley mapping project	43
4.2.3	GPS-based mapping	45
4.2.4	Unstructured outdoor mapping	45
4.2.5	2D outdoor feature maps	46
4.3	Mine mapping	46
4.3.1	Metric mine mapping	46
4.3.2	Topological mine mapping	47
5	The normal distributions transform	49
5.1	NDT	49
5.2	3D-NDT	52
5.3	Parameters	55
5.3.1	Sampling method	55
5.3.2	Cell size	56
5.3.3	Discretisation methods	56
6	Experiments	61
6.1	Data sets	62
6.2	Results	65
6.2.1	Sampling	66
6.2.2	Cell size	68
6.2.3	Initial error	69
6.2.4	Discretisation methods	70
6.2.5	Mobile robot registration	77
6.3	Qualitative comparison	77
6.4	Discussion	79
7	Conclusions and future work	81
7.1	Conclusions	81
7.2	Future work	81
7.2.1	Dynamic environments	82
7.2.2	Scanning while moving	83
7.2.3	Semantic analysis	83
7.2.4	Sensor fusion	84
7.2.5	Discussion	85
	Bibliography	87

Chapter 1

Introduction

“If I were the boss in China, I wouldn’t allow people to work in mines. I would let them plant trees in the suburbs instead.”

—*Anonymous Chinese miner*[55]

1.1 Motivation

Underground mining is and always has been a very dangerous enterprise. People working underground have had to endure many dangers. Suffocation, falling rocks, explosions, and gas poisoning are only a few examples. Many steps have been taken to improve safety, but even today a large number of lives are lost each year in mine accidents. In China alone, thousands of people are killed every year. According to official statistics from the Chinese state administration of work safety [13], no less than 8 726 people died in mine accidents in 2004 — that means an average of 167 persons per week! Despite the horrifying death rate, the numbers were around 6 % less than the year before. The numbers are vastly lower in the rest of the world, but safety for underground personnel in mines is still a big issue. Automation of mine vehicles is therefore of great importance.

Today’s tools for tunnel profile scanning are either very slow or very expensive, and profiling currently needs to be performed separately from any other activity in the tunnel. The rock drill industry has been searching for tools for a fast, active, and cheap solution to this problem for a long time.

In addition to profiling active mines and tunnels, mapping of abandoned mines is also desirable. In many cases, no maps are available for old mines, and even in the case where maps do exist, they are often inaccurate. This may also pose safety risks. An autonomous mobile robot with 3D scanning capabilities could be sent down in tunnel systems that are inaccessible to people.

There is also a great need for fast and reliable three-dimensional scanning and model building for many applications other than mine mapping and pro-

filing. Giving an exhaustive list of application areas is impossible, but the following examples form a small sample.

Mobile robots Robots can use 3D maps for route planning and obstacle avoidance. Automated modelling is especially useful in hard to reach areas, not least for extra-planetary vehicles such as the Mars rovers. Scan matching can also be used for localisation; that is, recognising a location within a previously built model of the world from a scan of the current surroundings.

Archaeology Three-dimensional scanning can also be used to measure archaeological artifacts and historical sites. A digital model will potentially preserve a deteriorating site indefinitely, and will also allow objects to be studied remotely or processed automatically. A good example is the Stanford Digital Forma Urbis Romae project, where fragments of a shattered enormous stone map of ancient Rome are analysed with scan matching methods, trying to reconstruct the puzzle [36].

Architecture and construction A 3D model of a building can be used to verify that it was constructed according to plans, and it can also be used by architects when planning a renovation or extension.

Traffic simulation The Swedish National Road Administration is using a tunnel simulator that gives rescue personnel, police and others the possibility of practising rescue missions and traffic control for different types of accidents in tunnels, without disturbing traffic. Automatic 3D modelling could make it easier to build true models of the sites used for simulation.

Medical applications 3D modelling is also needed in many medical applications; for example, when scanning interior body parts using magnetic resonance imaging or computed tomography. One of the goals of medical image registration is to merge different images in order to improve visual judgement for treatment or diagnosis. For instance, injected instruments can be visualised in 3D or tumour positions can be detected, both without opening the patient. Medical 3D modelling applications also include surgical training and patient education.

1.2 Objectives

The objective of this thesis is to investigate methods for registration of 3D range scans for volumetric profiling of an underground mine environment.

The long term goal of this work is to make it possible for mining vehicles to operate with minimal human intervention, or completely autonomously.



Figure 1.1: The Atlas Copco Boomer 195. This vehicle is equipped with rock drills mounted on telescopic booms, and is used for drilling holes in the working rock face before blasting.

1.3 Application to mine mapping

The main application considered in this thesis is tunnel profiling (that is, building a three-dimensional model) by a range sensor mounted on drill rigs that are commonly used for tunnel excavation (see figures 1.1 and 1.2). The thesis is part of a project that is a partnership between Örebro University; Atlas Copco Rock Drills AB, manufacturers of rock drills and excavation equipment; and Optab Optronikinnovation AB, who have built a new type of 3D laser range scanner.

Profiling of mine tunnels is necessary to check that new tunnels have the desired shape, to measure the volume of material removed, to survey old tunnels and investigate whether they are still safe, and to build three-dimensional maps that could be used to accommodate for autonomous operation of drill rigs and other mining vehicles.

Several problems are associated with this scanning procedure. Naturally, it is only possible to see a very limited part of the mine at a time. Furthermore, the vehicle itself occludes a large part of the potential field of view. No matter where on the rig the sensor is placed, the drill booms and other parts of the rig will always be in the line of sight between the sensor and the walls. Therefore it is crucial to have a reliable way to match, or register, the partial scans together to form a complete and consistent model. Additionally, the environment is often wet or dusty. Airborne dust and wet, specular surfaces can lead to quite noisy sensor readings, which makes scan registration more difficult.

The main work flow when excavating a new tunnel is as follows: The first step is to drive a drill rig up to the rock face at the current end of the tunnel (the so-called working face) and drill holes in a certain pattern. These holes are

Photo: Atlas Copco.



Figure 1.2: A drill rig in its natural environment, at the working face of a mine tunnel.

then filled with explosives, and the working face is blasted away. After venting out the dust and poisonous gases coming from the blast, the debris is hauled away, the rock in the new part of the tunnel is secured to minimise the risk of the tunnel collapsing, and then the process is repeated with the drill rig moving in again.

For obvious safety reasons, no people can be present in the tunnel before the venting is completed. If the drilling, blasting and removal processes were automated, the vehicles could be driven back shortly after the blast and the excavation would be much more safe and efficient.

Currently, the drilling can be performed either manually or in a limited automated mode. The booms on which the drills are mounted have no sensing capabilities. The rock face is often quite uneven, so to avoid hitting a protruding part of the surface with the drill, the booms must make quite large movements when running in automatic mode. Instead of following the closest possible path from one drill hole to the next, the drill must be moved straight back over a fairly large distance, then sideways to the new bore hole position, and then back in to the rock face again. If a detailed 3D model of the surface were available, this process could be performed much faster.

1.4 Main contributions

The focus of the thesis is on the first and most basic step of 3D mapping: scan registration — that is, fitting partial scans together based on their shape. The main contributions of this thesis are

- a survey of relevant sensor technology and registration algorithms,
- a new registration algorithm, the three-dimensional normal distributions transform (3D-NDT),
- a detailed comparison of 3D-NDT with existing registration algorithms, using actual mine data, including data collected with the new Optab 3D laser scanner and an existing laser scanner.

1.5 Structure of the thesis

The structure of the thesis is as follows:

Chapter 2 describes different technologies used for range sensing that can be used to create 3D models of natural environments.

Chapter 3 covers some basic algorithms for acquiring, registering, and visualising 3D scan data.

Chapter 4 shows applications of the algorithms from chapter 3.

Chapter 5 details the 3D normal distributions transform — a novel 3D registration algorithm.

Chapter 6 presents a quantitative and qualitative comparison of 3D registration algorithms and experimental results.

Chapter 7 concludes the thesis and shows different directions for future work.

Chapter 2

Sensors

There are several types of measuring and sensing devices that can be used for scanning and creating a 3D model of a scene. This chapter gives an overview of some common measuring principles and discusses their utility from the perspective of modelling an underground mine system.

This chapter is only concerned with range sensors. Although texture and other surface properties can be useful to incorporate in many application, that is not covered here.

Most of the methods in this chapter are based on the same principle, namely sending out an energy beam of some sort and measuring the reflected energy when it comes back. Radar, lidar and sonar all work this way, and they have several properties in common. They are all vulnerable to the effects of *specular reflection* to some degree. Most surfaces are *diffuse* with respect to visible light — that is, they reflect incoming light in all directions. For a perfectly diffuse surface, the intensity of the reflected energy is the same from all viewing angles, and is proportional to the cosine of the angle of incidence. The more specular, or shiny, a surface is, the more of incoming energy is reflected away in an angle equal to the angle of incidence. This effect can be seen when pointing the beam of a flashlight toward a mirror, which is a highly specular surface for visible light. The lit spot on the mirror itself is barely visible, because almost all light is deflected in a specular fashion, but when looking at a nearby wall, the spot of light shows up clearly because the wallpaper is diffuse. The proportions of specular and diffuse reflection for a given surface depend on the microscopic structure, and are different for different wavelengths. The longer the waves of the incoming energy, the rougher the surface has to be in order to be a diffuse reflector. As an analogy, one can imagine throwing a ball at a structured surface, as in figure 2.1. A large ball will bounce away (specular reflection), while a smaller ball is more likely to bounce back, or in some other direction (diffuse reflection). (These examples are taken from Everett [16].)

Specular reflection can lead to serious misreadings. For example, if the beam bounces off a wall at a shallow angle, the sensor's maximum distance will be reported instead of the actual distance to the wall. Or, if the beam originates

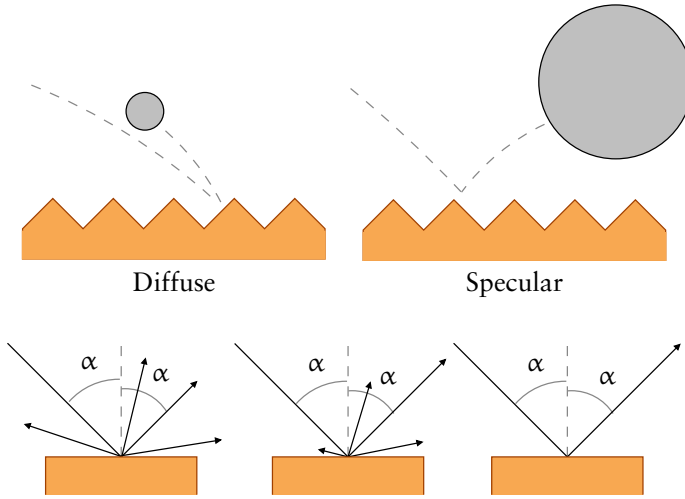


Figure 2.1: The same surface can be both a specular and diffuse reflector, depending on the wavelengths of the incoming energy. The images on the bottom row illustrate a perfectly diffuse, a moderately specular, and a perfectly specular surface, respectively.

from point A, bounces off object B and is then reflected diffusely from object C back to the sensor, the measured distance will be $(A \rightarrow B \rightarrow C \rightarrow A)/2$ instead of $A \rightarrow B$.

In general, a shorter wavelength leads to higher range resolution and less specular reflection but shorter maximum range due to absorption and scattering. Gas and particles in the air absorb and scatter the energy, making less of the beam return to the receiver, consequently decreasing the reliability of the measurement.

Diffraction is another problem common to these methods, and increases with longer waves. Diffraction takes place when energy passes an object or an opening which is close in size to the wavelength of the energy wave. The passing beam will bend around the edges and spread out from that point onwards. However, part of the wave will also be reflected back, so this is not a major problem for range scanning applications.

2.1 Radar

The term radar is an acronym for “radio detecting and ranging,” and has traditionally been used for range finding devices which use radio waves. However, in 1977 the IEEE redefined it so as to include all electromagnetic means for target location and tracking [16]. Nevertheless, this section only considers radio wave range finders.

Perhaps the most well-known application of radars is for military vessels at sea or in the air. This is also an area where it performs especially well, thanks to the long range which comes from the comparatively long wavelength. Radar is not vulnerable to dust, fog, rain, snow, vacuum, or difficult light conditions. A unique feature of radar is that it can detect multiple objects downrange. It does not penetrate steel or solid rock, however.

However, in a mine tunnel setting, there are several drawbacks of using radar for detailed model building.

The long wavelengths as compared to for example laser range finders means that the maximum angle of incidence for which they produce reliable results is rather limited. In a tunnel environment, this puts a limit on the distance ahead where the tunnel wall can be accurately measured.

Another possible limitation is the potentially large piece of hardware needed for an accurate radar. For precise modelling, a narrow so-called *pencil beam* is required. The resolution depends on the wavelength of the emitted beam as well as the aperture of the antenna. The beam width is inversely proportional to the antenna aperture for a given frequency. Using millimetre-wave radar, it is possible to get high resolution with a relatively small aperture, at the expense of range, but very narrow beams still require impractical antenna sizes for our application. At 77 GHz, which is a common millimetre-wave radar frequency, a 1° beam requires an aperture of 224 mm [19]. Larger apertures would make it difficult to mount the antenna on a mining vehicle.

Unfortunately, a single narrow beam is impossible to achieve in practice. The radiation pattern always includes a number of *side lobes* — less intense beams which spread out around the main beam (see figure 2.2). The reflections of these lobes will interfere with the main signal and lead to noisier results, not least in confined spaces, where all lobes will reflect off of nearby surfaces.

Alex Foessel-Bunting suggests fusing multiple observations to partly overcome these problems, and has presented a radar sensor model for that purpose [19]. However, this model is based on the assumptions that the radar antenna does not have strong side lobes and that there are no specular surfaces, which is not realistic.

2.2 Lidar

Even though ranging devices using laser can also be included under radar, they are also commonly referred to as lidars, “light detecting and ranging”, and in this text laser range finders are called lidars, to distinguish them from sensors using radio waves instead of light.

There are some safety concerns when using laser in environments with people, and powerful lasers can potentially be a source of ignition of methane gas or coal dust which can be found in mine tunnels. The European Committee for Electrotechnical Standardization (CENELEC) has set a limit of 150 mW or 20 mW/mm². Lasers operating above these levels are considered a potential ignition source. However, if there is a leakage of methane gas the mine needs

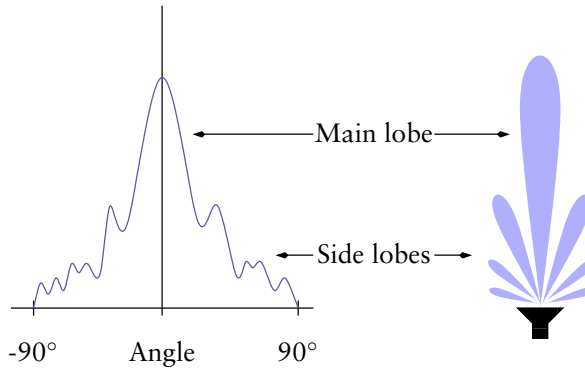


Figure 2.2: Two representations of a radar radiation pattern. The graph on the left shows the gain of the antenna at different angles from the front. To the right is a schematic representation of a top view of the antenna. The shaded regions are intended to show the gain and not the range of the lobes. This figure is adapted from Foessel-Bunting [19].

to be evacuated anyway, and current lidars with sufficient range and resolution use less effect than the specified limit, so this is generally not a real risk. A more important limitation is eye safety, which is regulated in the international standard IEC 60 825. However, because the beam rotates at high speed, it is possible to get sufficient performance without using a laser that could hurt the operators' eyes. Even if one would look straight into the beam, the short flash would not be harmful.

The accuracy and reliability of the measured data depend on the effect of the laser beam, the distance to the object surface, and its reflectance and angle of incidence; as well as the amount of time available for each measurement. Additional factors are the amount of background light, smoke, and dust; even though an infrared laser is better at penetrating smoke and dust than visible light.

The limitations stemming from attenuation are probably the main concerns when using a lidar in a mine tunnel, since it is not uncommon for the air to be quite dusty, especially if the sensor is to be used on an automated rig going into unventilated tunnels. Also, the main headlamps of a drill rig are quite powerful, and this could add some amount of noise to the sensor results if using a low-intensity laser. Simply turning the headlamps off while scanning is not an attractive solution, since it takes up to 15 minutes before the metal vapour discharge lamps that are used can be lit up again once they have been turned off. The reason for this is that when the lamps have reached their working temperature the gas pressure is so high that the voltage is not enough for lighting them again, until they have cooled down enough for the vapour to condense.

One way to compensate for difficult surroundings is to use a more powerful laser. This may not be acceptable in an environment with people, but it is also

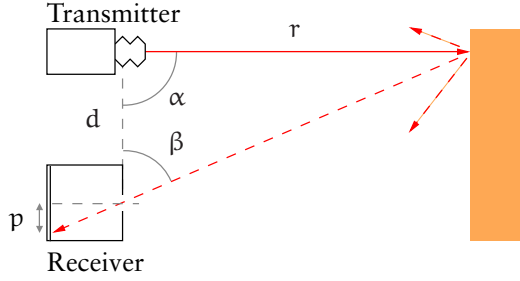


Figure 2.3: Active triangulation. The range r is measured by deducing the angle β from p and the known quantities α and d .

possible to use a laser with both higher effect and larger beam radius, thus trading a lower noise level for less lateral resolution, without affecting eye safety constraints. Similarly, one can opt to increase the aperture of the receiver unit. The strength of the received signal increases linearly with the receiver aperture, but the noise level increases with the square root of the aperture.

The distance to an object can be measured either using *triangulation* or by measuring the *time of flight* of the emitted beam, or its *phase shift*. A triangulation based lidar measures the lateral position of the spot illuminated by the laser beam as seen from the receiver, corresponding to the distance p in figure 2.3. Using this measurement, the angle β can be determined, and when β is known, calculating the range r is straightforward, using the known quantities α and d .

Another method for lidars is to emit rapid laser pulses and deduce the range by measuring the time needed for a pulse to return. Assuming that the laser travels through a known medium (such as air), measuring the distance is possible using timers with very high resolution. This is a very accurate method, but also quite expensive due to the electronics required.

The phase shift of the incoming beam compared with the outgoing can also be used to determine the distance to the closest surface, as illustrated in figure 2.4. If the phase of the light itself is measured, extremely high range resolution can be acquired at the expense of maximum range. More practical for large objects is to modulate the light with a given frequency and measure the phase shift of the modulated signal. This effectively increases the maximum range without any negative effects from increased specular reflection. If the measured shift in phase between the transmitted and the received signal is ϕ , the distance d to the target surface can be formulated as

$$d = \frac{\phi\lambda}{4\pi} = \frac{\phi c}{4\pi f}$$

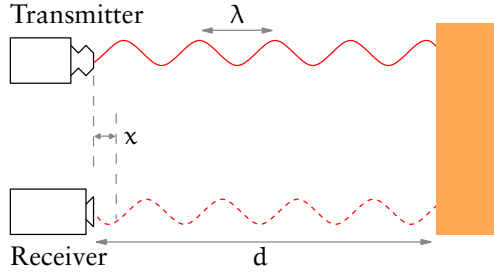


Figure 2.4: Phase shift measurement — x is the distance corresponding to the differential phase ϕ . This figure is adapted from Everett [16].



Figure 2.5: The SICK LMS 200 lidar.

where λ is the wavelength, c is the speed of light and f is the modulation frequency of the energy wave [16]. The phase shift can be measured by processing the two signals and averaging the result over several modulation cycles.

One popular type of lidar is the SICK LMS range scanners (see figure 2.5). These scanners measure range by time of flight. The SICK scanners are originally 2D sensors, but are mounted on pan/tilt units in many robotic research applications [3, 47, 48, 49, 64] to give 3D scans. An example of such a set-up can be seen in figure 6.4.

In contrast to the beams of radars and sonars, a laser beam can be made highly focused, without side lobes. This is also an effect of the short wavelength.

A lidar will have difficulty measuring the distance to a transparent surface. It will measure objects through a window, but with some added noise due to reflection and refraction. However, this is normally not something that one needs to ponder over in mining applications.

Also, highly specular surfaces can be problematic. In addition to the problem with signal loss described above, there will be an overload problem in the

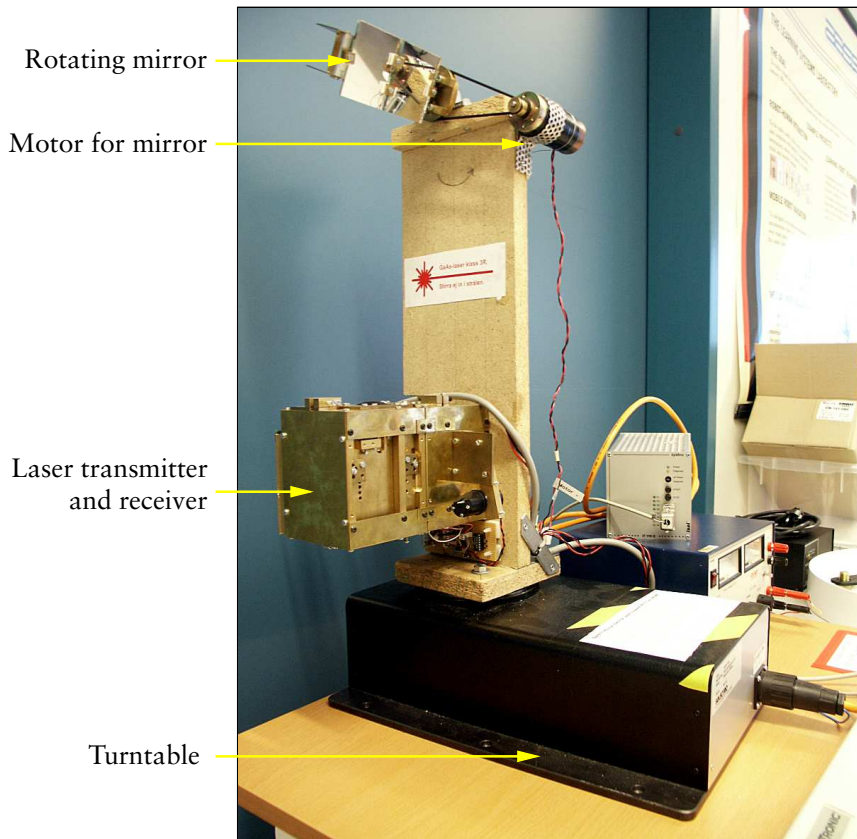


Figure 2.6: The Optab lidar lab prototype.

very rare occasions when the laser beam hits a specular surface head-on. Such surfaces are not common underground, but can occur, as certain minerals are very glossy and form flat surfaces when split. The high-intensity beam reflected back will be impossible to measure, and the electronics in the sensor will also need some time to recover; so a mirror-like surface at the exact right angle could cause a blind stripe in the sweep direction of the laser scanner. Pools of water on the ground may also give noisy readings.

2.2.1 The Optab lidar

For part of the work described later in this thesis, a lidar built by Optab Optronikinnovation AB was used. This sensor is not yet in production, but the lab prototype shown in figure 2.6 was used to gather data in the experiments.

