HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communications and Automation

Heikki Hyyti

Simultaneous Localization and Forest Mapping Using 2D and 3D Laser Scanners

Master's thesis

Espoo, September 11, 2009

Supervisor: Professor Arto Visala
Instructor: M.Sc. Mikko Miettinen

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author:	Heikki Hyyti		
Title of the thesis:	Simultaneous Localization and Forest Mapping Using 2D and 3D Laser Scanners		
Date:	9/11/2009	**Number of pages:**	7 + 93
Professorship:	Automation Technology	**Code:**	AS-84
Supervisor:	Professor Arto Visala		
Instructor:	M.Sc. Mikko Miettinen		

Simultaneous localization and forest mapping using 2D and 3D laser scanners demonstrates a complete solution to measure trees and map forest. In this thesis trees are measured locally in the forest using two- and three-dimensional laser scanners in motion. It presents a way to find tree-like objects from a cloud of measured points in real-time. Used measuring and mapping algorithms are specially fitted for this application to increase accuracy and performance.

This thesis begins to solve simultaneous localization and mapping (SLAM) from the pose of the measurement vehicle. The pose of the vehicle is estimated using laser odometry method that uses consecutive laser scanner measurements to track the movement. New heuristic methods are developed and tested for this application to maximize the accuracy of the pose estimate. This thesis introduces a new way to use cross-correlations in the laser odometry and shows how inertial measurements may be used to improve the quality of the odometry.

This thesis introduces a feature based map where 2D and 3D measurements are combined, and explains all the methods needed for building and updating the map. The map is used to readjust previously calculated pose estimate. All the collected measurements are used to calculated tree diameter statistics at different heights.

The resulting map is quite accurate. The mapping accuracy is a few centimeters for tree diameters and a few tens of centimeters for the location of trees. The localization error of the measurement vehicle is estimated to be smaller than the location error of the trees, as a large group of mapped trees is used to estimate the pose of the vehicle. The map is fixed to global map coordinates using a GPS receiver.

Keywords:	SLAM, pose tracking, tree measuring, laser scanner, forest mapping
Language:	English

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Heikki Hyyti		
Työn nimi:	Yhtäaikainen paikannus ja puuston kartoitus 2D- ja 3D-laserskannereilla		
Päivämäärä:	11.9.2009	**Sivujen määrä:**	7 + 93
Professuuri:	Automaatiotekniikka	**Koodi:**	AS-84
Työn valvoja:	Professori Arto Visala		
Työn ohjaaja:	DI Mikko Miettinen		

Yhtäaikainen paikannus ja puuston kartoitus 2D- ja 3D- laserskannereilla esittää tavan mitata ja kartoittaa metsän puut. Työssä mittaukset tehtiin paikallisesti metsässä käyttäen kaksi ja kolmiulotteisia laserskannereita liikkeestä mitaten. Työ esittää menetelmän puumaiset kohteiden tunnistamiseen mittalaitteiden tuottamasta pisteparvesta reaaliajassa. Työssä mittaus- ja kartoitusalgoritmit sovitetaan erityisesti metsän kartoitusta ja puuston mittausta varten.

Diplomityössä yhtäaikaisen paikoituksen ja kartoituksen ongelmaa lähdetään ratkaisemaan mittauslaitteen mahdollisimman tarkan paikan ja asennon estimoinnista metsäolosuhteissa. Mittauslaitteiston paikkaa mitataan laserodometrialla, jossa perättäisiä laserkeilauksia verrataan toisiinsa ja näiden väliltä tunnistetaan liike suhteessa ympäristöön. Työssä kehitetään uusia heuristisia paikannusmetodeja metsäympäristöön. Työ esittelee uuden tavan käyttää ristikorrelaatioita laserodometriassa ja näyttää miten gyro- ja kiihtyvyysantureita voidaan käyttää odometriatiedon parantamiseen.

Diplomityö esittää piirrepohjaisen puukartan, jonne 2D- ja 3D-laseretäisyysmittauksista lasketut piirteet lisätään. Karttaa päivitetään jatkuvasti ja uusinta kartan tietoa käytetään mitatun paikkatiedon parantamiseen. Samoin kaikkia kerättyjä mittauksia käytetään tilastollisesti parantamaan aikaisemmin eri korkeuksilta laskettuja puurunkojen läpimittaestimaatteja.

Lopputuloksena saatu kartta on melko tarkka. Kartoituksessa puun läpimitan tarkkuus on muutamia senttimetrejä ja puun paikan tarkkuus muuta kymmenen senttimetriä. Mittauslaitteen paikan arvioidaan olevan tarkempi kuin puiden, koska suurta määrää kartoitettuja puita käytetään mittauslaitteen paikan ja asennon sovittamiseen. Suhteellinen kartta on kiinnitetty globaaliin koordinaatistoon GPS mittalaitteen avulla.

Avainsanat: SLAM, paikan ja asennon seuranta, laserskanneri, metsän kartoitus

Kieli: englanti

Acknowledgments

First I would like to thank Professor Arto Visala for providing me with this challenging and rewarding subject for my master's thesis. I would also like to thank Mikko Miettinen and Matti Öhman for their help and assistance throughout the project.

I would also like to thank all the personnel of the laboratory for providing me with a relaxed and encouraging atmosphere to work in. Especially I would like to thank my roommates Jakke Kulovesi, Jouko Kalmari, Teemu Kemppainen and Visa Jokelainen for all the help and understanding.

Espoo, September 11, 2009

Heikki Hyyti

Contents

Notation and Symbols

x, X	scalar
$\mathbf{v}$, $\underline{v}$	vector
$\mathbf{M}$, $\underline{M}$	matrix
$\mathbf{M}_{(i,j)}$	component indexing, ith row, jth column
a_{name}, A_{name}	naming of objects
$a_{name,i}$, $\mathrm{A}_{name,i}$	indexing of named objects

C_{fg}	Discrete correlation between data sets f and g
d	tree diameter
ρ	tree radius
E	error
n, N	length
P	probability
q	quality
R	Earth's radius
r, θ	polar coordinates: radius, angle
W	weight
x, y, z	Cartesian coordinates
δ	angular sampling constant
λ, φ	geographic coordinates: longitude, latitude
ψ_e, θ_e, φ_e	Euler angles: yaw, pitch, roll
Ω	solid angle

ATV	All Terrain Vehicle
CM	Center of Mass
FOV	Field of view
GPS	Global Positioning System
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
SLAM	Simultaneous Localization and Mapping
STD	Standard deviation

Chapter 1

Introduction

Simultaneous localization and forest mapping using 2D and 3D laser scanners demonstrates a complete solution to measure trees and map forest. Measurements are taken locally using a vehicle equipped with scanning laser rangefinders often called laser scanners. The idea is quite new as previously forests are measured only from satellites or airplanes on a large scale. It is called Airborne Laser Scanning (ALS). In this work trees are measured using two- and three-dimensional laser scanners and the used measuring and mapping algorithms are fitted specially for this application. Similar methods are widely used in robotics for navigation and for building a model of robot's surroundings.

All the needed data could be collected with measuring equipment added to a forest harvester, because it moves slowly around while a modern woodsman chops trees or thins the forest. Forest harvesters have already lots of different sensors and electronics, and some of those could be used for aiding tree measuring and mapping. Rest of the needed measuring devices like laser scanners could be integrated to the harvester. However in this work the measuring equipment is installed on an All Terrain Vehicle (ATV) as it was easily available. The used measurement vehicle and sensors are described in Chapter 3.

The main part of this work focuses on algorithms needed to extract trees from 2D and 3D laser scanner measurements and methods to generate a map from the extracted features. Also algorithms and equations that are needed in the use of inertia and GPS measurements are described. All the main algorithms and methods in this work are described and explained so that the whole process can be understood. This work shows how data from different measuring devices can be transformed into an accurate forest map.

Mapping algorithms are usually used for robots which need the information about passable ways and obstacles. In this work the main goal is to measure trees while the possible obstacles are not that interesting. Therefore the existing algorithms cannot be straightforwardly used.

Instead many different algorithms and methods are combined and adapted particularly to this special case of measuring and mapping forest.

This thesis begins to solve simultaneous localization and mapping (SLAM) from the pose of the measurement vehicle. The pose of the vehicle is estimated using laser odometry method that uses consecutive laser scanner measurements to track the movement. The tested laser odometry methods are explained in Chapter 4. The same chapter also describes the methods for using inertia and GPS measurements. Subsequently tree features are extracted from collected measurements using methods described in Chapter 5. The chapter explains the different algorithms for finding and measuring trees from the horizontal 2D laser scans and from the 3D data collected using the self-made rotating 3D laser scanner.

A map is built from the gathered tree measurements and their locations along the travelled path in Chapter 6. This chapter introduces a feature based map where the 2D and 3D measurements are combined and explains the methods for building and updating the map. The gathered map is used to adjust the previously calculated localization estimate. Similarly all the gathered measurements are statistically used to estimate tree diameters at different heights measured from the ground level. Finally the results are covered in Chapter 7.

Chapter 2

Tree Localization and Mapping

The localization and mapping problem in this work is adapted for finding and measuring trees in the forest. This chapter introduces the simultaneous localization and mapping (SLAM) problem, and presents the special requirements that are needed in the forest environment. The algorithms developed in this work are also intended to be used in a forest harvester machine while the machine is working in a forest. Therefore the localization and mapping problem has to be solved using real-time capable and causal algorithms. The system is intended to be the basis of a more intelligent automated foresting application in the future.

2.1 Simultaneous Localization and Mapping

The term SLAM is an acronym for Simultaneous Localization and Mapping. It was originally developed by Hugh Durrant-Whyte and John J. Leonard [1] based on earlier work by Smith, Self and Cheeseman [2]. Durrant-Whyte and Leonard originally called it SMAL but it was later renamed to give a bigger impression [3].

Mapping is a problem of transforming the gathered information into a representation that a computer can understand. Main aspects in mapping are representation of the environment and interpretation of the sensor data. On the other hand, localization is the problem of estimating the pose of the robot relative to a map. Simultaneous localization and mapping is therefore defined as the problem of building a map while at the same time localizing the robot within that map. In practice, these two problems cannot be solved independently of each other. Therefore SLAM is often referred to as a chicken and egg problem: a good map is needed for accurate localization while a precise pose estimate is needed to build a proper map. [4]

Simultaneous localization and mapping can be implemented in many ways. First of all there is a huge amount of different hardware that can be used. Secondly SLAM is more like a concept than a single algorithm. There are multiple steps involved in SLAM and these different steps can be implemented using various algorithms. It is not uncommon that a different approach gives the best result depending on the application. Simultaneous localization and mapping has not been completely solved and there is still substantial research going on in this field. [3]

One of the popular algorithms for solving the simultaneous localization and mapping problem is the extended Kalman filter based EKF SLAM, which is usually used for feature based mapping in a two-dimensional case. Typically, EKF SLAM algorithms are featured based, and they use a maximum likelihood algorithm for data association. In resent years, the EKF SLAM has been the de facto standard method for simultaneous localization and mapping, until the introduction of FastSLAM. [5]

The FastSLAM by Michael Montemerlo is a more efficient way of solving the SLAM problem. The algorithm is based on a factorization of the posterior into a product of conditional landmark distributions and a distribution over robot paths. It estimates the full posterior distribution over the robot pose and landmark locations using a modified particle filter. Each particle possesses multiple Kalman filters that estimate all the landmark locations conditioned on the estimate of the path. [5]

2.2 State of the Art

Correlation based odometry is used in the paper from Gerhard Weiß et al. about indoor robot localization and orientation [6]. They successfully used angle correlation between old and new measurements when there was only rotation and no translation. For more complex cases they have introduced an angle histogram, which is a histogram of the angles between the adjacent measured points. Those histograms are correlated to the previously calculated histogram to find the rotation between the two poses.

Weiß et al. [6] has calculated the translation from the correlated x- and y-dimension histograms. In the indoor environment the edges of the room are visible in translation histograms, but in a forest the x- and y-dimensional data spread equally trough the histogram. Histogram based method cannot be straightforwardly used in a forest as there are no straight walls.

Artur Dubrawski and Barbara Siemiatkowska have studied correlation based laser odometry in their article "A Method for Tracking Pose of a Mobile Robot Equipped with a Scanning Laser Range Finder" [7]. They have solved the angular correlation problem using angle histograms.

Edouard Ivanjko, Bojana Dalbelo-Bašić, and Ivan Petrović have also studied a similar problem. They use histogram based solutions to solve the problem in their article: "Correlation Based Approach to Mobile Robot Pose Tracking in Unknown Environments" [8]. They both use a similar angular histogram solution to find the rotation between different scans as Weiß et al. in [6].

The tree measuring problem has been studied by Jaakko Jutila earlier in the same Forestrix-project in which this work is done. In his master's thesis, "Localization and Tree Measurement with Laser Scanner in a Forestry Machine Perception System" [9], the forest was measured using horizontal 2D laser scanner. Jutila has studied and compared different algorithms for tree measurement. His results are used in this work in the part concerning 2D laser scanner measurements.

The experience of research scientists in our laboratory was used as a starting point for studying different ways of mapping a forest. Nearly similar measuring setups have been previously built by Mikko Miettinen, Matti Öhman and Jaakko Jutila for gathering two-dimensional laser scanner data and maps from the forest. Mikko Miettinen, Matti Öhman, Arto Visala and Pekka Forsman have studied the SLAM-problem and released an article about "Simultaneous Localization and Mapping for Forest Harvesters" [10], where they have published their feature based 2D-mapping algorithms.

In our laboratory, SLAM has also been studied by Jorma Selkäinaho in his dissertation, "Adaptive Autonomous Navigation of Mobile Robots in Unknown Environments" [11]. Later with Pekka Forsman they have published an article about navigation by matching 2D laser scans [12], where they present a type of laser odometry using a horizontal laser scanner. The mentioned laser odometry has been first introduced by Tim Bailey and Eduardo Nebot in their article, "Localisation in Large-scale Environments" [13].

The SLAM problem in outdoor environment is also well covered in Tim Bailey's dissertation [14]. He has studied mobile robot SLAM in extensive outdoor environments and written a complete study of how two-dimensional localization and mapping should be done. Many other 2D and 3D robot localization and mapping methods have been published since then. For instance David Cole and Paul Newman had studied similar problem in their article: "Using Laser Range Data for 3D SLAM in Outdoor Environments" [15]. They have built complete three-dimensional environment model of the area where the robot has visited.

Another version of the 3D SLAM is done by Andreas Nüchter, Kai Lingemann, and Joachim Hertzberg. In their work about the 6D SLAM, they used a RIEGL laser range finder on the robot and the ICP-algorithm for scan matching [16]. Both of the previous SLAM works required a full stop while taking the 3D measurements. This is a huge drawback as an autonomous forest mapping system should not interfere with the normal machine operation. The autonomous system has to cope with the information that it gets, when it gets.

There are also other ways of measuring the forests than from the ground level. The forests are commonly measured from satellites and airplanes. The measurement resolution is usually not high enough for sorting out individual trees. Thus only general measurements like average height of trees or tree densities on an area can be estimated. P. Gong and Y. Sheng have measured trees from high resolution aerial images. They have built semiautomatic way to measure tree heights and sizes from aerial images. [17] Similarly D.S. Boyd and F.M. Danson have studied the measurement of forest resources from satellite images. They have been able to classify large areas with coarse detail and get some large scale information about forests and the surrounding environment. [18]

Tree trunks can neither be seen from satellite nor aerial images. Therefore tree measurement system should go into the woods and measure it from the ground level. The quantity and quality of the tree trunks is valuable information and therefore it is important to go underneath tree tops.

2.3 Challenges in Forest Environment

Navigation is difficult in an unknown and variable forest environment. There are no known features or landmarks and every found landmark may change as trees can be cut down or fall. Undergrowth, twigs and sprigs hide the tree trunks that could otherwise be used as stable features. The use of identified features leads to a need of more complex perception and modeling algorithms. Therefore a big variety of different mapping algorithms has previously been presented depending on the kind of environment, the degree of structuring and the target application [10].

Environment can and probably will change during measurements. Trees sway in the wind causing constant movement in the surrounding environment. There are large scale changes in a forest over time as trees and undergrowth grows. In addition some trees can fall or be cut down between measurements. There are always bushes, rocks, fences and human built objects among trees. Finding and recognizing a tree from the taken measurements is a challenge. To measure and map every tree is even harder. The variety and complexity of Finnish pine forest could be seen in Figure 1 below. The part of the forest from where this picture is taken is very sparse, but on the other hand younger trees at the background are grown next to each other forming extremely dense foliage.

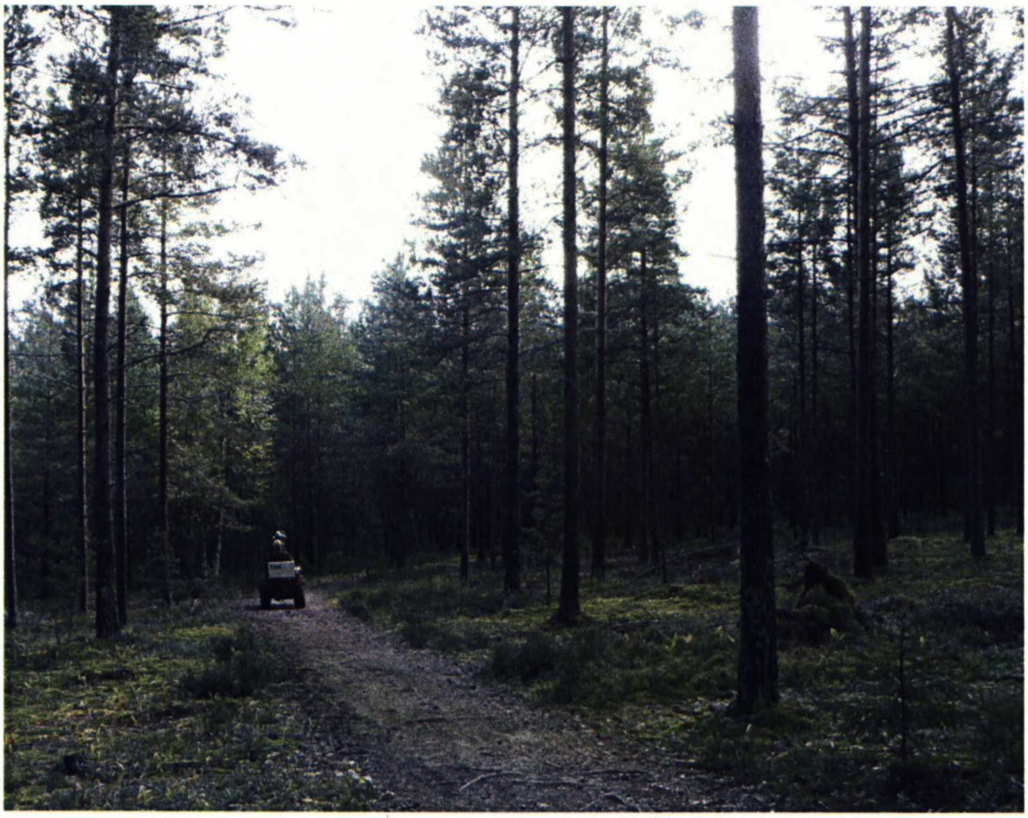

Figure 1: Measuring forest is hard in an unconstructed environment. There is sparse pine forest in the front and dense smaller pine trees at background. Ground can be hilly and there are usually plenty of obstacles and undergrowth.

Navigation can be complicated by difficult ground conditions like stumps and ditches. The ground might be wet and muddy causing sinking, skidding, and slippage. Therefore use of traditional odometry with measured wheel angles and rotations does not work reliably in the forest where the environmental conditions change. [19]

Difficult lightning and weather conditions affect many sensors, in particular vision systems [19]. Bright sunlight is interfering with laser scanners as well as cameras but laser scanners do not need any external lighting in contrast to cameras. The forest environment is hard for electronics, measurement devices and computers, which have to be shielded against moisture, dust, freezing and overheating. Snow, rain, and direct sunshine can damage measuring devices and disturb measurements [19]. Environment is different at winter time as snow and ice cover trees as well as the rest of the landscape.

2.4 Real-Time and Causality Requirements

In the work real-time requirements are taken into account while choosing and developing algorithms. In any autonomous robotic concept, all algorithms should be calculated in reasonable amount of time. Even though this work concentrates only on the autonomous tree measuring and mapping in the forest, results of the work could be used in real-time systems, where tree data would be collected and a map generated during measuring.

If a map had been generated using the whole data set afterwards, the map or the current location estimate would not be available during the harvesting operation. As no force majeure did arise with requirements, algorithms have been developed such a way that the real-time capability and causality of algorithms are ensured. Only those measurements are used to solve the localization and mapping problem, which are available at that time.

2.5 Future Applications

Nowadays the statistics and tree measurements are collected by hand, and thus there are only a little data to calculate statistics. With more accurate tree measurements and forest maps, the growth of trees and the need for thinning can be estimated more reliably. The forest owner would get a lot more accurate information from his (or her) forest and could better estimate the quantity and quality of trees. Also knowing the exact origin of every collected log would benefit corporations that need reliable traceability from their suppliers.

Later on a developed 3D mapping system could be used to make tree harvesting easier and more comfortable. For instance in a bad weather or at night time obstacles and trees could be visualized to driver such that the operator could better perceive the surroundings of the forest harvester. The need of harvesting could be planned a lot more detailed, as every tree would have a mapped identity and only trees of right size and species could be cut from forest as planned. The harvesting company could get more profit as workers could be aided to choose only trees that would give the best income.

Chapter 3

Measurement System and Measurements

In order to properly measure trees and to develop new methods for mapping forests, it is essential to design the measurement concept. The best solution seems to be a measurement device built on a forest harvester or like in our case on an ATV, because in that way the map generation can be done in parallel to normal work in a forest. Essential parts of the measurement system are a horizontal scanning laser rangefinder for measuring the pose change and a 3D scanner for measuring the trees at different heights. Alignment of the vehicle need to be known quite accurately to use the 3D scanner and thus an inertial measuring unit (IMU) has been added to the measurement system. Also a GPS receiver is attached to gain a global position for the gathered map.

In this chapter, the measurement vehicle is introduced with its attached measurement devices. Properties and measuring accuracies of different devices are analyzed and all needed functions and purposes of the measurement devices are explained. In the later part of the chapter, the used measuring methods and different typical finish forest types are shortly explained.

3.1 Measurement Vehicle

Obviously the best choice for a measurement vehicle would have been a forest harvester which is built for rough and hard forest environment. As no forest harvester was available for us, we built our measurement system on an all terrain vehicle (ATV). The vehicle is manufactured by Honda in the 1990s, but it is still working. Two large metal boxes seen in Figure 2 are fixed on the body of the ATV to give a platform for attaching equipment.

The measurement system consists of a computer with its accessories, a horizontal laser scanner, a rotating 3D scanner, an IMU, a GPS receiver, and an energy source. The measurement computer, taking care of data collection and storage, was placed in the front box and the batteries giving the power to all measurement devices are placed in the rear box. The horizontal laser scanner is fixed to front of the fore box. The initial versions of the same measurement vehicle included two horizontal laser scanners and they were both fixed to a rack fitted in the place of 3D scanner. They were aligned to the sides such that the maximal field of view (FOV) was acquired. The latest version includes a rotating 3D scanner that is fitted on top of the front box allowing large open FOV to the surrounding environment. The measurement vehicle with its subsystems is shown in Figure 2 below.

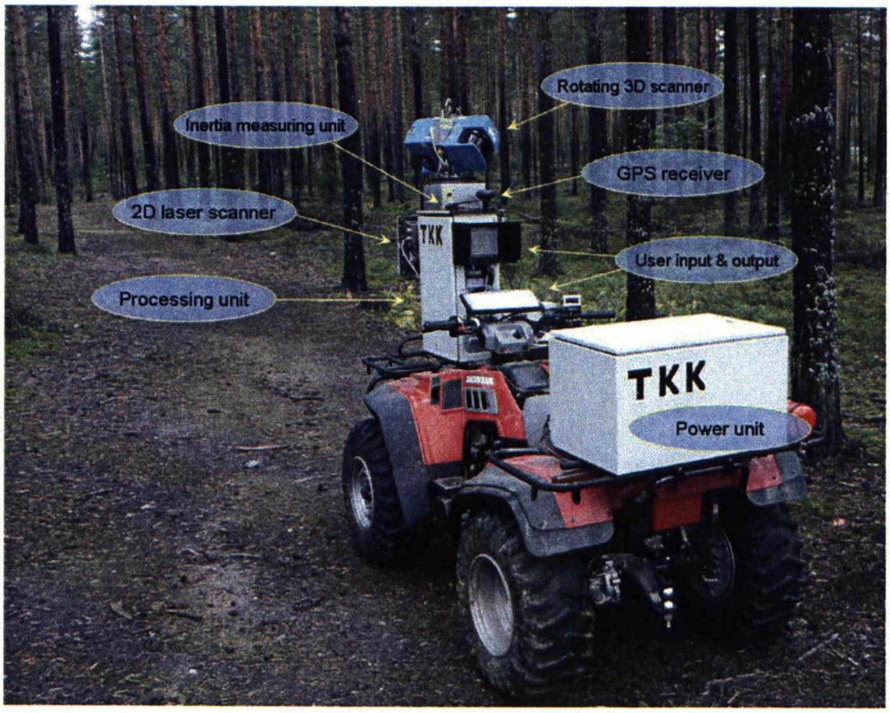

Figure 2: Honda all terrain vehicle and parts of the latest version of the measurement system. The difference between the initial and the latest versions is that the initial version had two 2D laser scanners pointing sideways in contrast to the latest version which has one 2D and one 3D scanner.

3.2 Horizontal 2D Laser Scanner

The initial measurement system consisted of two Sick LMS221 scanning laser rangefinders [20], or laser scanners for short. The later version of the measurement system has only one horizontal laser scanner. The field of view of the LMS221 scanner is 180° with the native resolution of 1°. It can also support higher resolution modes which are implemented using an interlacing technique. In the interlaced mode, the scanner takes 181 measurements during the first revolution of the rotating mirror. At the beginning of the second revolution the scanner adjusts its timing such that the new measurements are taken with 0.5° or 0.25° offsets relative to the first scan. The second scan consists of 180 measurements.

3.2.1 One 2D Scanner Pointing Ahead

In the later version of the measurement system, there is one Sick LMS221 laser scanner pointing ahead as seen in Figure 3. The second horizontal laser scanner has been left out to simplify the system and to make it more affordable for forest industry as every additional sensor increases complexity of the system. The laser odometry can be done with just one laser scanner as other studies show. For instance in Jorma Selkäinaho's dissertation, one horizontal laser scanner is sufficient for acquiring laser odometry [11].

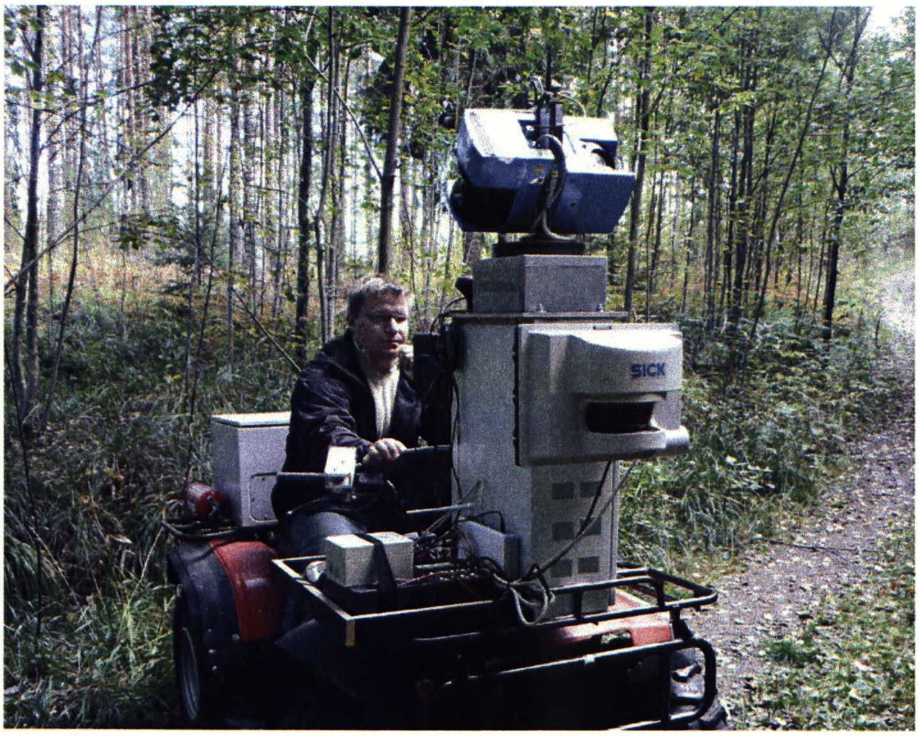

Figure 3: Image of the later version of the measurement vehicle at front. The horizontal 2D laser scanner was mounted to front of the system pointing forwards. The rotating 3D laser scanner is mounted on top of the vehicle. Researcher Mikko Miettinen is driving the vehicle in the picture.