This sensor uses the phase shift of two modulated signals to measure range. Rapid laser pulses that are modulated by a square wave at 50 MHz and 3 MHz,

respectively, are emitted. The high-frequency signal is used to get higher accuracy. Since the phase shift is cyclic and always less than 360° , the measured distance will also be cyclic and start to repeat for surfaces that are more than a certain distance from the scanner. The 3 MHz signal is therefore used to get a rough estimate of the range to the closest surface. This estimate is used to choose which possible cycle to choose from the high-frequency signal. The Optab lidar has a beam with a diameter of 4 cm, which maintains the same width over the 25 metre range for which it is dimensioned.

The data acquisition rate of the sensor is 560 000 points per second. Theoretically, all of these points could be used to get a scan with an very high angular resolution, but the range resolution would suffer. Each reported range measurement is therefore the average of a number of consecutive points. After averaging, the scanner produces 10 000 range measurements per second. By decreasing the speed at which the laser beam sweep across the surface, the accuracy can be increased proportionally, as more samples can be taken into account for each scan point. A simple statistical analysis is performed on the averaged points, and the standard deviation for the points that are used for each reported measurement is used to determine how reliable each measurement is.

2.3 Sonar

Sonar is based on a concept similar to that which is used in radar devices. However, instead of measuring the time of flight or phase shift of electromagnetic energy, sound pulses are used. Sound waves differ from electromagnetic energy in that they must propagate through a medium, and the propagation speed differs greatly between different media.

The wavelength of sound in air is very much longer than that of light or radio energy — for example, a sonar operating at 200 kHz has a wavelength of about 1.7 mm. Also, the speed of sound is of course vastly lower than the speed of electromagnetic energy. In warm air at sea level, sound travels at 357 m/s, and in sea water the speed is 1530 m/s [16]. The propagation speed and, consequently, the wavelength also depend on the temperature of the medium, and this is also something that needs to be considered when using sonars.

Traditionally, sonars have mostly been used for underwater applications, such as submarines or fishing equipment. Since the speed of sound is greater in water the wavelength is longer, and thus the range resolution is not as good as that of a wave with the same frequency in air. But the main advantage of underwater sonar is its range capacity. Water, being virtually incompressible, allows sound waves to travel hundreds or even thousands of kilometres. In air, a signal with 20 kHz loses approximately 0.06 dB/m, and the energy loss is doubled when measuring a reflected wave.

Due to the large spread of the beam, the measured distance is shorter than the true distance when the beam hits a surface at a shallow angle (see figure 2.7).

The nature of sound waves makes it difficult to focus a sound beam. The resulting lack of angular resolution disqualifies sonar for detailed 3D modelling,

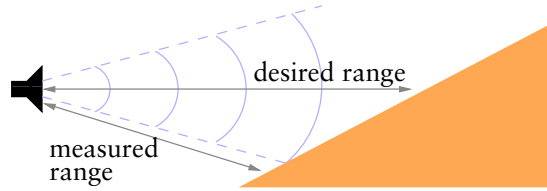


Figure 2.7: Wrong range measured because of sonar beam spread (the spread of the sonar beam is somewhat exaggerated in this figure).

but is an advantage when using sonar for example as a safety measure, to detect people in the vicinity of the drill rig.

2.4 Stereo vision

All of the sensor devices discussed above measure one point at a time. The ones described here and in section 2.5, however, produce a full two-dimensional range image at a time.

Stereo 3D sensing using two parallel cameras is a method which is rather simple regarding the hardware requirements, and exploits one of the effects used by humans and many other mammals to recognise depth.

Stereo vision uses *passive triangulation* to compute a range image. First, a point of interest is located in one image. A common method for picking interesting points is to locate pixels where the intensity gradient is high. The same point is recognised in the other image, based on its texture. Then, the lateral distances of both points are measured with respect to some common reference, and the range is calculated using the angles which can be derived from these distances, just as for active triangulation (figure 2.3). All pixels in the camera image can not be used for range measurements, only the ones that are recognisable as features, and that limits the attainable resolution for stereo vision. A laser pointer, or a grid of lasers, can be used to help getting reference points. According to Whitehorn's work [69], stereo vision can give high-resolution and fairly accurate results, in favourable conditions (when the object of interest is at a predefined and short distance). Software based methods [40] have also been used for feature detection, but since they recognise shapes from outlines and luminance data, they are difficult to use for rock wall views.

All cameras have limited depth of field, as most amateur photographers have experienced. It is only possible to have a limited range interval in focus at any given time, and the size of that interval is mainly determined by the aperture of the camera's shutter and the distance at which the focal point is set. The same is true for the human eye and the size of the pupil. Human beings use focus information as well as stereo image disparity to determine the distance to objects, but it has to my knowledge not yet been used for automated 3D sensing.

If the only light source is placed close to the cameras, and the texture of the measured surface is a uniform, diffuse colour, then the normal for each point can be derived directly from the light intensity. Such information can be useful for creating a polygon mesh from the range image, or for eliminating outlier points, though this is not likely to be of any use other than in a laboratory setting.

Stereo vision needs proper illumination, since it is a passive sensing method. To get a large depth of field, small camera apertures need to be used, and as the aperture decreases the required amount of light increases further. If only the region in front of the vehicle needs to be scanned, the lamps mounted on the rig produce sufficient illumination. If one would want to scan behind the vehicle as well, the sensor would need extra lighting in that direction.

The range accuracy decreases with the distance to the measured surface and increases with the baseline length. The baseline is the distance between the two cameras. Because the measured point must be seen by both cameras, the sensor will be blind at the closest range, unless the cameras can verge (so that the sensor can “cross its eyes”), and this minimum distance increases with the baseline length. So a stereo vision sensor with a long baseline will be accurate only at long distances, and vice versa.

Although the input rate of stereo vision is high, the main problems are that range accuracy is worse than that of electromagnetic range finders and that untextured surfaces are difficult or impossible to model, because there is no good way of recognising the same point from the two camera viewpoints [31].

2.5 Projected-light triangulation

Yet another method for 3D scanning is to project a light pattern onto the scene and analyse the shape of the pattern as seen by a video camera. Several different patterns are mentioned in the literature. Some examples include a bar code of sorts, with alternating black and white parallel stripes of different widths, a wedge, or a continuous colour gradient [58]. The projected pattern is observed by the camera, and the stripes are identified in the resulting image. For each pixel, the distance is determined with triangulation between the pixel viewing ray and the corresponding plane of light emitted by the projector. The resolution depends on the pattern and the resolution of the camera. If a stripe pattern is used, only points along the stripe borders can be measured. If a continuous gradient is used, the full resolution of the camera can be used, but only if the surface is single-coloured.

If a fixed stripe pattern is used, it is difficult to use projected-light triangulation for surfaces with discontinuities. The reason is that it is difficult to identify a certain stripe when it “jumps” between the two sides of an edge. This can be overcome by using a series of alternating patterns. Then, each pixel is identified by observing how it changes from light to dark over time, instead of identifying it from the light pattern showing in the neighbouring pixels. Each stripe has a unique on-off pattern, and the stripe can be identified by observing its history

over the last few frames. This way, discontinuous and moderately textured surfaces can be measured, but the method is on the other hand likely to fail if the object or sensor moves. This rules out using it for navigation or on a moving rig.

Using projected light makes it necessary to make certain assumptions about the texture and reflectivity of the scene, since the texture of the scene may interfere with the projected pattern. Another concern when using this method underground is the amount of background light. For the pattern to be seen clearly enough when the headlamps of the drill rig are on, the projector would have to be extremely powerful. For scanning scenes as large as mining tunnels it is also quite difficult to get sufficient edge sharpness for the light pattern. To get a bright image, the diameter of the projector lens must be large when using conventional projector methods. A large diameter leads to a shallow depth of field of the projected pattern, analogously to the effect the aperture of a camera shutter has on image focus.

The main advantage of this method is the same as for stereo vision — the speed of which range data can be acquired. A full range image can be produced 60 times per second. However, even if data is acquired at a high rate, registering and merging scans is a complex process which may still have to be performed off-line.

2.6 Discussion

It seems quite clear that only lidars will produce results with enough accuracy to be used for mine tunnel profiling, the main advantages being high resolution and relatively low sensitivity to specular reflection.

Instead of having a single sensor which is rotated to scan the full environment, it may be tempting to use multiple sensors measuring simultaneously in different directions to increase the scanning rate. Doing so can result in problems with *crosstalk* — bounced or direct impulses from nearby sensors — if the sensors are not properly shielded or the environment is highly specular. Crosstalk effects are especially pronounced in confined spaces.

Stereo vision and projected-light triangulation — and all scanning methods using a digital image device with a fixed resolution — will necessarily have lower angular resolution for surfaces at greater distances from the sensor, since each pixel covers a larger area. This is also the case with radars and sonars, because it is not possible to produce a beam which is sufficiently focused over a large distance. Because the Optab lidar works by averaging a large number of sample points, the number of points used for averaging could be decreased where the measured range is longer, thus increasing the angular resolution so that the spatial resolution (the distance between points on the surface) is maintained, at the expense of accuracy. This effect is not currently exploited.

Another class of sensors that is not covered here is contact sensing devices. Contact sensors consist of physical probes which are protruded, manually or automatically, and touch the target surface. While this method makes it possible

to measure with extreme accuracy, with an uncertainty of only a few micrometres, it is clearly not practical to use contact sensors in a mine setting.

Chapter 3

Basic techniques

This chapter serves as an introduction to previous work on mapping based on scan data. A number of methods for dealing with the various problems of the modelling process are described; starting with filtering the raw data from the sensors, to matching partial scans (which is the main topic of this thesis), to more elaborate matching methods and algorithms for visualising the results, and finally a few words on simultaneous localisation and mapping.

3.1 Dealing with noise

Noise is introduced in many steps of the modelling process. The sensor and the circumstances under which scanning is performed are noisy, and interpreting the measured data and building a 3D model can also introduce noise and inaccuracies. There are no perfect sensors, and any data acquired from a sensor will necessarily include some amount of noise.

In addition to the noise from the sensor itself, the fact that it is mounted on a moving drill rig may disturb the results further, due to vibrations and other movement. Normally, the drill rigs are equipped with two engines — a diesel engine used for transportation, and an electrical motor powering the drilling equipment. When doing tunnel profile scanning, the vehicle should not be moving and the diesel engine could be turned off. If, however, one would like to be able to perform quality scans even with the engine running, an accelerometer could be added to the sensor, and its output could possibly be used to compensate for the distortions caused by the vibrations. The frequency of the engine vibrations is not so high that a single measurement would be affected, but throughout a sweep of the laser beam, the level would be offset by different amounts. Moving the booms and drilling also gives rise to jolts and vibrations, but these could also most likely be compensated for with the use of an accelerometer. The electrical power plants do not produce vibrations that are significant when measuring on a centimetre scale. When driving and using the scanner for navigation and obstacle avoidance, most of the disturbances come from uneven terrain — not engine vibrations. One solution to compen-

sate for this source of noise would be to add an accelerometer, although a solitary accelerometer can not distinguish between acceleration and gravity. A sensor mounted on a stationary but tilted vehicle would register acceleration due to the earth's gravity. This problem can be overcome, however, by adding a gyroscope or an inclinometer. Using a less direct method, such as training an artificial neural network to recognise vibrations in an image, would probably be much more complicated and less effective; especially since one has to be able to compensate for different engine speeds and such. When using the sensor for autonomous navigation, there will necessarily be lots of vibrations and other disturbances, but the requirements for exact measurements are also much lower in that case.

A slow moving vehicle could use a 2D scanner and scan successive “slices” of the scene as it moves forward. Because of the vehicle motion, sequential scan lines would overlap unpredictably. Therefore the data points could not simply be connected, as when using a true 3D sensor. A method to cope with this is described by Huber [32]. The points from a series of scans are projected orthographically into a grid, and the range is calculated from the closest point in each grid element. This grid is then treated as a range image.

Parts of the surroundings will always be occluded — partly by the sensor itself, and partly by the large vehicle that it is mounted on. In particular, when scanning the rock face in front of the vehicle, where a highly detailed scan is desired to accommodate for automated drilling, the drill booms will cover some parts. The solution to this is to make multiple scans and register them, as described later in this chapter.

The discussion so far has only considered a moving craft in a stationary environment, but objects which are moving within the viewing area during a scan will also lead to errors. A simple solution is to have a motion sensor and abort the scan if motion is detected while scanning, and then starting the scan all over again [64]. Doing so works if the environment is mainly static with some rare occurrences of moving disturbances, but it would be better to detect moving objects and remove them from the model without restarting the scan. If a motion detection system is in place, it could also be used as a safety measure, producing a warning so that there is no risk for people to get hurt.

3.2 Registration

It is generally not possible to know the exact pose — that is, the position and orientation — of the scanner at each point where a 3D scan is made. The first scan in a sequence can be chosen as the origin, so that the global coordinate system is locked to the coordinate frame of that scan. In order to match subsequent scans, the change in pose between each scan is needed. When the scanner is mounted on a mobile robot, or other vehicle, a rough estimate of the pose change can be taken from the robot's odometry sensor, but this is not accurate enough to get a good match. Error from odometry accumulates especially fast

		Initial pose estimates known?	
		<i>yes</i>	<i>no</i>
Number of views	<i>two</i>	pair-wise registration	pair-wise surface matching
	<i>more</i>	multi-view registration	multi-view surface matching

Figure 3.1: A simple registration taxonomy

when turning. Fortunately, there are methods to deduce the change in pose by other means.

Pair-wise registration is the problem of matching two scans to build a complete model when the exact pose for one of the scans is unknown. Given two overlapping scans and an initial guess for a transformation that will bring one scan (the source) to the correct pose in the coordinate system of the other scan (the target), the output is a refined transformation. When the two scans match properly, they are said to be *in registration*.

In contrast to global surface matching algorithms, the class of registration algorithms search locally in pose space. Consequently, they may find an incorrect transformation if not given a good enough starting pose. This start pose can be selected manually, or in the case of a mobile robot, can be determined from odometry data.

Huber discusses registration and scan matching in his thesis [31], and provides the taxonomy shown in figure 3.1.

The class of registration algorithms can be further subdivided into point-to-point, point-to-feature, and feature-to-feature methods. ICP (section 3.2.1) is the most common point-to-point method, where individual points in both scans are matched to each other. It is also common to extract features such as line segments or other shapes from one or both scans. One example of this is the Cox registration algorithm [14], where points in one scan are matched to lines in the other. This is an attractive solution for structured environments, where certain shapes are known to be common. Inside an office building, for example, straight lines or planes at right angles to each other are very common, so extracting line segments may give a more usable representation of the scan data than using the actual points. In the HAYAI algorithm by Lingemann et al. [38], extreme values, corresponding to corners, are extracted from the polar coordinates of each 2D scan and matched to each other. This works best in a planar indoor environment, however. Another alternative is to extract more general features, as is done in NDT (section 3.2.5). In this case, the features describe the distribution of points within a locally bound volume.

3.2.1 ICP

The *iterative closest point* (ICP) algorithm is widely used today for registration of 3D point clouds and polygonal meshes. ICP iteratively updates and refines the relative pose by minimising the sum of squared distances between corresponding points in the two scans, and was introduced in 1992. The two seminal papers on ICP are by Besl and McKay [4] and Chen and Medioni [12]. Since its conception, a large number of variants have been developed, and a good survey of different variations of ICP was presented by Rusinkiewicz [57].

When matching high-resolution scans, one usually needs to choose a subset of points to compare. This can be done in a number of different ways. The simplest way is to use either a uniform subsample, where every n th point from the scan is selected, or to pick a random selection of points. If topology data in the form of mesh faces or surface normals at the points are available, it is also possible to subsample in a more selective manner; for example, choosing points where the normal gradient is high, choosing points so that the distributions of normal directions is as large as possible, or in a number of other ways. The preferred strategy for choosing points varies with the general shape of the surfaces. Surfaces that are generally flat and feature-less, such as long corridors, are notoriously difficult, but choosing samples so that the distribution of normals is as large as possible [57] forces the algorithm to pick more samples from the features that do exist (incisions, crevices and the like), and increases the chance of finding a correct match.

Once a set of points has been chosen, the algorithm proceeds to finding corresponding points in the other scan. This search is where most of the execution time is spent. The naive way to do this is to pick for each point its closest neighbour by Euclidean distance. While this will not in general be the correct corresponding point, especially if the scans are far apart from each other, successive iterations will still usually converge to a good solution. The matching of closest points at one ICP iteration is illustrated in figure 3.2.

Not all pairs that are found actually correspond to points that are close on the target surface. One may want to assign different weights to different point pairs, marking the confidence one has that they do indeed match. This can be done, for example, by setting the weight inversely proportional to the point-to-point distance and have lower weights for points further apart. For tunnel or corridor data, such linear weighting can degrade performance. Because most points along the walls and ceiling will generally be well-aligned, their influence will overwhelm point pairs with larger distances, which correspond to corners and other features that are important. When registering scans with different amount of occlusions, taken from the same position, linear weighting is preferable, to reduce the effect of outliers and other non-overlapping points.

Some “outlier” pairs will also need to be rejected entirely. This can be done using some heuristic to reject pairs where the points are far apart. Points lying on surface boundaries should always be rejected. Otherwise, points from non-overlapping sections of the data may cause a systematic “drag” bias —

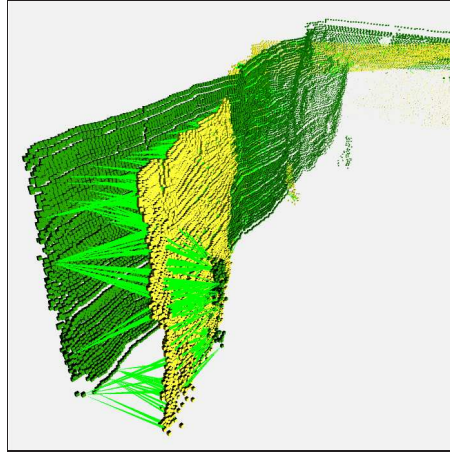


Figure 3.2: Matching two misaligned scans from a mine tunnel using ICP. The yellow scan is being matched to the dark green scan. The point-to-point correspondences are shown with bright green arrows.

see figure 3.3. Since it is difficult to determine the boundary points for point cloud data, the experiments shown later in the thesis suggest using a decreasing distance threshold for rejection instead. This way, pairs that are separated by a large distance are used in early iterations to bring the scans closer. In later iterations, pairs where the points are far from each other are likely to be pairs where a non-overlapping point is matched to a boundary point, and are rejected. The best choice of weighting and rejection strategies depends on the characteristics of the data.

Finally, the measured distances between the point pairs are minimised and the process is repeated again, with a new selection of points, until the algorithm has converged. There is a closed form solution for determining the transformation that minimises the total point-to-point error, which is described in the paper by Besl and McKay [4].

The two largest problems with ICP are that firstly, it is a point-based method that does not consider the local shape of the surface around each point; and secondly, that the frequent nearest neighbour searches are computationally expensive. Even though Chen and Medioni's version of ICP [12] uses point-to-plane correspondences, the planes used are just the tangent plane for each point. The tangent plane of a point includes minimal information about the shape surrounding the point.

Besl and McKay [4] also described an accelerated version of ICP, where the transformation vector is elongated if its direction during the last three iterations has been nearly the same. The most common convergence pattern for ICP is to take large steps towards the solution during the first few iterations, and then take smaller and smaller steps as it gets closer to an optimum. Accelerated

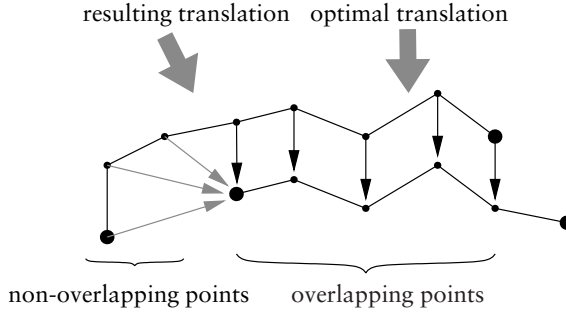


Figure 3.3: When the two scans do not overlap completely, allowing point pairs on the boundaries can introduce a systematic bias to the alignment process. The thin arrows in this figure show point pair correspondences. The shaded correspondences should be disregarded to get a good match.

ICP generally increases the step size during the later iterations, reducing the number of iterations needed. Each ICP iteration gives a point in pose space. Besl and McKay consider the angle between consecutive poses, and if the vector between the poses at iterations i and $i-1$ has a similar direction to that between $i-1$ and $i-2$, then two alternative candidates to the pose at iteration i are computed; one based on a linear approximation of the error as a function of the pose at iterations i , $i-1$ and $i-2$, and one based on a parabolic interpolant between the three points. The pose computed from the linear approximation is the zero crossing of the least-squares line, and the parabolic one is taken from the extreme point of the parabola. This is illustrated in figure 3.4.

Simon [61] improved this acceleration scheme by decoupling the rotation and translation components of the transformation. If rotation and translation are handled independently of each other, both components can be accelerated as much as possible at each step. If only the translation component has been consistent enough to be accelerated, the coupled acceleration scheme would not do anything, while the decoupled scheme would accelerate the translation vector and leave the rotation. Accelerated ICP was found to “overshoot” in some cases when testing with the mine data described in chapter 6, and is therefore not covered in the results.

The major bottleneck by far is the nearest-neighbour search when looking for the closest corresponding point of the selected subset at each iteration. If there are n points in the target data, and one wants to find the nearest neighbours of m points, a naive search requires ($O(nm)$) time. To speed up the nearest-neighbour search, the points in the target data set are commonly stored in a kd-tree structure. This is a strictly binary tree where each internal node represents a partition of the k -dimensional input space. The root node represents the entire space, and each leaf node represents a subspace that contains a subset of the points in the input data. The algorithm for a kd-tree search is shown in algorithm 3.1. The expected search time for the best matches is then propor-

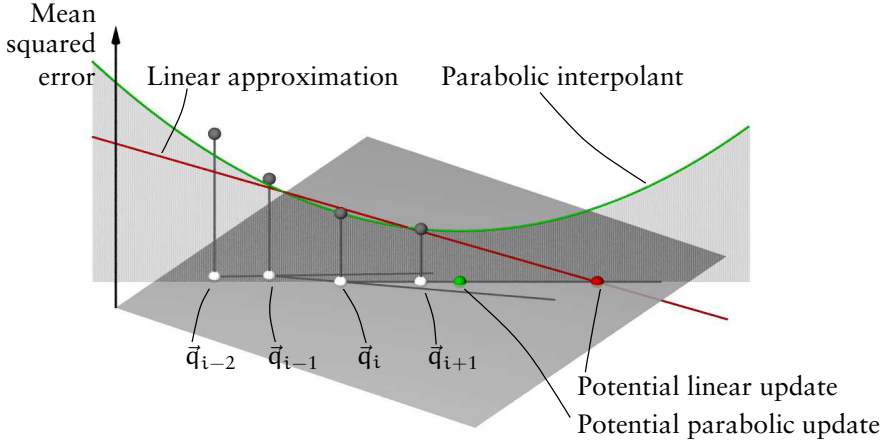


Figure 3.4: Accelerated ICP. The plane represents pose space, and the vertical axis shows the mean squared error between all point pairs. Point \bar{q}_i is the pose at iteration i . The parabolic interpolant and linear approximation are drawn along the direction specified by $\bar{q}_{i+1} - \bar{q}_i$. Unaccelerated ICP would choose \bar{q}_{i+1} as the next update, but accelerated ICP chooses either the linear or parabolic update.

tional to $m \log n$ [22]. Building the tree structure requires some additional time, proportional to $n \log n$. This is a great improvement compared to brute-force search, but the search for corresponding points still takes up a large portion of the total execution time. Several researchers have looked into improving the performance of this step. Greenspan and Godin [26] have developed a point search method, specifically tailored for ICP. They report that their method performs around 5 times faster than a kd-tree search on one sample data set where the two scans that are being matched are identical.

There are different ways of speeding up the nearest-neighbour search. It is possible to construct kd-trees in different ways to fine-tune performance. There are different possible criteria for where to split the space at each node. A common criterion is to split at the median point of all points contained in the sub-space corresponding to the current parent node. Because splitting is done along the main axes, all sub-spaces are cubic, and not spherical. Consequently, in some cases there may be a closer neighbour in a leaf other than the one that is visited first. This is determined with the so-called ball-within-bounds test (see line (i) of algorithm 3.1). In those cases the search algorithm needs to back-track and visit the other branches of the tree whose nodes intersect with the sphere centred at the query point and with the same radius as the distance between the query point and the closest match found. The test to determine this

Algorithm 3.1: KD-NN-SEARCH(\vec{p} , n)

Nearest neighbour search in a kd-tree, returning the closest point to point \vec{p} in the subtree rooted at node n .

Each internal node n stores two sub-nodes n_{left} and n_{right} , as well as an index k_n that specifies the dimension along which n splits the space, and a scalar d_n that determines the split point between the two sub-trees. Leaf nodes also contain a set \mathcal{Q} of data points, where $|\mathcal{Q}| > 0$.

```

if  $n$  is a leaf node
then
    { Examine all points  $\vec{q}_i \in \mathcal{Q}$ :
       $\vec{q} \leftarrow \arg \min_{q_i} |\vec{p} - \vec{q}_i|$ 
      if  $|\vec{p} - \vec{q}| > \text{distance from } \vec{p} \text{ to closest boundary of } n$  (i)
      then return  $(\vec{q}, |\vec{p} - \vec{q}|, \text{false})$ 
      else return  $(\vec{q}, |\vec{p} - \vec{q}|, \text{true})$ 
    }
else
    {  $d \leftarrow \vec{p}[k_n] - d_n$ 
      if  $d < 0$ 
      then  $(\vec{q}, r, \text{done}) \leftarrow \text{KD-NN-SEARCH}(\vec{p}, n_{\text{left}})$ 
      else  $(\vec{q}, r, \text{done}) \leftarrow \text{KD-NN-SEARCH}(\vec{p}, n_{\text{right}})$ 
      if not done
      Backtracking:
      then if  $d < 0$ 
      then { if the sphere centred at  $\vec{p}$  with radius  $r$  overlaps  $n_{\text{right}}$  (ii)
            then  $(\vec{q}, r, \text{done}) \leftarrow \text{KD-NN-SEARCH}(\vec{p}, n_{\text{right}})$ 
            else { if the sphere centred at  $\vec{p}$  with radius  $r$  overlaps  $n_{\text{left}}$  (iii)
                  then  $(\vec{q}, r, \text{done}) \leftarrow \text{KD-NN-SEARCH}(\vec{p}, n_{\text{left}})$ 
                }
            }
      return  $(\vec{q}, r, \text{done})$ 
    }

```

is called the bounds-overlap-ball test (see lines (ii) and (iii) of algorithm 3.1. A kd-tree with one point per leaf node is optimal in the sense that such a structure is guaranteed to require the least number of distance computations when performing the search. In practice, however, it is often more efficient to have larger bins in order to minimise the amount of back-tracking. The optimal bin size depends on the point distribution of the data, but somewhere between 10 and 20 points per bin seems to give good results for most data. A kd-tree nearest-neighbour query is illustrated in figure 3.5.

It is possible to improve the running time further by considering *approximate* nearest neighbours instead of searching exhaustively for the actual closest neighbour. Given the approximate and iterative nature of ICP, it is in practice not necessary to determine the actual nearest neighbour for each point. In fact,

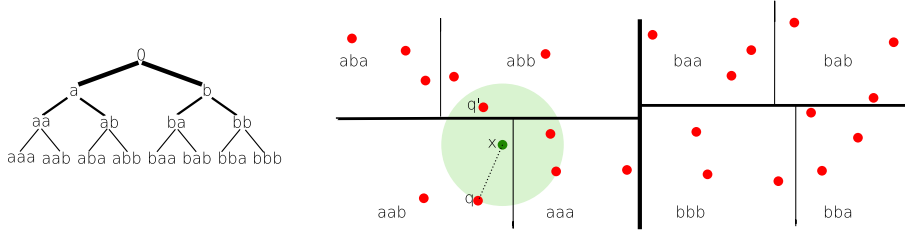


Figure 3.5: Nearest-neighbour search in a kd-tree with three points per bin. The tree structure is shown on the left, and the spatial subdivision is on the right. The query point is x , and it is contained by the leaf node aab . The closest point within this node is q , but because the minimum ball centred at x and containing q overlaps the bounds of the node, the search algorithm needs to backtrack and traverse nodes aaa and abb , too. The true nearest neighbour of x is q' in node abb .

given noisy input data with outliers, the true nearest neighbour may not be the best choice of correspondence for fast convergence [27]. If the demands on the neighbour search can be lowered, only requiring that the distance between the returned point and the actual nearest neighbour is less than some distance ϵ , line (i) can be changed to

if $|\vec{p} - \vec{q}| - \epsilon > \text{distance from } \vec{p} \text{ to closest boundary of } n$

With this modification, the algorithm performs less backtracking than algorithm 3.1, and therefore performs faster on most data. If the demands are relaxed even further, the linear search through the points in the bin leaves can also be skipped. Instead, the mean or median value of the points within each bin can be computed when the tree is created, and used as an approximation of the nearest neighbour, as done by Nüchter et al. [48].

Another class of search methods to be considered is the so-called *Elias* methods. These are strictly grid based methods, in which the space is subdivided into a lattice of congruent and non-overlapping cells, as opposed to the hierarchical and more adaptive structure of a kd-tree. For each query point, looking up which grid cell it belongs to is fast (it can be done in constant time if it is feasible to store the grid in an array). This cell and, if needed, non-empty cells around it are then searched in a concentric pattern to find the closest neighbour. See for example Greenspan et al. [28] for more details on this.

Elias methods are less attractive to use on data where most cells are unoccupied. Then, many query points will fall in unoccupied cells, and the search algorithm will have to investigate many surrounding cells before finding one with a potential nearest neighbour. With such data, the memory demands of Elias methods are also higher than for a tree structure. In the optimal case, however, finding the right cell is a constant-time operation, whereas a kd-tree query takes $O(\log n)$ time. Whether it is better to use an Elias structure or a tree depends on the shape of the data at hand.

3.2.2 IDC

The *iterative dual correspondences* (IDC) algorithm is an extension to ICP that primarily aims to speed up the convergence of the rotational part of the pose estimation [41]. With traditional ICP, the translational component usually converges relatively quickly, while the rotation is fine-tuned over more iterations.

IDC uses two rules for finding correspondences. In each iteration, the rotation–translation tuple $\tau_1 = (R_1, \vec{t}_1)$ is determined using the closest points, as for normal ICP. Without applying transformation τ_1 , a new set of corresponding points are selected using another criterion: the matching-range-point rule. This criterion uses the polar coordinates of points, and searches for corresponding points within an angular interval, which in two dimensions is formulated as $[\theta - b, \theta + b]$, where θ is the angle of point \vec{p} , and b is a bound for how far the algorithm should search. The matching-range-point rule is then formulated as follows:

$$\text{corresponding}(\vec{p}) = \arg \min_{\vec{p}'} (|\mathbf{r} - \mathbf{r}'|),$$

where $\vec{p} = [\theta, r]$ and $\vec{p}' = [\theta', r']$, and $\theta - b \leq \theta' \leq \theta + b$. In other words, the corresponding point is the one within the specified angular interval that has the most similar range coordinate. A second transformation $\tau_2 = (R_2, \vec{t}_2)$ is computed using the correspondences found with this method, and the transformation that is applied before the next iteration is $\tau_3 = (R_2, \vec{t}_1)$.

If this should be adapted to three dimensions, the interval $[\theta - b, \theta + b]$ would instead be a rectangular “window”. For a 3D point \vec{p} with polar coordinates $[\theta, \omega, r]$, where θ and ω are the latitudinal and longitudinal angles, the “window” would extend from $[\theta - b_\theta, \omega - b_\omega]$ to $[\theta + b_\theta, \omega + b_\omega]$.

3.2.3 pIC

Montesano, Minguez and Montano [45] have presented yet another scan registration algorithm, called the *probabilistic iterative correspondence* method, or pIC for short.

This is a probabilistic method, which tries to incorporate information about the uncertainties from both scanning noise and the uncertainty of the initial pose estimate. When using pIC, the initial set of possible correspondences is first reduced to a subset that contains all the points in one scan that are statistically compatible with the ones in the other, based on the Mahalanobis distance $d_M(\vec{p}, \vec{q})$ between point \vec{p} and \vec{q} , which is defined as

$$d_M^2(\vec{p}, \vec{q}) = |\vec{p} - \vec{q}| \mathbf{C}_{ij}^{-1} |\vec{p} - \vec{q}|,$$

where \mathbf{C} is a Taylor approximation of the transformation function that transforms the reference scan to the target coordinate system. If the noise is normal-distributed, the Mahalanobis distance has a χ^2 distribution with two degrees

of freedom for the 2D case, and three for 3D. Two points are considered compatible if the Mahalanobis distance between them is less than some specified confidence threshold. The set of points passing this confidence test defines a set A , and the expectation that point $\tilde{a}_i \in A$ is the best correspondence for point \tilde{p} is found by integrating over all possible locations of \tilde{p} and all possible locations of the sensor. More details on this can be found in the paper by Montesano et al. [45]. The results presented in this paper are based on experiments where two copies of the same scan were registered from a start pose with quite large error. Two scans were used, both of them from an indoor environment. The pIC algorithm converged to the correct solution in all cases, after about 25% as many iterations as ICP needed. Montesano et al. did not compare the execution time of the algorithms.

3.2.4 Point-based probabilistic registration

Hähnel and Burgard [30] present another probabilistic registration algorithm. This algorithm models the target scan with probability functions, each of which is a combination of a normal distribution and a uniform distribution.

To compute the likelihood of a scan point from the source scan, a ray is traced from the current estimate of the scanner pose, along the direction associated with each measurement from the source scan, to the closest surface in the target scan. The length of this ray is taken as the estimated range of this measurement. The likelihood of the current measurement is then computed from a combination of a normal distribution that is centred at the estimated range, with a variance tuned to the characteristics of the scanner, and a uniform distribution, that also is tuned to the accuracy of the scanner. The probability of scan point \tilde{s} from scan S at pose \tilde{p} , when matching to scan T , is $p(s, T, \tilde{p})$. The probability that the current source scan should be located at pose estimate \tilde{p} is then described as the product

$$\prod_{s \in S} p(s, T, \tilde{p}).$$

The algorithm tries to optimise the value of this function. In their paper, Hähnel and Burgard present results from one pair of 3D scans of a large building, and show that their algorithm gives more accurate matches than ICP on this data.

3.2.5 NDT

The normal distributions transform method for registration of 2D data was introduced by Biber and Straßer [6]. The key element in this algorithm is a new representation for the target point cloud. Instead of matching the source point cloud to the points in the target directly, the probability of finding a point at a certain position is modelled by a linear combination of normal distributions.

This gives a piecewise smooth representation of the target data, with continuous first and second order derivatives. Using this representation, it is possible to apply standard numerical optimisation methods for registration. Numerical optimisation is a problem that has been studied for centuries, and fast and reliable methods for optimising functions such as a sum of normal distributions have been developed and tested over time. Because the points in the target scan are not used directly for matching, there is no need for the computationally expensive nearest-neighbour search, as in ICP. Computing the normal distributions is a one-time task that is done during a single pass through the points of the target scan.

NDT is described in more detail in chapter 5.

3.2.6 Gaussian fields

Boughorbel et al. [9] have developed a registration criterion based on Gaussian fields, similar to the normal distributions of NDT. The basic idea of this approach is to use a Gaussian field to measure both the spatial distance and visual similarity of two points from two scans. Points are then compared in a multi-dimensional space that consists of the spatial dimensions plus a number of attribute dimensions. The attributes used to measure the visual similarity are 3D moments as described by Sharp et al. [59] (see section 3.3). The measure of proximity and similarity between two points \vec{p} and \vec{q} from different scans is formulated as

$$F(\vec{p}, \vec{q}) = \exp \left(-\frac{|\vec{p} - \vec{q}|^2}{\sigma^2} - (\mathbf{S}(\vec{p}) - \mathbf{S}(\vec{q}))^T \mathbf{\Sigma}^{-1} \times (\mathbf{S}(\vec{p}) - \mathbf{S}(\vec{q})) \right).$$

This expression describes a statistical normal distribution, centred at \vec{q} and decaying radially in Euclidean and attribute space. The parameter σ specifies the decay rate with respect to the spatial distance, and the diagonal matrix $\mathbf{\Sigma}$ punishes difference in attributes. The criterion for measuring the fitness of a pose is then defined as the sum $\sum_{i,j} F(\vec{p}_i, \vec{q}_j)$ over all possible point pairs.

3.2.7 Branch-and-bound registration

Some researchers [20, 52, 53] have used a branch-and-bound strategy for scan registration. The translation part of the pose space is discretised at several resolutions, with levels ordered from the coarsest to the finest. A number of positions are considered at some level of the pyramid. The best matches are considered at the next lower level of the pyramid (branching), and the others, along with all their subnodes in the pyramid, are discarded (bounding). In the paper by Forsman and Halme [20], this is used for a forestry application, and it can be useful for highly unstructured environments. Even so, it is mainly attractive for 2D applications. The branch-and-bound step is only applied for the translation part of the transformation. In the applications covered by these papers,

the orientation of the robot is deduced from other sensors, and rotation can be optimised simply by trying the registration for a number of sampled candidate rotations around the initial rotation estimate [53]. In unrestricted 3D-space, the number of candidate rotations that needs to be considered grows substantially, so this approach is not likely to scale very well.

3.3 Surface description

One weakness of ICP in its basic incarnation is that it is based on point-to-point matches. As such it does not make use of the local surface shape around each point, other than perhaps the estimated surface normal at the sample point. Some other description of the surface shape is needed for more robust scan matching. It should describe a limited local part of the surface, so that differences in the scans due to occlusion do not interfere more than necessary with the matching.

If the local surface descriptions are also invariant to rigid motion (that is, rotation and translation), corresponding surface parts can be found regardless of the initial poses of the scans. If sufficiently prominent features can be found, this makes global scan-matching (that is, registration without an initial pose estimate) possible.

One might also want to consider features that are invariant to additional affine transformations, for example scaling or shearing, but this is not covered here, and such features are not needed for rigid registration.

One surface description technique is *spin-images*, introduced by Andrew Johnson. Their utility for surface matching and object recognition is covered in detail in his thesis [35]. A spin-image is based around an oriented point on the surface — that is, a point with a normal. Spin-image creation can be thought of as placing an image raster at the oriented point with one of the image borders aligned along the normal of the point. The image plane is rotated around this axis. Each pixel then represents how much of the surface is passed by that pixel during its trajectory (see figure 3.6). In Johnson's original implementation, a small number is added for each vertex which the pixel passes as it sweeps through space. This simple method works for some data, but the points of the scans must be relatively uniformly spaced in order to make sensible comparisons. In tunnel scans, points that are far from the sensor location are much sparser than ones in the sensor's vicinity. One way to deal with this would be to refine the scan by interpolating new vertices in the outermost parts of the scan, but a better way, at least if the input data is a polygon mesh and not an unordered point cloud, is the face-based spin-images presented by Huber [31]. Each surface polygon is subsampled in a raster-scan pattern. Then the subsampled points, instead of the vertices, are used for creating spin-images. By subsampling the mesh faces, even scans with widely different resolutions but similar shape can be compared.

Cylindrical spin-images, as described above, are quite sensitive to error in the point normals. Two nearby points with slightly different normals can have

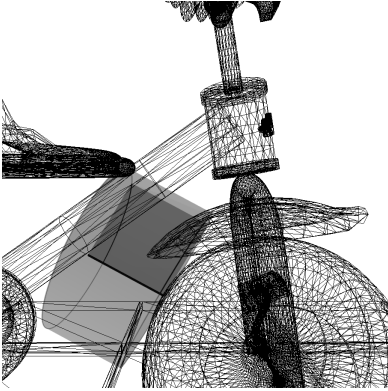


Figure 3.6: A spin-image is a projection of a surface onto a bounded cylindrical coordinate system.

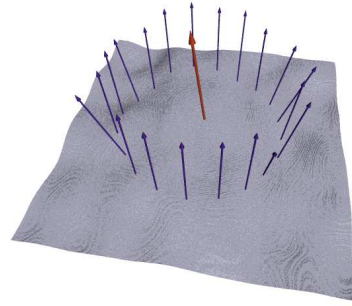


Figure 3.7: A splash is a collection of normals around a point.

very different spin-images. Unfortunately, it is difficult to compute reliable point normals for noisy scans. If normals are not available from the original scan, it can also be a time-consuming process to generate them for each vertex. As a possible solution, Johnson has also suggested spherical spin-images [33], which are less sensitive to this source of error. For cylindrical spin-images, the error for the outer pixels is much larger than for more central pixels when the point normal is poorly estimated. This effect is decreased with a spherical parametrisation. A further refinement is to compress the image memory footprint using principal component analysis [34]. While this is a lossy compression, and as such degrades the descriptive quality of the spin-image slightly, compressed spin-images are much cheaper to compare to each other, and this is the most computationally demanding step of spin-image scan matching. The speed-up gain is likely to outweigh the small decrease in accuracy for many applications, and this is the case for the data presented in the cited paper.

An alternative surface description is the *splash*, presented by Stein and Medioni [63]. Splashes are also based around oriented points. A circular path around the point is computed, so that each point on the path has the same distance to the centre point when measuring the distance along the mesh's surface. The normal of the surface along this circle is measured at equally spaced radial intervals, starting at an arbitrary point (see figure 3.7). This gives a one-dimensional record (in a circular coordinate system) of the local surface shape around a certain point. Several splashes with the same central point but different radii can also be combined in a super splash, thus giving a 2D image. Splashes do not handle discontinuities well, and Stein and Medioni complete their surface description with 3D curves, based on object edges. Edges are ex-

tracted from the range image and 3D polygonal line segments are fitted to each edge. Several lines, each with a different number of line segments, are created for each edge in order to increase robustness to noise.

Sameh Yamany and Aly Farag have presented yet another alternative: *surface signatures* [72]. These are similar to spin-images, but use surface curvature information instead of point density. A surface signature also constitutes a 2D representation of the surface as seen from a single point. In this case, however, the vertices in the mesh itself are not used, but a simplex mesh is created from the centre point of each polygon. If the original mesh is made of triangles, the resulting simplex mesh will in general consist of many different types of polygons, but each simplex point will have three neighbours. A simplex point and its three neighbours can be circumscribed by a sphere, and this sphere is used to determine the curvature in a point. Starting from a certain point \vec{p} with normal \vec{n}_p , a two-dimensional histogram is created with the Euclidean distance $d_i = |\vec{p} - \vec{p}_i|$ on one axis and the angle

$$\alpha_i = \cos^{-1} \left(\frac{\vec{n}_p(\vec{p} - \vec{p}_i)}{|\vec{p} - \vec{p}_i|} \right)$$

on the other. The local curvature of each other point \vec{p}_i is plotted at the corresponding position in the histogram. In the work of Yamani and Farag, the full mesh was encoded in this way. To make a local surface signature, one can just put a limit on the maximum distance d_i .

All of these descriptions are based on oriented points, so it is important that the normals of all points are accurate. If not, it is not possible to generate correct surface descriptions. Because of this, only points where the normal can be determined reliably should be selected. A consequence of this can be a very large overlap requirement when matching tunnel profile scans. Since the normal of the surface can not be measured directly, it has to be derived from several neighbouring points. At the far ends of the scan the uncertainty of range points is higher, which would lead to a very uncertain normal.

Sharp et al. [59] have used other kinds of invariant surface features for ICP registration. In their work, each point is represented with $k + 3$ parameters; three of which are the Euclidean position coordinates, and the other k are the feature coordinates. The features are based on surface curvature, moment invariants, and spherical harmonics. Corresponding points are then found in $k + 3$ dimensional space, using separate weights for the Euclidean and the feature parts of the point vector. This way, a point that is farther away in Euclidean space but has a more similar local surrounding may be considered as a better pairing than the closest point in 3D-space. In their article, Sharp et al. use $k = 8$ feature parameters; two for curvature, and three each for moments and harmonics. For more details, see their paper [59]. However, searching in 11-dimensional space is more problematic than searching in two or three dimensions. Use of kd-trees does not scale very well to higher dimensions, and in spaces with more than eight or so dimensions, using kd-trees does not generally improve performance over brute-force search. Furthermore, computing the

features described by Sharp et al. is quite expensive. According to the paper, spherical harmonics gave the largest increase in the region of convergence, but they took 32 s to compute for a 128×128 range image. Moment invariants took a further 9 s, and curvature (which is an unreliable feature) three more seconds. Additionally, if the weight for the feature components is scaled over time, any binned search structure used to accelerate neighbour search will have to be rebuilt every few iterations, increasing the time needed even more.

There are also non-local surface descriptions, such as *shape distributions* [54], but these are not useful for finding corresponding patches of partly overlapping surfaces. However, because they give a compact representation of a complete 3D shape, they are still useful for object classification and recognition in certain applications.

The so-called “regularised medial scaffolds” used by Chang et al. [11] are another attempt at developing 3D features for global scan-matching. This is essentially a way of generating a 3D skeleton of a scan using maximal balls, similar to what is commonly done in 2D image processing. The regularisation in this case is the process of simplifying the scaffold to reduce it to its most essential parts. The regularisation step is done to improve the search speed, but also takes away information about smaller details that are necessary for precise registration. Chang et al. used the regularised medial scaffolds to match pairs of scans approximately, and then used ICP to register them more accurately.

3.4 Localisation and mapping

Even using sophisticated techniques such as the ones described above for registration is not enough to build a coherent map of a large environment. Errors from each pair-wise scan registration will accumulate over distance; an effect that is most noticeable when the mapping agent comes back to a previously explored position after having traversed a loop. Even if the local error from each match is small, it is likely that the estimated pose of the current scan is far from the pose of the corresponding previous scan of the same location. This can be seen in figure 6.3.

This problem is referred to as the simultaneous localisation and mapping problem, or SLAM, for short. Given a map, it is easy to localise oneself in the environment. Conversely, it is easy to build a map as long as the localisation is accurate. Doing both things at once without external help, however, is very difficult, and an entire literature has sprung from this problem. The main reason why it is so difficult is that the same sensor data has to be used for both mapping and localisation. For a survey of SLAM algorithms, see Thrun [66].

There are three main types of localisation problems in mobile robotics:

Pose tracking, which is the problem of updating the estimated pose of a mobile robot when there is a prior estimate of the pose.

Global localisation — the problem of finding the pose when there is no prior estimate, and the robot could be anywhere within a defined space.

Kidnapped robot, when there is an erroneous prior pose estimate. This problem is related to a scenario in which the robot has successfully localised itself on the map, but is “blindfolded” and moved to another position. The robotics then has to re-localise itself without knowing that it has been transported.