3.2.2 Dual 2D Scanners Pointing Sideways

The earlier version of the measurement system included two separate laser scanners. A laser scanner was placed on both sides of the measurement vehicle giving nearly 360 degrees of horizontal measurement data at the highest interpolation mode of ¼ degrees. In this mode four consecutive partial scans are merged into one full scan. The higher resolution is obtained at the expense of the scan rate. The resulting virtual scanner measured 180 degrees at separation of ¼ degrees forming a vector of 721 radius measurements. These measurements from the both laser scanners are visualized in Figure 4 using different colors.

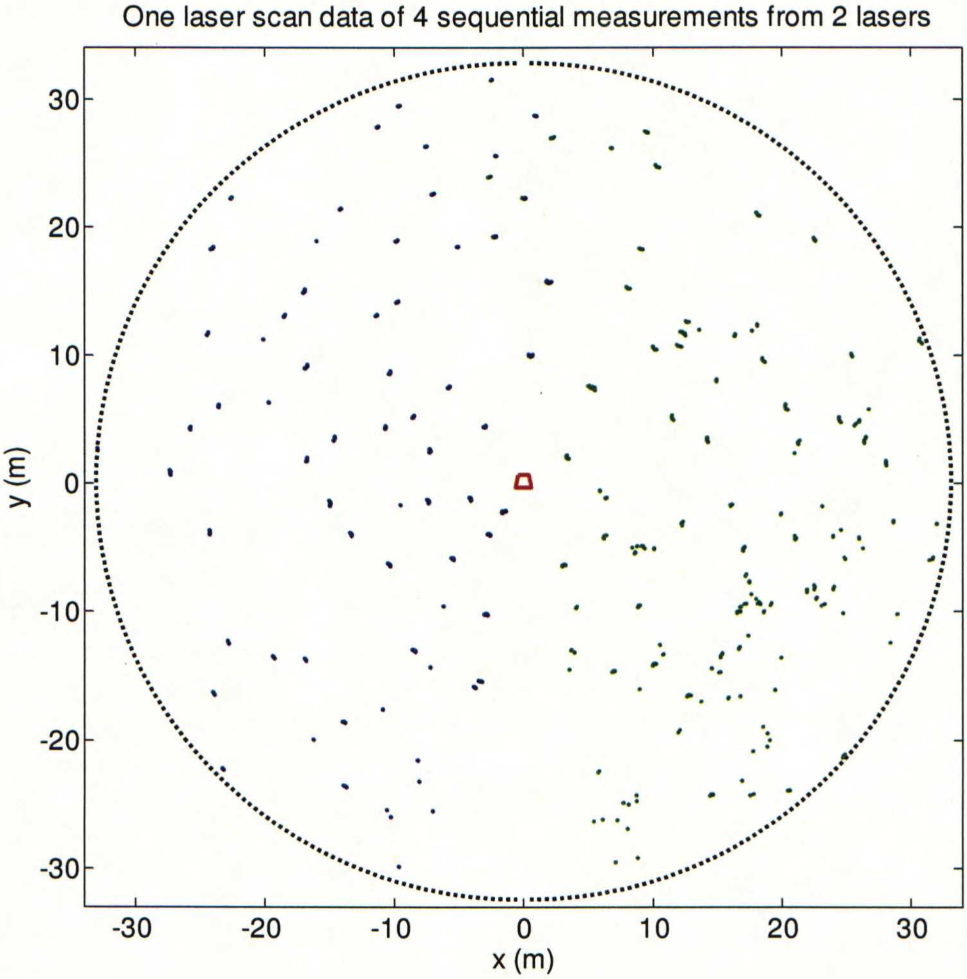

Figure 4: Visualization of the data from the first measurement system with two laser scanners. For each scanner there are 4 sequential laser measurements at ¼ degree resolution merged together. The dash lined circle symbolizes the maximum range of the laser scanners and the red marker in the center of the circle represents the measurement vehicle.

3.3 3D Laser Scanner

The 3D scanner is built from two separate Sick LMS200 laser scanners, which are technically very similar to other 2D scanners used in this thesis [20]. These 2D scanners are fixed to a rotating basis at a 45 degree angles as seen in Figure 3. The basis rotates around the main Z-axis at two selectable angular velocities, the faster is 39 RPM and the slower is 13 RPM. Thus the collection of a full 360 degree data will take a few seconds. It is a major challenge to combine the measurements from multiple scans to the same data set while the vehicle is moving and shaking. The data set is fixed using a pose and angle estimates during that time. Estimates are obtained from the IMU and laser odometry system, which are both explained later. Even small errors in these measurements will affect greatly the resulting accuracy of the measured trees and the generated forest map. Therefore it is important to get odometry and inertia information as accurately as possible.

The 3D laser scanner has been designed by researcher Matti Öhman in our laboratory. The device has an optical incremental encoder for the rotation, but the very first revision of the system, which was used to collect the data, did not have any absolute reference sensor to reset the value of the angle encoder. Therefore the calibration of the absolute angle and the correction of the errors in the optical encoder readings had to be calculated from the data of seen environment itself. Luckily the measurement vehicle has a unique shape which can be found from the collected 3D data set. Algorithms that are needed to re-orientate the acquired 3D data using the seen landmarks are skipped in this work as they would not be needed, if an absolute encoder was added in the 3D laser scanner.

The second problem with the 3D laser scanner is with timings. The data set is collected using a wireless link between the scanner system and the measurement computer. The wireless link has a random time delay that is not easy to predict. This delay is seen in the data when the measurement vehicle tilted fast, as it is not always synchronous with the measured inertial data. The IMU measurements are combined with the 3D scanner measurements to compensate any movement in the collected 3D data set and it does not work appropriately if the compensations are not exactly synchronized.

Another challenge with the delays is in the fact that the first version of the 3D scanner device did not have any synchronization between the two SICK laser scanning rangefinders. Therefore the data collected with the both scanners measuring at approximately 75 Hz frequency had a very small frequency jitter that slowly changed the phase of scanners' measuring cycle. Because the encoder measurement was added only to the range measurements collected with the first scanner and the angle of the second scanner was set to same as the angle of the nearest arrived encoder measurement, the second scanner's angle could have at maximum ±13ms error in the time of arrival. In the worst case this caused a rotational error of 2 degrees.

The second version of the rotating 3D laser scanner has synchronization between the two laser scanners and an absolute reference sensor. This version is used to estimate errors of the self-made 3D scanner. In the Appendix, Figure 46 shows the measured angular error between the full supposed loop of 4800 encoder ticks and the measured reference signal. This figure shows how uncertain cheap angular incremental and reference sensors are and that the incremental encoder did miss on average of 8 encoder ticks at every round. This corresponds to a bias angle of 0.6 degrees at every round against the direction of rotation if the nominal 4800 ticks were used.

3.4 Inertial Measurement Unit

The constant knowledge of the vehicle pose is vital for the 3D laser scanner to be able to measure and combine the data needed for the tree measurements. In the system setup, there is a MicroStrain 3DM-GX2 Inertia-Link Inertial Measurement Unit (IMU) [21] to measure the 3D inclination. It has a high update rate of a 250Hz and it gives the 3D data of angular pose and acceleration as well as axial acceleration. Its accuracy is sufficient for measuring current alignment of the measurement vehicle and to estimate the direction of gravity. These estimations are explained in Section 4.1.

3.5 GPS Receiver

A global position of the vehicle is needed for map building as the map is situated in the global coordinate frame. A GPS receiver from Ponsse's harvester is used to get the position measurements. Ponsse's receiver system is built from a U-Blox TIM-ST series GPS receiver using SiRFstar II chipset and Trimble's active GPS antenna designed for Ponsse. The receiver is a closed system, and the exact details are not available for us. The used receiver system with the active antenna works better than other GPS receivers in the forest although it has some inconvenient features. It for instance has quite long delay of approximately 8 seconds. The receiver system gives measurements in NMEA 0183 format [22] at a refresh rate of 1Hz.

3.6 Measurement Collection in Forest Environment

According to Finnish Forest Research Institute the four most common tree species in Finland are Scots pine (Pinus sylvestris) 65.5%, Norway spruce (Picea abies) 23.7%, Downy birch (Betula pubescens) 6.1%, and Silver birch (Betula pendula) 2.7% [23]. As the most common trees in Finland are pine and spruce it is reasonable to concentrate only on these two species.

The measurement sets are taken in the forest of Finnish Forest Research Institute, Metla [24]. The forest is growing mainly sparse pine tree, but there are denser areas with pines and spruces around. The terrain around the measuring area is diverse, with flat and hilly regions, so the measurements are taken at steep slopes and at flat regions.

Measurement runs for this work were taken as closed loops, where the start and end positions are near each other. While measuring at closed loops, the localization error can be estimated as the difference between the end and start locations in the generated map. Basic measurements were taken at easy sparse pine forests with pillar like trees and only little undergrowth. To test the capabilities of the algorithms, denser and tougher environments were also measured. However more test runs and study from various forests are still required, if these algorithms that are developed in this thesis are supposed to be used in a real measurement product.

Chapter 4

System Localization and Orientation

In the forest measurement system, it is important to know pose of the measurement vehicle. It is vital for map building and for measuring the trees. All tree measurements have to be associated to a specific identified tree in order to build a map of the trees. Combining measurements associated to a specific tree over a longer period of time increases the measurement accuracy as data is statistically analyzed and erroneous measurements are filtered out.

The first part of this chapter concentrates on the measurements of an inertial measurement device (IMU). It shows how orientation of the measuring device is calculated and how 3D orientation and acceleration can be used to improve the measurements collected by laser scanners. The second part is about finding the fastest and most accurate method to calculate pose change. Finally the third part presents a simple method to transform GPS coordinates to local coordinates.

4.1 Coordinate Systems and Transformations

The measurement vehicle has many separate measurement devices which all have their own native coordinate systems. These systems had to be measured and calibrated accurately to each other to allow the measurement system to function precisely. The different coordinate axes and fixed transformations are detailed in the next subsection. The 3D rotation of the measuring vehicle is calculated from IMU data in the second subsection. At last subsection, the direction of gravitation is estimated to minimize otherwise increasing rotation error in IMU data. In this thesis, every coordinate system has same scale and thus there is no scaling in transformations between coordinate systems.

4.1.1 Coordinate Systems of the Measurement vehicle

Nearly all coordinate axes in the system are headed such that y-direction is ahead; x is to the right and z upwards. The only exception is the inertial measurement unit, which has its native coordinate system based on "aircraft" formulation. [21] In this case the z-axis is pointing downwards and y backwards, because the IMU is fixed in that way to the measuring vehicle. All the coordinate systems, drawn in Figure 5, are right hand systems, and thus easily transformed to each other.

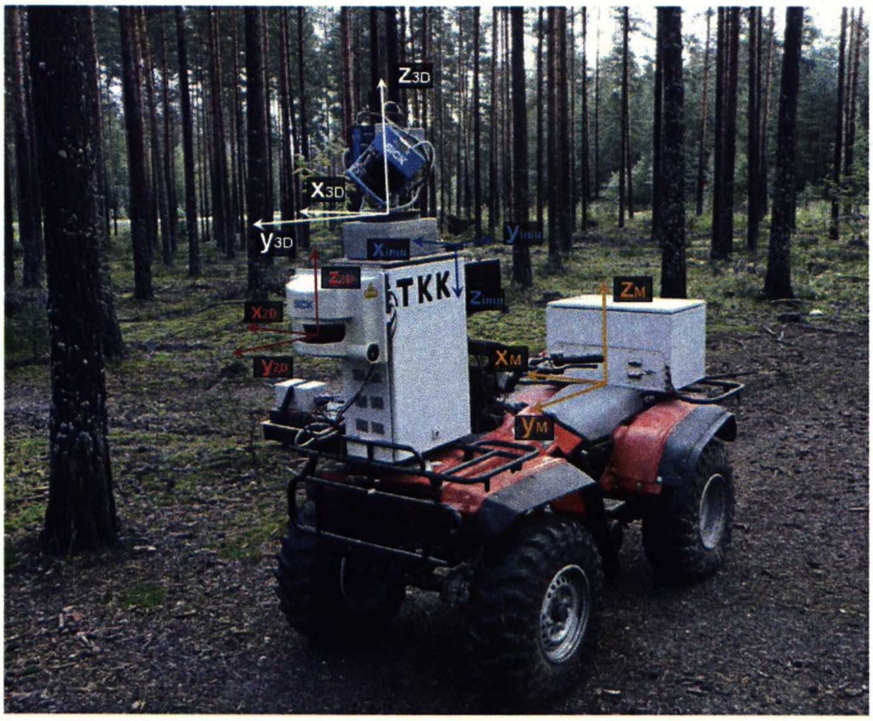

Figure 5: The all terrain vehicle with its relative coordinate systems is in the picture. Vehicle's main coordinates are drawn with yellow, IMU coordinates with blue, 3D scanner coordinates with white, and 2D scanner coordinates with red color. The y-axis is pointing to the heading direction in all other relative coordinates except IMU coordinates.

The vehicle's main coordinate frame, titled with subscript M in Figure 5, is placed above the center of the rear axis to minimize sideway translation in turns. According to Ackermann steering geometry, the centre of the turning circle is on the same axis as the rear wheels. [25] While the rotation origin is placed in the cross section of the rear axis and the y-axis pointing to the front, the driven path can be estimated using only rotation and y-directional translation. This simple origin replacement is used to simplify and increase the accuracy of the pose estimate.

The map coordinates are the same as vehicle's main coordinates at start. The vehicle's coordinates are at that time separated from the map coordinates as the vehicle's coordinates rotate and translate along the measuring vehicle. The z-axis is always pointing upwards against the estimated direction of gravitation in both of these two coordinate systems. The pose of the measuring vehicle is thus modeled only using two-dimensional translation and rotation.

The coordinates of 2D scanner are similar as the main coordinates of the vehicle; they are only moved towards the y-direction about 1.5 meters to meet the origin of the 2D scanner. The scanner is tilted to a small angle pointing upwards from its mounting bracket to calibrate it to the horizontal position when the vehicle is standing still. The used rotation is taken account while the transformation between 2D laser scanner and main coordinate system is calculated.

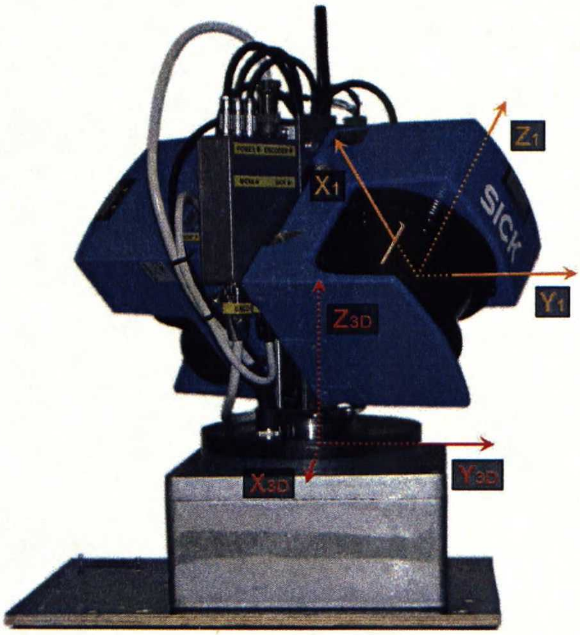

Figure 6: The coordinate system visualization for the 3D laser scanner consisting of two 2D scanners on a rotating basis. Coordinates of the first 2D scanner are drawn with yellow, and the main coordinates of the 3D scanner are drawn with red color. The coordinates of the first and second scanner are similar except they are rotated 180 degrees around the z-axis of the main coordinate system.

4.1.2 Orientation Estimate Using Inertial Measurements

The equations needed to estimate the vehicle pose are described in this section. The rotation matrix **M** can be presented in Euler angles [26] using (1) [21], where *s*( ) symbolizes sine and *c*( ) cosine function. In the equation, ψ_e symbolizes yaw, θ_e pitch, and φ_e roll. The equation is calculated using the "aircraft" formulation. [21]

$$\mathbf{M}=\begin{bmatrix} c(\psi_e)c(\theta_e) & s(\psi_e)c(\theta_e) & -s(\theta_e) \\ c(\psi_e)s(\theta_e)s(\varphi_e)-s(\psi_e)c(\varphi_e) & s(\psi_e)s(\theta_e)s(\varphi_e)+c(\psi_e)c(\varphi_e) & c(\theta_e)s(\varphi_e) \\ c(\psi_e)s(\theta_e)c(\varphi_e)+s(\psi_e)s(\varphi_e) & s(\psi_e)s(\theta_e)c(\varphi_e)-c(\psi_e)s(\varphi_e) & c(\theta_e)c(\varphi_e) \end{bmatrix} \tag{1}$$

Yaw ψ, pitch θ, and roll φ are calculated from rotation matrix **M**, using (2), (3) and (4) [21]. All of the listed equations are derived from (1).

$$\psi_e = \arctan\left(\frac{\mathbf{M}_{(1,2)}}{\mathbf{M}_{(1,1)}}\right) \tag{2}$$

$$\theta_e = \arctan\left(\frac{\mathbf{M}_{(2,3)}}{\mathbf{M}_{(3,3)}}\right) \tag{3}$$

$$\varphi_e = \arcsin\left(-\mathbf{M}_{(1,3)}\right) \tag{4}$$

The relation between the IMU coordinates and the global coordinates of IMU are presented in (5) [21]. In the equation $\mathbf{v}_{Link}$ is a vector in the IMU coordinates; $\mathbf{v}_{LinkGlobal}$ is the same vector in the global IMU coordinates. The $\mathbf{M}_{LE}$ is the rotation matrix that was got from the IMU as a measurement of its own rotation from the global IMU coordinates. The global IMU coordinates were calibrated before taking measurements while the system was standing still.

$$\mathbf{v}_{Link} = \mathbf{M}_{LE}\mathbf{v}_{LinkGlobal} \tag{5}$$

The IMU coordinates are different than coordinates used in the measurement vehicle. The IMU coordinates are transformed to the coordinates of the measurement vehicle using (6). In the equation $\mathbf{v}_{Link}$ is a vector in the IMU coordinates and $\mathbf{v}_{Vehicle}$ is the same vector in the coordinates of the measurement vehicle. The $\mathbf{M}_{VL}$ is the rotation matrix, constructed using (1) with measured Euler angles between coordinate frames.

$$\mathbf{v}_{Vehicle} = \mathbf{M}_{VL}\mathbf{v}_{Link} \tag{6}$$

The global IMU coordinates are similarly rotated to the vehicle's global coordinates as the IMU coordinates to the vehicle's coordinates before. That is why in (7), global IMU coordinates are rotated to vehicle's global coordinates using the same rotation matrix as in (6). In (7) $\mathbf{v}_{LinkGlobal}$ is a vector in the global IMU coordinates and $\mathbf{v}_{Global}$ is the same vector in the global coordinates seen from the measurement vehicle.

$$\mathbf{v}_{Global} = \mathbf{M}_{VL}\mathbf{v}_{LinkGlobal} \tag{7}$$

The previous equations (5), (6), and (7) are combined in (8), where the vectors in vehicle's global coordinates are rotated to the coordinates of the measurement vehicle using a rotation matrix measuring the rotation between the calibrated global and the rotated IMU coordinates. The rotation matrix, **M** in (9), is a product of the rotation matrices in (8). The original rotation matrix measured with IMU, $\mathbf{M}_{LE}$, is calculated to a rotation matrix giving the rotation of the measurement vehicle, **M**.

$$\mathbf{v}_{Vehicle} = \mathbf{M}_{VL}\mathbf{M}_{LE}\mathbf{M}_{VL}^{T}\mathbf{v}_{Global} \tag{8}$$

$$\mathbf{M} = \mathbf{M}_{VL}\mathbf{M}_{LE}\mathbf{M}_{VL}^{T} \tag{9}$$

4.1.3 Estimation of the Direction of Gravitation

The direction of gravitation is estimated as a long term average of the measured accelerations which are rotated to the global IMU coordinates using (5). The direction of the gravitation is calculated from the collected long term average of accelerations using (10) for the pitch and (11) for the roll. In these equations, $\boldsymbol{a}$ is the three-dimensional long term average acceleration with Cartesian components a_x, a_y, and a_z.

$$\theta_{e,G} = \arctan\left(\frac{a_x}{a_z}\right) \tag{10}$$

$$\varphi_{e,G} = \arctan\left(-\frac{a_y}{a_z}\right) \tag{11}$$

These pitch and roll angles of gravitation are first calculated to a rotation matrix using (1). The vector pointing to the direction of gravitation is then rotated back to the IMU coordinates using (5), as the calculated rotation matrix is in the global IMU coordinates. Finally the vector was rotated to the coordinates of the measurement vehicle using (6). Euler angles [26] for the direction of gravitation and the pose of the measurement vehicle are then calculated using (2), (3), and (4). Pitch and roll angles of the estimated direction of gravitation are then subtracted from these Euler angles to rotate the direction of gravitation to point downwards in the final rotation matrix. The last step was to calculate the final rotation matrix for the vehicle pose from the resulting yaw, roll and pitch angles using (1).

4.2 Pose Tracking

This section focuses on estimating the changes in the pose of the measurement system. In this section different laser odometry strategies are developed and compared to find out an accurate and still fast enough solution to find the pose change of the measuring device in forest environment. The pose estimation is a vital part of the SLAM process where the vehicle is simultaneously localized and the forest is mapped.

4.2.1 Correlation Based Laser Odometry

The correlation based laser odometry is a method where consecutive laser scans are compared to each other to reveal a pose change. A version of the method is used in a paper from Gerhard Weiß, Christopher Wetzler and Ewald von Puttkamer which focuses on indoor robot localization and orientation [6]. They have successfully used angle correlation between old and new measurements when there is only rotation and no translation. For more complex cases they have introduced an angle histogram, which was a histogram of the angles between the adjacent measured points. These histograms are correlated to the previous histogram to find the rotation. In forests, with only roughly round trees, the angle histogram does not find any main directions and thus cannot be used. Correlation of the raw angular data, which is also referred by Weiß et al., proved more useful. Annoyingly it only works when the translation is small.

A new way to represent the space around the measuring device is needed to successfully compare old and new measurements in a forest. Previously presented algorithms have used histogram based methods [6], [7], [8], or iterative and computationally expensive ICP methods [11] for estimating pose change. The problem is solved using minimum filtered projections to replace histograms. Angle histogram is not used at all. The rotation is estimated by cross-correlating compensated scans together in polar coordinates.

Estimated translation is compensated in data, when the rotation is estimated. Similarly rotation is compensated while the translation is estimated. Compensations are needed as the pose change consisted of both rotation and translation. Assuming no translation, the rotation estimation can be straightforwardly calculated using a cross-correlation between subsequent scans. As there normally is translation and rotation, the translation is first estimated and then reduced from the scan by simulating a new scan without the previously estimated translation. After the rotation is estimated, the result is used to simulate a new scan with only a translation in the pose change. This allows the translation to be estimated without significant interference from rotation. These phases are iterated three times to get more accurate translation and rotation estimates.

The combined FOV of the two laser scanners is nearly symmetric in every direction. The effect of the translation to rotation and vice versa can be minimized by calculating rotation and translation over both sides separately and averaging the results. This averaging together with the iterative compensation process makes it possible to estimate the rotation and translation more accurately as the measurement vehicle is moving in the forest.

The translation is usually estimated by cross-correlating histograms, like Weiß et al. [6]. The histogram method fails in the forest as there are no straight walls or large scale objects that would be seen in the x- or y-directional histograms. Histograms are replaced with projections calculated using a self developed minimum distance projection method. This method creates histogram type curve that can be cross-correlated with previously obtained one.

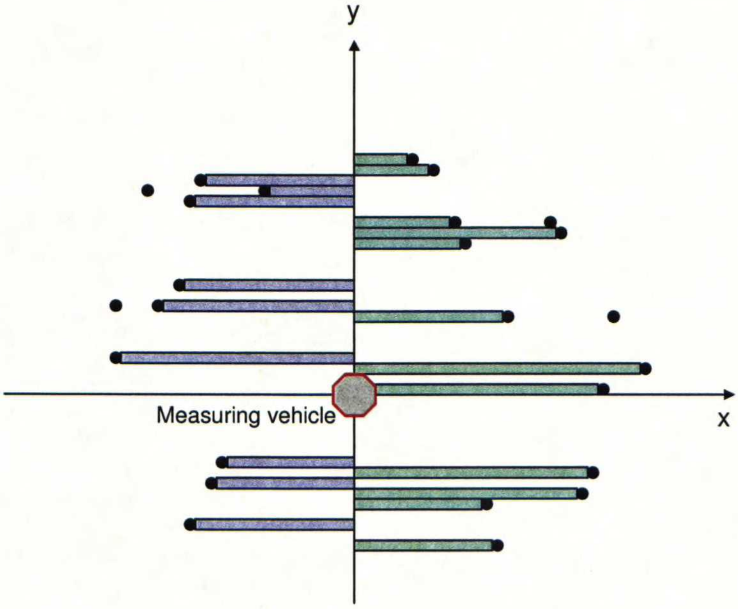

Figure 7: A visualization of the method of minimum distance calculation to reveal y-axis directional movement. The measurement points are discretized using a dense occupancy grid and the distance from the axis is measured to the nearest measurement point. After low-pass filtering, the resulting graph can be correlated with the previous one.

The minimum distance projection method is visualized in Figure 7. In this method, projections are done by first calculating dense occupancy grid from the measured points. The grid is then split into two pieces on both sides of the current axis. A minimum filter perpendicular to the current axis is applied to both pieces of the grid. The filter gives a distance to the nearest occupied grid point from the current axis. The operation results an *N*-length vector of distances from the current axis to the nearest measured point for the both divided pieces. *N* represents a number of occupancy grid cells in a row. Finally these vectors are low-pass filtered using a non-causal symmetric convolution filter to minimize the high frequency errors generated by the dense grid of only a few nonzero points.

Figure 7 shows how the method is used to reveal the y-directional projection. The x-directional projection can be generated similarly, only in the x-axis direction. In that case the x-axis is the current axis and the projections are done separately from the halves in front and behind the measurement vehicle. The cross-correlation can be calculated from these projections, but normal correlation gives only the results at the resolution that is used in the occupancy grid. More detailed correlation result can be acquired, if the whole correlation curve can be used in the calculations to gain better fit.

The cross-correlations are calculated using a real-valued discrete correlation shown in (13), which is derived from (12). Equation (12) is a continuous time version of a cross correlation equation, where *f* and *g* are cross correlated signals at time difference *t*. [27]

$$(f * g)(t) = \int_{-\infty}^{\infty} \overline{f}(\tau) g(t+\tau) d\tau \tag{12}$$

$$C_{fg}(n) = \sum_{m=-\infty}^{\infty} f(m) g(n+m) \tag{13}$$

In (13) *f* denotes the previous and *g* denotes the current filtered vector of the correlated inputs. In the equation, *m* is the correlation index and *n* is the discrete bias between the correlated signals.

The more detailed cross-correlation result is acquired using a center of mass of the correlation curve to point out distance of the best correlation. This method calculates the center of mass of the selected range of the cross-correlation graph. It allows the correlation result to be more accurate, than if only the largest correlation result would be selected. This method is called the

CM method, as it weights the correlation result by calculating the location of the center of mass. The CM method is explained in more detailed in equations (14) and (15).