Several algorithms for solving the SLAM problem have been developed, most of them for 2D maps, but the problem is still not solved. Existing algorithms fail when the environment is either too feature rich (cluttered with distracting or irrelevant features) or lacking features (bare and monotonous), some fail or become intractable when open-loop uncertainties become too large, and some consume too much memory or time as the map grows. SLAM in dynamic environments is also largely an unsolved problem.

3.4.1 Multi-view surface matching

One way to deal with the global localisation problem can be to use multi-view surface matching (defined in figure 3.1) in order to enforce a consistent model built from multiple scans.

When performing multi-view surface matching, the first step is to find out which scans overlap and also find approximate relative poses for them after which a multi-view registration algorithm aligns them properly.

Daniel F. Huber has presented a pioneering multi-view surface matching algorithm [31]. This algorithm has one local and one global phase. In the local phase, pair-wise surface matching and registration is performed on all pairs of views, building a graph of potential pose estimates. This graph may contain many incorrect matches. The surface matching stage is based on spin-images. Each match is refined with pair-wise registration. However, even matches where the local error measurement is zero may be incorrect, if parts of the model are symmetric, for example. In the global phase, this graph is scrutinised and global consistency constraints are enforced to weed out incorrect matches. This algorithm has $O(n^2)$ complexity, so it is limited to matching only a small number of views (around 50) simultaneously.

Although this is an interesting effort, it is of limited interest when considering a map building robot. The problem of multi-view surface matching is very general in that it does not assume any initial estimate of relations between different scans. When building a mine map, however, a good sense of which scans overlap with each other typically exists; both when scanning repeated views from a stationary drill rig (when building a model of a newly blasted area) or when scanning from a moving rig (when building a 3D map). Normally, a mine tunnel scanner would produce an *ordered* set of 3D views of a static scene. If scans are made at regular intervals along some trajectory through the tunnel, good initial alignment estimates already exist, and one can assume that sequential scans overlap.

3.4.2 3D occupancy grids

An occupancy grid, or evidence grid, is a discrete 2D or 3D grid in which each cell stores information about the probability of that area or volume being occupied by some object. The result is typically a low-resolution representation of the surface shape. The first approaches to robotic mapping were based on occupancy grids (see Moravec and Elfes [46]). They are well suited to integrate the noisy and low resolution input that is provided by sonar range finders, and this is one major reason why occupancy grids were used before laser range finders became more common. However, if sensors with higher resolution and accuracy are available, it is typically better to use another surface representation. One drawback of occupancy grids is that updating the grid with new data requires very accurate localisation. Once a scan at a certain pose has been incorporated into the occupancy grid it is not possible to improve its pose later. Another drawback is that occupancy grids do not scale very well with increased resolution (increasing the resolution of a 3D grid 10 times requires 1000 times more memory).

Fairfield et al. [17] used 3D occupancy grids for underwater mapping. For submarine environments, there are not many sensors that can provide the necessary resolution to recognise features that can be used by more sophisticated algorithms, so sonars and occupancy grids are good options for such applications. In the application of Fairfield et al., a probe with an array of sonars was submerged in a large natural water-filled sinkhole. The depth and heading of the probe were known with high accuracy from other sensors, so only the position in the current horizontal 2D plane needed to be determined from scan matching.

The actual localisation in their algorithm is performed with a particle filter, where the likelihood of the particles is evaluated with two methods: ray tracing and point correlation. For ray tracing, a ray is traced for each sonar beam from the sensor position through the surrounding grid cells until some surface evidence threshold is met. The point correlation method works by evaluating the probabilities at the grid cells where each sonar beam ends, given the current position hypothesis. So the desired probability that the sensor is at that pose is just the product of all the probabilities. The evidence in each cell can be iteratively updated during subsequent drops.

3.4.3 SLAM algorithms

The Kalman filter (see Welch and Bishop [68] for a good introduction) is central to many current SLAM algorithms. The Kalman filter is a set of mathematical equations that provides a recursive computational means to estimate the state of an uncertain process, in a way that minimises the mean of the squared error. The basic Kalman filter is used to estimate the state of a discrete-time process that is governed by a *linear* stochastic difference equation. A Kalman filter that is used for non-linear equations is referred to as an *extended Kalman filter*.

Each step of the process is represented by a mean vector, describing the most probable state of the process at that time (before applying the filter, this is the measured state), and a covariance matrix, describing the probability density for the estimate (taken from a model of the expected measurement error). The estimated pose of an autonomous robot is a good example of such a process. For SLAM, the state of the system includes the robot pose and the positions of a number of landmarks or previous poses where scans have been taken. Landmarks are either features extracted from the data, such as corners or other prominent structures, or select views. Together, the landmarks constitute the robot's representation of the map. The Kalman filter can then be used to recursively update the map and the estimate of the robot's pose, as well as the uncertainty measures for these variables. The largest drawback of this method in its basic form is that the time and memory requirements grow as $O(n^2)$ with the number n of landmarks, because a square $n \times n$ matrix for the covariances of all landmarks needs to be maintained.

One attempt to overcome this limitation is an algorithm named FastSLAM, presented by Montemerlo et al. [44]. FastSLAM is based on a particle filter — a method that is useful for solving global optimisation problems of high dimensionality. Every particle in the system represents a hypothetical robot trajectory. Each particle also stores a set of n Kalman filters that estimate the position for each landmark passed by its trajectory. Because these estimates are independent, n small covariance matrices are needed instead of one large matrix. The computation time for adding a new observation is $O(M \log n)$ if there are M particles, and the memory required grows as $O(Mn)$. Still, when closing a loop, a prohibitively large number of particles may be needed. This is because a particle filter integrates new measurements by choosing a subset of particles that is compatible with the measurements from the set of already existing particles (in other words, re-sampling). The particle filter is not capable of modifying the robot trajectory represented by the particles chosen to make it fit the new observation better, nor can it back-propagate the error along the loop. So the particle set must be large enough to contain a particle that is sufficiently close to the true pose of the robot at all times, and that can be difficult for environments with large loops.

Frese et al. [21] presented a multi-level relaxation algorithm for SLAM, inspired by the so-called multi-grid methods used for solving partial differential equations. This algorithm does away with Kalman filters altogether. The input to the multi-level relaxation algorithm is a set of pose estimates and a set of relations — that is, likelihood distributions of the relative pose change between a pair of poses. Naturally, only a subset of all n scans overlap with any given scan, so the total number of relations is kn , where k is number of scans that overlap an average scan. The algorithm then tries to maximise the total likelihood of all measurements through relaxation, which is an iterative process. Standard multi-grid methods for solving partial differential equations (PDEs) work by discretising the PDE at different resolutions for different iter-

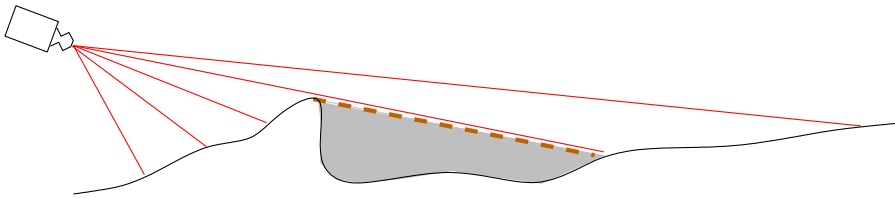


Figure 3.8: Range shadow caused by a bump at a shallow angle. The phantom surface marked by a thick dashed line should be removed.

ations, starting at one level of discretisation and then cycling in some pattern between finer and coarser levels. The error function can be oscillatory on certain levels of discretisation, and the multi-grid approach is meant to get the advantages of both smoothing the error on some iterations and getting a more accurate description of the problem on some iterations by using multiple levels of discretisation. The multi-level relaxation algorithm for SLAM implements different discretisation levels by using every m th pose of the robot’s measured trajectory. The algorithm uses a V-cycle, which means that the algorithm starts at the finest level, iterates through successively coarser levels and then back up to the finest level again. Two operators control the transition between levels: one fine-to-coarse *restriction* operator, and one coarse-to-fine *interpolation* operator. For each fine-to-coarse transition, every second frame is selected. The size of the levels therefore decreases exponentially, so the computation time is asymptotically the same as for the finest level, regardless of how many levels are used. The algorithm is shown to use only $O(kn)$ memory and time at all stages, both for incremental updates and when closing a loop.

3.5 Surface reconstruction and visualisation

For accurate and visually pleasing volumetric modelling, it is not enough just to register the raw scans. No two scans are likely to match seamlessly, due to scanner noise and movement, and there will also commonly be holes in the model due to range shadows. Reconstructing the surface for a pleasing end result is an interesting problem in itself.

Building a surface mesh from a single range image generated by a range sensor is not trivial. Simply “connecting the dots” — creating surface patches between all range pixels in the image — will create “phantom” surfaces where range shadows occur, as shown in figure 3.8. These will need to be detected and removed. See Huber’s work [32] for a robust way to fix this.

Merging multiple scans is another problem. Using the closest (innermost) points of overlapping tunnel scans leads to “thickening” of the walls [58]. Simply merging the point sets and creating a surface mesh where all points are used individually may lead to an unnecessarily rough and noisy surface. Curless and Levoy [15] presented a volumetric algorithm for reconstructing a surface com-

posed of aligned range scans that fills gaps in the reconstructions, is robust with respect to outliers, and can be used for incremental updating (adding new scans as they are made available to the system). Their algorithm maintains a voxel grid, similar to a 3D occupancy grid, that covers the volume occupied by the surface. The volumetric representation consists of a cumulative weighted signed distance function, such that the zero crossing of the distance function coincides with the surface. Incorporating one 3D scan at a time, it is first converted to a distance function, and then combined with the rest of the voxel grid using an additive scheme. The final surface is created by extracting an isosurface from the volumetric grid for all points where the distance function equals zero.

Chapter 4

Applications

This chapter describes a number of notable applications of 3D mapping and localisation.

4.1 Indoor mapping

A lot of work has been done with indoor mapping of static, office-like environments. Such structured, small-scale environments are comparatively easy, because a lot of assumptions can be made about the structure of the surroundings. For example, floors are generally assumed to be flat, and walls are assumed to be composed of flat, planar sections. It would be nearly impossible to cover all applications of indoor mapping here, but a few interesting examples are presented.

4.1.1 Kurt3D

The Fraunhofer centre for autonomous intelligent systems have a number of robot platforms, the most current of which is the Kurt3D robot.

Kurt3D is a relatively high-speed mobile robot platform, moving at a controlled pace of up to 4 m/s (14.4 km/h). The motors allow for speeds up to 5.4 m/s (19.4 km/h), but the current computer hardware and algorithms can not control it reliably at such speeds. The robot is equipped with a SICK laser scanner as well as two digital colour cameras.

For indoor use, approximate pose estimates are updated using the HAYAI algorithm [38] mentioned in section 3.2. The SICK scanner delivers 2D laser scans at up to 70 Hz, and efficient scan registration is required to be able to use the information given by the range sensor for localisation while leaving sufficient computer power to perform obstacle avoidance.



Figure 4.1: Kurt3D, a high-speed mobile robot.

4.1.2 Ariadne

Surmann et al. [64] presented another mobile robot that has been used for indoor 3D mapping. The Ariadne robot (figure 4.2) was also equipped with a SICK laser scanner mounted on a tilting head to produce pitching 3D scans, and a variant of ICP was used to register the partial scans of its surroundings. In addition to the 3D laser scanner, two fixed SICK scanners (one on the front and one at the back) were used for collision avoidance. The platform is quite large and stable, and can carry a payload of up to 200 kg. In the presented work, Ariadne was used to create a metric 3D map of an office corridor and the entrance hall of a large building, and used the maps for autonomous navigation and localisation.

4.2 Outdoor mapping

This section covers some outdoor mapping projects. Outdoor mapping is often more challenging than indoor mapping; partly because the distances travelled are usually larger, and partly because less assumptions can be made about the structure of the area that is to be mapped.

4.2.1 Kurt3D

Recently, Kurt3D (mentioned in section 4.1.1) has been shown doing outdoor 3D mapping and localisation as well. In this outdoor application, the robot has been driven on an outdoor garden path, performing full 3D scans every few meters, and using the 2D HAYAI algorithm for approximate localisation between 3D scan points. Reportedly, the HAYAI algorithm works relatively



Figure 4.2: The Ariadne robot platform.

well for obtaining a pose estimate for ICP scan matching in some conditions when odometry is not available or too poor to give a useful start pose [39].

Subsequent work [49] has improved on this method by using a more robust algorithm for acquiring the initial pose estimate used for registration. This approach is similar to the first stage of NDT in that the space occupied by the two scans is discretised using cubic cells. Octree representations are built for both scans. Starting with a super-cell, boxing in all points in the scan, it is recursively split into eight sub-cells (hence the name octree). Each cell is then split recursively until there is less than a threshold number of points within the cell. The remaining non-empty cells make up a coarse description of the general surface shapes. The octrees can then be matched to each other to get a quick estimate of the pose. ICP is then run starting from the pose given by this step.

This method is likely to fail for two mine tunnel scans with a poor initial pose estimate, because of the loss of small-scale features.

4.2.2 Berkeley mapping project

The video and image processing lab at the University of California has performed projects mapping dense urban environments.

Their experimental setup consists of two SICK laser scanners, a digital camera, and a heading sensor mounted on the back of a truck. The laser scanners are mounted so that the scanning plane of one sensor is horizontal and the other is vertical, and both sensors are pointing towards the right-hand side of the truck. The truck has been driven manually through the streets of Berkeley, modelling the facades of buildings as well as the street structure. Dynamic obstacles, such as pedestrians and other cars, are avoided simply by mounting the



Figure 4.3: The Berkeley urban mapping truck.

sensors on a high rack, so that they are about 3.6 m above the ground. The horizontal scanner is used for scan registration and position estimation, while the data from the vertical scanner is used for the actual modelling. Additionally, a true ground speed sensor, measuring the ground speed using Doppler shift, is mounted on the back of the car. See Früh and Zakhor [24] for more information.

Früh and Zakhor have also used aerial photos for accurate localisation [23]. High-resolution aerial photos are available for many larger cities, and such information can make it easier to solve the SLAM problem. Aerial images also make it possible to perform global 2D positioning without returning to a previously visited place, as is needed with traditional SLAM algorithms.

The scan registration algorithm used by Früh and Zakhor is a 2D line-based point-to-feature method. As long as the relative displacement between scans is small and the scans contain sufficient features, the pose update from scan matching has little error. The truck is also equipped with a speedometer and a heading device. These two sensors are too noisy to be of use for the continuous pose updates, but they can be used to check for errors resulting from scan registration. If there is a significant change in the translation computed from registration, the update is accepted only if both the speedometer indicates a speed change and successive updates change similarly. If there is a significant change in the orientation angle computed from registration, the change is accepted only if both succeeding angles change similarly and the heading sensor indicates an angle change. In all other cases, the new pose is computed by interpolating between valid preceding and succeeding poses.

Each scan registration introduces a small amount of error to the pose, so it is desirable to use scans as far apart as possible. Using fewer scans also decreases the computational cost. On the other hand, scans need a rather large amount



Figure 4.4: The Navlab 2 and 5 — two of CMU's outdoor mapping and navigation platforms.

of overlap to give a reliable match. Früh and Zakhor chose to use an adaptive scan rate, based on the speed of the truck, so that the estimated displacement between successive scans used for registration was between 80 and 100 cm.

Even though the models built are in 3D, the scan registration and localisation are performed in 2D. Scan registration is conducted on sequential horizontal 2D laser scans, and global localisation is performed with the help of a 2D aerial map. This approach is clearly not able to deal with tilt, and in the work covered here, it is assumed that the environment is flat, without any significant slopes. Driving through an environment where the ground is tilted, either laterally or in the anterior-posterior direction from the point of view of the scanners, would result in a skewed model.

4.2.3 GPS-based mapping

Boström et al. used a somewhat similar setup in their large-scale urban mapping project [8], but they did not use any scan registration at all. Instead they relied on accurate differential GPS positioning and interpolation between GPS updates, which arrived at a rate of approximately once per second. This approach is naturally only usable for applications above ground where it is possible to place a network of reference GPS stations near the area that is being mapped.

4.2.4 Unstructured outdoor mapping

Huber and Hebert [32] presented preliminary work on building large, high-resolution, 3D models of unstructured terrain. For this project, range scans from a slag heap were used. These were acquired with a high-resolution laser scanner (Ben Franklin 2) mounted on the two robotic platforms shown in figure 4.4. A second area was covered with a line laser range scanner mounted on an uninhabited helicopter. The algorithms used in this project were based on ICP and face-based spin-images for scan matching. Spin-image matching was used to roughly match the scans, and ICP was used in a second stage to refine the pose estimates from spin-image matching.



Figure 4.5: The ACFR outdoor mapping platform

4.2.5 2D outdoor feature maps

Guivant et al. [29] presented work regarding unstructured outdoor mapping, albeit in 2D. For their outdoor application, they mounted 2D SICK laser scanners on a truck (see figure 4.5). This truck was then driven around a large park. The tree trunks were used as features for localisation. Because they used 2D scan planes from a vehicle driving over rough terrain, it was not possible to recognise individual trunks from single scans. Therefore, a Kalman filter was used to track the centres and diameters of trunks. The trunk centres were then used as features that form a feature map of the park.

4.3 Mine mapping

The main application of this thesis is mine mapping, and this section presents a few other mine mapping experiments.

Mines are semi-structured environments. Mine mapping robots face some challenges not encountered by indoor robots; such as uneven floors and long feature-poor tunnel sections. Even though little can be assumed about the shape and form of walls and floors, and though they are not as regular as most indoor environments, the tunnels themselves form a certain structure.

4.3.1 Metric mine mapping

The CMU robotics institute has performed some experiments in underground mine mapping, in collaboration with the Fraunhofer institute for autonomous intelligent systems [3, 48, 65]. They have used a custom-built robot platform called “Groundhog”, built from two front-wheel pairs from four wheel drive vehicles (see figure 4.6).



Figure 4.6: The Groundhog underground exploration robot.

The Groundhog platform has been equipped with two SICK laser scanners mounted on pan-tilt units, as well as a number of other sensors. It is built for autonomous exploration of subterranean environments in general, but primarily abandoned underground mines.

The first successful autonomous run was made in 2003 in the abandoned Mathies mine. There, the robot managed to follow an underground tunnel that was considered dangerous for persons because of poisonous mud. It followed the corridor for 308 m before it noticed an impassable obstacle, decided to turn back, and returned through the tunnel. Groundhog has subsequently been used in several other mines in the USA as well. Groundhog uses ICP for 3D scan registration, and uses the 3D scans to build a 2D map, which is used for navigation.

Ferguson [18] used another platform for 3D underground mine mapping. In his work, a cart with four fixed SICK scanners (one vertical and one horizontal at each end of the cart) was used to build a 3D map of a tunnel system. Localisation was performed using 2D scan registration from the scans made by the horizontal scanners. A 3D map was built using data from the vertically mounted lasers together with the pose estimates acquired from 2D localisation. In the team's experiments, the vertical scanners were running at a slower rate than the horizontal ones. The 2D pose estimate was updated with every horizontal scan, and full 3D registration (using a “crosshair” combination of one horizontal and one vertical slice) was performed using ICP when vertical range data were available, to update the vertical component of the pose. Ferguson reported difficulties in creating a good map, mainly because of the lack of features in the mine tunnels and the small amount of data used for each registration (only two scan lines).

4.3.2 Topological mine mapping

The research group of Scott Thayer has also done some work regarding topological mapping and navigation in underground tunnel systems [47, 60]. These

efforts were based on the somewhat limiting assumption that the environment could be described as a tree; that is, a graph with no loops.

To identify a junction, they used a horizontal 2D slice of scan data, and created a Delaunay triangulation (see for example Matoušek [43]) of this set of points. Triangles with long and roughly equal sides tend to mark stable features along a tunnel. For an intersection of three tunnels, a Delaunay triangulation will normally contain a large triangle in the middle of the junction. The triangle's corresponding Voronoi edges are aligned with the directions of the three paths. To determine whether a large triangle marks a junction or not, one can see whether following all three Voronoi edges would lead into an unseen area. If any of the edges terminate close to an obstacle within the robot's sensor range, that edge is not worth exploring, and so the node is not considered an intersection. In short, the Voronoi node at the centre of the triangle is stored as the position of the feature (whether it is likely to be an intersection or not), and the number of Voronoi edges terminating in open space is stored as the degree (that is, the number of connected edges) of the node. In this manner, a graph containing undirected edges and nodes of degree three or less can be constructed, capturing the topological structure of the tunnel or corridor system.

This way of finding and storing information about junctions is more problematic when it comes to intersections of more than three tunnels. In this case, the Voronoi diagram will have a number of potential feature nodes within a small area at the centre of the junction. Silver et al. [60] dealt with this situation by considering the corresponding Delaunay triangles together, given that they share a side.

Chapter 5

The normal distributions transform

This chapter details the normal distributions transform. First, the basic algorithm as described by Biber and Straßer [6] is covered, and then a 3D version of the algorithm is presented, as well as a number of extensions.

5.1 NDT

As mentioned in section 3.2.5, the key concept of the normal distributions transform is that the target point cloud is represented by a combination of normal distributions, instead of using the points directly. This representation is a piecewise smooth function, with continuous first and second order derivatives. Consequently, it is possible to apply standard and time-tested numerical optimisation methods, such as Newton's method, for registration. Using this representation of the scan data comes with other advantages as well. For example, there is no need for the computationally expensive nearest-neighbour search that is central to ICP. Computing the normal distributions is a quick one-time task that is done during a single pass through the points of the target scan.

The first step of the algorithm is to subdivide the space occupied by the target scan into regularly sized cells (squares in the 2D case, or cubes in 3D). Then, for each cell b that contains more than some minimum number of points, the centroid \vec{q} of the points in the cell and the covariance matrix \mathbf{C} (also known as the dispersion matrix) are calculated as

$$\vec{q} = \frac{1}{n} \sum_{k=1}^n \vec{x}_k, \quad (5.1)$$

$$\mathbf{C} = \frac{1}{n-1} \sum_{k=1}^n (\vec{x}_k - \vec{q})(\vec{x}_k - \vec{q})^T, \quad (5.2)$$

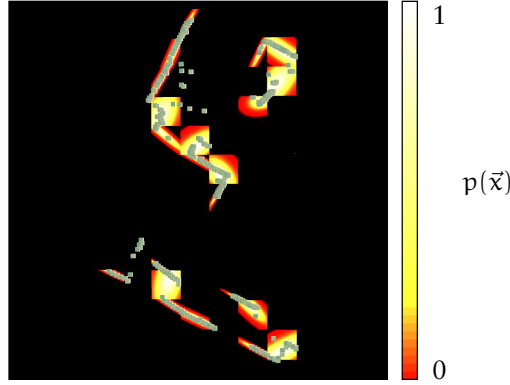


Figure 5.1: A 2D laser scan of part of a room (shown as points) and the PDFs describing the surface shape. Each cell is a square with 1 m side length. Brighter areas represent a higher probability.

where $\vec{x}_{k=1,\dots,n}$ are the points contained in the cell.

The probability that there is a point at position \vec{x} in cell b can then be modelled by the normal distribution $N(\vec{q}, \mathbf{C})$. The probability density function (PDF) is formulated as

$$p(\vec{x}) = \frac{1}{2} \exp \left(-\frac{(\vec{x} - \vec{q})^T \mathbf{C}^{-1} (\vec{x} - \vec{q})}{2} \right), \quad (5.3)$$

where \vec{q} and \mathbf{C} are the average and covariance for the cell that contains point \vec{x} . A 2D laser scan and its corresponding normal distributions are shown in figure 5.1.