The CM method is used in (14), where n is the correlation distance and A_y is the selected range of the correlation distances around the location of maximum correlation. C_{fg} is a cross-correlation calculated using (13). The center of mass of the correlation graph was then normalized with a sampling constant calculated from the length of the occupancy grid, L_y, divided by the corresponding number of slots in the grid, N.

$$\Delta y = \frac{L_y}{N} \frac{\sum_{n \in A_y} n C_{fg}(n)}{\sum_{n \in A_y} C_{fg}(n)} \tag{14}$$

Finally the resulting y-directional movement is calculated as an average of the correlation results from both sides of the current axis to minimize the interference of rotation to the translation. The phases of the presented y-directional cross-correlation are demonstrated in Appendix Figure **9** to visualize this quite complex procedure. The movement in x-direction is calculated similarly, only using x-axis as the current axis. Both x and y translations are calculated separately.

The estimated translation is used to simulate a new scan in which the previously estimated translation is compensated. The angle correlation cannot be calculated directly from the modified data set because the angles between measured points drift as the points were translated. This problem is solved by re-sampling the translated points using an interpolation algorithm. First the algorithm generates a new N_θ-length zero-valued vector of radii at discrete angular steps. Then a new vector is filled with the original radii according to the angular slot where the original angle belongs. After re-sampling, the data set could be cross-correlated with the previously calculated one.

$$\Delta \theta = \frac{\theta_{tot}}{N_\theta} \frac{\sum_{n \in A} n C_{fg}(n)}{\sum_{n \in A} C_{fg}(n)} \tag{15}$$

The rotation component is estimated using (15). Similarly as before, the method increases the correlation accuracy by using the previously introduced CM method. In the equation n is the correlation distance and A is the selected range of correlation distances. C_{fg} is the cross-correlation calculated using (13). The center of mass is normalized with the total angle of measurements θ_{tot} divided by N_{θ}, which is the amount of discrete angular steps in the correlated data.

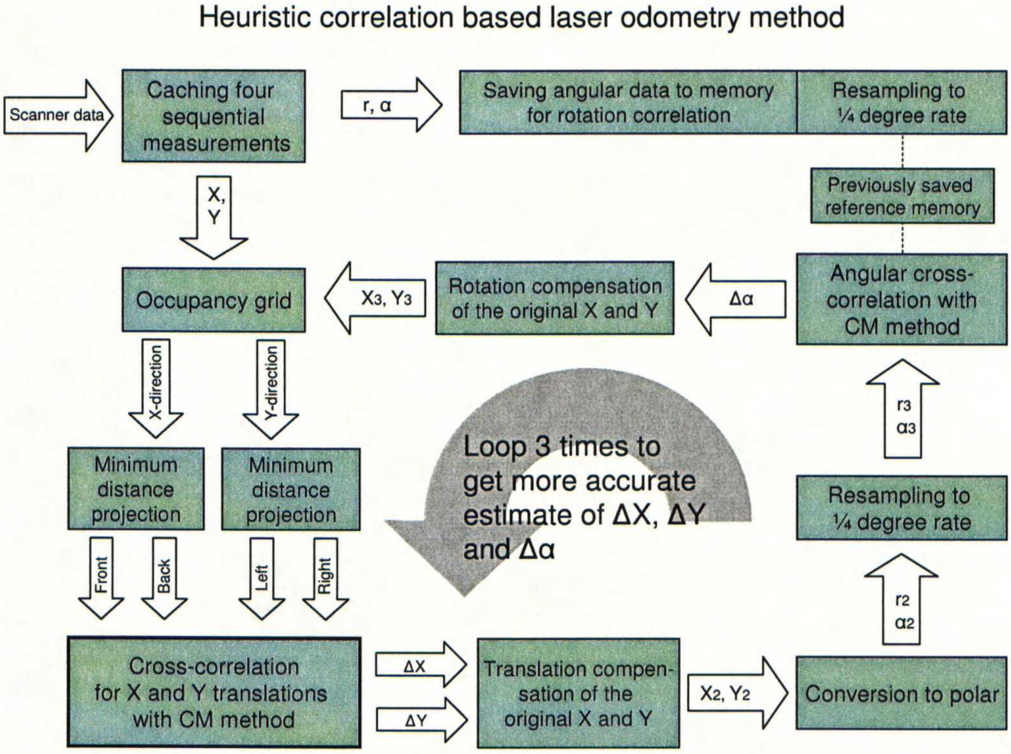
Heuristic correlation based laser odometry method
Scanner data
Caching four sequential measurements
r, α
Saving angular data to memory for rotation correlation
Resampling to ¼ degree rate
Previously saved reference memory
X, Y
Occupancy grid
X3, Y3
Rotation compensation of the original X and Y
Δα
Angular cross-correlation with CM method
X-direction
Y-direction
r3 α3
Minimum distance projection
Minimum distance projection
Loop 3 times to get more accurate estimate of ΔX, ΔY and Δα
Resampling to ¼ degree rate
Front
Back
Left
Right
r2 α2
Cross-correlation for X and Y translations with CM method
ΔX
ΔY
Translation compensation of the original X and Y
X2, Y2
Conversion to polar

Figure 8: The chart of a laser odometry calculation loop, where in a few iterative steps the best estimations of ΔX, ΔY and Δα are found. The two of the four different correlation calculations for translation are visualized in Figure 9.

A similar rotation is calculated for both laser scanners separately. The calculated results of the both scanners are averaged to reduce the effect of translation to rotation. The estimated angle is then used to compensate rotation from the measurements using the previously presented compensation method and then the translation was estimated again. This cycle is iterated three times to get an accurate estimate for the rotation and translation components. The whole calculation cycle is shown in chart in Figure 8. A part of the method where the y-directional translations is calculated is shown in chart in Figure 9.

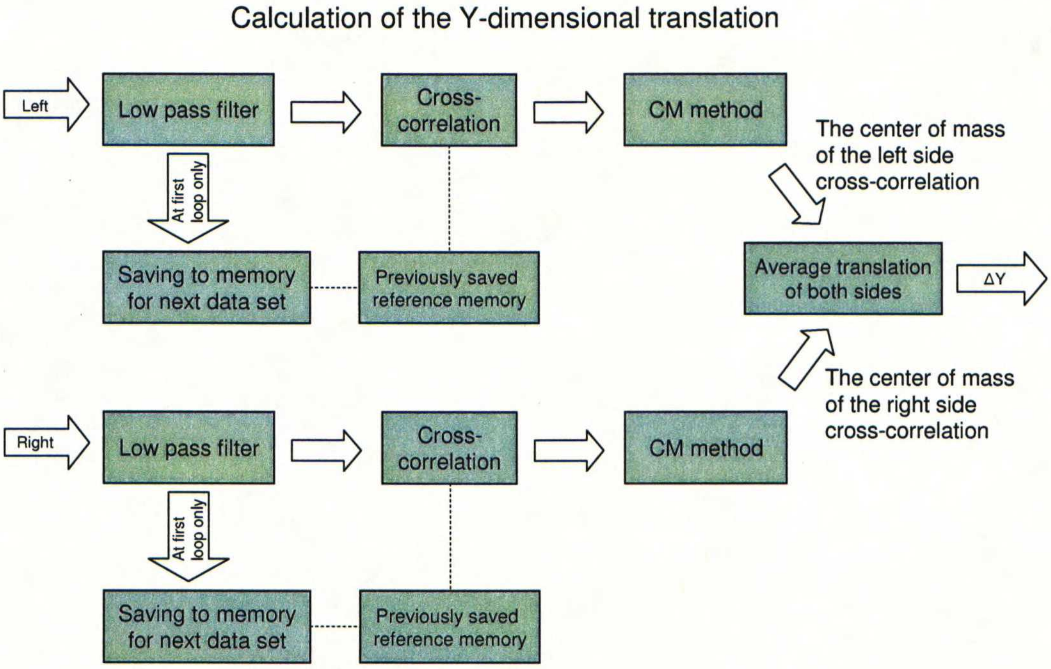
Calculation of the Y-dimensional translation
Left
Low pass filter
At first loop only
Saving to memory for next data set
Cross-correlation
Previously saved reference memory
CM method
The center of mass of the left side cross-correlation
Average translation of both sides
ΔY
The center of mass of the right side cross-correlation
Right
Low pass filter
At first loop only
Saving to memory for next data set
Cross-correlation
Previously saved reference memory
CM method

Figure 9: The chart of the correlation calculations for the y-directional translation. An x-directional translation is calculated in analogous fashion. These all correlations are calculated in the largest chart element in Figure 8.

4.2.2 Iterative Closest Point Method for Laser Odometry

The iterative closest point (ICP) method described in the article of "A Method for Registration of 3-D Shapes" from Paul J. Besl, and Neil D. McKay [28] is tested as it is widely used method for acquiring the difference of the pose. The iterative closest point method consists of a few iteration steps that are:

1) Associate points by the nearest neighbor criteria
2) Estimate the parameters using a mean square cost function
3) Transform the points using the estimated parameters
4) Terminate the iteration when the change in the mean square error falls below a preset threshold specifying the desired precision of the parameter estimation

Calculations are done using the same scan data as in the next section 4.2.3. In the data set there was only one Sick LMS221 scanner pointing ahead. Four partial scans are merged together forming a 721 point complete scan at ¼ degree resolution. This dataset is compared to the previously obtained data set. The distances between points are determined simply by calculating all distances between previous and current data set points. The nearest points were associated as pairs.

The pose change between these pairs is estimated by minimizing the sum of distances between the paired points. This minimization is done by using an optimal method of absolute orientation using unit quaternions by Berthold K. P. Horn [29]. Newly measured points are then transformed using an estimated pose change. After that a new iteration cycle is started with a search of the nearest points between the previous and modified current data sets. The number of needed iterations to achieve accurate fit varies much and therefore it has to be limited even though it decreases the accuracy of the method.

4.2.3 Feature Based Laser Odometry with Inertial Measurement

Since the previously introduced ICP method for full scan data described in Section 4.2.2, is too slow to iterate enough to meet the required accuracy, the amount of compared points has to be decreased. One method is to extract some features from the scanner data and compare the features instead of all measured points.

Because of the goal in mapping and measuring trees, all 2-dimensional laser scanner data has to be filtered to find tree edges. That is why the identified trees can be easily and efficiently used as traceable features in this case. Initially the tree edges are filtered out using the edge detection filter and then the points are logically separated to groups of points between the left and right edges as described later in the section 5.1, which shows the tree recognition process. The simplest way to calculate a reference point for a group of points representing a tree between left and right edges is to determine the center of mass of those points. It can be effectively used as a feature to represent a tree when a pose change between consecutive scans remains quite small.

These features are not comparable, if the same tree is seen from a very different angle because the measurement points hit the tree on different sides and therefore the calculated feature is not the same. Fortunately translation and rotation were only small between compared feature sets when only the sequential measurements are used. This limitation has to be taken into account when center of mass method is used to reduce the amount of compared points.

To reduce this effect, a center point of a fitted circle, representing a tree trunk, has been tried as a feature as well. However the center of mass method proved to be more robust than the circle fitting method, because the fitted circle is not accurate when the measurements are noisy and there are some outliers. For instance a miss of one measurement point of a tree only changes the center of mass a little, when the same erroneous point will change the position of the circle a lot more.

In this laser odometry method rotation was directly acquired from the IMU introduced in 3.4. The rotation can be measured more accurately using inertial measurements described in Section 4.1.2 than by using only scanner data, and therefore the translation-rotation problem can be reduced to a translation problem. The remaining translation can be simply calculated as the difference between paired features. The pair association is done similarly as pairing in the next section, 4.2.4. The translation estimate was filtered out of the noisy set of translations between consecutive paired features using a median filter for the x and y axes separately. The previously described feature based laser odometry method is visualized in a chart drawn in Appendix Figure 45.

4.2.4 Map Based Pose Correction

The pose of the measurement vehicle can be estimated more accurately, if the estimate is fine-tuned using the knowledge of tree locations from the map, which is constantly being updated. The map is built from the trees found from the laser scanner data using the methods described later in Chapter 6 and the trees are found using the methods described in Chapter 5.

The pose estimate is fine-tuned by fitting found trees to the map. At first, the found trees are paired with the nearest trees in the map. The pairing is done using a simple but effective trick called mutual consistency check [30]. In the pairing method, first the nearest mapped tree is paired for all found trees. Then the same procedure is done reversed for all the mapped trees that are close enough the found trees. The final pairs, 'the married couples' are the trees that have both selected each other. The method is visualized in Figure 10 in three steps.

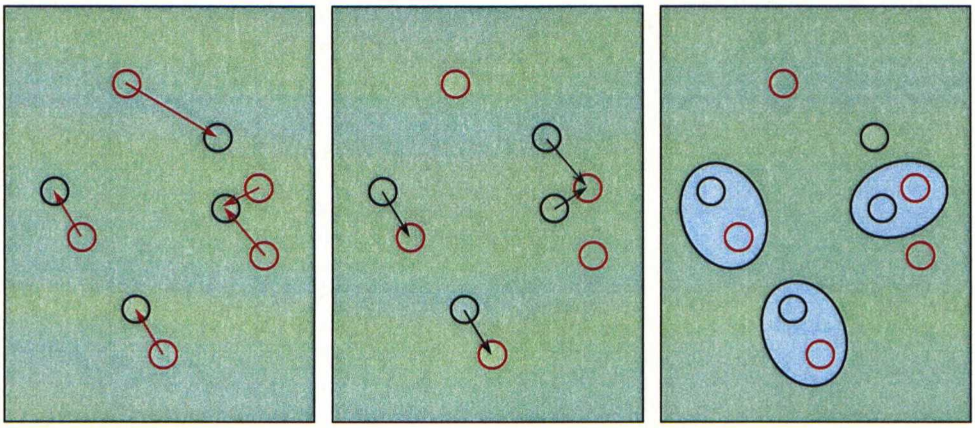

Figure 10: The visualization of mutual consistency check of the mapped and found trees. The found trees are marked with red and mapped trees with black color. In the left subfigure nearest trees are found for all the found trees. In the center nearest trees are found for all the mapped trees. The resulting mutually consistent pairs are shown in the right.

The pose error between the odometry estimate and the map is minimized as a sum of biased errors between the married tree pairs. The error is biased with the amount of measurements focused to the each mapped tree. Finally the pose of the measurement vehicle and the locations of found trees are moved to the new adjusted location. The rotation is estimated as a median error angle between the paired trees in the polar coordinates, where origin is placed on the measurement vehicle. A median is used instead of linear minimization, because the median filters out errors generated by incorrectly married tree pairs and it is thus more robust method.

4.3 Global Localization

At first it seems like the GPS data can be easily used to improve the accuracy of the SLAM process although localization errors in the GPS data seem to be quite large compared to the localization errors in the SLAM process. However GPS measurements are delayed in Ponsse's GPS receiver as it has some type of filtering delaying measurements approximately 8 seconds. Such a large delay prevented the use of the GPS data aiding the SLAM process.

Instead of using the GPS data to improve the accuracy of the SLAM process, it is used only to fix the map to the global coordinate system. It is done by pairing 8 seconds old points from the travelled path of the map with the GPS path. These pairs are placed to a list of a thousand sequential measurement pairs. The GPS coordinates are transformed to planar coordinates using the location of the starting point as a reference where other GPS measurements are compared using a method that is described in Section 4.3.1.

After transformation the paired xy-coordinates from the map and the GPS measurements can be compared with each other. A sum of distances between the paired points is then minimized resulting an estimated translation and rotation between the GPS path and the map path. The calculations are done the method of absolute orientation using unit quaternions by Berthold K. P. Horn [29] similarly as in some other minimization tasks in this work.

4.3.1 GPS Coordinate Projection to Rectangular Coordinates

In order to compare the GPS path and the map path generated by the SLAM process, a projection from WGS84 coordinates to a plane is needed. The simplest and still accurate enough projection is a linearization the spherical coordinates around the selected reference point. The magnitude of the error is shown in the end of this section. The projection method is quite similar to the sinusoidal projection, as if the x-coordinate is measured as longitude from the reference point instead of Greenwich, and the y-coordinate is measured as latitude from the reference point instead of the equator. This method is visualized in Figure 11, where the changes of longitude and latitude are transformed to x representing the distance to East and y representing the distance to North from the reference point.

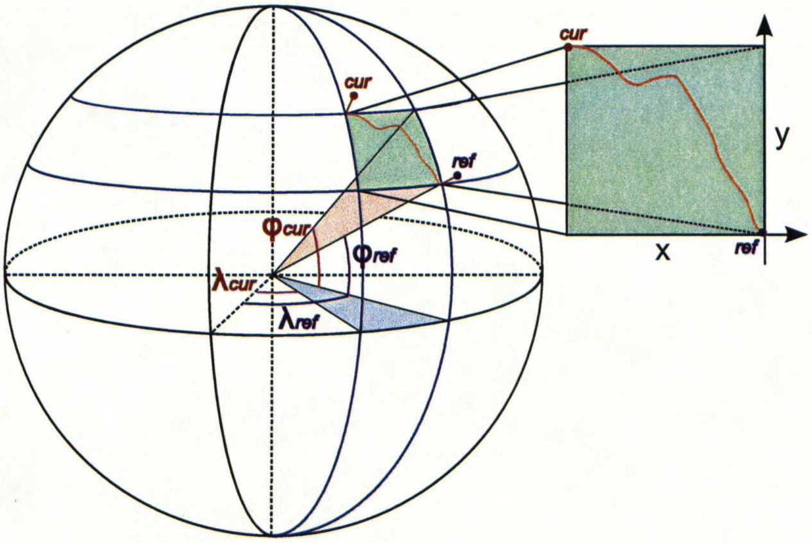

Figure 11: The model of the coordinate transform from the WGS84 coordinates to the xy-coordinates using modified sinusoidal projection. In the figure φ is the latitude and λ is the longitude. The subscript *ref* is for reference point and *cur* is for the current location of the measurement vehicle.

The projection from the GPS coordinates to the plane is done using (16) and (17), and it is shown in Figure 11. In the method x is calculated as the longitudinal difference between the current and the reference points scaled with Earth's radius at the latitude of reference point. Similarly y is calculated as the latitudinal difference scaled with Earth's radius, R. In (16) and (17), φ_{ref} is the latitude and λ_{ref} is the longitude of the reference point, and φ_{cur} is the latitude and λ_{cur} is the longitude of the measurement vehicle.

$$x = (\lambda_{cur} - \lambda_{ref})R(\varphi_{ref})\cos(\varphi_{ref}) \tag{16}$$

$$y = (\varphi_{cur} - \varphi_{ref})R(\varphi_{ref}) \tag{17}$$

Earth's radius is calculated using (18), where a is the semi-major axis, i.e. Earth's equatorial radius, and b is the semi-minor axis, i.e. Earth's polar radius, and φ is latitude of the reference point. The method is based on the ellipsoidal model of the WGS84 coordinate system. [31]

$$R(\varphi) = \sqrt{\frac{\left(a^2 \cos(\varphi)\right)^2 + \left(b^2 \sin(\varphi)\right)^2}{\left(a \cos(\varphi)\right)^2 + \left(b \sin(\varphi)\right)^2}} \qquad (18)$$

This simple model linearizes angular coordinates around the reference point and thus there is a potential error source. The magnitude of the errors is calculated using a great circle distance [32] as a reference distance compared to the Euclidean distance calculated from the projected coordinates, *x* and *y*. While the compared coordinates are at a distance of approximately 400 meters, the maximum error of the coordinate transform is approximately 1cm. The measurement error of the GPS receiver is usually a lot more and thus the error of this projection method can be neglected.

Earth's radius estimation is more significant error source. The maximum and minimum Earth radii used in the WGS84 system are a = 6378137.0m and b = 6356752.3m. To show the magnitude of error an example was calculated using an estimated average size of a forest map of 400m. A great circle distance between the same coordinates at the distance of 400m was calculated using minimum and maximum radiuses. The distance between the same coordinates at the maximum Earth's radius was 1.3m longer than while using the Earth's minimum radius. To minimize the error caused by erroneous Earth's radius, the radius for the reference point is calculated using (18). [31] To increase the accuracy of the method it might be reasonable in the future versions to use the measured GPS height and add it to the Earth's radius estimate.

Chapter 5

Laser Scanner Data Analysis

One important aspect of the forest mapping is processing the measurement data produced by the laser scanners. The first part of this chapter is quite simple as it is concentrating on searching and measuring trees from the horizontal laser scanner. Later part is more complicated as it is focusing to the methods needed for extracting and measuring trees from the data produced by the rotating 3D laser scanner.

5.1 2D Scanner Data Analysis

The 2D laser scanner data analysis is employed only shortly in this work because it is discussed in more detailed in our laboratory's previous studies, e.g. Jaakko Jutila's master's thesis about the localization and tree measurement with a laser scanner in a forestry machine perception system [9]. The same work is documented also in an article from Jaakko Jutila, Kosti Kannas, and Arto Visala [33]. Similar methods are used in this work, and methods are described only shortly.

The Sick LMS221 scanning laser rangefinder gives measurements at 1 degree resolution [20]. In this system double interlacing mode is used and four partial scans are merged together to form 721 point data set at an angular resolution of ¼ degrees. The trees are found by detecting the edges of the trees from the merged data set. The edge detection is discussed in the next section. The measurement points between the detected tree edges are associated to the same tree object. A circle is fitted to represent a tree trunk using the methods described in Section 5.1.2. A diameter is measured and a tree location is estimated using circle fitting. Trees are abstracted as circles for the mapping process which is described in Chapter 6.

5.1.1 Tree Search and Edge Detection

The previous studies have used discrete derivative to filter out the edges of trees from the laser scanner measurements [9] [14]. It is not used in this study as the filter is not symmetric in both directions and it is sensitive to noise in measurement data. Instead of the first derivative, the second derivative is used since it is symmetric and more robust edge detector. Because of its properties, this filter automatically removes trees that have less than three measurements.

The edges are detected using an absolute value of the discrete second derivative edge detection filter presented by William Pratt [34]. It is modified for angular data in (19). The measurements at all angles are present at filtering time so the filter does not have to be causal. In (19) θ is an angle of measured radius r, and δ is an angular sampling constant of ¼ degrees.

$$F_r(\theta) = \left| r(\theta - \delta) - 2r(\theta) + r(\theta + \delta) \right| \tag{19}$$

The tree search and edge detection algorithm is visualized in Figure 12. The polar data shown in the left subfigure is limited using a maximum value r_{limit}, which is smaller than the maximum range of the measurements. The data set and the limit are shown in the middle subfigure, where the radius r, is shown as a function of the angle θ. The measurements that are smaller than the limit are colored with brown. The whole data set is filtered using the edge detection filter F_r, shown in (19), and the filtered result is illustrated in the right part of the figure as the function of the angle.

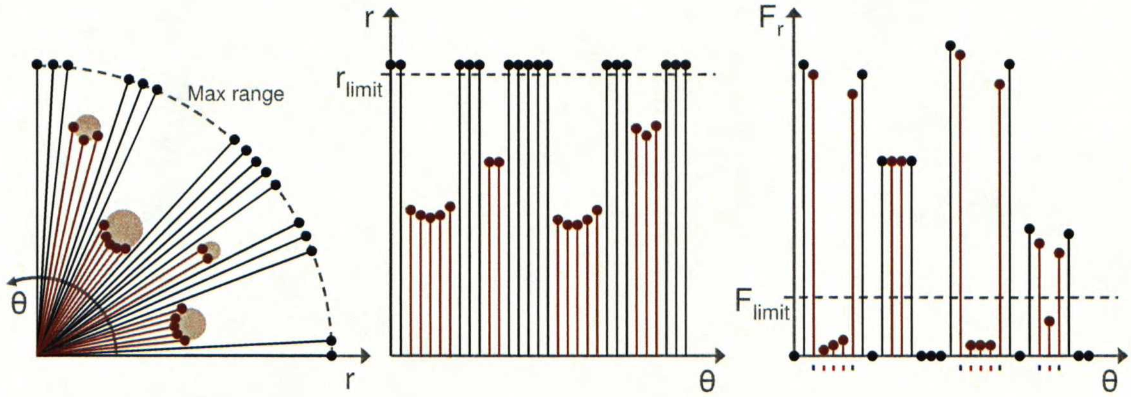

Figure 12: Tree search and edge detection from the polar laser scanner data. The left subfigure shows the laser range measurements with some trees. The middle subfigure shows measured radius as a function of the angle and the right subfigure shows the edge-filtered radius as a function of the angle. Brown color indicates that the radius is smaller than the limit value. Blue and green marks represent the right and left edges of a tree leaving the red center points in-between.

The trees are found using a maximum limit value for the filtered radiuses, F_{limit}. Measured points at certain discrete angles are selected to represent trees if the radii are smaller than r_{limit} and the filtered radii are smaller than F_{limit}. These measurement points are marked with red color in the right part of Figure 12. All the adjacent red points are linked to the same tree just like the edge points on both sides. These edge points are drawn with green and blue markers in the figure.

5.1.2 Measuring Tree Diameter and Location

The two dimensional tree measurements are done in a similar manner as in Jaakko Jutila's master's thesis [9]. He shows in his work that the method, where a tree is estimated as a circle fitted inside a sector of the selected laser scanner measurements, is the most accurate method to fit a circle with only a few measured points. The diameter of the estimated circle is always smaller than the cross-section of the original tree. Jutila shows that the method is a lot more accurate than least squares circle fitting techniques [9]. The used method is visualized in Figure 13, where the original tree is shown with a solid circle, the laser scanner measurements are shown as brown dots, and the measured edges of a tree are marked with green lines.

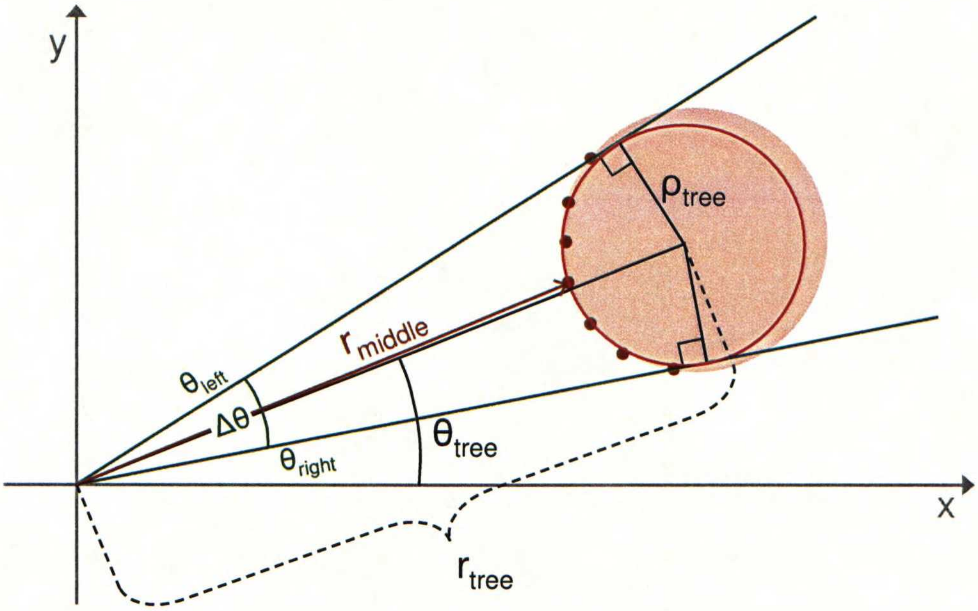

Figure 13: The visualization of the tree measuring algorithm, where the brown points are the laser scanner measurements from the tree plotted with the solid circle, green lines are the filtered tree edges, and the red circle is the estimated tree fitted using (20), (21), and (22).