The parameters to be optimised, the rotation and translation of the current pose estimate, can be encoded in a vector \vec{p} . For 2D registration, there are three transformation parameters to optimise. Let $\vec{p} = [t_x, t_y, \phi]$ be the parameter vector, where t_x and t_y are the translation parameters and ϕ is the rotation angle. Using counter-clockwise rotation, the 2D transformation function is then

$$T_3(\vec{p}, \vec{x}) = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \vec{x} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (5.4)$$

The algorithm measures the fitness of a particular pose by evaluating the sum of PDFs at all points of the source scan. The function value is called the score of the pose. Since optimisation problems are generally formulated as minimisation problems, the score function is defined so that good parameters yield a large negative number. Given a set of points $\mathcal{X} = \{\vec{x}_1, \dots, \vec{x}_n\}$, a pose encoded in a

vector \vec{p} , and a transformation function $T(\vec{p}, \vec{x})$ to transform a point in space, the score $s(\vec{p})$ for the current set of parameters is defined as

$$s(\vec{p}) = - \sum_{k=1}^n p(T(\vec{p}, \vec{x}_k)), \quad (5.5)$$

that is, the negated sum of probabilities that the transformed points of the source scan are actually lying on the target surface.

Given the vector of transformation parameters \vec{p} , Newton's algorithm can be used to iteratively solve the equation $\mathbf{H}\Delta\vec{p} = -\vec{g}$, where \mathbf{H} and \vec{g} are the Hessian and gradient of s . The increment $\Delta\vec{p}$ is then added to the current estimate of the parameters in each iteration, so that $\vec{p} \leftarrow \vec{p} + \Delta\vec{p}$.

For brevity, let $\vec{x}' \equiv T(\vec{p}, \vec{x}) - \vec{q}$. In other words, \vec{x}' is the transformed point \vec{x} , relative to the centre of the point distribution of the cell to which it belongs. The entries for the gradient of the score function can be written as

$$g_i = \frac{\delta s}{\delta p_i} = \sum_{k=1}^n \vec{x}'_k{}^T \mathbf{C}^{-1} \frac{\delta \vec{x}'_k}{\delta p_i} \exp\left(\frac{-\vec{x}'_k{}^T \mathbf{C}^{-1} \vec{x}'_k}{2}\right). \quad (5.6)$$

The entries of the Hessian are

$$\begin{aligned} H_{ij} = \frac{\delta^2 s}{\delta p_i \delta p_j} = & \sum_{k=1}^n \exp\left(\frac{-\vec{x}'_k{}^T \mathbf{C}^{-1} \vec{x}'_k}{2}\right) \left(\left(\vec{x}'_k{}^T \mathbf{C}^{-1} \frac{\delta \vec{x}'_k}{\delta p_i} \right) \left(-\vec{x}'_k{}^T \mathbf{C}^{-1} \frac{\delta \vec{x}'_k}{\delta p_j} \right) + \right. \\ & \left. \vec{x}'_k{}^T \mathbf{C}^{-1} \frac{\delta^2 \vec{x}'_k}{\delta p_i \delta p_j} + \frac{\delta \vec{x}'_k}{\delta p_j}{}^T \mathbf{C}^{-1} \frac{\delta \vec{x}'_k}{\delta p_i} \right). \end{aligned} \quad (5.7)$$

The first-order and second-order partial derivatives in equations (5.6) and (5.7) depend on the transformation function. Using the 2D transformation function from equation (5.4), the first-order derivatives $\frac{\delta \vec{x}'}{\delta p_i}$ are given by column i of the Jacobian matrix

$$\mathbf{J}_3 = \begin{bmatrix} 1 & 0 & -x'_1 \sin \phi - x'_2 \cos \phi \\ 0 & 1 & x'_1 \cos \phi - x'_2 \sin \phi \end{bmatrix}, \quad (5.8)$$

and the second-order derivatives are

$$\frac{\delta^2 \vec{x}'}{\delta p_i \delta p_j} = \begin{cases} \begin{pmatrix} -x'_1 \cos \phi + x'_2 \cos \phi \\ -x'_1 \sin \phi - x'_2 \cos \phi \end{pmatrix} & \text{if } i = j = 3 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{otherwise.} \end{cases} \quad (5.9)$$

The NDT algorithm for registering two point sets \mathcal{S} and \mathcal{T} (finding the pose \vec{p} that moves \mathcal{S} into registration with \mathcal{T}) is given in algorithm 5.1.

Algorithm 5.1: $\text{NDT}(\mathcal{S}, \mathcal{T}, \vec{p})$

```

Build cell structure  $\mathcal{B}$ 
for all points  $\vec{t}_i \in \mathcal{T}$ 
  do { Find the cell  $b_k$  that contains  $\vec{t}_i$ 
       Store  $\vec{t}_i$  in  $b_k$ 
for all bins  $b_i \in \mathcal{B}$ 
  do {  $\mathcal{T}' = \{\vec{t}'_1, \dots, \vec{t}'_n\} \leftarrow$  all points in  $b_i$ 
        $\vec{q}_i \leftarrow |\mathcal{T}'|^{-1} \sum_{j=1}^n \vec{t}'_j$ 
        $\mathbf{C}_i =$  covariance of all points in  $\mathcal{T}'$ 
while not converged
  { score  $\leftarrow 0$ 
     $\vec{g} \leftarrow 0$ 
     $\mathbf{H} \leftarrow 0$ 
    for all points  $\vec{s}_i \in \mathcal{S}$ 
      do { Find the cell  $b_k$  that contains  $T(\vec{p}, \vec{s}_i)$ 
            $\vec{s}'_i \leftarrow T(\vec{p}, \vec{s}_i) - \vec{q}_k$ 
           do { score  $\leftarrow$  score  $- p(\vec{s}'_i)$  (see equation (5.3))
                Update  $\vec{g}$  (see equation (5.6))
                Update  $\mathbf{H}$  (see equation (5.7))
           Solve  $\mathbf{H}\Delta\vec{p} = -\vec{g}$ 
            $\vec{p} \leftarrow \vec{p} + \Delta\vec{p}$ 

```

In very recent work carried out independently by Ripperda and Brenner [56], a semi-3D version of NDT was used to register large high-resolution outdoor scans. Ripperda and Brenner divided each 3D scan into several horizontal slices and used NDT on each pair of slices. Using N number of slices, and denoting the score for slice n $s_n(\vec{p})$, the score function used in their paper was the sum over all slice pairs

$$s(\vec{p}) = \sum_{n=1}^N s_n(\vec{p}). \quad (5.10)$$

This approach can only perform registration in the horizontal plane, and only works under the assumption that the scanner was upright at each scan pose — an assumption that does not hold for the majority of mobile robot applications.

5.2 3D-NDT

The main difference between 2D and 3D registration with NDT lies in the spatial transformation function and its partial derivatives. In two dimensions, rotation is represented with a single value for the angle of rotation around the

origin. General rotation in 3D is more complex. A robust 3D rotation representation requires both an axis and an angle. This leads to a seven-dimensional optimisation problem (three parameters for the translation, three for the rotation axis, and one for the rotation angle), with quite expensive expressions for the partial derivatives. Using a right-handed coordinate system and counter-clockwise rotations, a general transformation function of a 3D point \vec{x} using a parameter vector \vec{p} can then be formulated as

$$T_7(\vec{p}, \vec{x}) = \begin{bmatrix} \text{tr}_x^2 + c & \text{tr}_x r_y - s r_z & \text{tr}_x r_z + s r_y \\ \text{tr}_x r_y + s r_z & \text{tr}_y^2 + c & \text{tr}_y r_z - s r_x \\ \text{tr}_x r_z - s r_y & \text{tr}_y r_z + s r_x & \text{tr}_z^2 + c \end{bmatrix} \vec{x} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad (5.11)$$

where $\vec{p} = [\vec{t} \mid \vec{r} \mid \phi]$, $\vec{t} = [t_x \ t_y \ t_z]$ is the translation, $\vec{r} = [r_x \ r_y \ r_z]$ is the axis of rotation, $s = \sin \phi$, $c = \cos \phi$, $t = 1 - \cos \phi$, and ϕ is the rotation angle.

The score function of the NDT algorithm is shown in equation (5.3). Optimising this function with Newton's method requires the partial derivatives of the transformation function with respect to the transformation parameters \vec{p} , as shown in equations (5.6) and (5.7). The partial derivatives when using T_7 can be found in the Jacobian and Hessian matrices (5.12) and (5.13). The Hessian is presented as a block matrix with 7×7 blocks, where each block is a three-element vector. For a 3D point \vec{x} , the average \vec{q} of the cell in which it lies, and a vector \vec{p} of transformation parameters, let $\vec{x}' \equiv T_7(\vec{p}, \vec{x}) - \vec{q}$. Then, $\frac{\delta \vec{x}'}{\delta p_i}$ is the i -th column of \mathbf{J}_7 and $\frac{\delta^2 \vec{x}'}{\delta p_i \delta p_j} = \mathbf{H}_{ij}$. Figure 5.2 illustrates the 3D normal distributions for a mine tunnel scan.

If only small angles are considered, the rotation representation can be simplified substantially by using Euler angles instead of the axis-angle representation described above. In that case, rotations are represented as the product of three rotation matrices with angles ϕ_x , ϕ_y , and ϕ_z , rotating points around the principal coordinate axes in sequence. Using Euler angles as a representation of general rotation has a number of defects: mainly that Euler angles are not always unique, and that under certain conditions, they can lead to a situation called gimbal lock, where one degree of freedom is lost. See Altmann [1] for an exhaustive reference on rotations.

Using Euler angles, there are only six transformation parameters to optimise. The 6D transformation function is shown in equation (5.15). Further trigonometric simplifications can be made if only small angles are considered. For small ϕ , $\sin \phi \approx \phi$ and $\cos \phi \approx 1 - 0.5\phi^2$. Also, $\phi^2 \approx 0$, and when these facts are exploited, most terms of the derivatives reduce to zero. The simplified transformation function is shown in equation (5.16).

$$\mathbf{J}_7 =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ t(2r_x x_1 + r_y x_2 + r_z x_3) & tr_y x_1 - sx_3 & tr_z x_1 + sx_2 \\ tr_x x_2 + sx_3 & t(r_x x_1 + 2r_y x_2 + r_z x_3) & tr_z x_2 - sx_1 \\ tr_x x_3 - sx_2 & tr_y x_3 + sx_1 & t(r_x x_1 + r_y x_2 + 2r_z x_3) \\ sA - cB & sC - cD & sE - cF \end{bmatrix}^T \quad (5.12)$$

$$\begin{aligned} A &= (r_x^2 - 1)x_1 + r_x r_y x_2 + r_x r_z x_3, & B &= r_z x_2 - r_y x_3, \\ C &= r_x r_y x_1 + (r_y^2 - 1)x_2 + r_y r_z x_3, & D &= -r_z x_1 + r_x x_3, \\ E &= r_x r_z x_1 + r_y r_z x_2 + (r_z^2 - 1)x_3, & F &= r_y x_1 - r_x x_2 \end{aligned}$$

$$\mathbf{H}_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \vec{a} & \vec{b} & \vec{c} & \vec{d} \\ 0 & 0 & 0 & \vec{b} & \vec{e} & \vec{f} & \vec{g} \\ 0 & 0 & 0 & \vec{c} & \vec{f} & \vec{h} & \vec{i} \\ 0 & 0 & 0 & \vec{d} & \vec{g} & \vec{i} & \vec{j} \end{bmatrix}$$

$$\begin{aligned} \vec{a} &= \begin{bmatrix} 2tx_1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} tx_2 \\ tx_1 \\ 0 \end{bmatrix}, \quad \vec{c} = \begin{bmatrix} tx_3 \\ 0 \\ tx_1 \end{bmatrix}, \quad \vec{d} = \begin{bmatrix} s(2r_x x_1 + r_y x_2 + r_z x_3) \\ sr_y x_1 - cx_3 \\ sr_z x_1 + cx_2 \end{bmatrix}, \\ \vec{e} &= \begin{bmatrix} 0 \\ 2tx_2 \\ 0 \end{bmatrix}, \quad \vec{f} = \begin{bmatrix} 0 \\ tx_3 \\ tx_2 \end{bmatrix}, \quad \vec{g} = \begin{bmatrix} sr_x x_2 + cx_3 \\ s(r_x x_1 + 2r_y x_2 + r_z x_3) \\ sr_z x_2 - cx_1 \end{bmatrix}, \\ \vec{h} &= \begin{bmatrix} 0 \\ 0 \\ 2x_3 t \end{bmatrix}, \quad \vec{i} = \begin{bmatrix} sr_x x_3 - cx_2 \\ sr_y x_3 + cx_1 \\ s(r_x x_1 + r_y x_2 + 2r_z x_3) \end{bmatrix}, \quad \vec{j} = \begin{bmatrix} cA + sB \\ cC + sD \\ cE + sF \end{bmatrix} \end{aligned} \quad (5.13)$$

$$T_6(\vec{x}) = \vec{R}_x \vec{R}_y \vec{R}_z \vec{x} + \vec{t} \quad (5.14)$$

$$= \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ s_x s_y c_z + c_x s_z & -s_x s_y s_z + c_x c_z & -s_x c_y \\ -c_x s_y c_z + s_x s_z & c_x s_y s_z + s_x c_z & c_x c_z \end{bmatrix} \vec{x} + \vec{t} \quad (5.15)$$

$$\approx \begin{bmatrix} 1 & -\phi_z & \phi_y \\ \phi_z & 1 & -\phi_x \\ -\phi_y & \phi_x & 1 \end{bmatrix} \vec{x} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad (5.16)$$

where $c_i = \cos \phi_i$ and $s_i = \sin \phi_i$.

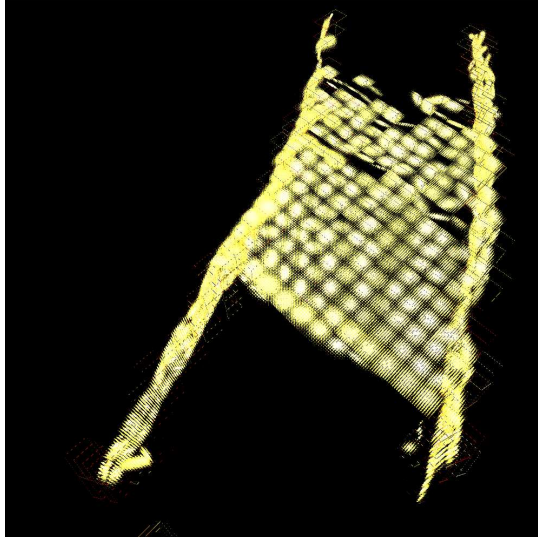


Figure 5.2: The probability functions used by 3D-NDT for one of the tunnel sections shown in figure 6.2, seen from above. Brighter, denser parts represent higher probabilities. The cells here have a side length of 1 m.

The first-order derivatives of equation (5.16) with respect to the transformation parameters in $\vec{p} = [t_x, t_y, t_z, \phi_x, \phi_y, \phi_z]$ can be found in equation (5.17). The i -th column of J_6 is $\frac{\delta \vec{x}'}{\delta p_i}$. The second-order partial derivatives all reduce to zero if the small-angle approximations are applied.

$$J_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & x_3 & -x_2 \\ 0 & 1 & 0 & -x_3 & 0 & x_1 \\ 0 & 0 & 1 & x_2 & -x_1 & 0 \end{bmatrix} \quad (5.17)$$

Using these equations improves the running time of the algorithm, at the cost of lower stability when registering scans with a large initial error.

5.3 Parameters

Several choices need to be made for a practical implementation of 3D-NDT. This section describes different parameters and their influence on the algorithm.

5.3.1 Sampling method

When using 3D-NDT, the target scan is converted to a set of normal distributions. The points in the source scan are then matched to these functions. As discussed earlier, it is normally desirable to subsample the source scan in order to improve running time. Usually, a large number of scan points are redundant

for the purpose of describing the scanned surface shape. In many cases, not least when scanning in corridors and tunnels, as well as in unstructured outdoor environments, the distribution of points is very much denser near the scanner location than farther out. If points are sampled in a uniformly random manner, the sampled subset will have a similar distribution. A sensible alternative method is to make sure that the spatial distribution of points in the subsample is as even as possible. This can be done by grouping the points in equally sized cells, similarly to what is done when the normal distributions are generated for the target scan. Then, a number of points are drawn from each cell. If the distribution of cells is adequate, this strategy will give an even distribution of points.

It is also quite possible to implement subsampling methods that consider the normals as well as positions of points, either making the distribution of normals as spread out as possible, or primarily choosing points with “unusual” normals. In the work covered by this thesis, point clouds that have no additional information have been used, so such sampling methods have not been investigated in detail.

5.3.2 Cell size

Choosing a good cell size is important when using NDT. Any feature that is much smaller than the size of a cell will be blurred out by the PDF that describes the local surface shape around it. Choosing a cell size that is too large therefore often leads to less accurate registration. On the other hand, the region of influence of a cell only extends as far as its boundaries. That is, the cell will only contribute to the score function for scan points within its bounds. The consequence of this is that if the cells are too small, the two scans must be close together before registration. Using smaller cells also requires more memory. The optimal size and distribution of cells are highly dependent on the shape of the input data and on the application.

5.3.3 Discretisation methods

The versions of 3D-NDT described so far beg for a more adaptive cell structure. Using a fixed lattice of square or cubic cells burdens the user with the task of choosing a good cell size. This section presents a number of alternative methods for handling the cells and their PDFs.

Fixed subdivision

The benefit of using a fixed lattice of cells, as described above, is that the overhead for initialising the cell structure is small. Only one set of PDF parameters needs to be computed for each cell, and the position of each cell is straightforward. Even more important for the performance of the algorithm is that point-

to-cell look-up is also a very quick operation that can be done in constant time, regardless of the number of cells.

Octree subdivision

An octree is a tree structure that can be used to store a hierarchical discretisation of 3D space. In an octree, each node represents a bounded partition of the space. Each internal node has eight children that represent congruent and non-overlapping subdivisions of the space partition that corresponds to their parent node.

When creating an octree, the root node is sized to encompass the whole target scan. The tree is then built recursively, so that all nodes containing more than a threshold number of points are split into eight equally sized sub-cells.

The “octree” version of 3D-NDT starts with fixed regular cells, as described before, with the difference that each cell is the root node of an octree. Each cell in which the spread of the distribution is larger than a certain threshold is then recursively split, thus making a forest of octrees.

When traversing the cell structure, looking for the corresponding cell to a point in the source scan, the leaf node that contains the point is chosen and its PDF is used to compute the score function.

Additive subdivision

Using adaptive subdivision (using finer subdivision in places where a single normal distribution can not describe the surface satisfyingly) gives a better representation of the surface shape in areas where large cells would hide many details, while keeping large cells where the surface is largely planar and further subdivision is unnecessary. However, the problem that small cells have a smaller region of influence remains: if corresponding points of the two scans are not within the same cell, the extra fidelity is of no use.

A slight change to the octree subdivision scheme can overcome this limitation. Instead of using only one leaf of the octrees, each point from the source scan has its score function evaluated for all of the distributions in the leaf cells. This effectively increases the support size of the leaf cells to that of their root cell, without sacrificing the extra refinement of the surface description that they give. This is illustrated in figure 5.3.

Iterative subdivision

Another option is simply to perform a number of NDT runs with successively finer cell resolution, so that the start pose for each iteration other than the first one is the end pose of the previous run. The first runs are good for bringing badly aligned scans closer together, and later runs improve the rough initial match.

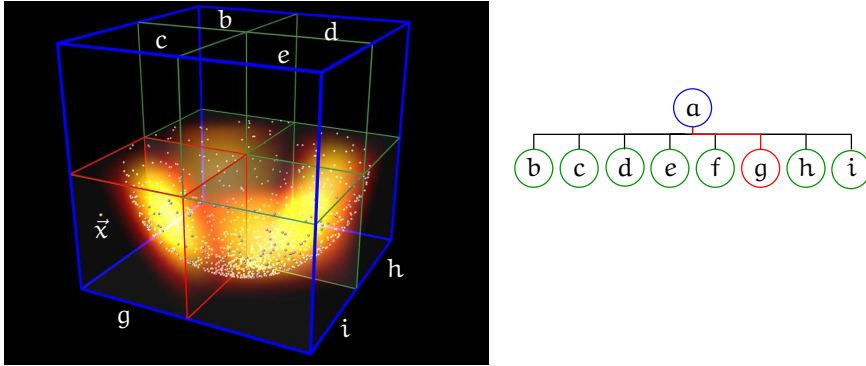


Figure 5.3: Comparing octree and additive subdivision. A subdivided grid cell is shown on the left, and the tree structure is shown on the right. The PDF of cell *a* has a large spread, because the points within the cell are not aligned along a planar region. Therefore it is split, and the PDFs of eight sub-regions *b*–*i* are computed instead. Point \bar{x} is within cell *a*, and, more specifically, within sub-cell *g*. Using octree subdivision, \bar{x} ’s contribution to the score function is computed from *g* alone. Using additive subdivision, the score is a weighted sum computed from nodes *b*–*i*. In this example, nodes *b*–*e* are empty and will not add anything to the score.

Linked cells and infinite outer bounds

Using the discretisation methods described so far, points lying in unoccupied cells are discarded, thus rendering large parts of the input space “dead”. Instead of doing so, the PDF from the closest occupied cell can be used for those points. This increases the region of influence of cells, and is illustrated in figure 5.4.

The same idea can also be applied to points falling outside of the cell lattice altogether. The score for those points can be computed using the closest cell on the edge of the lattice. However, this effectively introduces the problem of matching non-overlapping points to border points, as shown in figure 3.3.

Linked bins can be implemented either by letting each cell store a pointer to the nearest occupied cell, or by storing only occupied cells and putting them in a kd-tree. The latter should be preferable if there are many unoccupied cells.

Multiple adaptive distributions

A yet more adaptive discretisation method is to use a clustering algorithm that divides the points of the scan into a number of clusters, based on their positions, and then use one NDT cell for each cluster.

A common clustering algorithm that is easy to implement is k-means clustering. A set of *k* clusters is initialised, and the points of the scan are assigned at random to the clusters. The clustering algorithm then proceeds iteratively. In each step, the centre point of each cluster is computed from the centroid of the points it currently contains. Each point is then moved to the cluster that

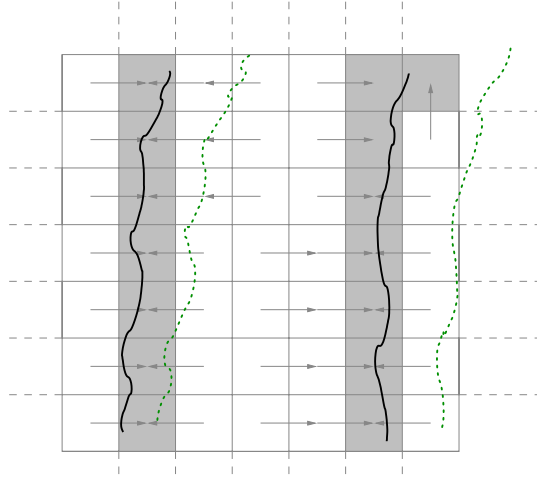


Figure 5.4: Matching two 2D scans of a tunnel section. The dotted scan is being registered to the solid scan. Occupied cells are shaded. If linked bins are not used, the parts of the scan that are in unshaded cells will be skipped. Otherwise the linked cell (shown with arrows) will be used. If using infinite outer bounds, the outer cells extend as shown with dashed lines.

has the closest centre. These two steps are iterated until no more points have changed clusters between two iterations, or until the number of changes falls below some threshold value.