The tree diameter is estimated using Jutila's circle fitting technique, and thus the tree radius ρ_{tree} can be estimated using (20) [9]. In (20) and in Figure 13, $\Delta\theta$ is the angular difference of edges, θ_{left} and θ_{right}, and r_{middle} is a measured range to the measured middle point of the tree. If there is an even amount of measured points, range of the middle point is calculated as an average of ranges of the innermost points.

$$\rho_{tree} = r_{middle} \frac{\sin(\Delta\theta / 2)}{1 - \sin(\Delta\theta / 2)} \tag{20}$$

The location of a fitted circle can be estimated using (21) and (22). In equations, r_{tree} and θ_{tree} are polar coordinates of a measured tree. The range is calculated as a sum of the estimated tree radius ρ_{tree} and the range to the middle point. The angle is calculated as an angular average of the tree edges, θ_{left} and θ_{right}.

$$r_{tree} = r_{middle} + \rho_{tree} \tag{21}$$

$$\theta_{tree} = \frac{\theta_{left} + \theta_{right}}{2} \tag{22}$$

The method is improved by calculating the xyz-location for every measurement point in the global vehicle coordinates using the rotation method described in Section 4.1.2. After reorientation measurement points are stated in a vertical cylindrical coordinate system, and the angles and ranges of the measured points are used similarly as original polar data from the laser scanner. This improvement reduces the distortion that is caused as the laser scanner sways along the measuring vehicle. Before this improvement the rotation of the laser scanner was affecting to the measured properties of the trees.

5.2 3D Scanner Data Analysis

Three dimensional tree filtering is a lot more complicated than the two dimensional version described earlier in this chapter. The data set from the surrounding environment is collected using two similar 2D scanners, which are rotating around at 45° angles to collect measurements at different directions acquiring a horizontal field of view of 360° and vertical FOV of 90°. The data rate of a single laser scanner is the same in both 2D and 3D setups.

Large amount of measurements had to be collected together to get enough information about surroundings. Trees have to be searched from the set of collected points and all the other objects and parts of trees except the tree trunk have to be filtered out. Trunks have to be measured from noisy filtered and grouped points associated to a trunk from different heights measured from the estimated ground level. In addition the measurement vehicle is constantly moving during the measurement and the movement has to be taken into account. This section tells how it is done.

5.2.1 Measurement Grouping Using Estimated Orientation and Movement

Two dimensional measurement data is first gathered from both rotating laser scanners and it is transformed to vehicle coordinates. One laser scanner measurement consists of 181 range measurements at one degree resolution. Both of the lasers are mounted to a rotating fixture which is rotating around the z-axis. These measurements are transformed to the vehicle coordinates using coordinate transformations viewed in Section 4.1.1.

The data set is collected from a number of sequential measurements and measured points are fixed to the same coordinates using the estimated pose and movement of the vehicle. The fixing is done by caching all the collected points to a measurement cache in vehicle coordinates. Old values in the cache are moved and rotated to the opposite direction than the estimated pose change of the vehicle. After that the new points are added to the end of the cache. In effect, the algorithm integrates the measured pose changes to older collected measurements as current change of the pose is always updated to the cached points.

An error in the estimated pose increases as the cached points get older. The longer measurements are cached the larger the total error is. If the errors are to be minimized, the cache must be short. The caching of the measured points is essential for the tree search and measuring methods as accurate measurements cannot be calculated from only a few scans.

5.2.2 Edge Detection from Raw Scanner Data

As previously stated in Section 3.3, the 3D laser scanner is built from two rotating 2D laser scanners which are rotating at 45° angles around the vertical axis. Thus a single scan forms a plane which cuts trees at 45°. In a single scan trees can be seen somewhat similarly as in the horizontal laser scanner. The same edge detection algorithm is used as in the 2D case presented in Section 5.1.1. Every 3D data point is classified using the edge detector and the edge detector values are linked to each 3D measurement points. These edge detector values had different coding for a point measured beyond the maximum distance limit, a normal hit from non tree point, a point associated to a tree, and left and right edges of the tree. Different codes and edge detector values are listed in Table 1.

Table 1: The used edge detector values with explanations

Code	Explanation
-1	point beyond the maximum limit
0	point not associated to a tree
1	right edge of a tree
2	point associated to a tree
3	left edge of a tree

As the edge detector values are added to every measured point, the 3D measurements can be later filtered using these pre-calculated values. Points beyond the maximum range can be filtered out by accepting only points where the calculated value is greater or equal than zero. Points associated to a tree can be selected easily if only the points where the filtered value is greater than zero are accepted. This is later used to filter out measurements from branches and undergrowth and simplify the tree trunk measuring process. It can also be used to reduce the amount of points that are needed in later calculations.

5.2.3 Ground Estimation and Ground Model Generation

It is essential to find the 3D laser scanner measurements, which have reflected from ground to discard those measurement points that have not reflected from trees. If measurements reflected from ground are left in the data set where trees are searched, there would be a lot more wrong points between the searched trees. The easiest solution seems to be recognition of the ground echoes from single scans. The filtered ground measurements are separated to cache of measured ground points. Other points are left to the main cache, which is used in tree search.

The gathered and filtered measurements from the ground are used to generate the ground model, as knowledge about the ground level is needed later in the measurement process. A height of a measurement associated to a tree is needed when the tree is later modeled. The heights of measurements associated to a tree are calculated from the estimated ground level around the location of the tree. In addition the ground model can be visualized to the user or used to give information about the surrounding environment and a driven path. It, along the tree map, can be used to estimate the open path in future applications.

The ground level is found from the single laser scans by recursively fitting a line to the measurement points nearest to the ground. 3D data is scanned at 45 degrees and thus the leftmost measurements in a scanned plane are the undermost measured points in the vehicle coordinates. A line is fitted to these points iteratively with gradually diminishing margins. The process is iterated three times to get a sufficiently accurate estimate for the ground level.

The fitted line and separation of the ground and tree measurement groups are shown in Figure 14, where the fitted line is drawn with red, the ground points are drawn with orange and the other points that took part to the fitting are brown. All other measurements that are not used in the line fitting process are drawn with black color. Some of the highest points are removed from the fitting to hasten the process as the ground would never be above the measurement vehicle. Points that took part to the fitting process are shown at right in Figure 14.

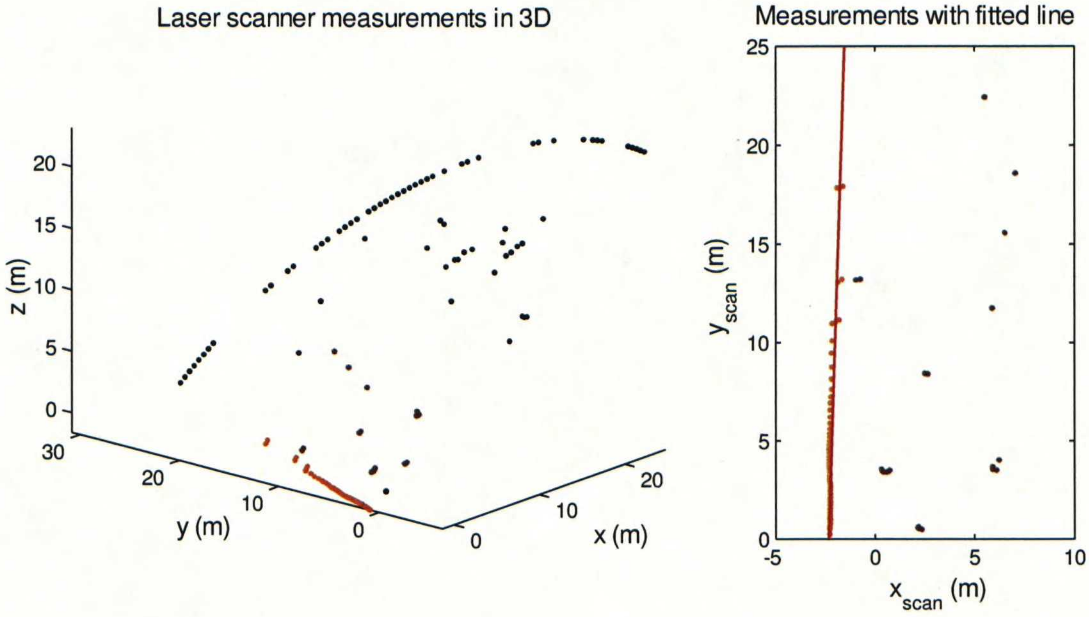

Figure 14: Filtering method for finding the ground points from a single laser scanner data set. The left part of the figure shows the measured data in the vehicle coordinate system. The right part of the figure shows a selected window of the same points in the coordinates of the laser scanner, where x is the direction of zero angle. The points for the ground line fitting are shown in brown and the selected ground points are shown in orange. All other points of the single laser scan are shown with black color.

The actual line fitting is done using the least squares line fitting technique [35] modified to accept weighting of points. The error function E of the modified biased line fitting method is presented in (23), and the used fitted line $f(y)$ is parameterized in (24). An error function used to fit the line is the sum of squares of the x-directional differences between the fitted line and the measured points x. Weighting is done using the weight function W, which is a function of y-directional distance y at the current index i.

$$E = \sum_{i=1}^{n} \left[(x_i - f(y_i)) W(y_i) \right]^2 \tag{23}$$

$$f(y) = ay + b \tag{24}$$

The line parameters a and b are fitted by minimizing E by calculating the zero values of the partial derivatives of (23) with respect to constants a and b separately. These partial derivatives are shown in (25).

$$\begin{pmatrix} \frac{dE}{da} & = & -2\sum_{i=1}^{n} \left[(W(y_i)x_i - W(y_i)y_i a - W(y_i)b) W(y_i)y_i \right] & = & 0 \\ \frac{dE}{db} & = & -2\sum_{i=1}^{n} \left[(W(y_i)x_i - W(y_i)y_i a - W(y_i)b) W(y_i) \right] & = & 0 \end{pmatrix} \tag{25}$$

The partial derivatives are set to zero and the equations are rewritten in matrix form in (26). The line parameters a and b can be calculated as $\mathbf{A}^{-1}\mathbf{B}$ using the numerical matrix calculation methods.

$$\underbrace{\begin{bmatrix} \sum_{i=1}^{n} W^2(y_i) y_i^{\,2} & \sum_{i=1}^{n} W^2(y_i) y_i \\ \sum_{i=1}^{n} W^2(y_i) y_i & \sum_{i=1}^{n} W^2(y_i) \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} a \\ b \end{bmatrix} = \underbrace{\begin{bmatrix} \sum_{i=1}^{n} W^2(y_i) y_i x_i \\ \sum_{i=1}^{n} W^2(y_i) x_i \end{bmatrix}}_{\mathbf{B}} \tag{26}$$

The position of the ground line is initially selected to be parallel to y-axis and go trough the leftmost points in the selected window as shown in Figure 14. Then the points that are near the fitted line are taken as the new points to where the line is refitted. This process is iterated three times to get a sufficiently reliable estimate for the ground point group. The quality of the fitted line is estimated as the size of the ground group. If the number of the resulting ground points is smaller than ten, the ground fitting is assumed to be incorrect and the filtering is not done at all.

To get a better estimate, more weight is given to the points that are further away. Many different weighting functions have been tested. The best way to weight points seems to be a linear weighting function, where all the points are given some basic weight and the weight is increased with a small multiplier respect to the distance y. The resulting function with the chosen constants is shown in (27). The weight W is squared similarly as in (26) causing the real weighting function to be the square root of the function presented in the equation. This result is experimentally fitted to this particular application and it is dependent of the geometry of the measuring system and the properties of the 3D scanner.

$$W^2(y) = 0.25 + 0.03y \tag{27}$$

The measurements that are selected as the ground points are not equally distributed as the 3D point cloud is denser near the scanner and sparser further away. While optimizing the ground level calculations for speed, the extra points must be removed. A simple way to do this is to model the distribution of points hitting to the ground by calculating a distance histogram of points hitting ground over the whole measuring drive in a typical forest. This measured histogram is shown in Figure 15.

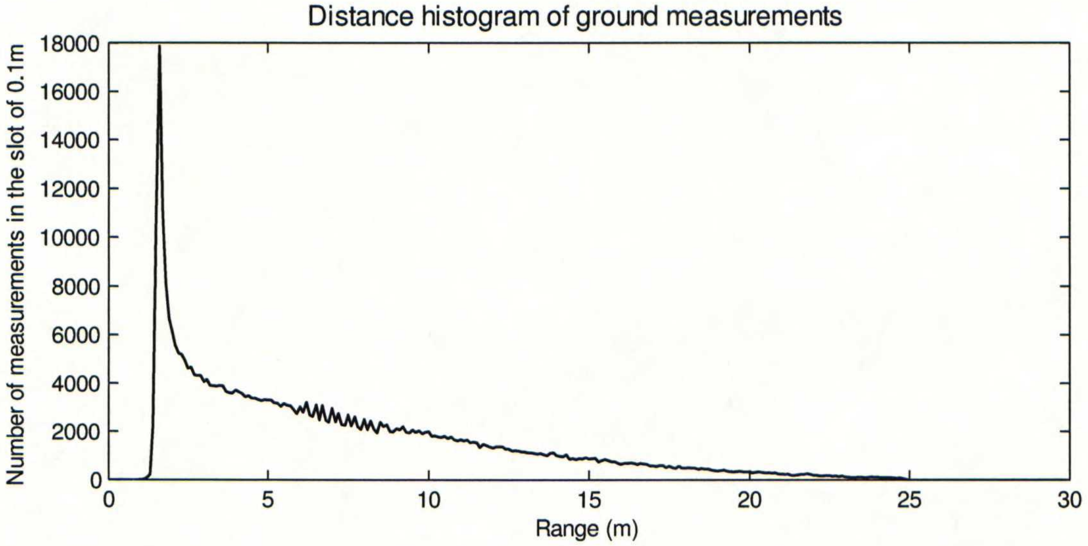

Figure 15: A distance histogram of measurements from ground after selecting the points by the line fitting method.

The optimal distribution of the measured ground points is a uniform distribution. To achieve it, the distance distribution should be increasing linearly along the radius. Because there are a lot more points than needed in ground plane fitting, extra points can be discarded randomly. The probability for accepting points is calculated as a function of the range based on the previously measured histogram-data. This is done in the following manner. First the inverse of the distance histogram is calculated and then it is multiplied with the value of the histogram around the range of 24m, as it was the most distant range from where the ground measurements were in this case acquired. The resulting graph is shown in Figure 16, and it is representing the probability to keep measurements to get a uniform distribution along the measurement range.

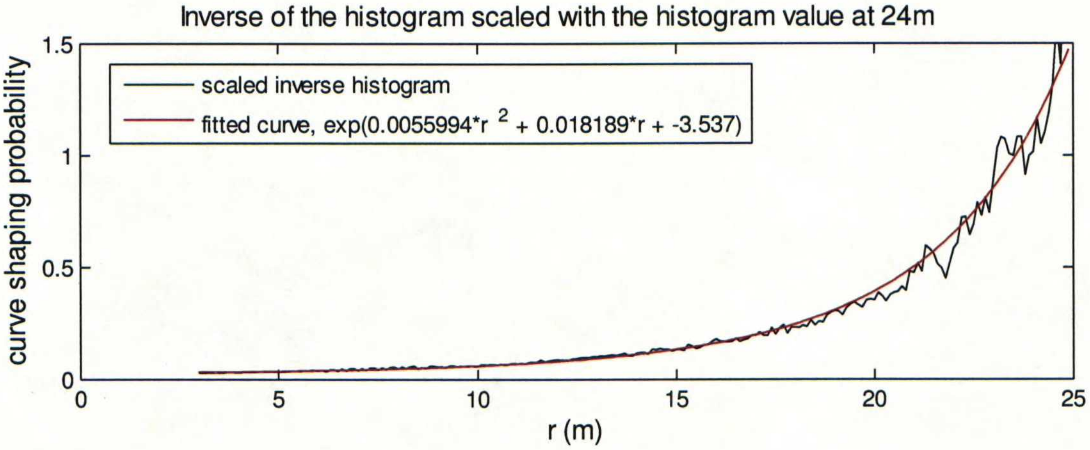

Figure 16: The curve fitting procedure to get a probability density function of the points needed to be kept so that the density of the original data would be shaped to uniform distribution along the measurement range.

A curve is fitted to the inverse histogram data using Matlab's line fitting tools. The best fit seems to be the exponent function of a second order polynomial. The fitted curve is drawn over the scaled inverse of the histogram in Figure 16 with red color. This curve estimates the probability to get uniform distance distribution of measured points from ground. To get the uniform distribution over the whole area, it is multiplied with the range as shown in (28).

$$P_{equalArea}(r) = rP_{flat}(r) = re^{0.0055994r^2+0.018189r-3.537} \tag{28}$$

The filtered ground measurements are discarded using the previously calculated probability density function. Nearly all points near of the measurement vehicle and only a few points more distant are discarded. The remaining ground points are added to the ground cache, which is moved and rotated similarly as the main measurement cache using information about the pose changes of the measurement vehicle. The used motion compensation method is described in Section 5.2.1.

The final part of the ground modeling is to fit a grid of partially continuous planes to the remaining equally distributed ground cache. The ground model is estimated by creating a grid with one meter squares around the measurement vehicle up to 20m. The grid heights are calculated by averaging all the measured height values in the ground cache around the same grid point. An example of a ground model is drawn in Section 7.3.

5.2.4 Tree Localization Using 2D Histogram

Trees are searched from the cached measurement points by calculating the 2D histogram over the data set using bin width of 0.2m in both directions. The trees are supposed to be nearly vertical objects so they should be easily seen, while all the measurement points with nearly same x and y coordinates are grouped. The number of measured points hitting to a bin is highly biased by the measurement range. To separate other measurements from the trees, there must be some kind of edge value that decides when the measurement point group is classified as a tree.

The edge value is determined using a simulated model of a tree calculated as a function of the measurement range. The number of measurement points is assumed to be equal at every steradian at the FOV of the 3D scanner. A solid angle of the model tree is estimated as a function of the measurement range. The solid angle can be calculated in spherical coordinates using (29) [36]. It is an area integral of the unit sphere, where the model of a tree is projected. In (29) φ is the polar angle, θ is the azimuthal angle, and S is the area of the measured object.

$$\Omega = \iint_S \sin\varphi \partial\theta\partial\varphi \tag{29}$$

The model calculation is visualized from side and from above in Figure 17. In the figure, the azimuthal angle is calculated from parameters of a model tree. The minimum diameter for a qualified tree, d, is selected as 80mm, and the horizontal angle of the tree as a function of range, r, can be calculated using (30).

$$\Delta\theta = 2\arctan\frac{d}{2r} \tag{30}$$

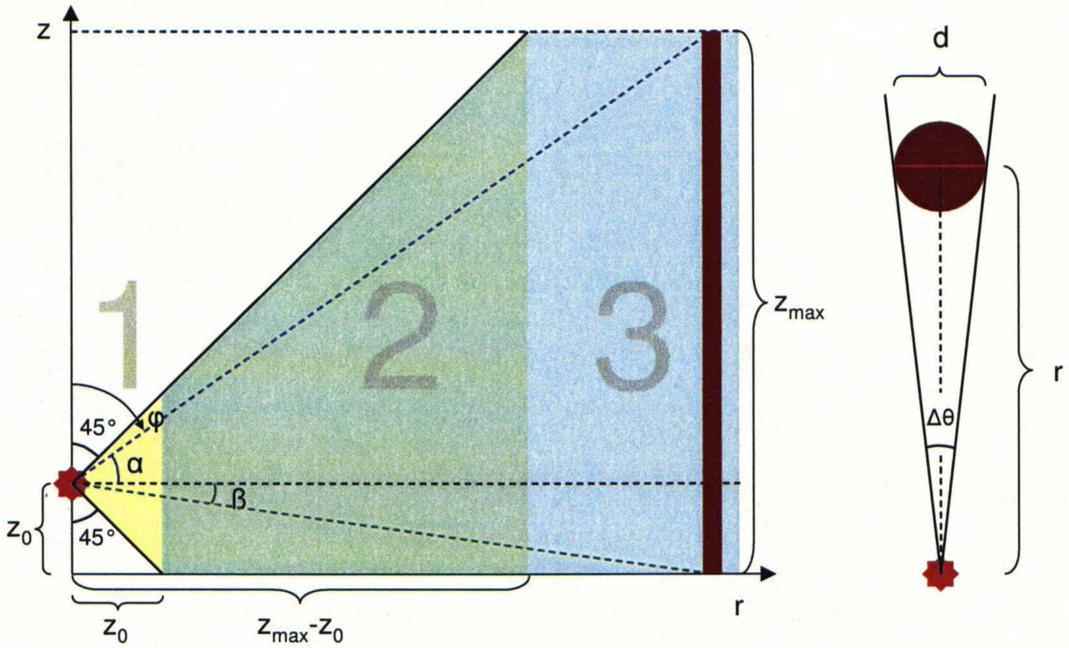

Figure 17: Estimation of the number of points measured from a model tree. The height of the tree, h is calculated from the field of view of the 3D scanner. It is assumed that the density of the measurements is equal per steradian. At the first region in the figure, FOV is limiting the number of measured points from a tree. At the second region the ground is limiting measurements, and at the third one also the upper limit of the measurements is decreasing the number of measurements that the model tree can reflect.

The solid angle of the model tree in (31) is calculated using (29) and (30). The angles α and β in the equation are the same angles as seen in Figure 17. The FOV of the 3D scanner is limiting angles to ±45 degrees. The calculation of these angles is presented in (32) and (33). In the equations, r is the measurement range, z_0 is the mounting height of the 3D scanner and z_{max} is the height limit for the measurement points. The height limit is fixed to 10m, as the trunk of tree is not usually seen in the data above 10m.

$$\Omega = 2\arctan\frac{d}{2r}(\sin\alpha + \sin\beta) \tag{31}$$

$$\alpha = \min\left[\frac{\pi}{4}, \arctan\left(\frac{z_{\max} - z_0}{r}\right)\right] \tag{32}$$

$$\beta = \min\left[\frac{\pi}{4}, \arctan\left(\frac{z_0}{r}\right)\right] \tag{33}$$

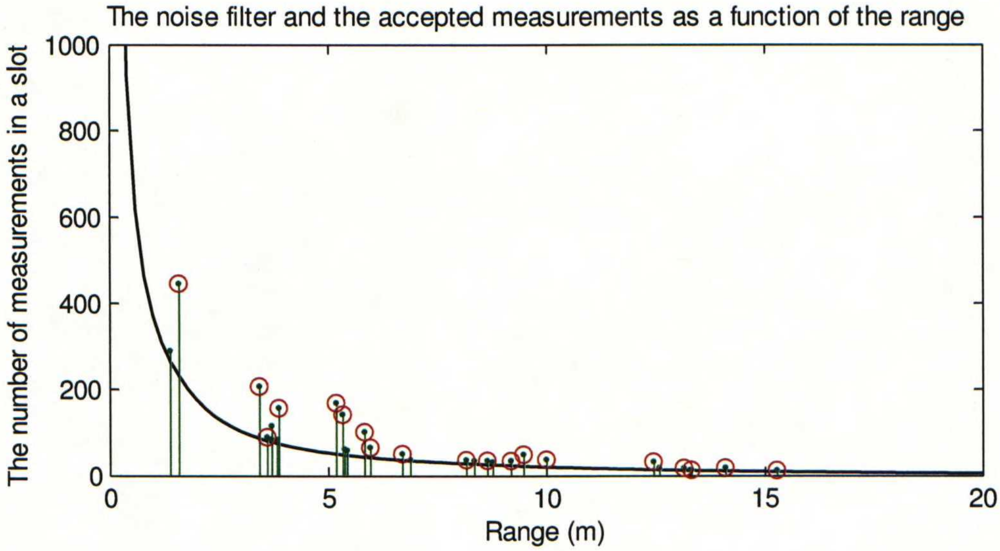

Figure 18: The calculated noise filter curve and accepted trees as a function of the measurement range. All of the accepted measurements are marked with green color. Red color indicates the local maxima. The same measurements are visualized in Figure 19.

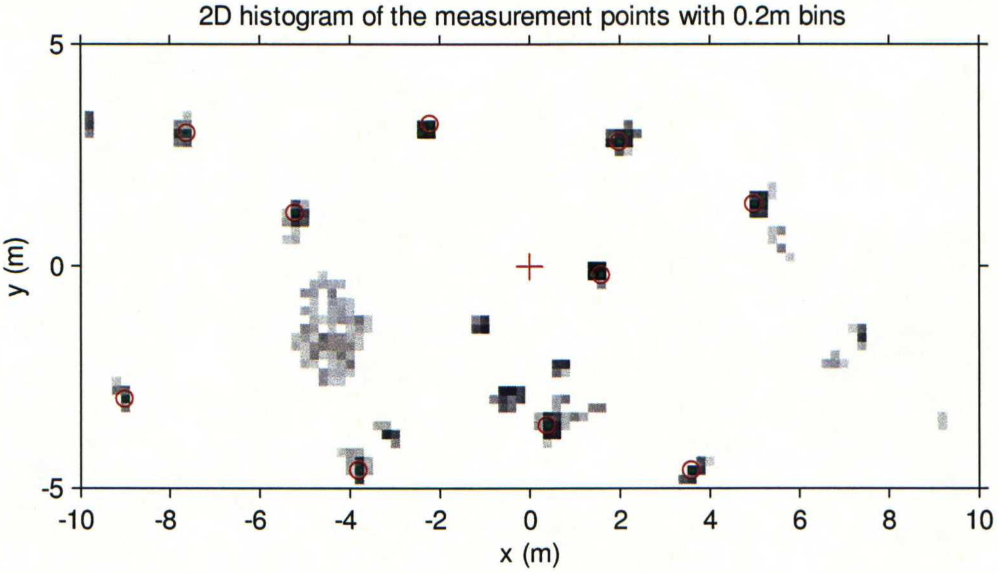

Figure 19: The 2D histogram of the measurement points from trees. The histogram is calculated using a bin width of 0.2m in both x- and y-directions. The intensities in the figure reveal the number of measurements in the bin. The green marks indicate acceptance after the noise filter and the red circles surround accepted local maxima at the local neighborhood. The color scale of the image is logarithmic, because otherwise the bins with only a few measurements would not be visible in the figure.

The calculated solid angle of the model tree is biased with an average measurement density of the 3D scanner and the resulting limit is used to specify the bins from where a tree is found. The limiting curve, which is called the noise filter, is visualized in Figure 18, where the black curve represents the calculated filter limit, and the green lines are the accepted histogram bins. The red circles surround local maxima. The local maximum is calculated using a range limit of one meter. The local maxima locations in the histogram are revealing the tree locations. The histogram is visualized in Figure 19. The darker the bin in the image is the larger is the number of the measurement points. The green dots and red circles are same as those in Figure 18.

5.2.5 Tree Pose Adjustment by Line Fitting

The found trees are reoriented by fitting a vertical line to the points at the found area of a tree and rotating the data so that the line is aligned vertically. The line is fitted using least squares method. First the line is parameterized using (34) [37].

$$\begin{aligned} x &= x_0 + v_x t \\ y &= y_0 + v_y t \\ z &= z_0 + v_z t \end{aligned} \tag{34}$$

Requiring the parameter t to be the same for all Cartesian coordinates x, y and z, (35) is derived. In the equations, x_0, y_0, and z_0 are the coordinates of the reference point through which the line is going. $\boldsymbol{v}$ is the direction vector of the line.

$$\frac{x - x_0}{v_x} = \frac{y - y_0}{v_y} = \frac{z - z_0}{v_z} \tag{35}$$

The previous equation is derived to two independent equations by combining v_x with v_y and v_z. Formed constants k_x and k_y in (36) are equivalent to the slopes of the xz- and yz-planes as shown in Figure 20.

$$\begin{aligned} z - z_0 &= \frac{v_z}{v_x}(x - x_0) = k_x(x - x_0) \\ z - z_0 &= \frac{v_z}{v_y}(y - y_0) = k_y(y - y_0) \end{aligned} \tag{36}$$

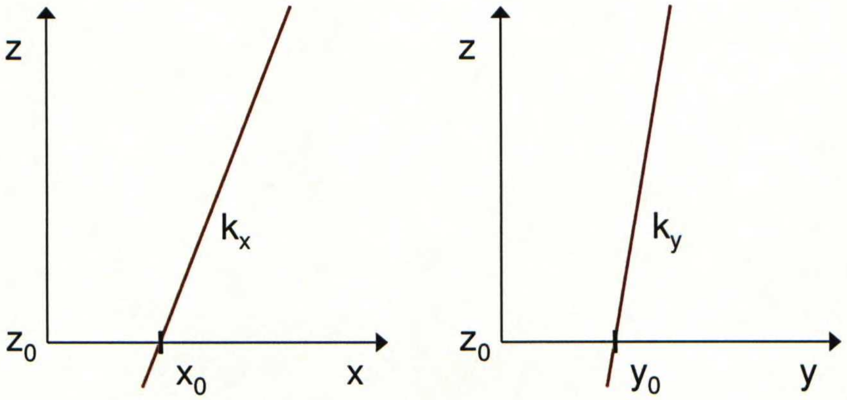

Figure 20: The fitted line projected into the xz- and yz-planes. z_0 is chosen to be zero to ease the fitting process and the calculation of the coordinates for the tree in the ground level. The measured points are previously elevated to meet ground level at z_0.

The z-directional reference value is chosen to be zero and the equations are flipped so that the coordinates x and y can be calculated as a function of z from the parameters k_x, x_0, k_y, and y_0 in (37).

$$\begin{aligned} x(z) &= \frac{1}{k_x} z + x_0 \\ y(z) &= \frac{1}{k_y} z + y_0 \end{aligned} \tag{37}$$

The line is fitted by minimizing the Euclidean distance from the line to the points along the xy-plane using an error function presented in (38). The Euclidean distance is multiplied using W, which is the weight function of height, z.

$$e_r = W(z)\sqrt{(x - x(z))^2 + (y - y(z))^2} \tag{38}$$

The sum of squares of these Euclidean errors is shown in (39). It is the function that is minimized to fit the line. The x and y parts can be separated as shown in (40). They do not have any shared parameters, and both separated parts are similar as the two-dimensional version, described in Section 5.2.3. Thus the same method can be used separately for both sums, and the total error can be minimized.

$$E = \sum_{i=1}^{n} {e_r}^2 = \sum_{i=1}^{n} \left[\left[(x_i - x(z_i))^2 + (y_i - y(z_i))^2\right] W(z_i)^2\right] \tag{39}$$

$$E=\sum_{i=1}^{n}\left[\left((x_i-\frac{1}{k_x}z_i-x_0\right)W(z_i)\right]^2+\sum_{i=1}^{n}\left[\left(y_i-\frac{1}{k_y}z_i-y_0\right)W(z_i)\right]^2 \qquad (40)$$

The line is fitted iteratively while dropping the furthest points out. The radial error to the line without the weight function, shown in (38), is calculated for every point. The density of measurement points as a function of the calculated radial error is used to determine points that do not belong to the tree trunk. These points are filtered out from the trunk to improve the quality of the tree diameter measurement. The process is iterated three times as it is enough for fitting the line sufficiently well.

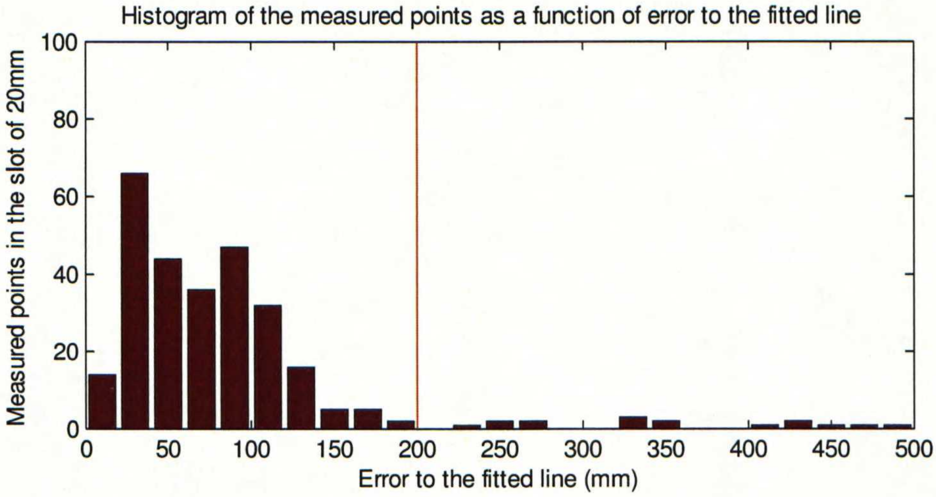

Figure 21: The histogram of measured points as a function of the error to the fitted line. The red line represents a selected limit, which is selected along the radius of the first zero element after the center of mass of the histogram.

Points that are filtered out are shown at the right side of the red line in Figure 21. The limit is decided using the first zero element in the calculated histogram after the center of mass of the histogram. This method finds the first radial cap larger than 20mm and heuristically cuts the tree out of the possible surrounding objects and erroneous measurements.

5.2.6 Multiple Horizontal 2D Projections

The filtered and reoriented data of a single tree is used to generate ten separate 2D projections of that tree. The projections are taken from one to ten meters at one meter intervals. A projection overlaps its neighboring projections as it is 2m height. The projections are accepted only if at least 15 measurements are counted in the current slice. All of these accepted projections of the same tree are shown in Figure 22 with fitted circles. The figure shows the process of tree filtering, projection and circle fitting in three phases. The used circle fitting process is described in Section 5.2.7.

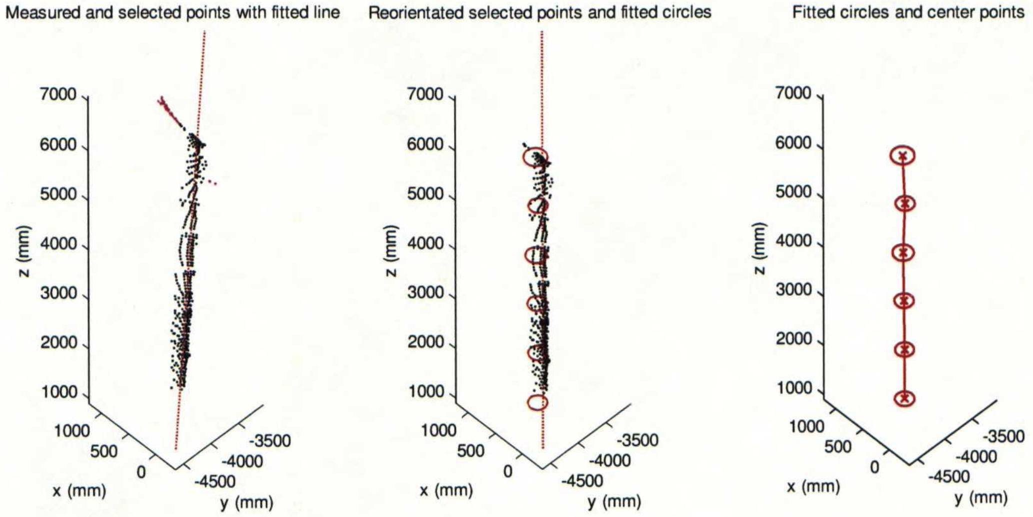

Figure 22: Tree extraction, position fine tuning, circle fitting, and tree model generation in three phases. In the left subfigure, the line is recursively fitted to the data set, and some outliers are filtered out. In middle subfigure, same data set is re-orientated and divided into 10 slots. A circle is fitted to every slot if there is enough measured points. In the right subfigure, the wire model of the tree is visualized using the fitted circles and center points. Notice that the lowest and the three highest circles are missing as there are too few measurement points around those regions.

5.2.7 Tree Diameter Measurement with Circle Fitting

The circle fitting is done using the linear circle fitting procedure presented by Ian Coope [38]. According to his method the nonlinear circle fitting procedure presented in (41) is made to linear using the substitutions shown in (42). The resulting minimized error function is presented in (43).

$$E = \sum_{j=1}^{n} \left[\left\| \underline{x} - \underline{a}_j \right\|_2^2 - \rho^2 \right]^2 \tag{41}$$

$$\underline{b}_j = \begin{bmatrix} \underline{a}_j \\ 1 \end{bmatrix}, \quad \underline{y} = \begin{bmatrix} 2\underline{x} \\ \rho^2 - \underline{x}^T \underline{x} \end{bmatrix} \tag{42}$$

$$E = \sum_{j=1}^{n} \left[\underline{a}_j^T \underline{a}_j - \underline{b}_j^T \underline{y} \right]^2 \tag{43}$$

In the previous equations E is a minimized error function, n is the number of points in data, $\underline{x}$ is the vector to the center point of a circle, ρ is the circle radius, and $\underline{a}_\mathrm{i}$ is the vector to the measured point at the index j. In the new enhanced space, $\underline{b}_j$ is a vector to the measured point at index j, and $\underline{y}$ is a vector to the fitted circle. In 2D case (43) can be expressed using (44), where $a_{x,j}$ is x and $a_{y,j}$ is y axial component of the vector $\underline{a}_j$. Similarly y_1, y_2 and y_3 are the components of the vector $\underline{y}$.

$$E = \sum_{j=1}^{n} \left[a_{x,j}{}^{2} + a_{y,j}{}^{2} - a_{x,j} y_1 - a_{y,j} y_2 - y_3 \right]^2 \tag{44}$$

The error function E can be minimized by calculating the roots of the partial derivatives. These partial derivatives are shown in (45) and they can be expressed in matrix form using (46).

$$\left(\begin{array}{ccccc} \dfrac{dE}{dy_1} & = & -2\sum\limits_{j=1}^{n} \left[a_{x,j}{}^{2} + a_{y,j}{}^{2} - a_{x,j} y_1 - a_{y,j} y_2 - y_3 \right] a_{x,j} & = & 0 \\ \dfrac{dE}{dy_2} & = & -2\sum\limits_{j=1}^{n} \left[a_{x,j}{}^{2} + a_{y,j}{}^{2} - a_{x,j} y_1 - a_{y,j} y_2 - y_3 \right] a_{y,j} & = & 0 \\ \dfrac{dE}{dy_3} & = & -2\sum\limits_{j=1}^{n} \left[a_{x,j}{}^{2} + a_{y,j}{}^{2} - a_{x,j} y_1 - a_{y,j} y_2 - y_3 \right] & = & 0 \end{array} \right) \tag{45}$$

$$\underbrace{\begin{bmatrix} \sum\limits_{j=1}^{n} a_{x,j}{}^{2} & \sum\limits_{j=1}^{n} a_{x,j} a_{y,j} & \sum\limits_{j=1}^{n} a_{x,j} \\ \sum\limits_{j=1}^{n} a_{x,j} a_{y,j} & \sum\limits_{j=1}^{n} a_{y,j}{}^{2} & \sum\limits_{j=1}^{n} a_{y,j} \\ \sum\limits_{j=1}^{n} a_{x,j} & \sum\limits_{j=1}^{n} a_{y,j} & n \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \sum\limits_{j=1}^{n} a_{x,j}{}^{3} + \sum\limits_{j=1}^{n} a_{y,j}{}^{2} \\ \sum\limits_{j=1}^{n} a_{x,j}{}^{2} + \sum\limits_{j=1}^{n} a_{y,j}{}^{3} \\ \sum\limits_{j=1}^{n} a_{x,j}{}^{2} + \sum\limits_{j=1}^{n} a_{y,j}{}^{2} \end{bmatrix}}_{\mathbf{D}} \tag{46}$$

The parameters can be calculated from (46) by finding the solution for $\underline{y} = \mathbf{C}^{-1}\mathbf{D}$. The parameters of the circle can be calculated from the vector $\underline{y}$ using (42). The original parameters $\underline{x}$ and ρ are then calculated with (47), where the center point of the circle is marked with x_0 and y_0 and the circle radius with ρ.

$$\underline{x} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \dfrac{y_1}{2} \\ \dfrac{y_2}{2} \end{bmatrix}, \quad \rho = \sqrt{y_3 + \frac{y_1{}^{2} + y_2{}^{2}}{4}} \tag{47}$$

The circle fitting is not accurate enough while fitting circles only once, because of the outliers that remain in the data set. A statistical outlier filtering method is used to mark the farthest measurements as outliers. First a standard deviation, STD, of the distance from the circle to measurement points is calculated. The 1.2 times the STD is taken as the cut off limit from the fitted circle. This limit is able to discard outliers, as they are a lot farter than the selected limit. The circle is then fitted using the same method again and a better approximation of the tree is obtained. The data points and the area of acceptance are shown in Figure 23 with the iterated circle drawn in red.

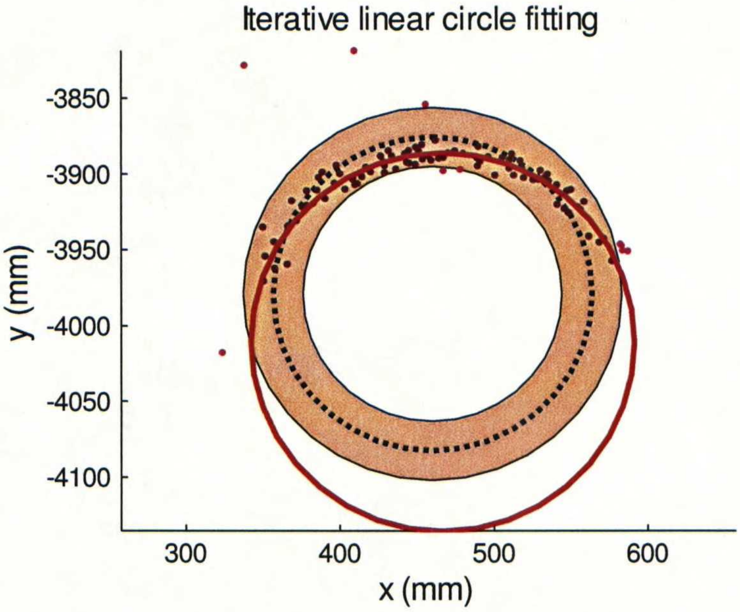

Figure 23: The iterative circle fitting method using linear circle fitting. The outliers are selected statistically by calculating the standard deviation of the distances from the fitted circle. The black dotted line is the first fitted circle and the yellow area around it is the region of accepted points. The red circle is the fitted circle after the outliers are filtered out.

5.3 Tree Feature Measurements and Measurement Errors

All the collected 3D measurements from a single tree and the corresponding 2D measurement are linked to a same entity. Circle fitting measurements are used as features for the tree model. Every tree model can have one 2D and ten 3D features. Each of these features has radius and location measurements as parameters. This kind of an abstraction helps the tree mapping process to decrease the amount of data and adjust the different measurements together from very different scanner devices. The features from 2D and 3D laser scanners at different heights can be associated to the same tree without mixing data from different heights or measuring devices.

The measurement errors are calculated using the estimated tree radius and location as the known values and all measurements are compared to these estimated values. There are no high precision reference measurements available and therefore estimated values have to be used as a baseline. While only studying the statistics of the measured data, the bias error remains unknown. The amount of bias error can be still estimated by comparing the 2D and 3D measurements together as has been done in Section 7.2.

Chapter 6

Mapping

The mapping process is needed for linking the measurements from different scans to the same trees. While the selected 2D and 3D laser rangefinder scans are combined and the translation and rotation between the scans is corrected, a map of the environment is obtained as a by-product. In this work the mapping is done using the high level features, fitted circles of measured trees at different heights using the measurements from the 2D and 3D scanners. Usually the mapping is done using a lot lower level approach, i.e. using FastSLAM-algorithm by Michael Montemerlo [5], where the optimal fit of a single scan is calculated using probabilistic methods.

A similar mapping technique is presented in an article from Andreas Nüchter et al. about "6D SLAM – 3D Mapping Outdoor Environments" [16]. In their article a map is generated by fitting scans together in 6 dimensions (three for translation and the other three for rotation). These methods can be used for mapping nearly anything, but they are not effective in more complex mapping scenarios, like measuring trees in a forest. In addition, these methods increase the size of a map data set and they also need a lot more computational power.

The size of the map can be reduced by extracting high level features from the data set and using these features for mapping. The accuracy of the previously mapped features is improved by associating the same trees between different scans and using statistical methods for calculating the best effort estimate for the collection of linked features. Simultaneously the pose estimates of the measurement vehicle and the previously mapped trees are adjusted using a two dimensional modification of the Berthold K. P. Horn's Absolute orientation based on unit quaternions [29]. In the end, the resulting map only contains the estimated tree features and thus size of the data set can be kept small.

This process is not mathematically as exact as usually used SLAM algorithms like fastSLAM or EKF SLAM [5], but it is computationally inexpensive and sufficiently accurate for mapping forest where trees can be used as features. The mapping is faster when using only less mapped objects. The recognition and extraction of features for found objects from multiple scans is the computationally most expensive part of the whole process.

6.1 Tree Models with High Level Features

Tree models are generated by combining extracted features from the 2D and 3D scanner data, which is collected using methods described in Chapter 5. The features in this work are the fitted circles with their parameters. There is one 2D feature and ten 3D features in a tree model. The 3D features have been measured at one meter intervals from zero to nine meters height. The Nearest 2D and 3D trees are associated using the mutual consistency check, which is explained in Section 4.2.4 and in Figure 10.

Tree measurements are not perfect and thus not every height at every 3D measuring cycle is perfectly measured. Only the features that can be measured reliably are taken into account in the mapping process. Tree objects are accepted to the mapping if there are at least three 3D features at different heights or a 3D feature and a 2D feature associated to the same tree. These identified trees are added to the map, and a new estimate of a mapped tree is calculated every time a new set of features is added to the map.

6.2 Map Data Structure

The map is built for 2D and 3D scanner data separately, because rotation of the 3D scanner is only estimated from the obtained 3D data and the estimated angle of 3D scanner is not absolutely accurate. Thus the measured locations of the 3D and 2D tree features may have a minor difference. To minimize the effect of this difference between the different measuring devices the map is generated separately for 2D and 3D features and those features of the same tree object are only linked to each other in the map.

The 3D version of the map has an estimate for tree's x and y location and fitted circle radius. The estimate is calculated as a continuous average separately for x, y and r, because the average is an adequate estimate for the location of a tree, and the calculation of a continuous average is fast. The estimate can be updated just by multiplying the previous value with the number of measurements and adding the new measurement and dividing it with a new updated number of measurements. The 2D version of the map has similar estimated x, y locations of the tree and the estimated fitted circle radius measurement. These estimates are updated similarly as the 3D estimates. The measurements from the 3D and 2D laser scanners are only affecting to each other while other estimate is not available for that tree, as then 3D map estimate is used for finding the 2D tree from the newest scan and vice versa.

The Matlab's structure array is used with fields for every 2D and 3D estimate, field for numbers of 2D and 3D measurements and fields for structures of 2D and 3D measurements. All the measured features are collected from every measuring cycle affecting the current tree and all the feature data is saved to the array for the tree. The stored measurement data in these inner structures is not used in the mapping process as it is slow. Only the continuously calculated estimates are used to compare the found trees with the mapped trees. The all collected features from every measuring cycle are used only in the post processing of the map and while the looped path is closed as described in Section 6.4.

$$forest = struct\begin{bmatrix} time \\ xyr\_estimate\_2d \\ amount\_2d \\ xyr\_estimate\_3d \\ amount\_3d \\ measurements\_2d = struct\begin{bmatrix} time \\ from\_xy \\ xyzr \end{bmatrix} \\ measurements\_3d = struct\begin{bmatrix} time \\ from\_xy \\ xyzr \end{bmatrix} \end{bmatrix} \tag{48}$$

A tree-object stored in an array *forest* is visualized in (48), where the word *struct* means a structure array. In the structure, *time* is an integer that shows when this tree was found, feature estimates *xyr_estimate_2d* and *xyr_estimate_3d* are three dimensional vectors with the average estimates of the location and the fitted circle radius. Variables *amount_2d* and *amount_3d* are integers that store information about the number of the 2D or 3D measurements stored for that tree. Variables *measurements_2d* and *measurements_3d* are structure arrays, in which every measured feature is stored. Those structures have fields for measuring time, an estimate of the location of the measurement vehicle, *from_xy*, and the measured features that are stored in *xyzr*. In the 2D case *xyzr* is a four-dimensional vector where the three first dimensions are for location, and the last dimension is for fitted circle radius. In the 3D case the same variable is a 4 by 10 matrix with four rows for x, y, z and r and ten columns for every measured height.

The data structure must be searched by the estimated x and y locations, so the Matlab's structure array is not the best possible solution as the search procedure must go trough all objects in the map. The better solution would be to use a kd-tree, k-dimensional tree similarly as in an article about 6D SLAM from Andreas Nüchter et al. [16]. However the feature based map does not have too many objects and therefore the search can be done in real-time with Matlab software.

6.3 Mapping Algorithm

A mapping algorithm used in this work consists of a few steps that are done for the found trees to update them into the map. The algorithm has the following steps:

1) Associate the found and mapped objects to pairs
2) Calculate the optimal transformation between the pairs
3) Update the pose of the vehicle and find pairs iteratively
4) Add found objects, which do not have pair in the map, to the map
5) Update map with objects that have pair in the map

6.3.1 Associating the Found and Mapped Objects

All the distances from found trees to the mapped trees around the measurement vehicle are calculated and the nearest trees are paired using the mutual consistency check if the distance between the paired trees is smaller than one meter. These requirements, mutual consistency check and the distance limit are able to pair found trees with mapped trees. The coordinates of the found trees are calculated using the measured locations of the trees and the measured location of the vehicle. The vehicle pose is updated using the laser odometry and the previous map position.

6.3.2 Transformation between Pairs

The optimal transformation between paired, the found and the mapped trees is calculated using an idea of Berthold K. P. Horn's "Absolute orientation using unit quaternions" [29]. In the 2D case and when the scale is always the same, the problem can be simplified. Using Horn's proven fact that the optimal translation is always the translation between the mass centers of the found and the mapped tree locations. If the locations of found trees are represented with n 2D vectors $\mathbf{v}_{f,i}$ and the locations of the mapped tree pairs with the same number of 2D vectors $\mathbf{v}_{m,i}$, the new coordinates that use the mass centers as an origin can be represented as in (49), where the vectors $\mathbf{v}'_{f,i}$ represents the locations of the found trees in the new coordinates similarly as $\mathbf{v}'_{m,i}$ does the same for the mapped trees.

$$\begin{aligned} \mathbf{v}'_{f,i} &= \mathbf{v}_{f,i} - \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_{f,i} \\ \mathbf{v}'_{m,i} &= \mathbf{v}_{m,i} - \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_{m,i} \end{aligned} \tag{49}$$

In the 2D case where the scale is always the same, the approximately best fit could be calculated as a rotation that minimizes the sum of distances between pairs of the found and mapped trees. The rotation between pairs can be minimized using polar coordinates θ and r of the calculated mass center coordinates. The rotation can be estimated by minimizing the sum of the curve lengths between the pairs. The rotation can then be calculated as an average angular error biased with the average range of the paired points. The average angle can be calculated only if the angle between paired points is always smaller than pi/2. This ensures that the average is always calculated correctly in the periodic coordinates between –pi and pi. This can easily be ensured as there are only small rotation differences between the map and the odometry estimate.

The rotation is now calculated in a few phases. First a vector of error angles $\mathbf{\Delta\theta}$ is calculated in (50), where a n-dimensional vector of the angular differences between the angle coordinates of the pairs is wrapped between $-\pi$ and π. The wrapping is necessary, because angles are later averaged and averaging does not give the correct angle unless all the calculated angles are from the same cycle. All the calculated angles have to be larger than –pi/2 and smaller than pi/2 to ensure that the average is calculated correctly. Secondly a n-dimensional vector of the average ranges $\underline{r}_{avg}$ is calculated in (51). Finally the rotation $\Delta\theta$ between the found and the mapped trees can be calculated using (52).

$$\Delta\theta_i = \mathrm{mod}\left(\theta_{m,i} - \theta_{f,i} + \pi, \quad 2\pi\right) - \pi \tag{50}$$

$$r_{avg,i} = \frac{r_{m,i} + r_{f,i}}{2} \tag{51}$$

$$\Delta\theta = \frac{\sum_{i=1}^{n} \Delta\theta_i r_{avg,i}}{\sum_{i=1}^{n} r_{avg,i}} \tag{52}$$

Because the coordinates of the found trees are rotated, the optimal translation is changing depending on the center of rotation. The center of rotation is taken into account in the translation calculations in (53), where ***R*** is the rotation matrix calculated in (54). $\mathbf{v}_{r0}$ is a two-dimensional vector to the location of the rotation center in the map coordinates. The resulting **Δ*v*** is the vector pointing out the needed translation of the found trees after they are rotated using the rotation matrix ***R***.

$$\mathbf{\Delta v} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_{f,i} - \mathbf{v}_{r0} - \mathbf{R}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_{m,i} - \mathbf{v}_{r0}\right) \tag{53}$$

$$\mathbf{R} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{54}$$

This angular representation is used instead of Horn's absolute orientation method because this type of approach allows the use of nonlinear filters in these calculated angles and distances between pairs. This kind of filtering is used in this work to reduce the effect of wrong associations.

6.3.3 Updating Pose of the Vehicle and Finding Pairs Iteratively

The found trees and the estimated pose of the measurement vehicle are first rotated with the rotation matrix **R**, calculated in (54) using the angle from (52), and then translated with **Δv**, calculated in (53). The distances between the paired transformed found trees and mapped trees are calculated and the only a half of the paired points, with the smallest error, are accepted. The transformation is calculated again using the same method, but only with the previously selected pairs. This iteration discards trees that do not fit accurately with the other tree-pairs. Finally the iterated transformation is accepted for all the found trees and for the pose of the measurement vehicle.

6.3.4 Adding New and Updating Old Trees to Map

After the transformation is corrected, the found trees that do not have any pair, but have enough features are accepted as new trees to the map. These new trees are added to the map using the methods explained in Section 6.2. The estimates are calculated for 2D and 3D parts of the map separately. All the features are added to the 2D and 3D arrays of features. The data from 2D features is not affecting the 3D features and vice versa.

The found trees that have pair in the map are also rotated and translated similarly as the other new trees. The corresponding mapped trees are updated with the information from these paired trees. Estimated 2D and 3D map locations and circle radius estimates are updated using the continuous averaging method described in Section 6.2. Also the features extracted from the found trees are added to the feature arrays of the corresponding mapped trees.

6.4 Closing the Loop

After laser odometry and mapping there is still some errors left. While the vehicle moves, noisy estimates will increase the accumulated errors. Accuracy of the measured map is always decreasing while the measurement vehicle is moving towards an unmapped region. Because of the increasing uncertainties in the current location, the resulting map is not equally accurate at all locations. One way to model the increasing uncertainty is to estimate the uncertainty along the driven path. When the previously mapped path is found again, the pose error between the old mapped trees and the found trees can be measured. The map can then be corrected using this measured pose error.

The phases of the map correcting algorithm are:

1) Find a right association between the old mapped and the found trees
2) Calculate needed transformation between associated pairs
3) Scale the measured rotation error estimate to meet the measured rotation error
4) Calculate a new estimate for the driven path correcting the angular error
5) Calculate remaining translation error

6) Calculate a new estimate for the driven path correcting the remaining translation error
7) Rebuild the path
8) Link mapped objects to the old path using all the locations where the same tree has been measured
9) Calculate new object locations using the links from the rebuilt path

6.4.1 Finding and Accepting Transformation

The map is divided into a two sets; the old trees and the new trees. The old trees are compared to the found trees, which are the trees that are just found using the sensors. The pose error between the old mapped trees and the found trees is extremely hard to find so that possibility of making wrong associations is minimized. The maximum search space is estimated using the previously estimated and constantly incremented error estimate. When these two sets are found to intersect, a few found trees at the center of the intersection are selected. All combinations of translations from the centermost found tree to the old mapped trees, where translation distance is smaller than the estimated maximum localization error, are tested for the best fit.

The translation of a one selected relation is done for all the found trees. Then translated found trees are associated with the mapped trees and the optimal transformation is calculated similarly as in Section 6.3.2. The total error of the association is calculated to estimate the quality of the fit between the paired objects using (55). In the equation $\underline{r}_{m,i}$ is the location of the mapped tree with a pairing index *i* and $\underline{r}_{t,i}$ is the location of the associated translated found tree.

$$E_{fitting} = \sum_{i=1}^{n} \left\| \mathbf{v}_{m,i} - \mathbf{v}_{t,i} \right\| \tag{55}$$

The previous error is smallest when there are no paired objects at all, so the total error cannot be directly used to select the best fit. A quality of fit is introduced to ease the selection of the best fit. It is calculated using an idea that the best fit would have the smallest error and the largest number of associated objects. The quality of fit is then calculated using (56), where the number of pairs, *n* is divided by the average of fitting error, $E_{fitting}$.

$$q_{fitting} = \frac{n^2}{E_{fitting}} \tag{56}$$

The association with the largest quality of fit would probably be the best transformation, but it also might also be a wrong association. To avoid wrong decisions, the a few best transformations are stored to the memory and later compared to the other stored transformations. If nearly the same transformation is found more than three times or the number of paired objects in the association is remarkably high compared to the number of all trees at the intersection, the current transformation is accepted.