In order to get a good distribution of cells, without pathologically large or small cells, the initial distribution should be even across the volume occupied by the scan. The clustering pass then tries to move the cells to where they are needed.

Chapter 6

Experiments

This chapter presents the results of quantitative and qualitative comparisons between ICP and 3D-NDT, described in chapters 3 and 5. The purpose of these experiments is to evaluate the accuracy and robustness of the different algorithms under real operating conditions in a mine environment.

There are many parameters that can be changed, both for ICP and 3D-NDT. To make a quantitative comparison between the registration algorithms, a number of test sequences were run on different data sets, each with different parameter settings. A batch of runs were performed for each set of parameter values. To avoid a combinatorial explosion in the number of possibilities, a “baseline” combination of variants has been chosen, that incorporates the following features:

- Common parameters
 - spatially distributed sampling of points on the source scan,
 - no subsampling of the target scans (all points are used),
 - 10% sample rate,
 - initial translation error of 1 m,
 - initial rotation error of 0.1 radians,
 - 100 tests for each set of parameters.
- ICP parameters
 - constant weighting of point pairs,
 - fixed distance threshold of 1 m,
 - convergence threshold when the change in translation is below 0.0001 m or the change in rotation angle is below 0.0001 radians.
- NDT parameters
 - fixed cells with 1 m side length,

- Newton’s method with line search for optimisation, with a maximum step length of 0.05 ($|\Delta \vec{p}| = 0.05$) so that the max change in the pose vector is 5 cm or 0.05 radians at each iteration.

The magnitude of the initial pose errors was kept the same over each trial sequence, but the directions were different for each run. In other words, the translation displacement for each test run was a point on a sphere with a fixed radius. The added rotation error had its axis pointing in a random direction for each run and the angle (that is, the amount of rotation) fixed for each batch of runs.

The results of the experiments are presented with box plots, with a line connecting the median values of each set of runs. The box extends to the upper and lower quartile of the data, and the “whiskers” extend to the maximum and minimum values of the sequence. The limits for what is considered a “good match” are shown with dashed horizontal lines. These are not hard limits, but were chosen according to what was considered acceptable for the application and the accuracy with which the ground truth pose could be estimated.

In the following presentation, translations are written as three-dimensional vectors $\vec{t} = [t_x, t_y, t_z]$, where the quantities are measured in metres. Rotations are shown with a unit vector, representing the axis of rotation, and an angle, measured in radians, so that $R = ([r_x, r_y, r_z], \theta)$.

6.1 Data sets

Three different mine data sets were used in the comparison and evaluation of the registration algorithms. They were all collected in the Kvarntorp mine, south of Örebro in Sweden. This mine is no longer in production, but was once used to mine sandstone. The mine consists of more than 40 km of tunnels, all in one plane. Parts of the mine are currently used as archives and storage facilities, while others are used as a test bed for mining equipment.

Because of the natural layers of sandstone, the tunnels have a rather characteristic shape, with flat ceiling and relatively straight walls. Even though the floor and ceiling are flat compared to many other mines and natural environments, the unevenness of the floor makes a wheeled vehicle tilt considerably while driving over it. The roughness is comparable to that of a gravel road, and if scans were being registered with only three degrees of freedom (disregarding tilt and changes in floor height), there would be large discrepancies between some scans.

The JUNCTION data set (figure 6.1) consists of two scans from the end section of a tunnel. At the far end of the tunnel, there is a flat cast concrete structure; and on one of the side walls there is a passage to a neighbouring tunnel. Both the end face and this passage are salient and large-scale features. These two scans were taken from the same pose, and only differ in resolution. In other words, the ground truth pose for the source scan with respect to the tar-

get scan is $\vec{t} = \vec{0}$, $R = ([0, 0, 1], 0)$. The source scan contains 139 642 points and the target scan contains 72 417 points.

The TUNNEL data set (figure 6.2) was collected further down the same tunnel. Four scans were taken approximately four metres apart. The scans contain around 27 500 points each. The scans in this set have much less obvious features. The only large-scale features; that is, the walls and ceiling; are not enough to give accurate registration, as the scans can “slide” along the direction of the tunnel, and still have a large amount of overlap and close proximity of all surfaces — the usual criteria for a good match. The small-scale features, such as bumps on the walls and light fixtures in the ceiling, need to be matched in order to properly register these scans. The results reported here are for the first two scans in this sequence. For these two scans, the ground truth was determined visually, by running a number of registration attempts and picking one that looked like the closest match. This was determined to be

$$\vec{t} = [3.33, 1.65, -0.06], R = ([-0.289, -0.047, -0.956], 0.034).$$

When collecting this data set, we tried to measure the relative displacement between the scans using a total station (that is, a tripod mounted laser measurement device). The resulting measurements were not accurate enough to simply paste the scans together, but they were good enough as an initial estimate for the registration algorithms. Equally good starting points could have been found just using a tape measure, for example.

Both the JUNCTION and TUNNEL data sets were collected with an early prototype of the Optab scanner, described in section 2.2.1, but the configuration of the scanner was changed between the two. For the TUNNEL data set, the scanner was oriented so that the first scan plane was horizontal. The scanner was then tilted upwards. This is a so-called pitching scan. Because of this configuration, the floor is not visible in the TUNNEL data. For the JUNCTION data set, the scanner was mounted so that each scan plane was vertical, and the scanner was rotated around the vertical axis. That is called a yawing scan.

A larger data set, KVARNTORP-LOOP, was collected at a later date, using a SICK LMS 200 mounted on our mobile robot platform called Tjorven, shown in figure 6.4 (see Lindgren [37] for the origin of its name). For the KVARNTORP-LOOP data set, the robot was driven along two tunnels, forming a loop, with 3D scans being taken about four to five metres apart. The first 65 scans from this set are shown in figure 6.3. The scans contain between 60 000 and 90 000 points each (the resolution was increased after a few scans). The scanner on Tjorven is configured for pitching scans, as can be seen in figure 6.4. Tjorven can also tilt the scanner downwards, so that both the floor and ceiling can be scanned.

The scans of the KVARNTORP-LOOP data set are more accurate than those of the TUNNEL and JUNCTION sets. The latter show some disturbances due to the somewhat unstable experimental setup of the scanner.

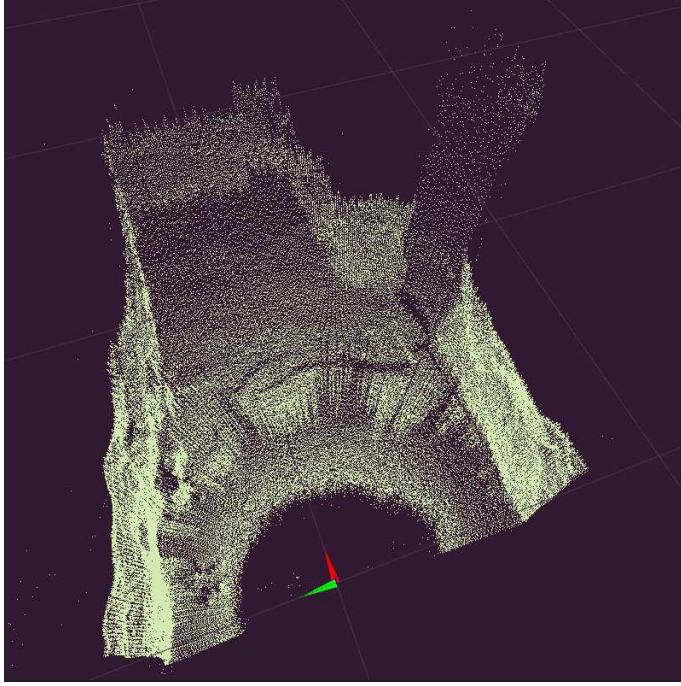


Figure 6.1: The source scan from the JUNCTION data set, seen at an angle from above. The green and red arrows show the local coordinate system, and the scanner was placed where the arrows meet. The background grid lines are 10 m apart.

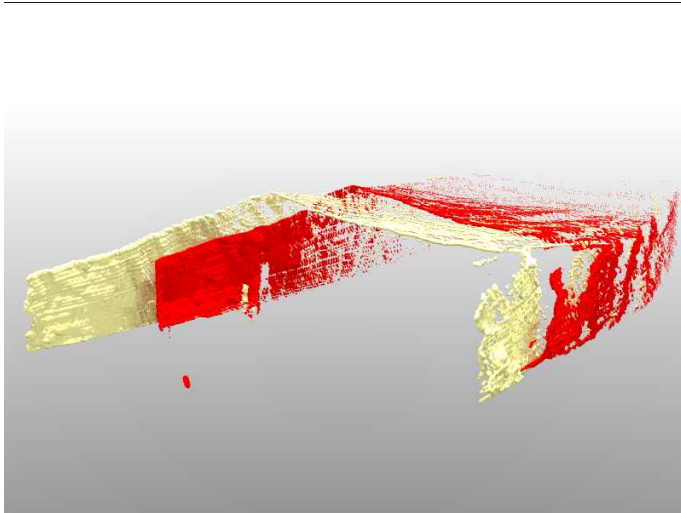


Figure 6.2: The two scans of the TUNNEL data set. The free-floating points in the middle of the tunnel are noise.

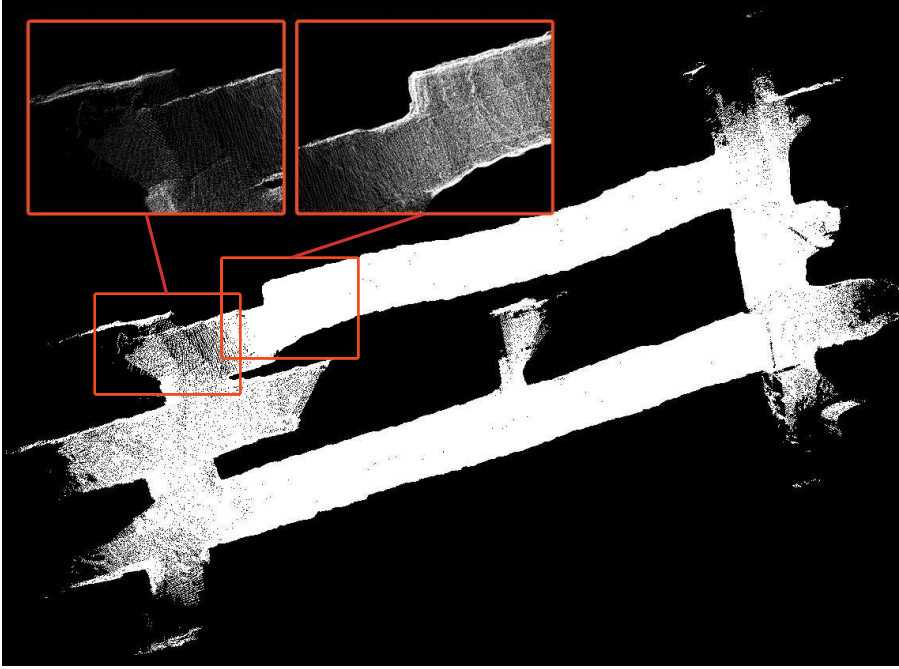


Figure 6.3: The first 65 scans from the KVARNTORP-LOOP data set, seen from above. The model measures approximately 55 m by 155 m, and is around 6 m high. The top left corner shows the accumulated error after coming back to a previously visited location after completing the loop. The error there is about 2.7 m. To the right of this section is a clear “offset” in the tunnel. This is not a registration error, but shows the actual shape of the tunnel. That shape is probably due to a mistake on part of the excavation crew when they were trying to “close the loop”.

6.2 Results

For all of these experiments, rotations were represented using axes and angles instead of Euler angles, and the transformation function from equation (5.11) was used for 3D-NDT. The initial error in the pose estimate is denoted \bar{e}_t (metres) for the translation and e_r (radians) for the rotation. The different parameters that were tested are also described in section 5.3.

Moderate effort was made to optimise the efficiency of the programs. The algorithms are implemented in C++. The ICP implementation uses the quite efficient approximate nearest neighbour library ANN. The numerical optimisation code used in 3D-NDT makes use of the C linear algebra library newmat. This library claims to be most efficient for large matrices, but the matrices involved in the computations for 3D-NDT are no larger than 7×7 . It is likely that the numerical optimisation can be performed faster. The experiments were

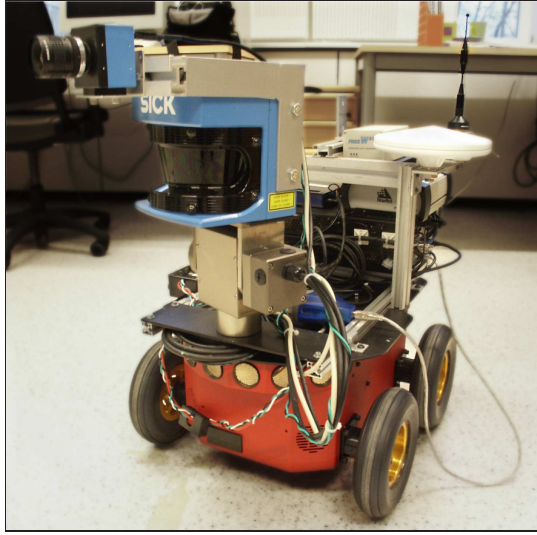


Figure 6.4: Tjorven, our mobile robot platform. In addition to the laser scanner used for 3D mapping, it is also equipped with a digital camera, an array of sonars, an omnidirectional camera (not shown here), and a differential GPS system. In this picture, it has its indoor wheels on.

run on a computer with an AMD Athlon processor running at 1950 MHz and 512 MB of memory.

6.2.1 Sampling

Sample ratio

To test the sensitivity to the amount of samples being used for registration, a number of test sequences were run with increasing numbers of samples. From 0.1% up to 50% of the points in the source scans were sampled and used for matching with all of the points in the target scans. Figures 6.5 and 6.6 show the results of tests where all other parameters were set according to the baseline setup.

The conclusion is that 3D-NDT is more sensitive when using very low sample ratios (less than a few percent). Even though it succeeds at registering the two scans from most of the start poses tried, it fails for some when using a very low sample ratio. However, around 15 % of the total number of points is enough to give reliable results for the JUNCTION data set when the initial error is moderate. The median error is lower for 3D-NDT in all cases with larger sampling ratios, but there are some outlier cases where the error is much larger. The execution time is around one third of that of ICP for this data set and these settings. ICP gives acceptable results down to around 8 % on the same data and

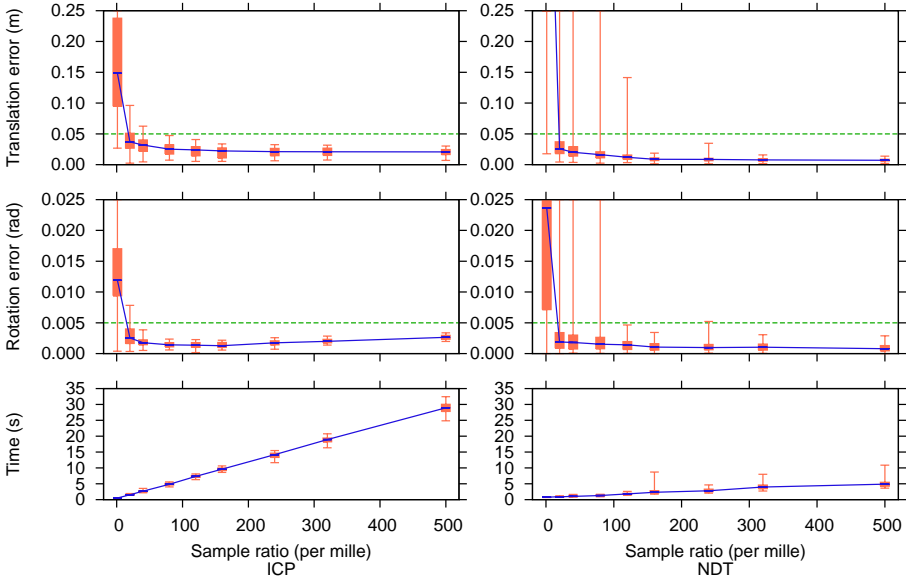


Figure 6.5: Performance of ICP and 3D-NDT using different sample ratios on the JUNCTION data, with $|\vec{e}_t|=1$ m and $|e_r|=0.1$ rad.

initial error. As can be seen from figure 6.6, the TUNNEL data set is much more challenging than the JUNCTION set, both for ICP and 3D-NDT. The median error is still smaller for 3D-NDT than for ICP, but with an initial translation error of 1 m and 0.1 radians rotation error, the algorithms fail to register the scans from a rather large number of the initial pose estimates. Figure 6.5 shows that the rotation error actually increases for ICP as the sample ratio goes above 20 % for the JUNCTION data set. The reason for this is most likely that more of the scan noise is used; in other words, overfitting. A similar effect can be seen for 3D-NDT on the TUNNEL data set in figure 6.6. Because the two scans in this data set are only partially overlapping, ICP tends to move the source scan a bit too much towards the centre of the target scan to maximise the amount of overlap. The pose that 3D-NDT converges to when using a high sample ratio is similar to the one that ICP converges to.

If both scans are subsampled using the same ratio, and not just the source scan, the required sample ratio is much higher.

Sampling method

Spatially distributed sampling is generally more robust than uniformly random sampling. The results of using a uniform probability distribution when selecting the subset for matching is shown in figure 6.7. As discussed earlier, using uniformly random sampling will preserve the general distribution of points in

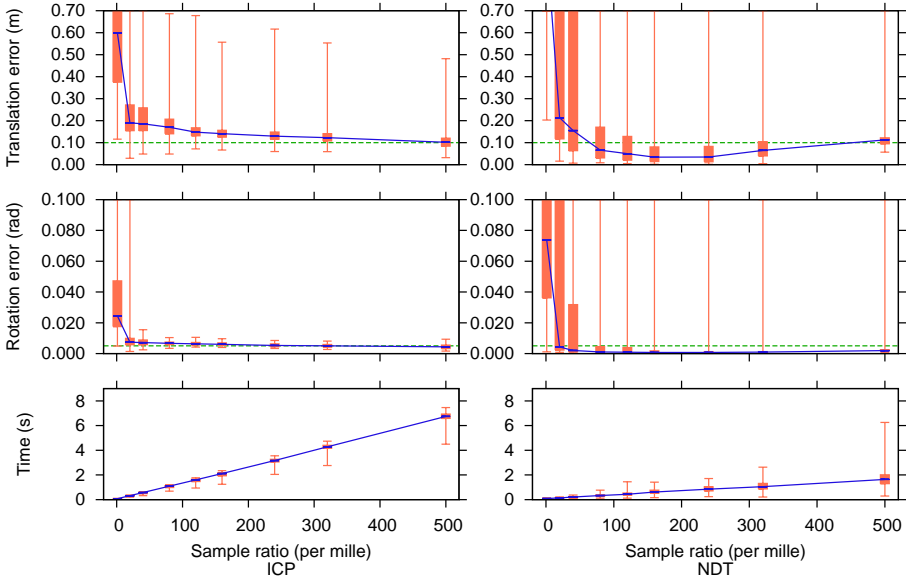


Figure 6.6: Performance of ICP and 3D-NDT using different sample ratios on the TUNNEL data, with $|\vec{e}_t|=1$ m and $|\vec{e}_r|=0.1$ rad.

the scan, and that is not optimal for tunnel scans, where the concentration of points is much higher near the sensor location than further away.

Comparing figures 6.6 and 6.7, it is clear that the registration errors are larger when using uniformly random sampling, at least for 3D-NDT. Using spatially distributed sampling with ICP, the range between the upper and lower quartiles was quite small in all cases where more than 8% of the points were used. This range was much larger when using uniformly random sampling. For 3D-NDT, the interquartile range was rather large for both sampling methods, but the median translation and rotation errors were significantly lower when using spatially distributed sampling.

All other experiments discussed in this chapter use spatially distributed sampling.

6.2.2 Cell size

To show the effect of different cell sizes for 3D-NDT, registration with sizes ranging from 0.5 m up to 3 m are shown in figure 6.8. Each box plot shows the results of 50 test runs.

The running times are shorter when the cells are larger (and fewer). The translation error is at its smallest within a certain cell size range, and increases with both smaller and larger cells. For smaller cells, many runs fail to register the two scans completely, because of small regions of influence. This can be seen

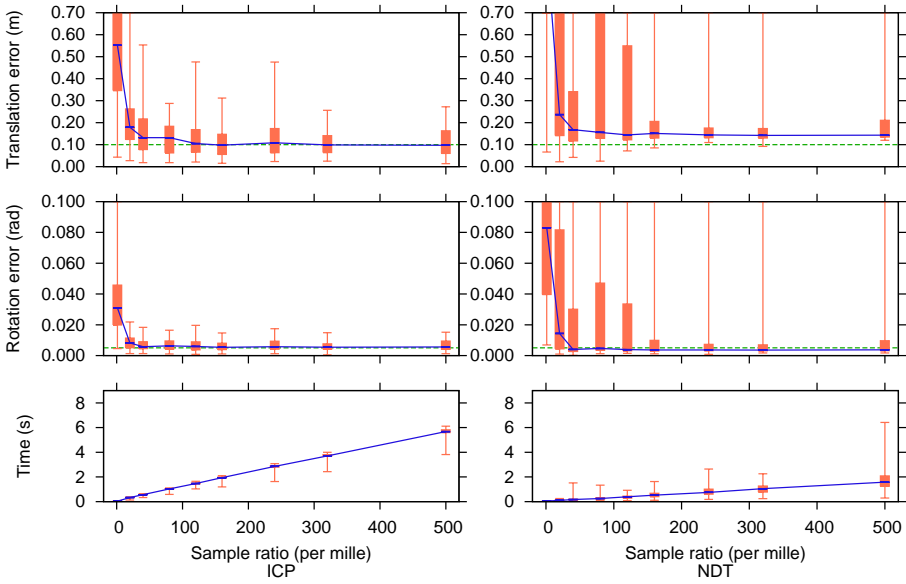


Figure 6.7: Using uniform random distribution instead of spatially distributed selection. The data set is TUNNEL, and the other settings are the same as in figure 6.6.

in figure 6.8, where the upper quartile of the tests with JUNCTION and 0.75 m cells is comfortably below the acceptable threshold, but the error of the worst few runs is much larger. This result has to do with the fact that, depending on the direction of the initial pose error, for some test runs the small cells will not be able to “attract” enough points. For larger cells, the accuracy decreases because of loss of surface shape information. Based on these results, a cell size between 1 m and 2 m is most suitable for the given environment.

Because static arrays were used for cell storage, memory usage increased drastically for the tests with smaller cells. This also leads to slower performance because of memory swapping and page faults, particularly for the JUNCTION data set. The times reported here only measure CPU time. The actual time was much larger for the smaller cell sizes (at 0.5 m cell size, the target scan in the JUNCTION data set needs around 1 800 000 cells).

6.2.3 Initial error

The sensitivity of the algorithms with respect to the amount of error in the initial pose estimate was also tested, both for the translational and rotational components. The results are shown in figures 6.9–6.13. For the translation error tests, the initial rotation error was set to zero, and the translation error was zero when testing the sensitivity to the initial rotation error.