6.4.2 Rebuilding the Map

After the right transformation is found, the map has to be rebuilt to minimize pose errors. As already described the accumulated pose error is estimated and recorded during the drive. These error estimates are linked to the driven path, and thus the gathered errors can be corrected more accurately than by only linearly dividing the needed transformation along the driven path.

The map rebuilding algorithm is designed to use the gathered errors and divide their effect along the driven path. The locations of measured trees are linked to the path and dragged along while the path is changed. This method is not optimal as the distances and angles between the trees could change. However this method uses the distance and angle of mapped trees from all of the measuring locations for every tree seen during measurements. After the path is changed, the same tree would be measured similarly as if the rebuilt path would be the original path without any rebuilding.

First the accumulated error along the driven path between the current location of the measurement vehicle and the nearest location of the previous path is scaled to meet the measured error. The scaling is done similarly for the rotation and translation errors using (57), where the scaled error at index *i* is marked with $E_{scaled,i}$ and the measured error with E_{meas}. The estimated errors along the path are marked with E_i, where *i* is the path index from the first to the last path step. The same formula can be used for scaling the estimated rotation and translation errors.

$$E_{scaled,i} = \frac{E_i - E_1}{E_n - E_1} E_{meas} \tag{57}$$

The driven path is rebuilt so that every path step is modeled with a step length and an angle change. The scaled accumulated rotation error is added to step angles and the path is reconstructed starting from the beginning to reduce the measured angle error from the resulting path. After the angular error of the driven path is cancelled, the end of the path has changed its location. The remaining translation is calculated between the target location of the original translation and the end location of the path after angular correction. Similarly the remaining translation is divided along the driven path using the knowledge about the accumulated and estimated translational errors.

Finally the estimated errors along the fixed path are replaced with the realized uncertainty u_i, which is calculated using (58). In the equation E_0 is the estimated error at the last index at the driven path before any transformation. $E_{scaled,1}$ and $E_{scaled,n}$ are calculated using (57). Similarly this equation can be used for calculating the angular and translational errors. The equation divides the realized error along the path so that in both directions along the realized loop, the furthest path index has the largest uncertainty.

$$u_i = E_0 + \left| E_{scaled,i} - \frac{E_{scaled,1} + E_{scaled,n}}{2} \right| \tag{58}$$

The trees are linked to the old path by their original locations that has been used to build the map. The links to the trees are modeled using polar coordinates, distance and angle, from every step of the driven path. Same relations are found for the rebuilt path and the new tree locations are calculated as the average of the locations where the tree links from the path are pointing. The location repair is changing the estimated locations of the mapped trees and all of the gathered and measured feature data in that map object. The whole map rebuilding process is demonstrated in three phases in Figure 24.

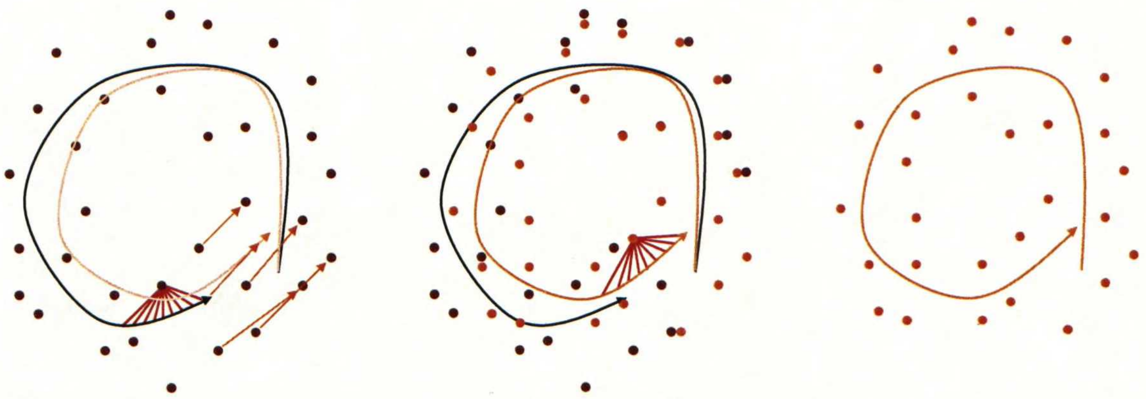

Figure 24: The map rebuilding process in three phases. First the right transformation between the mapped trees and the found trees is found and the new path is calculated in the leftmost image. The trees are linked in that image using the red lines from every point along the driven path where it has been seen. The links are drawn at the same length and angle to the new path and the tree is moved to that location in the middle image. The rightmost image represents the modified map after the rebuilding process. The brown points represent the original trees and orange points are the modified trees in the map.

6.5 Post Processing the Map after Measuring Run

A lot of measurement data is collected to the map data structure while driving and measuring. The best estimate of the map is calculated using all the collected data with all features. The locations of the trees and circle radius estimates at every measured height are estimated using the median of data. The median is used as it takes only the measurement in the middle of the set and thus measuring errors do not affect to the final result so much.

The mapped forest data set has many measured features from every mapped tree, from ten to a nearly hundred feature sets at every measured height at normal measurement run. If the features are measured without statistically biased errors, or the biases are measured using a large amount of data and the biases are reduced from the data afterwards, the estimate can be given accurately. In this case when there is a lot of noise in the measurements the median is quite good estimate for the parameters.

Chapter 7

Results

Forests can be measured and mapped with 2D and 3D laser scanners simultaneously using the algorithms described in this work. The process requires data from both laser scanners as well as from IMU. The 2D scanner data is mainly used to calculate the movement of the measurement vehicle. The inertial data is used to estimate the 3D rotations of the vehicle. Using estimated pose information 3D scanner data can be collected so that a 3D image of environment can be generated while the measurement vehicle is moving. Finally a map is generated from the measured features extracted from the 2D and 3D scans. The data from the 3D and 2D maps is used to adjust the accuracy of the pose estimate. The GPS data is used only for fixing the map to the global coordinates.

7.1 Quality of the Tree Recognition

The trees are found from the data using the different methods described in the previous chapters. The method for the 2D scanner is documented in Section 5.1 and it is quite similar to the earlier methods used in Jaakko Jutila's master's thesis [9]. The main difference is that the trees are searched from the combination of four partial scans instead of using only singular scans.

The quality of the found trees from a single scan is shown in Figure 25. The tree is measured only in the horizontal plane of a 2D scanner. While the scanner moves and sways along the measurement vehicle, the plane moves and rotates along. This plane may cut the surface of ground within the measuring range of the scanner and thus the ground might be falsely recognized as a large tree. In addition all bushes and branches that fall into the way of laser scanner measurements are interfering with the recognition and measurement of the tree trunks. These effects can be seen in Figure 25, where laser measurements have reflected from the ground in the right part of the figure. Some branches are confusing the tree diameter measurement at the middle part of the figure where the red circles are quite large.

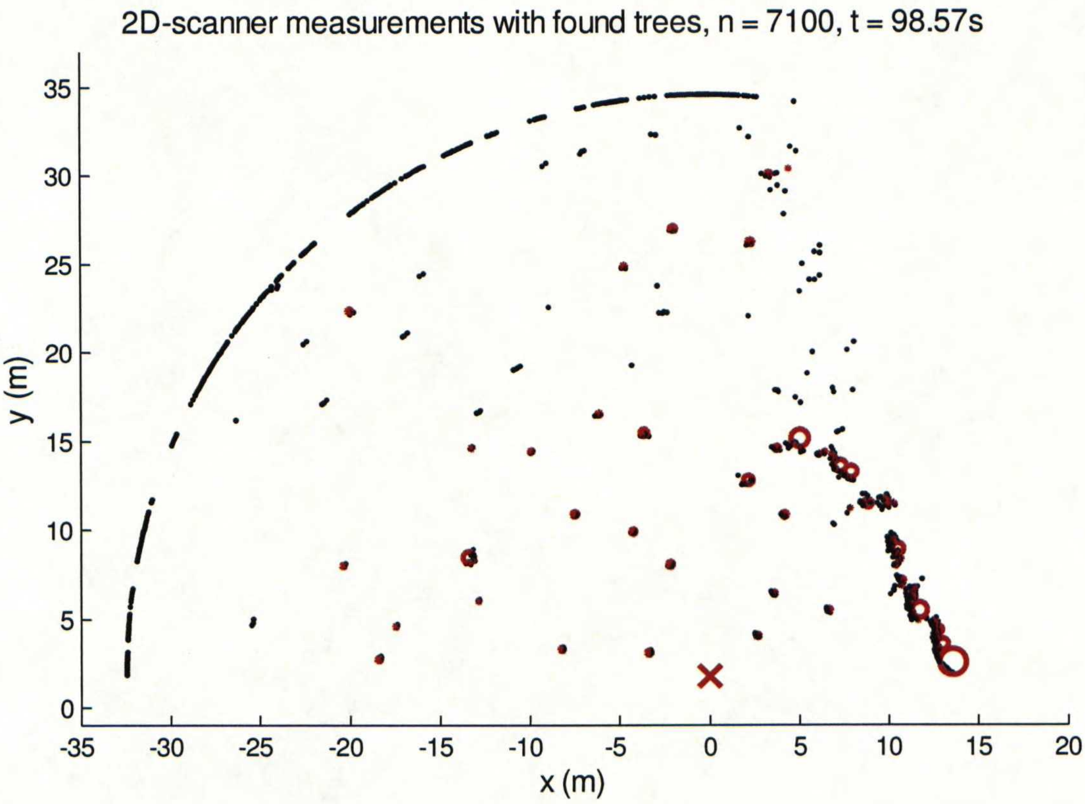

Figure 25: The 2D scanner measurements are drawn with black dots and the found trees with red circles. The measurement is done by combining four sequential 2D laser rangefinder partial scans. The laser scanner has measured ground at the right side of the image as there are lots of wrongly found trees next to each other. The red cross is representing a location of the 2d-scanner.

If the forest does not have a lot of undergrowth, branches or twigs, the 2D laser scanner can be used to identify and measure the trees. Usually well cared forests are environments where the 2D scanner system could manage, especially when there are only a few lower branches. However when measuring denser and more complicated environments a 2D scanner is not giving enough measurements from the trees to recognize and measure them reliably.

Because the 2D data is not usable in all cases alone, the 3D data obtained with the self-made 3D scanner is used to get a lot more detailed information about the surrounding environment. Methods for handling the 3D scanner data are documented in Section 5.2. Using the laser odometry information described in Section 4.2 and inertial information described in Section 4.1 the 3D scanner data can be collected while the measurement vehicle is constantly moving. The

3D data is collected using the shortest possible time for getting a solid data set from the surrounding environment, as the accuracy of estimated pose is decreasing as a function of time. The measured changes in the estimated pose are used to eliminate the effects of the movement in the data of the 3D scanner as described in Section 5.2.1.

The quality of the 3D scanner data can be seen in Figure 26, while mapping a pine forest with a few lower branches, and in Figure 27, while measuring a harder spruce forest. In the first image the tree trunks are easily found since there are only a few measured reflections from undergrowth or branches. The measured reflections from the ground are filtered out from the image using methods described in Section 5.2.3. The image consists of only the measured edge points of the trees as then the reflections from branches and twigs can be nearly avoided. The edge detection and filtering method can be used as the 3D scanner is self-made and has dual 2D scanners at an angle of 45° towards the rotation axis. This method is documented in Section 5.2.2.

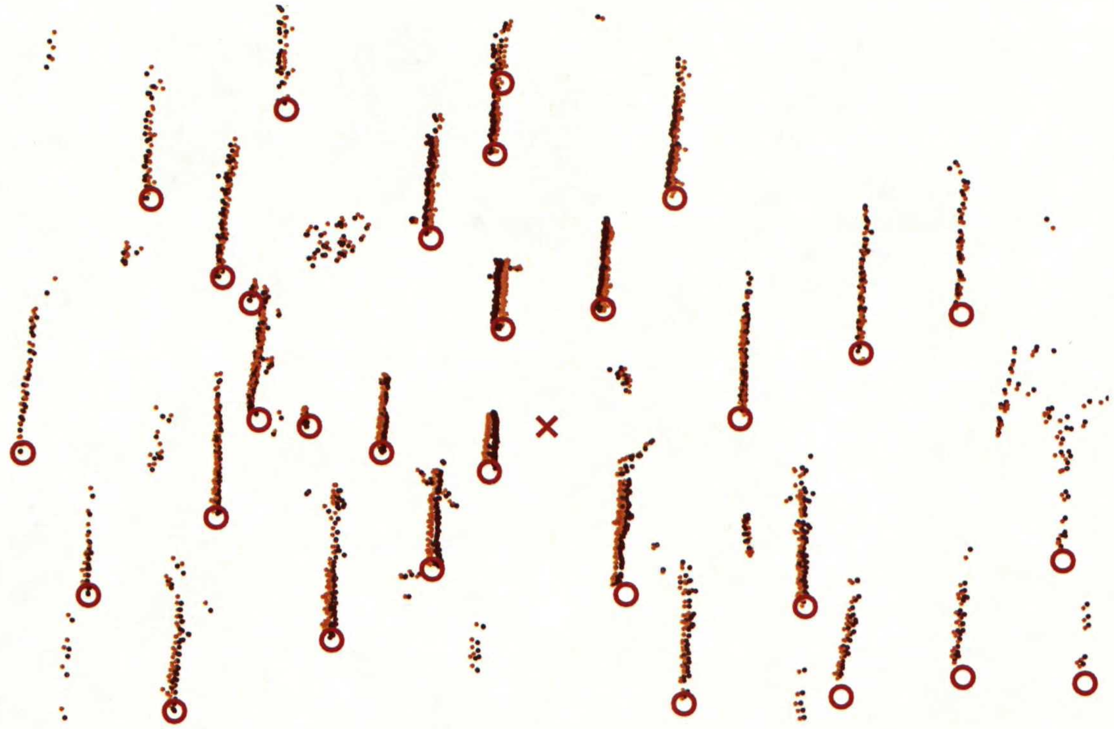

Figure 26: A bird's eye view of the filtered 3D scanner measurements during a short time. The measured points at the left and right edges of the tree are drawn using brown and orange colors. The recognized trees are marked with red circles in the ground level. The red cross in the middle of the image represents the location of the 3D scanner.

The red circles in Figure 26 and Figure 27 are showing locations of the recognized trees plotted in the ground level. Both images are viewed indirectly from above the measuring location and the ground level is scaled to zero height. By comparing the figures, it can be seen that most of the tree trunks can be found even though there are lots of noise in a 3D data set. The red circles are the locations of the recognized trees and the red cross marks the location of the 3D scanner. Both figures are scaled so that the visible trees are at most ten meters high and the visible area is about 35 meters wide. The tree recognition procedure is documented in Section 5.2.4 and 5.2.5.

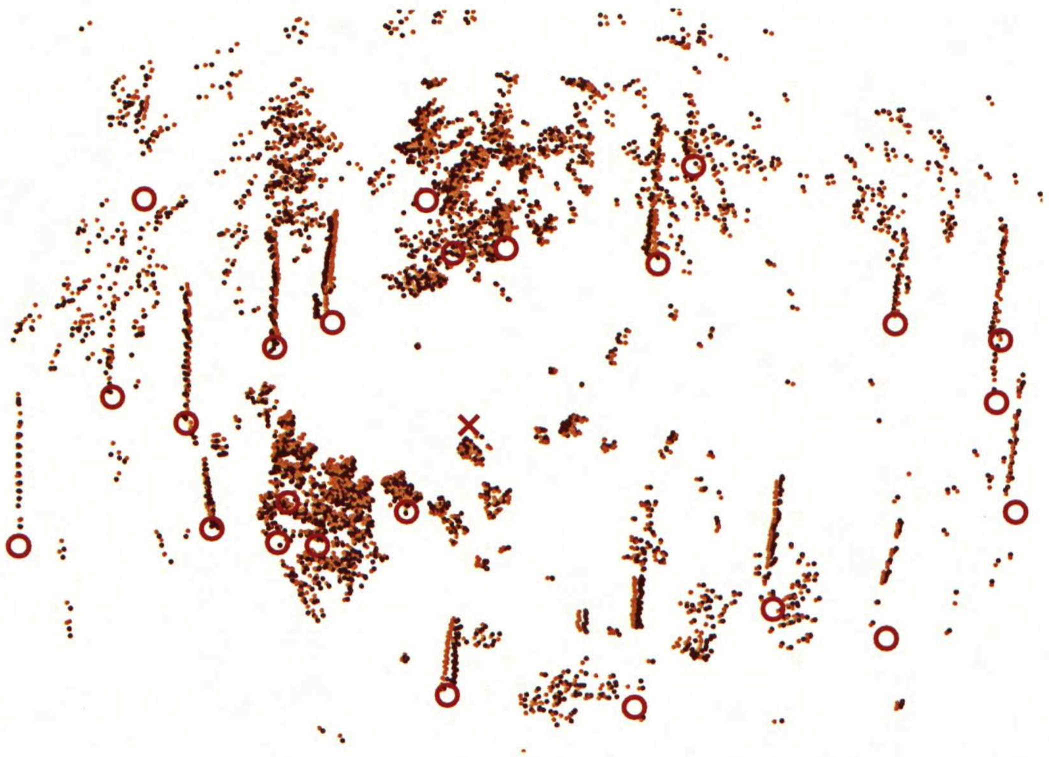

Figure 27: A bird's eye view of the 3D scanner measurements during a short time in the spruce forest. The trees can be found even though there are a lot of noise in the measurements from the branches and twigs around the trunks. All markings are similar as in Figure 26.

7.2 Tree Measuring Accuracy

The tree diameters are measured similarly as the tree locations. The 2D trees are found by fitting circles into the 2D scan data as previously stated. The 3D trees are found using more complicated methods from the collected 3D data sets. All the tree measuring methods and algorithms are documented in Chapter 5.

Tree diameter and location measurements have some unreliability in the measuring accuracy. As there are no proper absolute reference measurements available from the measured forests, it seems better to use mapped estimates as absolute values to evaluate measured data. This method is not giving any estimate for the bias errors, but it is still reasonable to calculate some statistics of the mapped data as the accuracy of the measurement system can be evaluated using the statistics of the whole measurement data. In addition the bias errors can be reduced from the final map afterwards if the bias errors are calibrated with accurately measured reference trees.

The diameter measurements of a single tree measured with the 2D scanner while approaching the tree are drawn as a function of measuring range in Figure 28. It can be seen that the diameter, modeled using the fitted circle radius, is a lot smaller as the tree is measured in short range and the variation in the measurements as well as the calculated average radius is increasing as measuring range increases. There are some outliers below and above the main group of the measured points as the system has failed to measure a tree accurately and therefore the standard deviation is such a large. The estimated radius is the median of the whole data set.

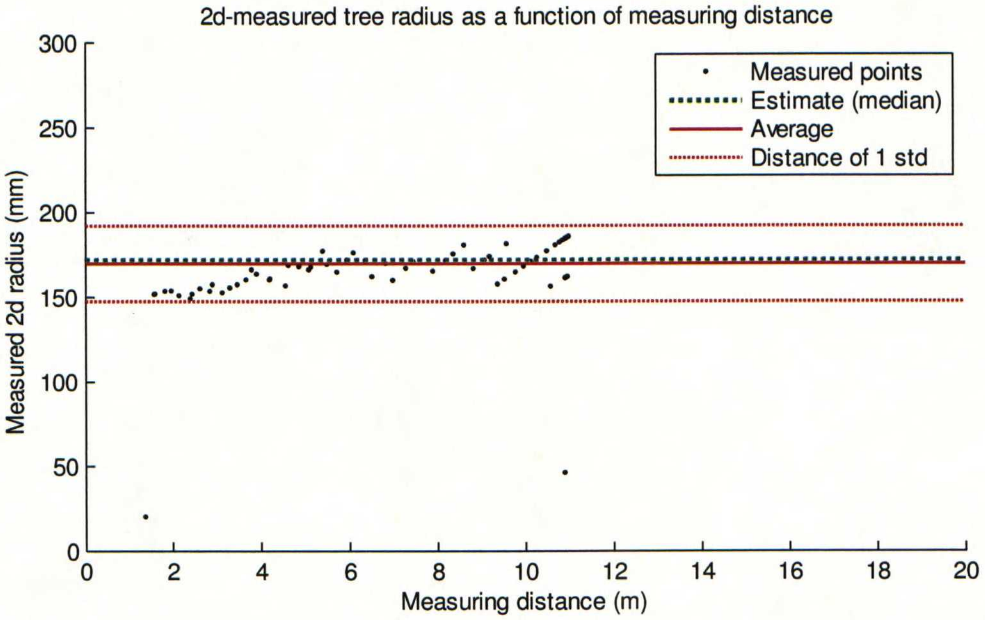

Figure 28: The 2D measurements for a single mapped tree measured at different ranges. This is the same tree than in Figure 29.

The tree diameters measured using the 3D scanner are similarly collected from the whole data set and drawn to Figure 29. The tree diameters were modeled with the fitted circle radii from ten different heights at one meter intervals, but only radius measurements at one meter height are drawn here as they are the closest ones compared to the 2D scanner measurements. As seen from Figure 29, the 3D measurements are systematically smaller and they have larger variation than the 2D measurements in Figure 28. The regions in the measuring data that have a large amount of radius measurements at the same measuring range have formed when the measurement vehicle has been stopped in the beginning and in the end of the approach drive.

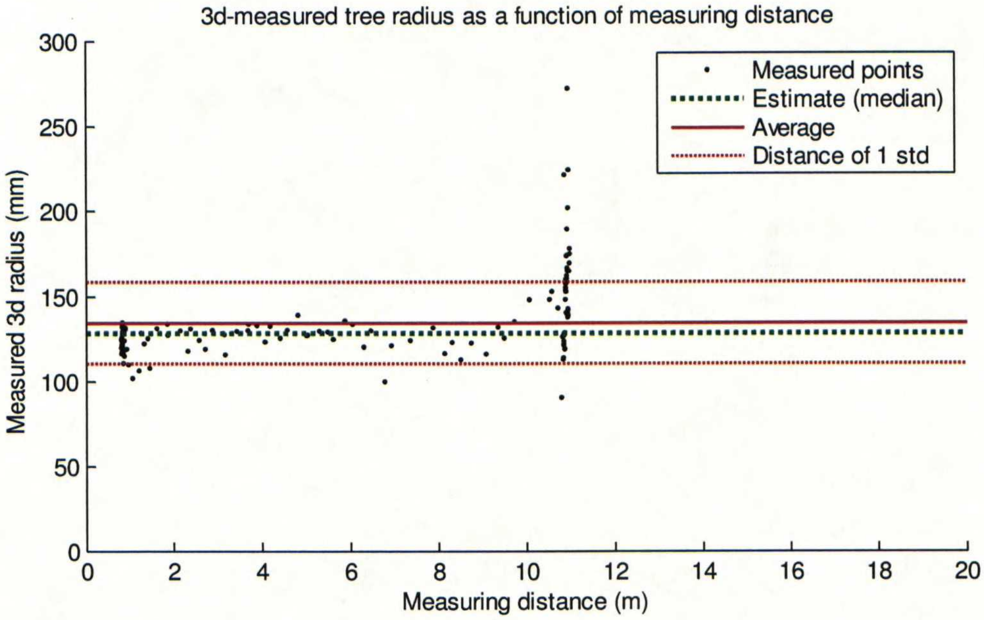

Figure 29: The 3D measurements for a single mapped tree measured at different ranges. This is the same tree than in Figure 28.

The previous measurements are only concerning the same single tree and thus they do not represent the whole data. They are viewed to describe the differences between the 2D and 3D laser scanners and the different tree measuring methods. By comparing these two figures, it can be seen that the median of the 3D measurements is approximately 40mm smaller than the median of the 2D measurements. By knowing the measuring algorithms, it is likely that the 2D measurements are overestimated such that the tree diameters in short measuring range are nearer the correct tree diameter. The 3D measurements are probably underestimating the real tree diameter as circles are fitted to the noisy measurement data using the method described in Section 5.2.7.

The measurements cannot be straightforwardly compared between the trees as they have different size. Therefore all the tree measurements are first reduced by the estimated median radius and then plotted in the same graph. The error to the estimated radius is more comparable between the trees and more data can be used to calculate the statistics for the measurement system. The errors in the estimated measurements are plotted in Figure 30 for the 2D and in Figure 31 for the 3D measurements.

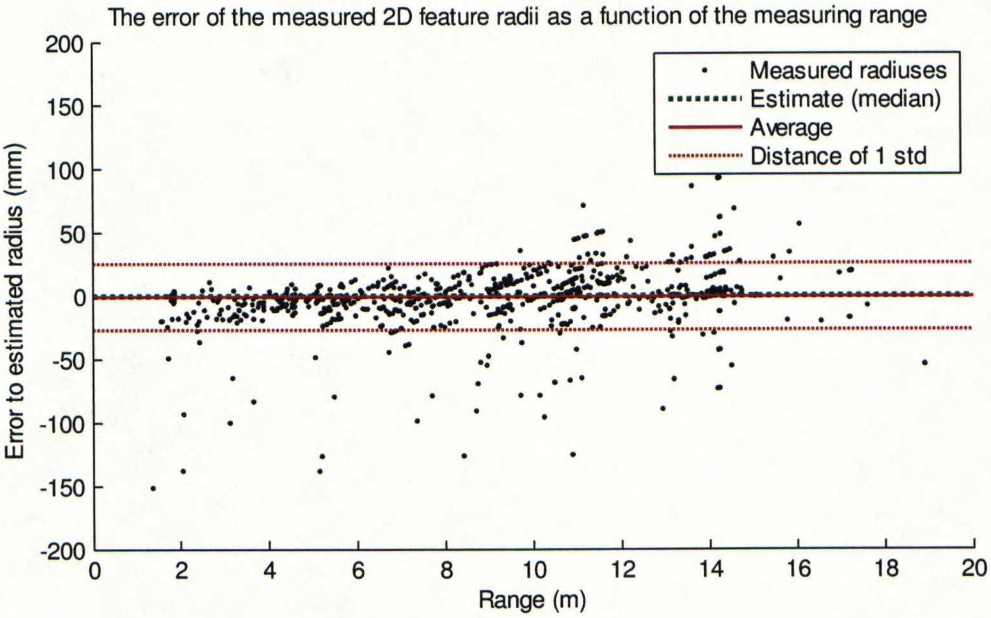

Figure 30: The error of 2D tree diameter measurements of the whole mapped forest as a function of the range.

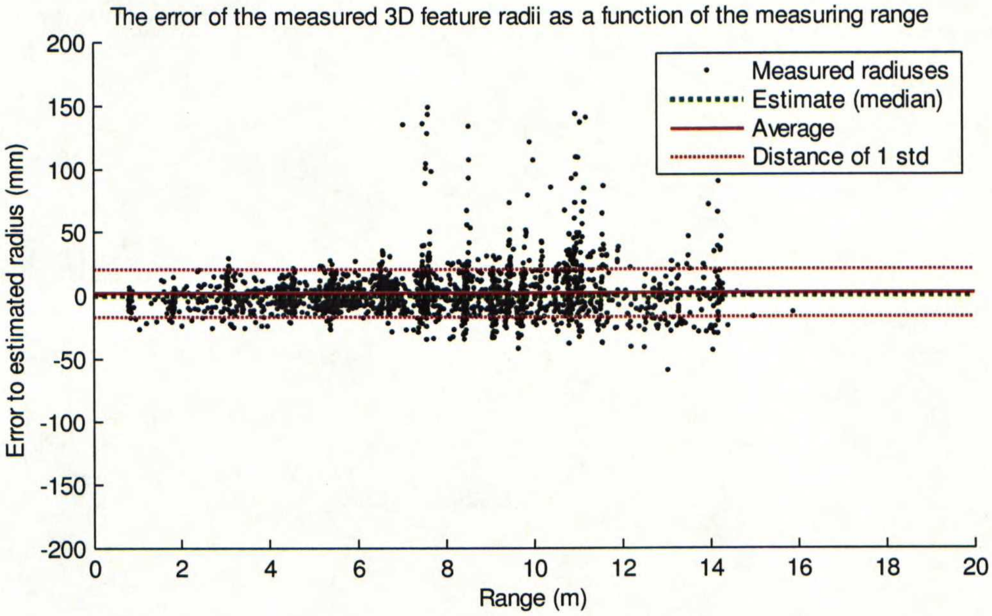

Figure 31: The error of 3D tree diameter measurements of the whole mapped forest as a function of the range.