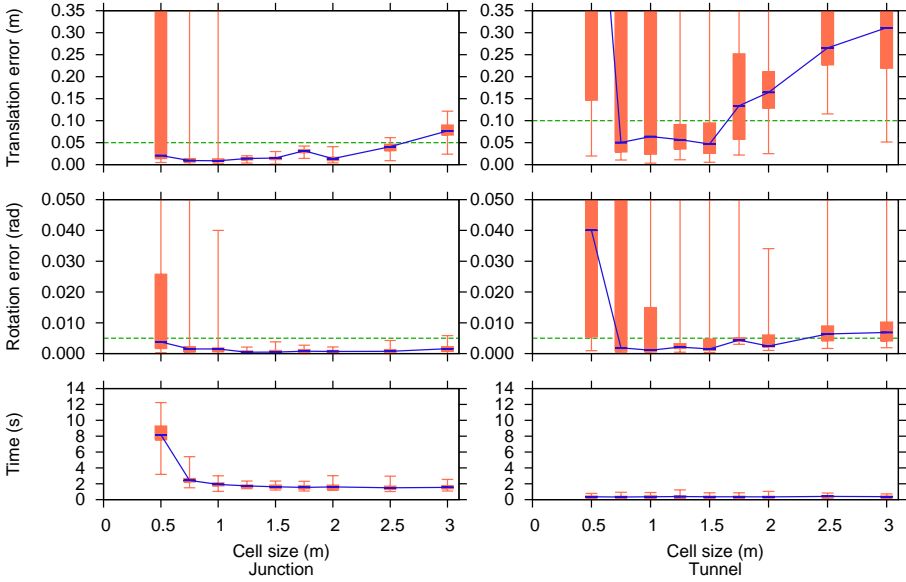


Figure 6.8: Comparing the effect of different cell sizes, using fixed cells. Each test sequence is 50 runs.

Again, 3D-NDT shows a smaller median error in most cases, although failed registrations start to occur at smaller values for the initial error than is the case for ICP. With the baseline settings for ICP, an initial translation error of up to 2.5 m (when the error in rotation is zero) or a rotation error of up to 0.35 radians (when the translation error is zero) can be handled reliably for the JUNCTION data set. Using 3D-NDT, failed registrations start to occur at 2 m translation error or 0.3 radians rotation error.

As shown in figure 6.12, when a large cell size is used, 3D-NDT's tolerance for a large initial translation error in the TUNNEL data set is slightly better than when using 1 m cells, at the expense of less accurate matches for runs with smaller error. However, the end results are still not close enough to be considered good matches.

6.2.4 Discretisation methods

Test results for different discretisation methods on the JUNCTION and TUNNEL data sets are shown in figure 6.14. For these tests, cell sizes varying between 2 and 1 metres were used. The results for fixed 2 m cells are shown for comparison. With the initial error set according to the baseline setup, all methods perform equally well on the JUNCTION set. To show the differences in the methods' efficiency, the initial pose error was increased a little for the tests on the

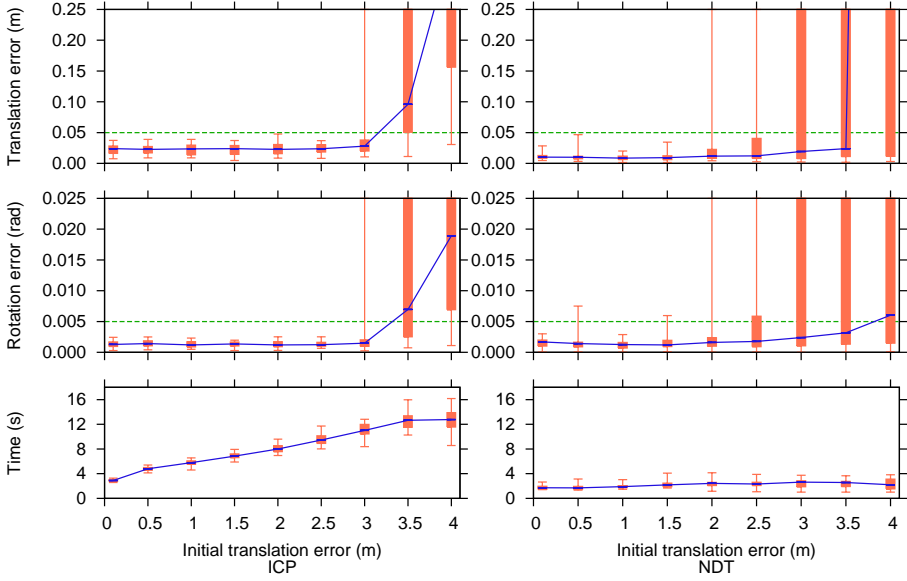


Figure 6.9: Comparing the sensitivity to the initial error in the translation estimate for the JUNCTION set. Here, 50 runs were performed with $|e_r|=0$.

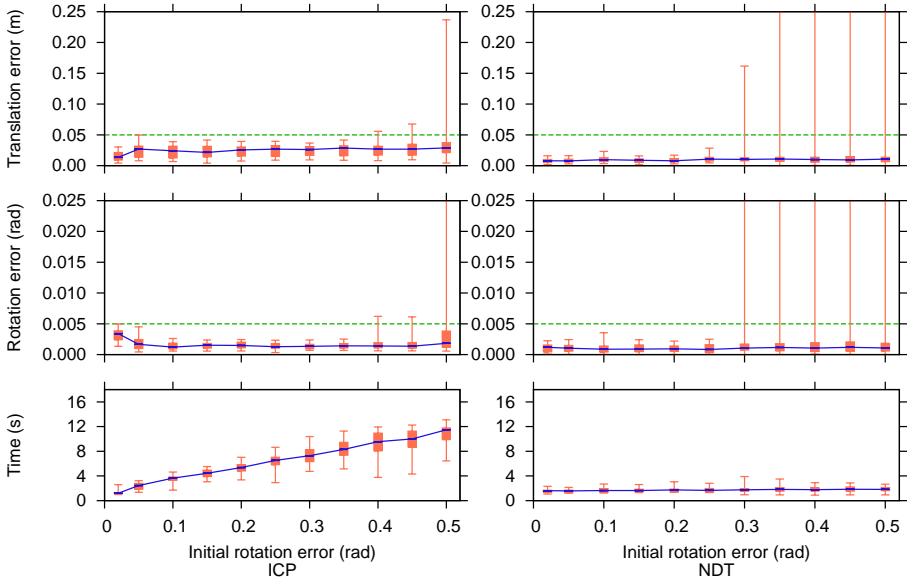


Figure 6.10: Comparing the sensitivity to the initial error in the rotation estimate for the JUNCTION set. 50 runs were performed with $|\vec{e}_t|=0$.

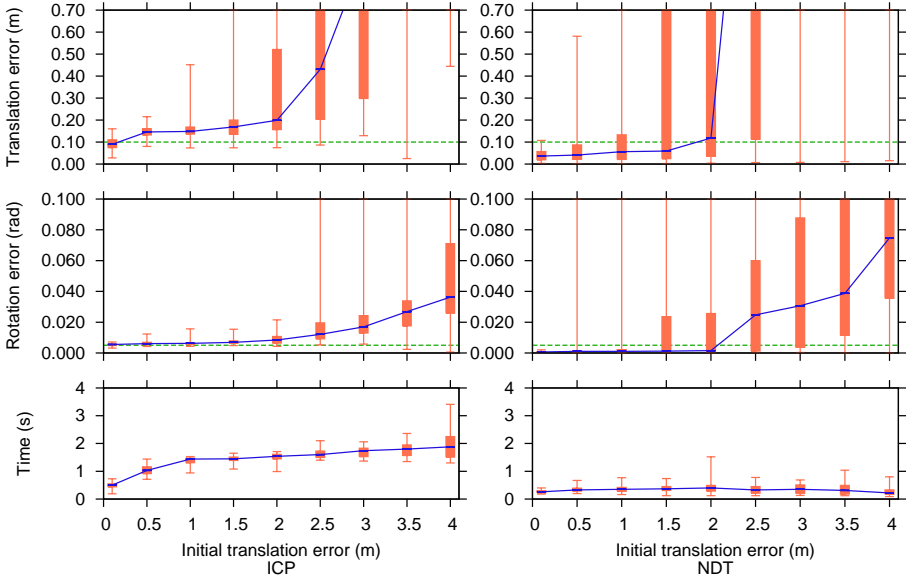


Figure 6.11: Varying initial translation error for the TUNNEL set. Here, 100 runs were performed with $|e_r|=0$.

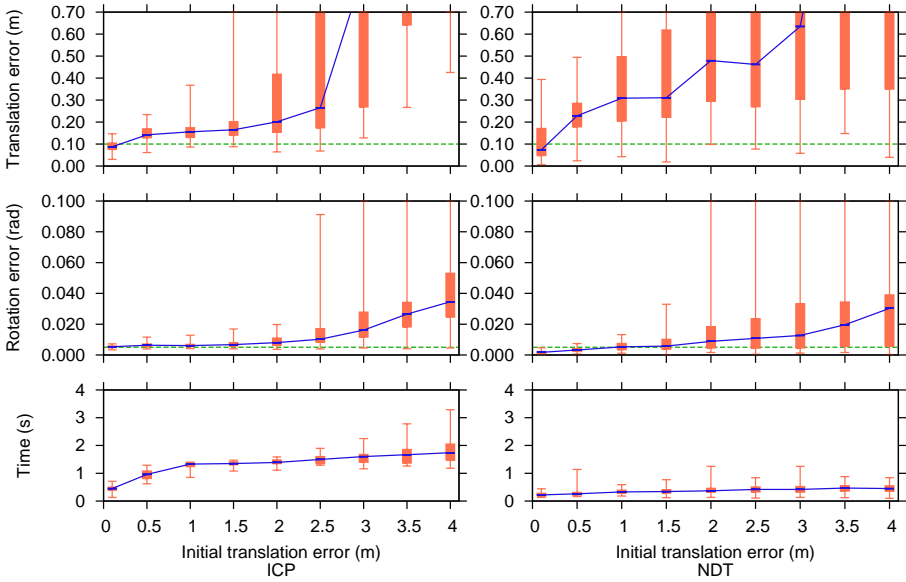


Figure 6.12: Varying initial translation error for the TUNNEL set. These graphs show the results of the same settings as in figure 6.11, but using a 3 m distance threshold instead of 1 m. 100 runs were performed.

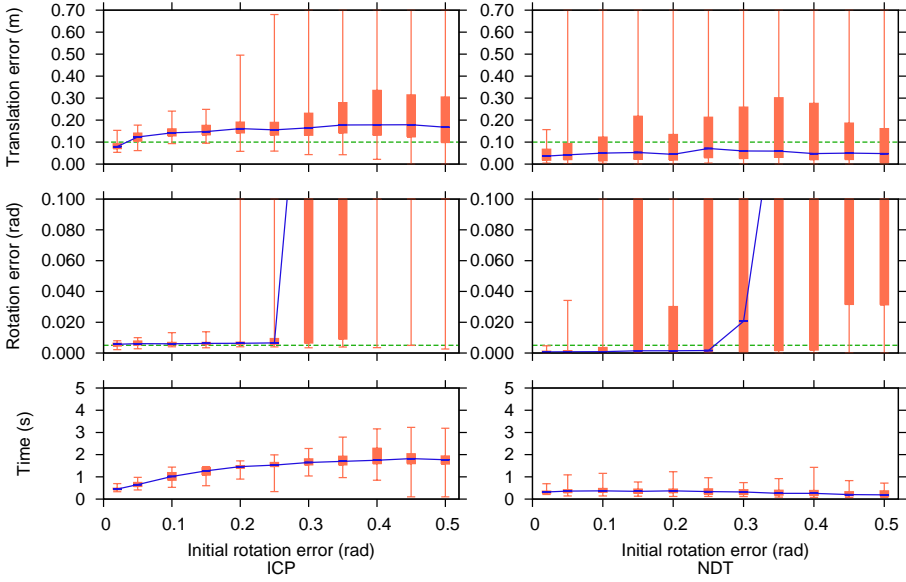


Figure 6.13: Varying initial rotation error for the TUNNEL set. 100 runs were performed with $|\vec{e}_t|=0$ and 1 m distance threshold.

JUNCTION set, so that $|\vec{e}_r| = 0.2$. The different discretisation methods are described in section 5.3.3.

Octree subdivision

Octree subdivision was implemented to make the normal distributions follow the surface shape better. However, this did not lead to a noticeable improvement for the JUNCTION data set. A probable reason for this is that the added detail was not needed for this data set, as it has clear and large features. Octree subdivision did improve the result for the TUNNEL data set, approximately halving the median error compared to using fixed cells.

Additive octrees

Additive octrees, when the score for each point is computed based on a weighted sum of all leaves in the octree where it belongs instead of using a single leaf, improved the result of the TUNNEL set slightly, at the cost of a minor increase in execution time, because more cells were investigated for each point. However, 3D-NDT with additive octree subdivision failed for two of the initial pose estimates when running on the JUNCTION data set. The results for the other 98 poses were satisfactory.

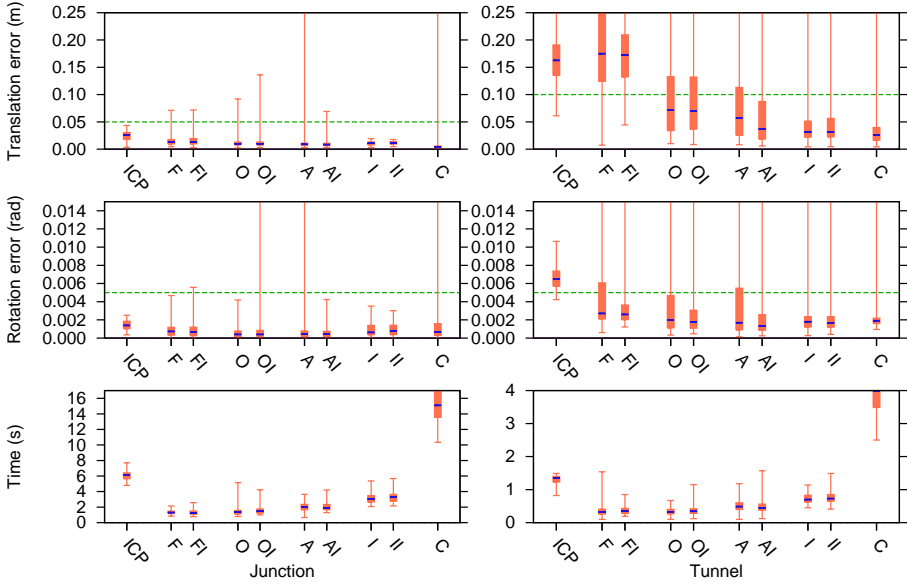


Figure 6.14: Comparing different discretisation methods for 3D-NDT on the JUNCTION and TUNNEL data sets. For the JUNCTION tests, $|\vec{e}_t|=1$ m and $|\vec{e}_r|=0.2$ rad. For TUNNEL, $|\vec{e}_t|=1$ m and $|\vec{e}_r|=0.1$ rad. Baseline ICP is on the left. The next two plots (F and FI) show 3D-NDT with fixed cells, O and OI show octree subdivision, A and AI show additive subdivision, I and II show iterative subdivision, and C shows 3D-NDT with k-means clustering (500 clusters). The rightmost plot in each pair (OI) uses infinite outer bounds (not applicable for ICP and clustered 3D-NDT).

Iterative subdivision

The more “brute-force” method of iterative runs with varying cell size improved the result for both data sets, at the price of longer execution times. Iterative subdivision and additive subdivision with infinite outer bounds were the only methods that succeeded in accurately registering the TUNNEL data set from at least 75% of the initial poses.

For the tests shown here, the first iteration used 2 m cells. For each subsequent iteration, the cell size was multiplied by 0.75, and the registration was stopped when the size was smaller than 1 m. In other words, the cell sizes used were 2 m, 1.5 m, and 1.125 m, respectively. Consequently, the total running time is three times longer than when using fixed cells.

Linked cells and infinite outer bounds

Linked cells and infinite outer bounds did not give any better results on the JUNCTION set, but led to a slight improvement on the more challenging TUNNEL set.

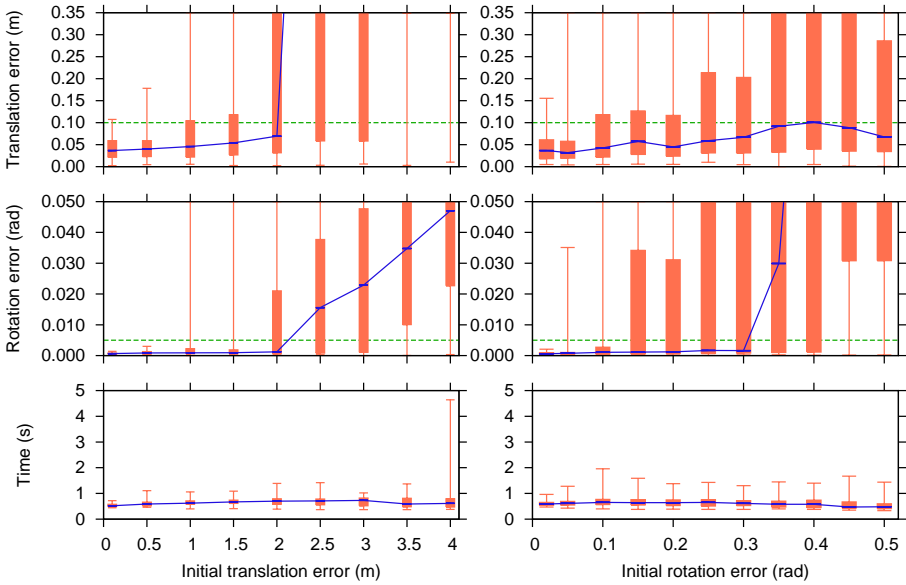


Figure 6.15: Stability with respect to initial error, using 3D-NDT with linked bins and infinite bounds on the TUNNEL set. The initial rotation error is 0 for the plots on the left side, and the initial translation error is 0 on the right side.

Results using linked cells are presented in figure 6.15. The plots in this figure should be compared to the plots in figures 6.11 and 6.13. Registration with linked cells improved the results for pose estimates where the initial error was moderate. The upper quartile of the error for registration with 3D-NDT is lower than the median for the corresponding ICP registrations when the initial error is less than 2 m and 0.2 rad, respectively.

Using infinite bounds on the outermost cells generally led to a slight improvement for the TUNNEL data set, especially for the rotation component of the pose. Interestingly, it did not lead to an improvement for the JUNCTION data set. The likely reason for this is that the error in the initial pose estimate was not large enough for the outer cells to have any significant effect.

Multiple adaptive distributions

A k-means clustering approach was also implemented, which attempts to distribute cells so that they fit the data as well as possible. This method is different from the other discretisation methods in a few ways. The user has no direct control over the size of cells, but only chooses the amount k of cells to use. Because there is no regularity in the cell structure, as there is for the other discretisation methods described, storing the cells in a static array is not appropriate. Instead, the cells were stored in a kd-tree. (The k in k-means clustering and kd-tree have no correspondence. They are named like this only because that is the standard

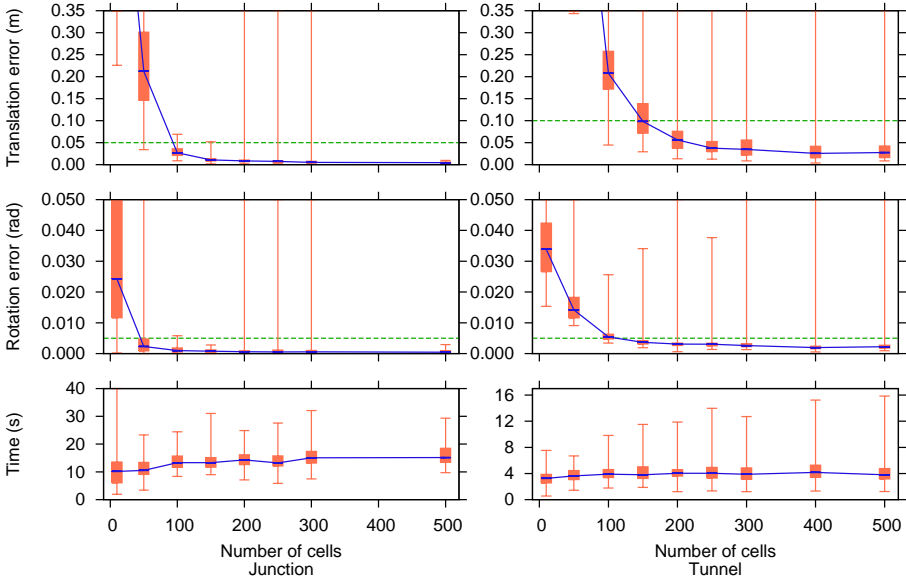


Figure 6.16: Different number of cells for 3D-NDT with k-means clustering.

notation.) Using a tree search structure has the same effect as enabling linked bins and infinite outer bounds, since the closest occupied cell is always found when searching the kd-tree. The clustering algorithm was set to stop after the first iteration when the number of points re-assigned was less than 1% of the total number of points in the scan.

This method takes much more time than standard 3D-NDT. The pre-processing takes longer time because of the clustering step, and searching for the closest cell in a kd-tree is not as fast as a simple array lookup. Results for tests with different numbers of cells are shown in figure 6.16. The execution time does not increase significantly when increasing the number of cells, but the accuracy does, so there is no reason to use a small number of cells.

As can be seen from figure 6.14, the increased time needed for this method did not pay off; there was no higher accuracy than for iterative subdivision.

Summary

A comparison of all methods described in this section can be seen in figure 6.14. It is clear that the TUNNEL data set is simply too difficult to register reliably if the error of the initial pose estimate is too large and if the error is along a difficult direction. However, most of the 3D-NDT versions outperform ICP in this case. In the cases where both ICP and 3D-NDT converge, 3D-NDT tends to give more accurate results in less time.

6.2.5 Mobile robot registration

The experiments discussed so far use an artificial setup, with randomised start poses. The KVARNTORP-LOOP data set contains scans collected by a mobile robot, together with pose estimates for each scan, derived from the robot's odometry. This is more like the actual data that can be expected in the mine mapping application. A few examples of registering scans from this data set with different algorithms are shown in figure 6.17. The odometry errors for these scan pairs were found to be

$$\vec{e}_t = [-0.56, 0.03, 0.12] \ (|\vec{e}_t| = 0.573), \ |e_r| = 0.06 \text{ for scans 1-2,}$$

$$\vec{e}_t = [-0.34, 0.22, -0.13] \ (|\vec{e}_t| = 0.422), \ |e_r| = 0.111 \text{ for scans 17-18,}$$

$$\vec{e}_t = [-0.52, 0.10, -0.06] \ (|\vec{e}_t| = 0.535), \ |e_r| = 0.102 \text{ for scans 51-52.}$$

3D-NDT with iterative discretisation is the algorithm that gives the best results for the three scan pairs. It successfully registers all three pairs, except that the rotation error for the first pair is slightly larger than desired. The execution times were slightly shorter than that of ICP. Another notable result from these registrations is that linked bins and infinite outer bounds gave worse results with the fixed, octree, and additive discretisation schemes.

6.3 Qualitative comparison

The experiments covered in this chapter make it clear that, as long as iterative subdivision or clustering are not used for the discretisation step, 3D-NDT is much faster than ICP, which is the most common method for 3D registration today. When 3D-NDT converges, it often gives a closer match than ICP, especially when registering partly overlapping straight tunnel scans. When registering such data, ICP has a tendency to move the scans too close together in an attempt to maximise the amount of overlap between the two scans, because non-overlapping points are being matched to points near the edges.

3D-NDT with iterative subdivision is the most reliable of the investigated methods. When the algorithm is run in this fashion, it gives good results much more consistently, and it can still be performed at no greater cost than ICP. Which algorithm and which parameters to choose depends on the data. In many cases, iterative subdivision is not necessary. 3D-NDT sometimes has problems even when large features are available, if too many points are well-aligned to the walls. This seems to be not only because of the limited region of influence stemming from discretisation. It might be advantageous to weight cells according to their uniqueness within the scan, so that PDFs with a close distribution along a wall gets less weight if there are many such cells, and cells with PDFs with other alignments or other amounts of spread would get more weight.

One can argue that 3D-NDT inherently includes a basic noise filter, since scattered outlier points do not count towards the PDF. Only cells that contain

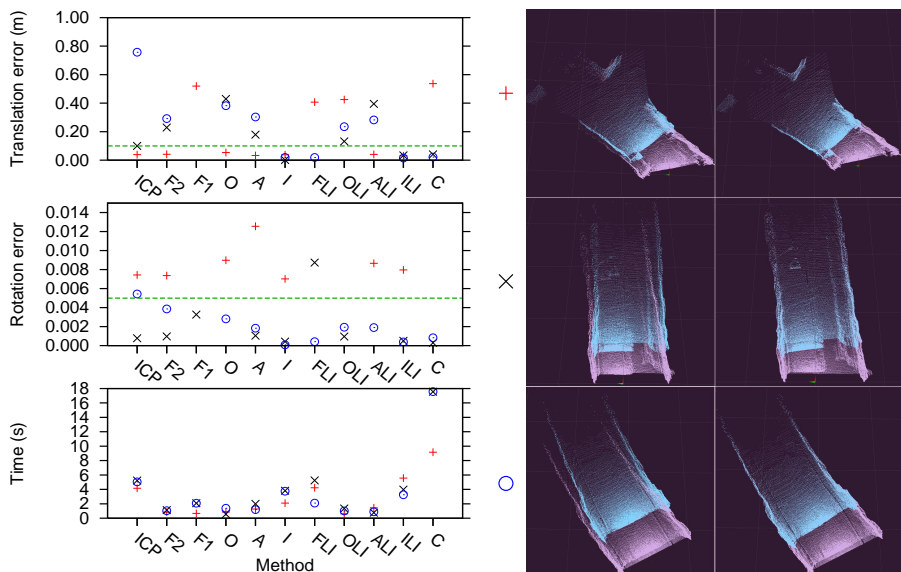


Figure 6.17: Matching scans from the KVARNTORP-LOOP data set with different registration algorithms, using robot odometry for the start pose. The leftmost one is the baseline ICP algorithm and the others are different 3D-NDT variants. F2 uses fixed 2 m cells; F1 uses 1 m cells; O, A, and I use octree, additive, and iterative subdivision of 2–0.75 m cells; the next four (oLI) use linked bins and infinite bounds; and C uses k-means clustering with 500 clusters. The red points correspond to scans 1 and 2, the dark crosses to scans 17 and 18, and the blue circles to scans 51 and 52. The scan pairs are shown to the right: on the left is the pose from odometry, and on the right is the pose after successful registration.

a sufficient number of points (more than 5) are considered occupied. However, only a few stray points that lie within a bin where most other points are coplanar will disrupt the covariance so that the distribution is more spherical than planar.

To conclude, the main advantage of ICP is that it is well known and used. It can also potentially make better use of small-scale features. 3D-NDT, on the other hand, is faster, and can produce results that are as accurate or better than those of ICP. However, 3D-NDT introduces other parameters, most importantly the size of the individual cells, and is more sensitive to error in the initial pose estimate. When the initial error in the pose estimate is large, 3D-NDT with iterative subdivision is preferable. Otherwise, 3D-NDT with additive subdivision generally gives good results in shorter time.

6.4 Discussion

When performing autonomous mapping of environments with little salient features, it is important to make scans at close intervals. None of the algorithms investigated in this chapter can be used as a “hands-free” solution for the KVARNTORP-LOOP data set, where parts of an underground mine system were scanned at approximately 5 m intervals using a small mobile robot. Manual intervention is needed to make sure that all pair-wise registrations are correct. It is likely that the registration process would be much smoother if the scans had been taken at closer intervals; both because they then would have more overlap and more features in common, and because the relative error in the pose estimate taken from the robot’s odometry readings would be smaller.

Chapter 7

Conclusions and future work

7.1 Conclusions

This thesis has investigated methods for registration of 3D range scans for volumetric profiling of an underground mine environment. A survey of relevant sensor technology and registration algorithms has also been presented.

Mine mapping with a mobile robot comes with several challenges not encountered with indoor mapping. One main difficulty is that the floors are uneven, which leads to the need for scan registration with 3D translations and rotations. Long tunnel sections with very little prominent features are also commonly encountered, and such environments are very challenging to current registration algorithms. Furthermore, the scale of the environment is typically large: mine tunnels are commonly 10–50 m wide, and mines often cover several square kilometres.

The thesis has presented a novel registration algorithm: the 3D normal distributions transform. An analysis of the algorithm with respect to different parameters based on real-world experiments in a mine has also been presented, alongside with a comparison with ICP, the most common registration algorithm used today. 3D-NDT has been shown to be a fast registration algorithm that is capable of very accurate registration, though it is more sensitive to large errors in the initial pose estimate.

7.2 Future work

Scan registration is only the first step when building and using a 3D map. The following sections discuss other problems that need to be solved before autonomous mining vehicles can be used in reality.

One application of underground mine automation that is of particular interest to the mining industry is the operation of load–haul–dump (LHD) vehicles. These vehicles dig out the ore from the blast site and haul it to a dump site, where it can be processed further. LHD vehicle automation needs work in all of these areas. LHD vehicles work in dynamic environments, they need to work

swiftly in order not to hinder other vehicles in the area, and consequently must be able to acquire 3D data while moving, and they also need to have semantic understanding of the environments they inhabit in order to load ore efficiently and safely.

7.2.1 Dynamic environments

An active mine changes substantially over time. On a larger scale, the shape changes in a rather predictable manner as new tunnels are built according to specified plans. On a smaller scale, the perceived look and shape of tunnels change as vehicles move back and forth, gravel heaps and boulders appear and are hauled away, permanent or less permanent structures are built, and so on.

It would be interesting to look into methods for maintaining an accurate and reliable 3D model even as the environment changes. The main problem in this kind of scenario is the so-called stability–plasticity dilemma — that is, the trade-off between adapting to new patterns (recognising significant changes) and preserving old patterns (disregarding insignificant changes). Biber and Duckett [5] did some preliminary work on this, representing 2D maps on multiple timescales simultaneously in order to solve this dilemma. However, they did not consider topological changes — something that is likely to occur in a production mine. Yamauchi and Beer [73] and Zimmer [75] presented systems that attempt to handle topological changes. Yamauchi and Beer did this using what they call adaptive place networks. Here, specified locations and the probability that paths between them can be traversed are stored in a directed graph. The probabilities are then updated gradually as the robot attempts to traverse the links. In this way, they are able to delete paths that have been blocked, and new paths can be added during exploration.

Mapping and localisation in a dynamic environment includes being able to filter out persons and other moving objects from the background. Such filtering can be used both as a safety measure, in order to avoid colliding with people and other objects while driving; and also to filter out moving objects from the final map. Maintaining a usable map is made even more difficult by temporarily stationary objects such as parked cars and non-permanent fixtures. This is the so-called DATMO (detection and tracking of moving objects) problem. Both SLAM and DATMO have been studied in isolation for many years, but only recently have efforts been made to integrate them (see for example Wang et al. [67]).

Dynamic environments can be found in a vast number of other applications. In fact, it is hard to contrive a real-world application where one does not have to account for dynamic changes in the environment when building and using maps. Changing light conditions, changing seasons, and moving people, furniture, or other vehicles — all of these are changes that may need to be taken into account.

7.2.2 Scanning while moving

Current 3D laser scanners are not fast enough to give high-resolution scans at interactive frame rates. Therefore it is necessary to stop the vehicle while making a 3D scan. If the scanner could be running continuously while the vehicle is moving, 3D mapping would be made much more efficient.

The problem is that without any extra sensors measuring the exact velocity of the vehicle at each moment, it is quite difficult to filter a scan made while driving over a bumpy surface. It is not realistic to expect that it would be possible to get an equally accurate scan while moving as is possible when standing still, but it would be interesting to see how much information can be extracted and to what extent the disturbances can be filtered.

As long as a reference object, for example a floor that is known to be flat and horizontal, is visible during the whole scan, it should be relatively easy to compensate for the motion. In the general case, however, it is likely to be much more difficult, using 3D range data alone.

There have been some efforts in this direction, but not many. Wulf et al. mounted a SICK scanner, with the scanning plane oriented vertically, on a special turntable constructed so that the scanner could be rotated around the vertical axis continuously. This is shown in figure 7.1. Using this setup, they can make a full 3D scan in between 1.2 s and 3.2 s, depending on the required resolution [70, 71]. Assuming that the robot is moving on a flat floor and that the odometry readings are accurate enough over the time needed for one scan, it is possible simply to interpolate the sensor position for each scan line between the start and end pose of the scan. This method is likely to give sub-optimal results when driving over a rough surface, such as a typical mine floor. In that case, a 3D gyro is likely to be needed, to give the relative orientation of each scan line [10].

7.2.3 Semantic analysis

A 3D model only gives a description of the shape of the measured area. Having a map with higher level information could also be very useful, making it possible to recognise and classify which parts of the model are junctions and how many tunnels join at the junction, which parts are entrances and exits, and so on. A map that includes such information is much more useful for navigation than a plain 3D model, and is clearly a prerequisite for autonomous operation in underground mining applications.

In addition to topological information about the structure of the mine, a lot of other semantic information can be extracted from the 3D model as well — for example, determining whether a tunnel subsection is secured or not, or whether or not a certain tunnel wall is a working face (that is to be drilled and blasted) or not. For autonomous load-haul-dump operation, where ore gravel is transported from the blast site to a dump area, it is necessary to discriminate the



Figure 7.1: Continuously rotating 3D SICK scanner setup.

muck pile from solid walls and floor around it. In the case of manual operation, this information can be presented as an aid to the vehicle operator.

Several authors have previously published work on semantic maps. Stachniss et al. [62] use a combination of a 2D laser range finder and a camera to discriminate between different kinds of rooms. The wall structure is measured by the 2D range slice, and the classification is also based on object detection from the camera images, looking for objects typically found in different kinds of rooms. Nüchter et al. [50] have used the Kurt3D platform to detect walls and ceiling in an indoor office environment, as shown in figure 7.2. For such an environment, the task can be achieved mainly by looking at the gradient between consecutive points, with some extra measures to account for noisy sensor data. Mine tunnels with their more uneven contours are more challenging, and it is clear that other methods are needed to tell the difference between the ore gravel heap and the surrounding solid walls.

7.2.4 Sensor fusion

Combining different types of sensors makes it possible to build a map with substantially more information than is possible with only a range finder.

A model where colour and texture (from a digital camera) is added might be used to map the distribution of rock types within a mine. Examples of previous work on colour-range data fusion includes Andreasson et al. [2] and Yoshida

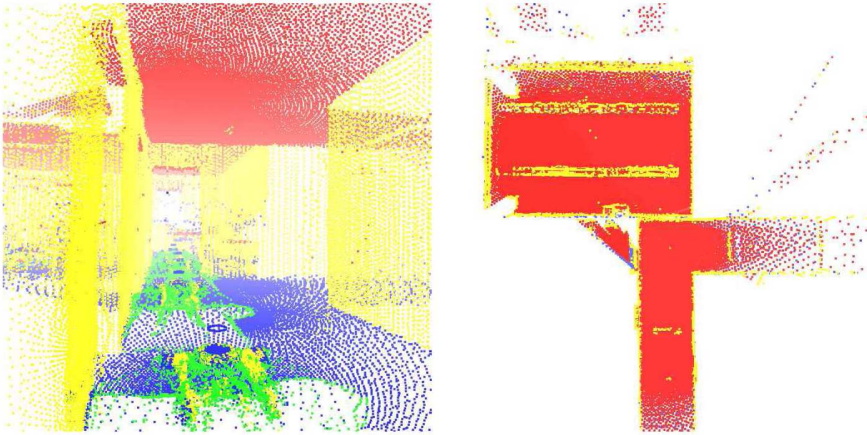


Figure 7.2: Semantically labelled 3D point cloud from an office environment produced by the Kurt3D platform. Red points mark the ceiling, blue points the floor, yellow points are other objects, and green points correspond to artifacts from scanning the robot itself. On the left is a frontal view, and the right picture is a view from the top.

and Saito [74]. Andreasson et al. extract planar structures from range data by fusing colour information with the laser data. The camera and range sensor are calibrated using the intensity of the camera pixels and the remission values of the range scanner. Yoshida and Saito use an ICP method that incorporates colour information, and also warp the textures of the registered scans to give the model a better appearance. Other researchers have used colour data not to aid registration but to map textures to already existing models [7, 25, 51].

Combining a range finder with either a regular camera or a thermal camera would also make it much easier to filter out movable objects, such as people or vehicles, even though this is not impossible to do using only 3D range data. Less work has been published on thermal–range data fusion, but the thermal images and colour images have many similar properties.

Martin et al. [42] used a combination of a lidar, a sonar, and an omnidirectional camera for people detection on a mobile robot platform. Each of the three sensors was used to generate a probability distribution for the number of persons surrounding the robot and their positions. The distributions were aggregated using Bayes’ rule to produce a more probable hypothesis of the distribution of people in the robot’s immediate vicinity.

It might also be interesting to include a gas sensor and an anemometer, and put such information into the map as well.

7.2.5 Discussion

All of these approaches would benefit from sensor fusion — having access to more information also makes it easier to perform many tasks. There is a sub-

stantial body of work concerning object recognition, both in 2D and 3D, although not so much in a mining context. Dynamic maps and scanning while moving are more novel areas. All these areas are interesting both from an academic point of view and from the perspective of the rock drilling and mining industry.

Bibliography

- [1] Simon L. Altmann. *Rotations, Quaternions, and Double Groups*. Oxford Science Publications, 1986.
- [2] Henrik Andreasson, Rudolph Triebel, and Wolfram Burgard. Improving plane extraction from 3D data by fusing laser data and vision. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Alberta, Canada, 2005. IEEE.
- [3] C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D. Hähnel, and S. Thrun. A campaign in autonomous mine mapping. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, April 2004.
- [4] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), February 1992.
- [5] Peter Biber and Tom Duckett. Dynamic maps for long-term operation of mobile service robots. In *Robotics: Science and Systems*, June 2005.
- [6] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2003.
- [7] Peter Biber, Sven Fleck, Michael Wand, Dirk Staneker, and Wolfgang Straßer. First experiences with a mobile platform for flexible 3D model acquisition in indoor and outdoor environments – the Wägele. In *3D-ARCH'2005: 3D Virtual Reconstruction and Visualization of Complex Architectures*, 2005.
- [8] G. Boström, M. Fiocco, D. Puig, A. Rossini, J. G. M. Gonçalves, and V. Sequeira. Acquisition, modelling and rendering of very large urban environments. In *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2004.

- [9] Faysal Boughorbel, Andreas Koschan, Besma Abidi, and Mongi Abidi. Gaussian fields: a new criterion for 3D rigid registration. *Pattern Recognition*, 37(7):1567–1571, 2004.
- [10] Christian Brenneke, Oliver Wulf, and Bernardo Wagner. Using 3D laser range data for SLAM in outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
- [11] Ming-Ching Chang, Frederic F. Leymarie, and Benjamin B. Kimia. 3D shape registration using regularized medial scaffolds. In *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, pages 987–994, 2004.
- [12] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992.
- [13] Chinese State Administration of Work Safety, January 3 2006. URL <http://www.chinasafety.gov.cn/anquanfenxi/anquanfenxi.htm>
- [14] Ingemar J. Cox. Blanche: An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, April 1991.
- [15] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, 1996.
- [16] H. R. Everett. *Sensors for Mobile Robots: Theory and Application*. A K Peters, Ltd, 1995. ISBN 1-56881-048-2.
- [17] N. Fairfield, G.A. Kantor, and D. Wettergreen. Three dimensional evidence grids for SLAM in complex underwater environments. In *Proceedings of the 14th International Symposium of Unmanned Untethered Submersible Technology (UUST)*, August 2005.
- [18] Dave Ferguson. Volumetric mapping with 3D registration. Technical report, Robotics Institute, Carnegie Mellon University, November 2002.
- [19] Alex Foessel-Bunting. Radar sensor model for three dimensional map building. In *Proc. SPIE, Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, volume 4195, November 2000.
- [20] Pekka Forsman and Aarne Halme. Feature based registration of range images for mapping of natural outdoor environments. In *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2004.
- [21] Udo Frese, Per Larsson, and Tom Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, April 2005.

- [22] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.
- [23] Christian Früh and Avidesh Zakhor. 3D model generation for cities using aerial photographs and ground level laser scans. In *Vision and Pattern Recognition Conference*, December 2001.
- [24] Christian Früh and Avidesh Zakhor. Fast 3D model generation in urban environments. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 165–170, April 2001.
- [25] Christian Früh and Avidesh Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60:5–24, October 2004.
- [26] Michael Greenspan and Guy Godin. A nearest neighbor method for efficient ICP. In *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, 2001.
- [27] Michael Greenspan and Mike Yurick. Approximate k-d tree search for efficient ICP. In *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM '03)*, 2003.
- [28] Michael Greenspan, Guy Godin, and Jimmy Talbot. Acceleration of binning nearest neighbour methods. In *Proceedings of Vision Interface*, pages 337–344, May 2000.
- [29] Jose Guivant, Eduardo Nebot, and Hugh Durrant Whyte. Simultaneous localization and map building using natural features in outdoor environments. In *Intelligent Autonomous Systems 6*, pages 581–588, July 2000.
- [30] Dirk Hähnel and Wolfram Burgard. Probabilistic matching for 3d scan registration. In *Proc. of the VDI-Conference Robotik 2002 (Robotik)*, 2002.
- [31] Daniel F. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Carnegie Mellon University, 2002.
- [32] Daniel F. Huber and Martial Hebert. A new approach to 3-D terrain mapping. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1999.
- [33] Andrew E. Johnson. Surface landmark selection and matching in natural terrain. In *IEEE Computer Vision and Pattern Recognition*, 2000.
- [34] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

- [35] Andrew Edie Johnson. *Spin Images: A Representation for 3-D Surface Matching*. PhD thesis, Carnegie Mellon University, 1997.
- [36] Geoff Koch. Technology speeds up efforts to piece together ancient marble map of rome. *Stanford Report*, April 19 2004.
- [37] Astrid Lindgren. *Vi på Saltkråkan*. Rabén & Sjögren, 1964.
- [38] Kai Lingemann, Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. Indoor and outdoor localization for fast mobile robots. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.
- [39] Kai Lingemann, Andreas Nüchter, Joachim Herzberg, and Hartmut Surmann. About the control of high speed mobile indoor robots. In *Proceedings of the European Conference on Mobile Robots*, September 2005.
- [40] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
- [41] Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, March 1997.
- [42] Christian Martin, Erik Schaffernicht, Andrea Scheidig, and Horst-Michael Gross. Sensor fusion using a probabilistic aggregation scheme for people detection and tracking. In *Proceedings of the European Conference on Mobile Robots*, 2005.
- [43] Jiří Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.
- [44] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.
- [45] Luis Montesano, Javier Minguéz, and Luis Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2005.
- [46] Hans Moravec and A. E. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pages 116–121, March 1985.
- [47] Aaron Christopher Morris, David Silver, David Ferguson, and Scott Thayer. Towards topological exploration of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

- [48] Andreas Nüchter, Hartmut Surmann, Kai Lingemann, and Joachim Hertzberg. 6D SLAM with an application to autonomous mine mapping. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, April 2004.
- [49] Andreas Nüchter, Kai Lingemann, Joachim Herzberg, and Hartmut Surmann. Heuristic-based laser scan matching for outdoor 6D SLAM. In *KI 2005: 28th Annual German Conference on AI*, pages 304–319. Springer, September 2005.
- [50] Andreas Nüchter, Oliver Wulf, Kai Lingemann, Joachim Hertzberg, Bernardo Wagner, and Hartmut Surmann. 3D mapping with semantic knowledge. In *Proceedings of the RoboCup International Symposium*, July 2005.
- [51] Kazunori Ohno and Satoshi Tadokoro. Dense 3D map building based on lrf data and color image fusion. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [52] Clark F. Olson. Probabilistic self-localization for mobile robots. In *IEEE Transactions on Robotics and Automation*, pages 55–66, February 2000.
- [53] Clark F. Olson and Daniel P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, January 1997.
- [54] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4): 807–832, October 2002.
- [55] Torbjörn Petersson. Gruvdöden tabu i Fuxin. *Dagens Nyheter*, pages A18–, April 24 2005.
- [56] Nora Ripperda and Claus Brenner. Marker-free registration of terrestrial laser scans using the normal distribution transform. In *Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005*, August 2005.
- [57] Szymon Marek Rusinkiewicz. Efficient variants of the ICP algorithm. In *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, 2001.
- [58] Szymon Marek Rusinkiewicz. *Real-time acquisition and rendering of large 3D models*. PhD thesis, Stanford University, 2001.
- [59] Gregory C. Sharp, Sang W. Lee, and David K. Wehe. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102, January 2002.

- [60] David Silver, David Ferguson, Aaron Christopher Morris, and Scott Thayer. Feature extraction for topological mine maps. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, September 2004.
- [61] David A. Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Carnegie Mellon University, 1996.
- [62] Cyrill Stachniss, Oscar Martínez-Mozos, Axel Rottmann, and Wolfram Burgard. Semantic labeling of places. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, San Francisco, CA, USA, 2005.
- [63] Fridtjof Stein and Gérard Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, February 1992.
- [64] Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45:181–198, 2003.
- [65] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of underground mines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [66] Sebastian Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [67] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [68] Greg Welch and Gary Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, April 2004. TR 95-041.
- [69] Mark Whitehorn, Tyrone Vincent, Christian H. Debrunner, and John Steele. Stereo vision in LHD automation. *IEEE Transactions on Industry Applications*, 39(1), 2003.
- [70] Oliver Wulf and Bernardo Wagner. Fast 3D-scanning methods for laser measurement systems. In *International Conference on Control Systems and Computer Science (CSCS14)*, 2003.

- [71] Oliver Wulf, Bernardo Wagner, and Mohamed Khalaf-Allah. Using 3D data for Monte Carlo localization in complex indoor environments. In *Proceedings of the European Conference on Mobile Robots*, 2005.
- [72] Sameh M. Yamany and Aly A. Farag. Free-form surface registration using surface signatures. In *IEEE International Conference on Computer Vision (ICCV'99)*, September 1999.
- [73] Brian Yamauchi and Randall Beer. Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man and Cybernetics*, 26(3):496–505, June 1996.
- [74] K. Yoshida and H. Saito. Registration of range images using texture of high-resolution color images. In *IAPR Workshop on Machine Vision Application (MVA02)*, December 2002.
- [75] Uwe Zimmer. *Adaptive Approaches to Basic Mobile Robot Tasks*. PhD thesis, University of Kaiserslautern, 1995.