While comparing these two figures, it can be seen that the data from the 2D and 3D laser scanners and the measuring algorithms for them behave quite dissimilarly. The 2D scanner measurements have many measurements where a tree is measured a lot smaller than it really is. It can also be seen that the most of the 2D features are measured larger as the measuring range is increasing. This kind of behavior cannot be seen in Figure 31 concerning the 3D measurements, where the underestimated measurements are rare. On the other hand overestimated measurements are more common in the 3D scanner, but the estimated median is quite the same regardless of the measuring distance. The standard deviation is a lot larger in the 2D graph as there is more noise in the measured 2D features and a lot less points affecting to the estimation process.

The tree locations are similarly compared using the estimated locations as the tree diameter measurements. The exact reference data from the measured forest is not available and thus the correct tree locations must be estimated and all the location measurements are then compared to these estimated values. The estimates, which are visible in Figure 32 with all the location measurements for the same tree, are calculated using median in both directions separately. These tree locations are the center points of the fitted circles which are collected during a tree approach run. The 2D trees are usually measured a little further away than the 3D trees as shown in Figure 32. The tree shown in Figure 32 is the same tree that has been approached in the measuring run and which has been used as an example in the previous figures.

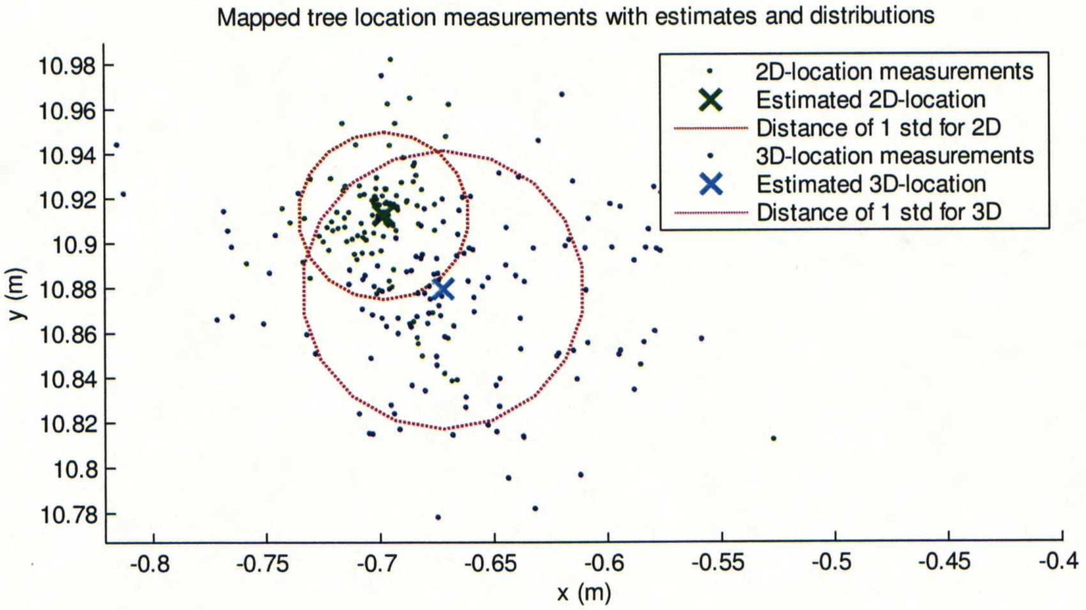

Figure 32: The mapped tree location measurements with the estimated locations and the calculated standard deviation error ranges around the estimated location. The tree here is the same tree number 15 than used in the previous figures in this section and the same tree is later on mapped using the median estimates in Figure 35.

Figure 32 shows the typical differences between the location measurements of the 2D and 3D scanner systems. In both point clouds the distributions of the measured locations of the same tree are similarly round-shaped. The only difference is that the deviation in the 3D location measurements is larger than in the corresponding 2D location measurements. There is some difference between the estimated locations in the tree shown in Figure 32 as well as in the all other paired location estimates. That is why the 3D and 2D maps are not mixed together as it is known that there might be some errors between these two maps. The error sources for both devices are different and therefore there are some bias-errors in both measurements. Those biases can be calibrated out of the measurement system only when the exact reference data from the forest is available.

Similarly as the tree diameter measurements, the location error measurements are plotted as a function of the measuring range to visualize the error distribution. The errors are calculated as the Euclidean distances to the estimated tree locations separately from the 2D and 3D measurements. Figure 33 shows errors of the 2D locations and Figure 34 the same errors for the 3D locations. The data used to generate both figures is same that is used to generate the previous graphs and an upcoming map on the approach run in Figure 35.

By comparing these figures, it can be seen that the 2D scanner can measure the tree locations quite accurately from greater ranges than the 3D scanner. It can also be seen that the location error is smaller in the 2D measurements at all ranges while the accuracies of the 3D location measurements are degreasing quickly as the measuring range increases. The measuring accuracy of the rotation encoder of the self-made 3D scanner is maybe the key factor for the degraded location measuring accuracy in this case. On the other hand the used 3D data is collected during a longer time than the 2D data and thus the inaccuracies caused by the motion compensation may increase the uncertainties in the 3D location measurements.

In addition to the diameter measurement figures, these two location measurement graphs have a standard deviation calculated in a moving window of ±5m. It clearly shows the increasing uncertainty in the location measurements while the measuring range is growing. The location measurements in Figure 33 and Figure 34 are calculated using the whole mapped measurement data from all the trees separately from the 2D and 3D scanners. The errors in the tree location and diameter measurements are quite large, but they could be reduced by using more accurate or faster sensors. This work shows that by using only an inaccurate data, localization and mapping can anyway be done quite accurately.

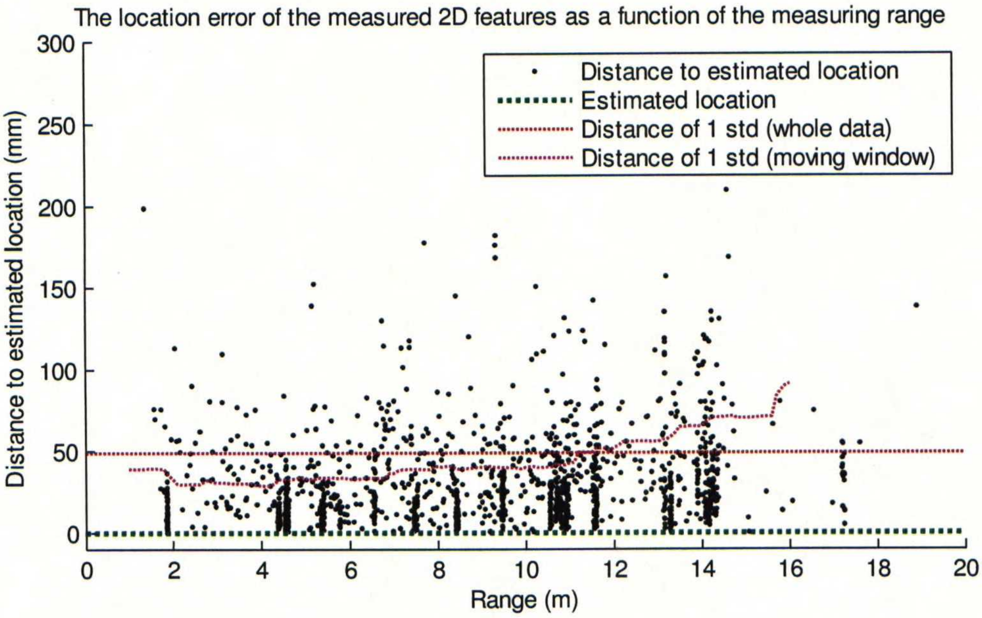

Figure 33: The location error of the trees measured using the 2D laser scanner. The data set is acquired from the whole measured forest.

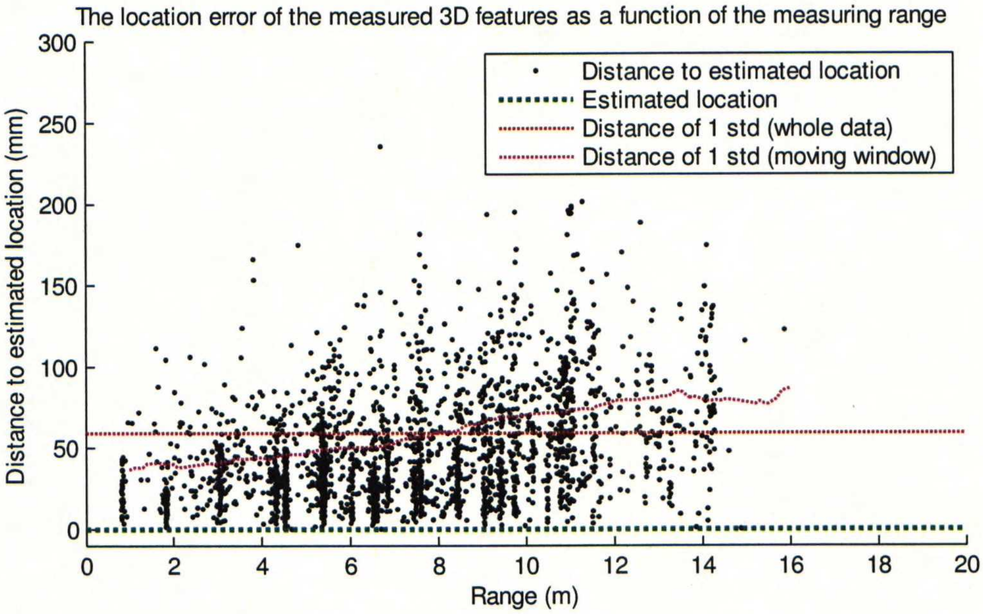

Figure 34: The location error of the trees measured using the 3D laser scanner. The data set is the same as used in Figure 33.

The map and the path that has been used while calculating these errors is shown in Figure 35, where the red line is representing the driven path approaching the tree number 15. The brown poles in the figure are the estimated 3D trees and the black circles are the estimated 2D trees. Numbers in Figure 35 are indexing trees. The drive consisted of the approach of the tree number

15 and the reverse drive back to start. The measurement run was used to measure a single tree at different ranges. More information about the accuracy of mapping is presented in Section 7.5.

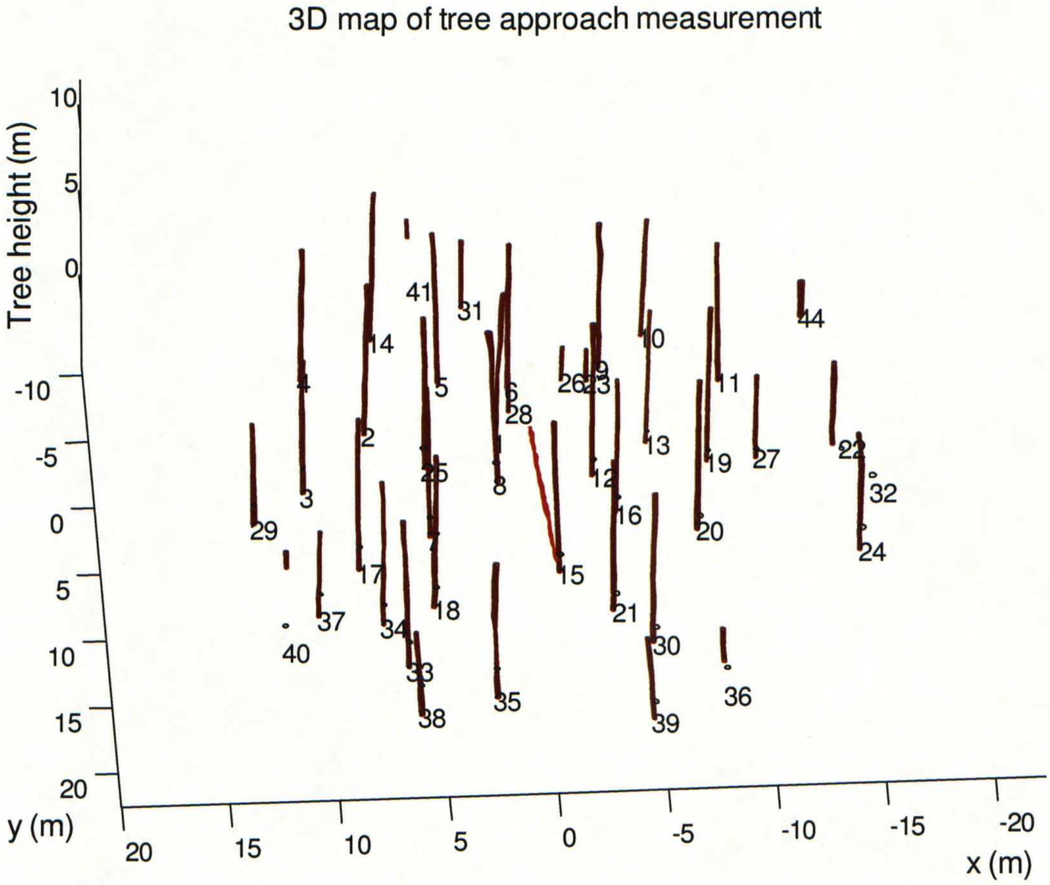

Figure 35: The map of the tree approach measuring drive, from which the tree measurement data is calculated for the previous figures. The tree number 15 is the approached tree, which is also shown in the previous figures. The tree models in the figure are drawn using the median of the features. Only those features that have been managed to measure sufficiently accurately are drawn in the figure.

7.3 Quality of the Ground Estimation

The main part of the measurement points are reduced from the ground measurement group to speed up the ground model calculation process as stated in Section 5.2.3. A lot more measurements are got from the ground around the measurement vehicle than from farther. The difference between the point sets measured from the same forest before and after the use of the statistical measurement selection method is shown in Figure 36. The target density of the measurements which is needed to the uniform distribution is also drawn to the same figure to demonstrate the quality of the method.

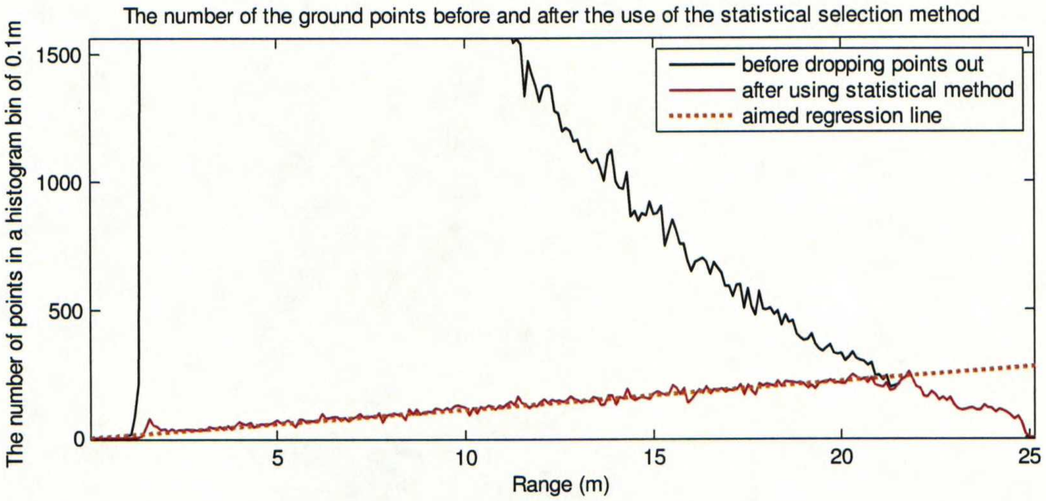

Figure 36: The histogram of the measured points associated to the ground and the histogram from the same measurement run after randomly discarding unnecessary measurement points using the statistical method described in the section 5.2.3. The black curve is peaking to the value of 18000 measurements around the range of 1.6m.

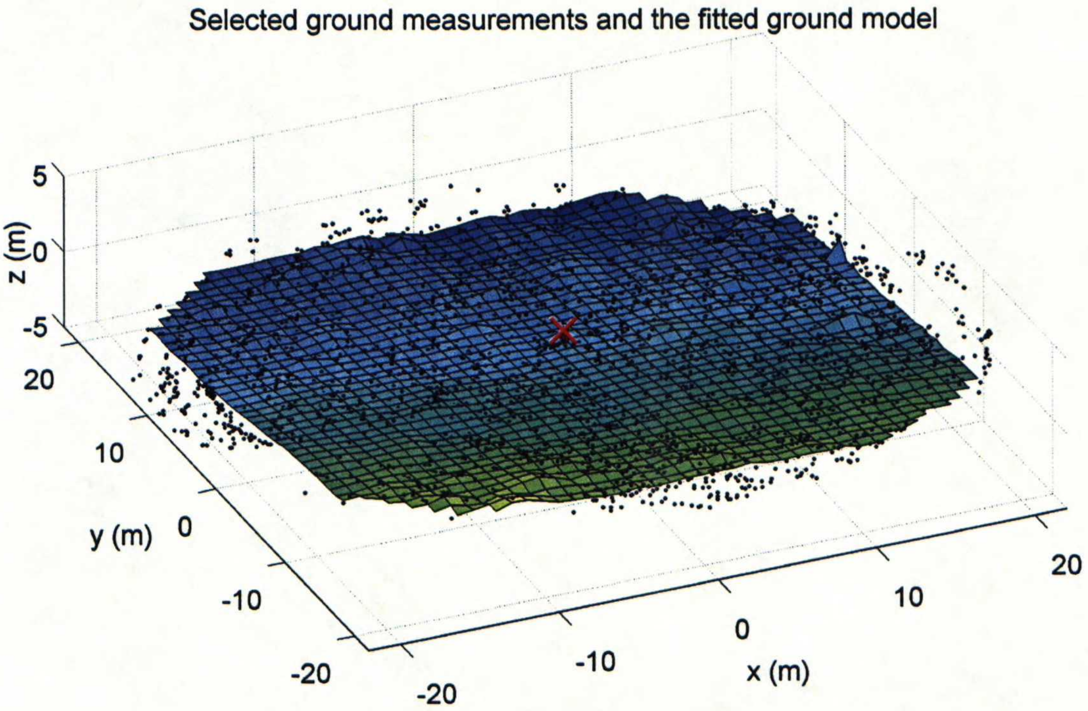

Figure 37: The fitted ground model with the selected measurements from the ground associated to it. The red cross in the middle is representing the location of the measurement vehicle.

After the ground points are gathered to the ground cache during a few seconds, a ground model is generated before the trees are searched from the main measurement cache. The ground model is calculated by dividing the area to one square meter grids and averaging the heights of the measurements around every grid to estimate the ground height around that grid point. An example of the ground model with all the measured points associated to it is drawn in Figure 37. If there are no ground measurements in around some of the grid points, the height is calculated using the height-values of the surrounding grids.

7.4 Localization Accuracy

7.4.1 Correlation Based Laser Odometry

The correlation based laser odometry method, presented in Section 4.2.1, is used either with one or with two scanning laser rangefinders. The method works well as two scanners are used to get nearly 360° FOV. In these circumstances the rotation and translation components can be separated from each other and the results are accurate. The errors are drawn in Figure 38 for the iterated and non-iterated versions of the same algorithm with a version of the algorithm where the CM method is not used. The real path used as a reference is estimated using the mapping algorithm described in Chapter 6.

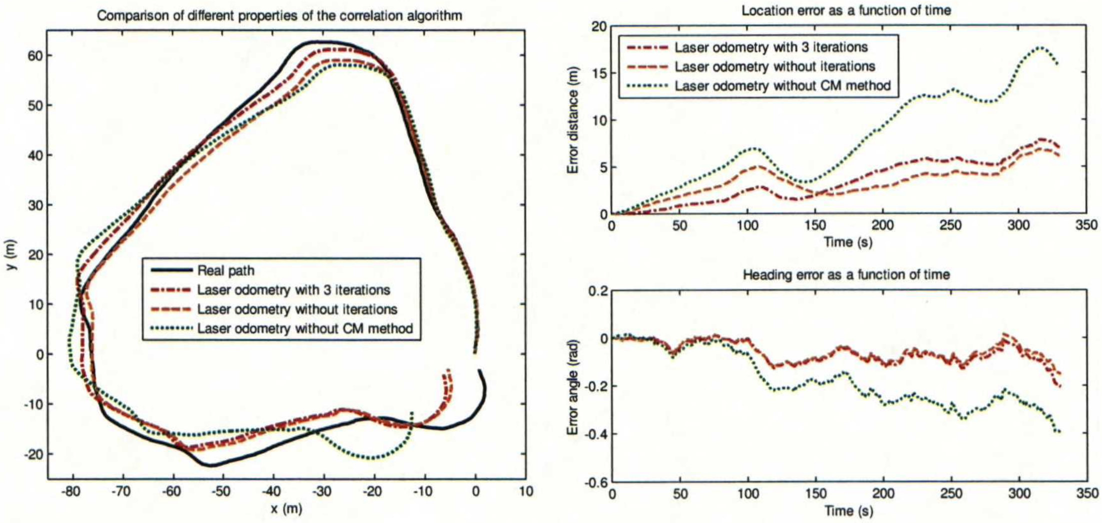

Figure 38: Comparison of the different properties of the laser odometry method along the 260m long test loop with the calculated location and heading errors. The real path is estimated using the forest mapping method described in Chapter 6.

The localization errors in Figure 38 are measured as the Euclidean distances between the paths of compared methods and the mapped path. The heading errors are measured as the differences of rotation estimates between the compared methods and the rotation estimates along the mapped path. All the errors are shown as a function of time in the same Figure 38.

The total odometry based localization error after closing the 260m test loop in 330 seconds is 6.2m when the algorithm is not iterated. When three iterations are done, the localization error is 7.1m. Angular errors in the heading are also small: 0.15 radians for the not iterated version and 0.2 radians for the version with three iterations. By this far the used method seems to work better without the iterations. All measured errors are represented in Figure 38.

While observing only the previously listed end results, the idea of iterating the algorithm might seem useless. However the first one hundred seconds in Figure 38 show that the iterated version of the algorithm is anyway more accurate. The first larger error in the heading around the time of 100s is adding the largest error to the iterated version dropping the total measured accuracy of the method.

When the center of mass (CM) method is not used to get a sub-integer accuracy estimate for the cross-correlation result, the accuracy of the laser odometry dropped a lot. The localization error was over 15m and the heading error was 0.4 radians. As the CM method is not used, the estimated transformation is decided using the largest correlation value. There is therefore more bias error in those measurements, and there are larger errors in the heading and in the location estimate.

While using only one laser scanner either pointing ahead or sideways, the rotation and translation components cannot be separated sufficiently accurately and the iteration converged randomly to a slightly wrong point between the correct rotation and translation components. For instance, when the rotation is the most significant component, the measured result translates too much and rotates too little and vice versa. This problem can be avoided using the inertial measurements to estimate the rotation component as described in Section 4.2.3.

7.4.2 Iterative Closest Point Method for Laser Odometry

There are a few drawbacks that complicated the use of the ICP method in the forest measuring case described in Section 4.2.2. Firstly the method is iterative with usually many iteration steps to gain a sufficiently accurate estimate of the pose change. Secondly the method has too high probability to converge into a local minimum instead of the global minimum. That means that the method can occasionally result into a large localization error as normally the method is functioning accurately.

The results of the ICP method for calculating the pose change of the measurement vehicle are not as good as the results of the other used methods. The heading angle jumps occasionally to wrong angles and the accuracy of the translation is not as good as in the other methods that are used in this work. The resulting path did not resemble the mapped path as there are too much occasional errors. Besides the low accuracy, the ICP method is computationally expensive.

7.4.3 Feature Based Laser Odometry with Inertial Measurement

The rotation and translation problem cannot be solved satisfactorily, when using only one scanning laser rangefinder using only the scanner data. However while using the heading angle of the measurement vehicle measured with IMU, the pose change problem can be reduced to the translation problem as stated in Section 4.2.3. The localization accuracy using one laser scanner pointing ahead and IMU for acquiring the heading angle is shown in Figure 39. As seen in the figure, the heading angle is slowly drifting as the inertial measurements have some noise. The most important thing is that there is no irregularities like fast turns or hops in this measured laser odometry path, and thus the method can be safely used to estimate the location although the method is not absolute accurate.

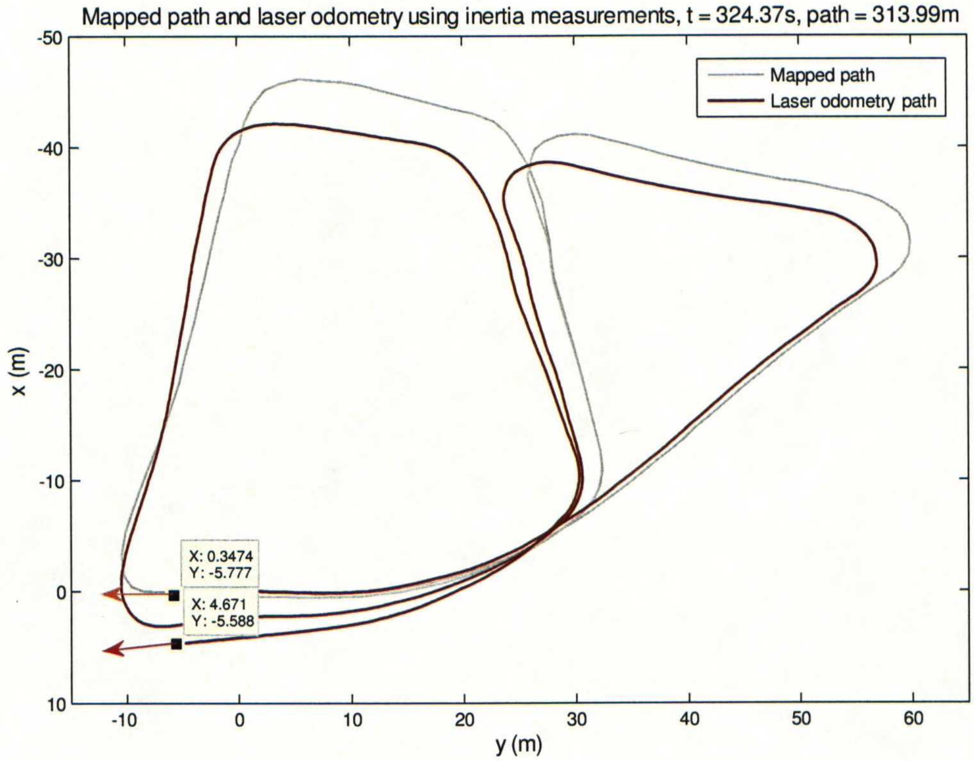

Figure 39: The localization accuracy using the inertial heading angle with laser odometry. The heading angle is used to simplify the process of finding the right rotation and only the translation is calculated using the laser odometry method described in Section 4.2.3. The mapped path is assumed to be a lot more accurate than the laser odometry path and thus an accumulating localization error can be measured between them.

The localization error on the test run shown in Figure 39 is 4,33m as a Euclidean distance. The error is measured between the end locations of the driven and the mapped path, which is assumed to be near the real path in the forest. More important here is the measured angular error of 7.1 degrees between the orange and red arrows. It is caused by the inaccuracies in the IMU measurements. If the error is generated equally trough measuring time the angular error rate of the IMU is 0.022 degrees per second.

7.4.4 Global Localization

The global localization of the map is done using GPS receiver. Because of the deficiencies of the GPS receiver listed in Section 3.5, the accurate location of the measurement vehicle is not available without a long term average of the data. The averaging is done using the SLAM path as reference position for the GPS measurements and minimizing the distances between the projected GPS and the map paths as described in Section 4.3. The end result of the fitted GPS path with the map path after a full test run is shown in Figure 40.

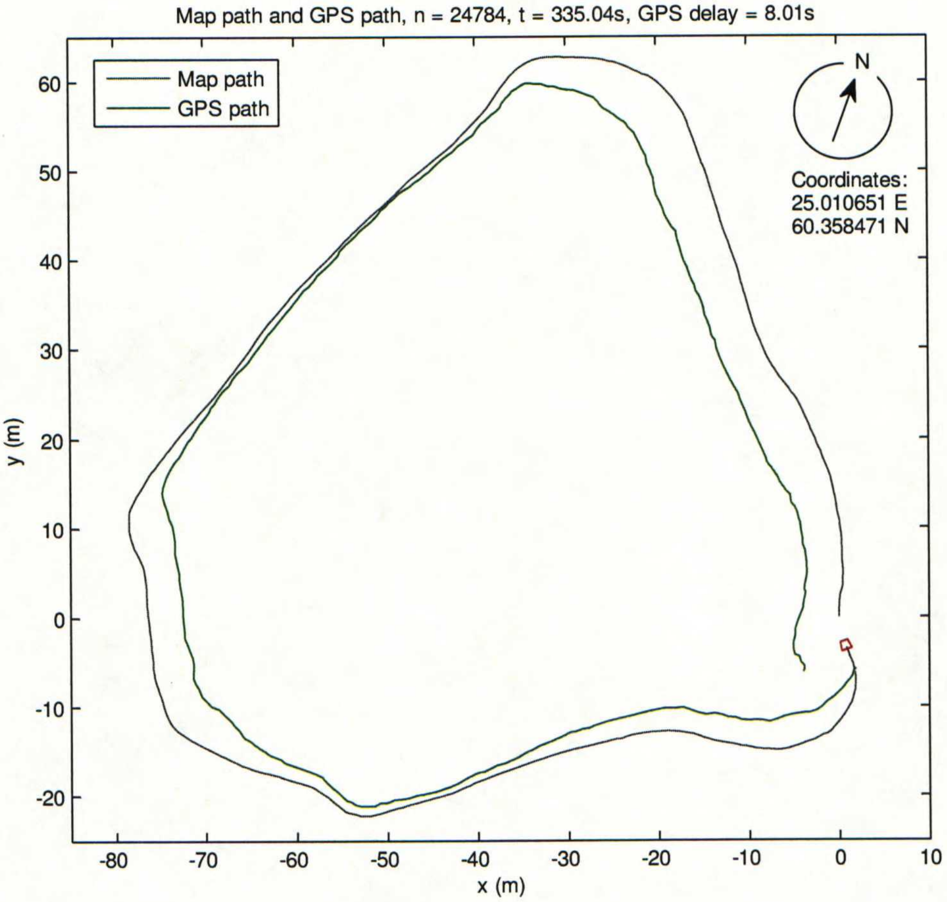

Figure 40: The map path and the GPS path measurements are plotted using the global coordinate projection method to fit the map path to global coordinates. The used GPS receiver has about 8 seconds delay on its measurements. The coordinates shown in the figure are representing the estimated global location of the measurement vehicle.

As shown in Figure 40, the GPS path is not very accurate; as the map path fitted using the laser odometry method and the mapping method is much likely nearer to the right path in the forest than the GPS path. The error seen in the projected GPS path is made up firstly by the localization error of the GPS receiver. Secondly the error is generated by the projection method used to project the global GPS data to the plane around the selected reference point. As shown in Section 4.3.1, the maximal error of the projection method is only a few centimeters on maps like the one in Figure 40.

The averaging of the GPS location is minimizing the effect of the bias error in the global fixing of the map, if the measuring time is long enough. This is only a hypothesis, and maybe the measurement time is never long enough to average out the GPS bias errors. The reference point for GPS is fixed to the origin of Figure 40 and the coordinates drawn in the upper right corner of the figure are representing the current global location of the measurement vehicle. This global fixing is saved to the resulting map and it can later be used to fit multiple maps together in the larger global collection of forest maps.

7.5 Mapping

The quality of the mapping process is difficult to demonstrate. All different parts of the system are affecting to the quality of mapping. Inaccuracies in the tree measurements are making the whole mapping process noisier as the same trees are not measured exactly in the same place at every measuring cycle. Similarly the imperfect pose estimate of the measurement vehicle is adding more noise to the tree location measurements and consequently impairing the quality of the map. Therefore the quality of mapping had to be measured as a quality of the whole system.

At first a typical fit of the found trees to the mapped trees is shown in Figure 41 to demonstrate the localization, the mapping and the tree search together. In the figure, the 2D and 3D maps are drawn on top of each other; the mapped 2D trees are drawn with the green and 3D ones with the brown circles. Both maps are related by links between the mapped trees so that every tree has an identity in the both maps. The 2D map is only used to fit the found 2D trees and similarly the 3D map for the found 3D trees. The mapped tree objects which are on the range of the scanners are surrounded by light blue diamonds. Similarly the new trees that are just added to the map are surrounded with red diamonds.

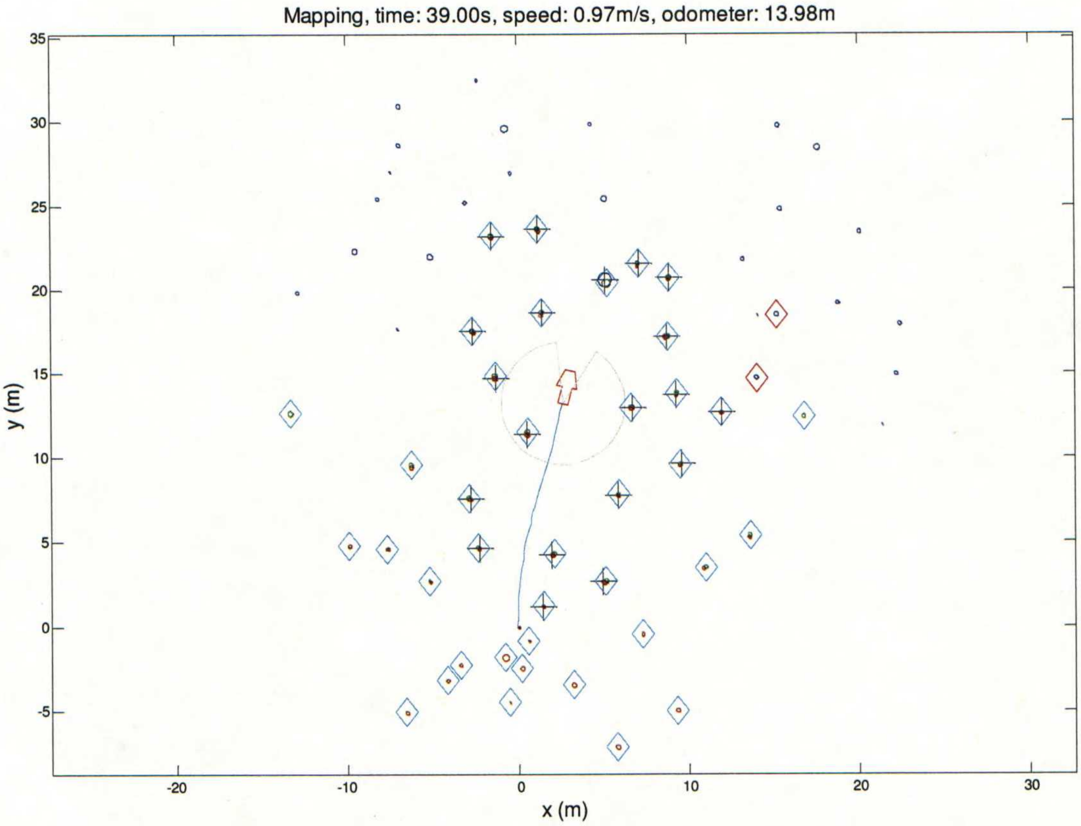

Figure 41: The ongoing mapping process while driving in the forest. Blue diamonds are marking the mapped trees inside; Plus signs are marking the found trees that have been paired with the corresponding mapped trees and red diamonds are drawn around the trees which have been just added to the map. The grey circle shows the accumulated maximal localization error and the cut slice in it shows the estimate of the accumulated maximal angular error. These estimates are used to limit the search of the old trees as the old path is encountered.

The plus signs in Figure 41 show the locations of the found trees from the current tree recognition process. These found trees are paired with the corresponding mapped trees and the location estimate is updated. The dark-blue circles in the front sector are representing the found 2D trees which are so noisy that they are not alone trusted to add new trees to the map. The gray circle around the red arrow, which represents the measurement vehicle, shows the estimated accumulated maximal localization error. The cut sector in the circle shows equivalently the estimated accumulating maximal angular error. The estimated maximum errors are greatly overestimated to ensure that it would be the absolute maximum estimate as it is mainly used to limit the search of the pose error while the previous path is encountered.

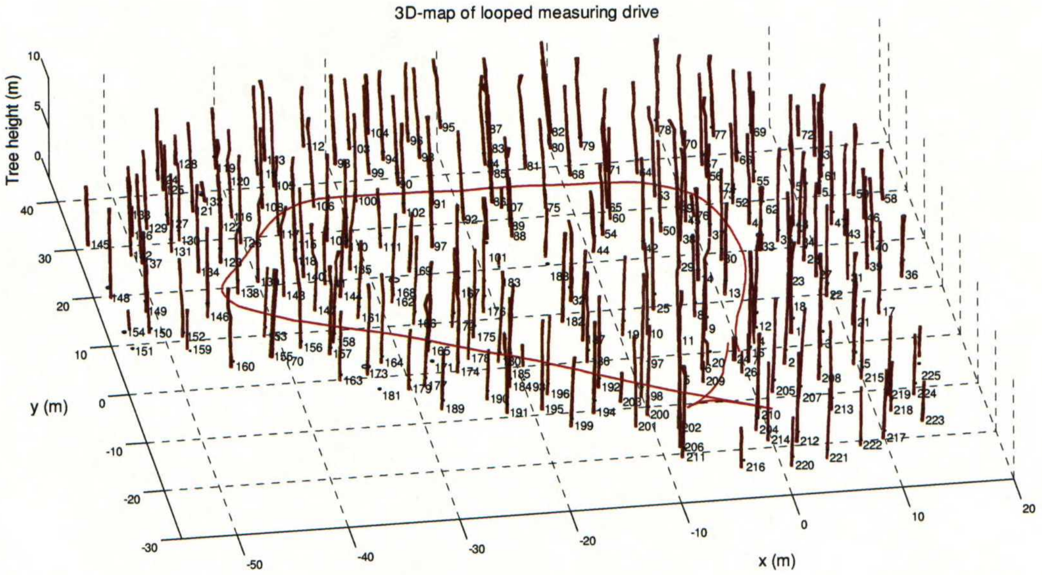

Figure 42: The collected 3D map from a closed measuring run. Every identified tree is numbered in the finding order.

The resulting 3D map is drawn in Figure 42. The path is drawn with the red line and the 3D tree models are drawn using brown pillars. Trees are measured by fitting ten circles one on the other at one meter intervals to the 3D measurements as previously stated in Section 5.2.7. The tree models are drawn to the figure using those fitted circles with faces between them. As seen in the figure the highest measurements have most noise and erroneous measurements. This can be explained by the smaller amount of measurements from the highest regions. Also the inaccurately estimated rotation is affecting more to higher tree measurements than to the measurements nearer the ground, as the rotation center is near the ground.

7.5.1 Closing the Loop

The path closing method works sufficiently well as the errors in the mapping are usually so small that the generated map is not needed to stretch much. To test the quality of the reconstruction process of the map as the needed map rebuilding transformation is large, a large angular bias error was added to the pose estimation system to ensure that there is enough error in the final pose when the previous map is encountered. The encounter phase is seen in Figure 43, where the same trees are found as duplicates around the red arrow which is marking the measurement vehicle. The 2D scanner is seeing the same trees at a few meters above the previously mapped trees. The obtained error is so large that the right transformation is required to be searched longer than only from the nearest trees.

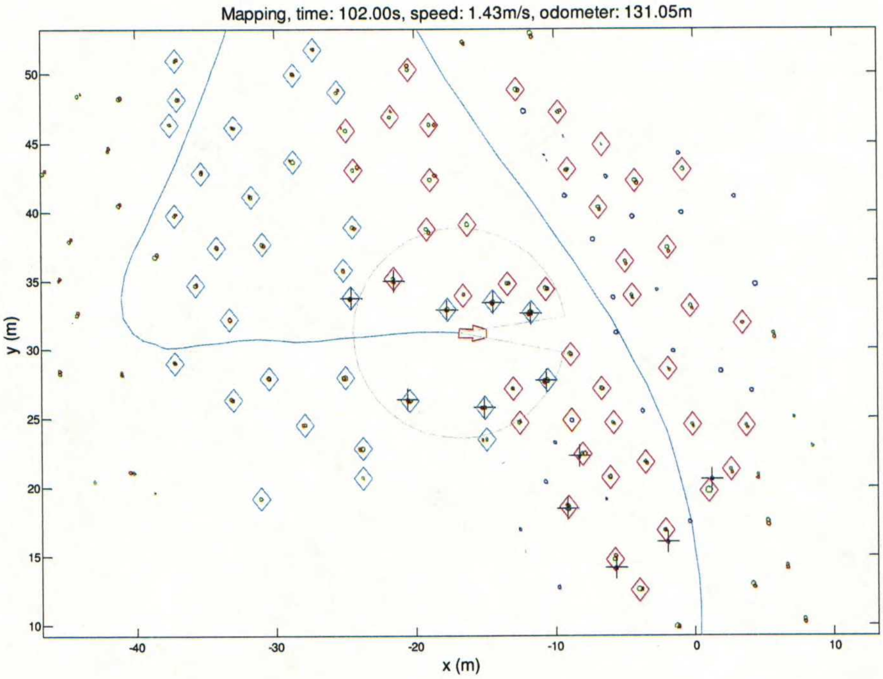

Figure 43: The first image of the map rebuild system in action. After a huge angular bias is added to the system, as normally errors are too small to test the map repairing algorithm. The old mapped trees are drawn inside magenta colored diamonds and the new mapped trees are surrounded by the blue diamonds.

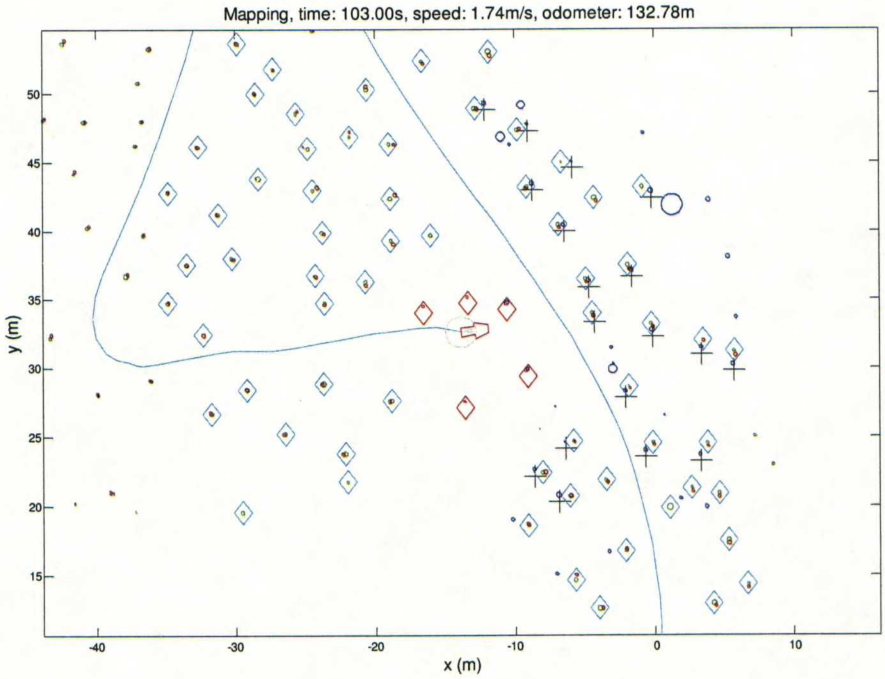

Figure 44: The second part of the path repairing system in action. The transformation is detected and the path and mapped trees are rebuilt. The trees that are duplicated are removed. The size of the created angular bias error can be seen between the paired found and the mapped trees. They are marked with the plus signs and the blue diamonds.

The pose errors are found and repaired in Figure 44 using the method described in Section 6.4. The trees which are mapped multiple times are marked with red diamonds in the figure as they are removed in the process. While comparing these two figures, it can be seen that the path is reconstructed to meet the recognized transformation. The path is in different angle between the figures as the bias has been removed and the remaining map is stretched so that the measured pose error is cancelled. The errors between the plus signs and the corresponding blue diamonds show the size of the repaired error.

Chapter 8

Conclusions

The tree localization and mapping cannot be done accurately with only the 2D laser scanner or the 3D scanner alone. The 2D measurements are usable if the forest does not have a lot of undergrowth or lower branches in trees. The 3D scanner is needed in the more complicated environments, but it cannot be moved during the measurements without an accurate odometry method and IMU measurements. The best result is acquired as the both laser scanners are used together in the laser odometry and in the tree measuring and recognition.

The system is measuring most accurately as it is standing in place, but all the algorithms and methods worked sufficiently well as the machine was moving in the forest at slow speeds. The used methods worked sufficiently accurately as driving at all tested speeds from zero to over two meters per second. While using a fast modern computer, the whole process can be calculated in real-time.

The mapping accuracy is estimated to be a few centimeters for the diameter and a few tens of centimeters for the location of the trees. The pose of the measurement vehicle is firstly estimated using a laser odometry system and then readjusted using the collected map. The localization error of the measurement vehicle is estimated to be smaller than the location error of the trees, as the large group of mapped trees is used to fit the pose of the vehicle.

The global location accuracy depended greatly on the quality of GPS measurements. As the used GPS receiver has slowly changing bias and the measurements are taken only during a short time, the measurements could not be collected enough to remove the slowly changing bias error. The GPS measurements are heavily filtered in the used receiver and thus there are no hops or large discontinuities in the data.

8.1 Method Comparison

8.1.1 Comparison between Different System Setups

The measurements in this thesis were gathered using two different system setups. The earlier setup had only two horizontal two-dimensional laser scanners as the later setup has one 2D scanner and a rotating 3D scanner. An inertial measuring unit and a GPS receiver were used in the both setups. The first setup worked well when the measured forest was flat enough and there were only a little undergrowth or lower branches to hide the tree shapes as there was no 3D scanner to measure the trees. The 3D scanner was a great help for finding and measuring the trees in the forest, as it could see the ground and the trees at different heights around the measurement vehicle. Branches are easier to filter out from the measurements as trees are seen as a whole and not from a single horizontal cut as in the earlier setup. The latest setup with the rotating 3D scanner proved to be a sufficiently accurate mapping system for many different forests.

8.1.2 Laser Odometry Method Comparison

The pose change is estimated using the different laser odometry methods described in Section 4.2. As the 360 degree FOV of the horizontal range measurements was available in the first correlation based method, the measurements were accurate. Then the system had only a little bias error in the pose estimate and thus the estimated location was quite good. The only drawback of the method was that it only worked sufficiently with two 2D scanners and with the FOV of 360 degrees. The method did not work with only one laser scanner with the FOV of 180 degrees. It needs measurements symmetrically from the surrounding environment.

The heading angle can be estimated using the IMU measurements. As the rotation is known, the translation can be easily separated from the laser odometry measurements. As the correlation measurements proven to be inaccurate while only one 2D scanner was used, a feature based method was used to find the groups of points from the measurement data. These groups were successfully compared between consecutive scans. This method is described in Section 4.2.3. This method using the IMU data and the 2D scanner data seems to be the best solution for estimating the pose change of the measurement vehicle.

The ICP method was also tested to fit consecutive 2D scanner measurements together as it is a standard procedure in many papers. The method worked usually, while a fit was good and the transformation was small. As there was too much noise or erroneous reflections from the ground the ICP algorithm failed and resulted to a wrong association. In addition it is quite slow to calculate with much iteration.

8.2 Suggestion of Improvements

The laser odometry method could be improved by either using two horizontal 2D scanners with or using a single 2D scanner that has larger FOV than 180 degrees to replace the old 2D scanner. It would help separating the translation from the rotation between consecutive scans if the presented cross-correlation method is used. It would also give more 2D data to use with other presented laser odometry methods.

The used 3D scanner was self-made. The collected data contained many deficiencies, like variable time delays that were caused by a wireless data transfer link and delays between the 2D scanners from which the 3D scanner was built. Also the lack of an absolute angle encoder complicated the 3D mapping process as the exact rotation of the 3D scanner could not be known. All these deficiencies were later patched up, but the data from which forest maps were calculated in this thesis was collected with the earlier version of the 3D scanner.

The global localization accuracy could be improved a lot, if the bias errors from the GPS data could be corrected. The use of a Real-Time Kinematic (RTK) GPS with a centimeter-level accuracy would increase the global localization accuracy of the collected map. The GPS signal has to be received in a forest, so the quality of the antenna is an important factor as well. In this work a GPS receiver with a simpler filter could be used in the system as the map path could be used as a reference for the GPS location filter. In that manner the GPS coordinate could be calculated faster with nearly no delay. If the GPS data would be more accurate and available with smaller delays, it could be used directly to improve the mapping accuracy instead of only using it to fix the global position of the map.

To summarize the work, it can be said that the quality of the mapping and the whole process could be improved with more accurate, reliable and faster sensors.

Chapter 9

References

[1] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *Robotics and Automation, IEEE Transactions on,* vol. 7, pp. 376-382, 1991.

[2] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Robotics and Automation. Proceedings. 1987 IEEE International Conference on,* vol. 4, pp. 850-850, 1987.

[3] M.R. Blas and S. Riisgaard. (2004), *SLAM for dummies.* [Online]. Available: http://www.ocw.cn/NR/rdonlyres/Aeronautics-and-Astronautics/16-412JSpring-2005/9D8DB59F-24EC-4B75-BA7A-F0916BAB2440/0/1aslam_blas_repo.pdf

[4] C. Stachniss, U. Frese and G. Grisetti. (2001, 2001-12-01). What is SLAM? [Online]. *2009(6/23),* pp. 1. Available: http://openslam.org/slam.html

[5] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," in *In Proceedings of the AAAI National Conference on Artificial Intelligence,* 2003, pp. 593-598.

[6] G. Weiss, C. Wetzler and E. von Puttkamer, "Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans," *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on,* vol. 1, pp. 595-601 vol.1, 1994.

[7] A. Dubrawski and I. Siemiatkowska, "A method for tracking the pose of a mobile robot equipped with a scanning laser range finder," *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on,* vol. 3, pp. 2518-2523 vol.3, 1998.

[8] E. Ivanjko, B. D. Basic and I. Petrovic, "Correlation Based Approach to Mobile Robot Pose Tracking in Unknown Environments," *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on,* pp. 445-450, 2007.

[9] J. Jutila, "Localization and Tree Measurement with Laser Scanner in a Forestry Machine Perception System," 2006-07-10. 2006.

[10] M. Miettinen, M. Ohman, A. Visala and P. Forsman, "Simultaneous Localization and Mapping for Forest Harvesters," *Robotics and Automation, 2007 IEEE International Conference on,* pp. 517-522, 2007.

[11] J. Selkäinaho, "Adaptive autonomous navigation of mobile robots in unknown environments," 2002.

[12] J. Selkäinaho and P. Forsman, "Navigation of an outdoor service robot by matching 2D laser scans," *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on,* pp. 353-358, 2005.

[13] T. Bailey and E. Nebot, "Localisation in large-scale environments," *Robotics and Autonomous Systems,* vol. 37, pp. 261, 2001.

[14] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," 2002.

[15] D. M. Cole and P. M. Newman, "Using laser range data for 3D SLAM in outdoor environments," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on,* pp. 1556-1563, 2006.

[16] A. Nüchter, K. Lingemann, J. Hertzberg and H. Surmann, "6D SLAM - 3D mapping outdoor environments," *Journal of Field Robotics,* vol. 24, pp. 699-722, 2007.

[17] P. Gong and Y. Sheng. (2002), 3D model-based tree measurement from high-resolution aerial imagery. *Photogrammetric engineering and remote sensing 68(11),* pp. 1203. Available: http://www.slrss.cn/download/38.pdf

[18] D. S. Boyd and F. M. Danson. (2005), Satellite remote sensing of forest resources: Three decades of research development. *Progress in physical geography 29(1),* pp. 1. Available: http://web.ics.purdue.edu/~shao/publications/Forest_RS2005.pdf

[19] T. Hellström. (2002, 2002-10-18). Autonomous navigation for forest machines. Department of Computing Science, Umeå University. [Online]. Available: http://www.cs.umu.se/~thomash/reports/navigation-project-prestudy%20CSreport.pdf

[20] S. I. SICK. (2008), Laser measurement sensors. [Online]. *2009(1/9),* Available: http://mysick.com/eCat.aspx?go=DataSheet&Cat=Gus&At=Fa&Cult=English&Category=Produktfinder&ProductID=9173

[21] I. MicroStrain. (2007), 3DM-GX2 data communications protocol. [Online]. Available: http://www.microstrain.com/pdf/dcp/Inertia-Link-3DM-GX2-data-communications-protocol.pdf

[22] D. DePriest. (2009), NMEA data. [Online]. *2009(7/6),* Available: http://www.gpsinformation.org/dale/nmea.htm

[23] Metla. (2007), Forest finland in brief. [Online]. Onlinepp. 24.8.2009. Available: http://www.metla.fi/metinfo/tilasto/julkaisut/muut/brief2007.pdf

[24] Metla. (2009), Finnish forest research institute. [Online]. *2009(7/6),* Available: http://www.metla.fi/index-en.html

[25] W. F. Milliken, "Race car vehicle dynamics," in D. L. Milliken, Ed. Warrendale: SAE, 1995., pp. 715.

[26] O. Chocron. (2000), Euler ZYX convention. [Online]. Available: http://stuff.mit.edu/afs/athena/course/2/2.05/www/Handout/HO2.PDF

[27] E. W. Weisstein. (2009, 2009-03-02). Cross-correlation theorem. [Online]. *2009(3/2),* Available: http://mathworld.wolfram.com/Cross-CorrelationTheorem.html

[28] P. J. Besl and N. D. McKay. (1992), A method for registration of 3-D shapes. *IEEE transactions on pattern analysis and machine intelligence 14(2),* pp. 239.

[29] B. K. P. Horn. (1987), Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America. B, Optical physics 4(4),* pp. 629.

[30] D. Nistér, O. Naroditsky and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics,* vol. 23, pp. 3-20, 2006.

[31] Wikipedia. (2009, 2009-03-07). Earth radius. [Online]. *2009(3/9),* Available: http://en.wikipedia.org/wiki/Earth_radius

[32] E. W. Weisstein. (2009, 2009-03-09). Great circle. [Online]. *2009(3/9),* Available: http://mathworld.wolfram.com/GreatCircle.html

[33] J. Jutila, K. Kannas and A. Visala, "Tree Measurement in Forest by 2D Laser Scanning," *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on,* pp. 491-496, 2007.

[34] W. K. Pratt, "Edge detection," in *Digital Image Processing (Third Edition)* Anonymous 2002, pp. 443-508.

[35] E. W. Weisstein. (2009, 2009-03-09). Least squares fitting. [Online]. *2009(3/9),* Available: http://mathworld.wolfram.com/LeastSquaresFitting.html

[36] E. W. Weisstein. (2009, 2009-03-26). Solid angle. [Online]. *2009(3/26),* Available: http://mathworld.wolfram.com/SolidAngle.html

[37] E. W. Weisstein. (2009, 2009-04-01). Line. [Online]. *2009(4/1),* Available: http://mathworld.wolfram.com/Line.html

[38] I. D. Coope, "Circle fitting by linear and nonlinear least squares," *J. Optimiz. Theory Appl.,* vol. 76, pp. 381-388, 02/01. 1993.

Chapter 10

Appendix

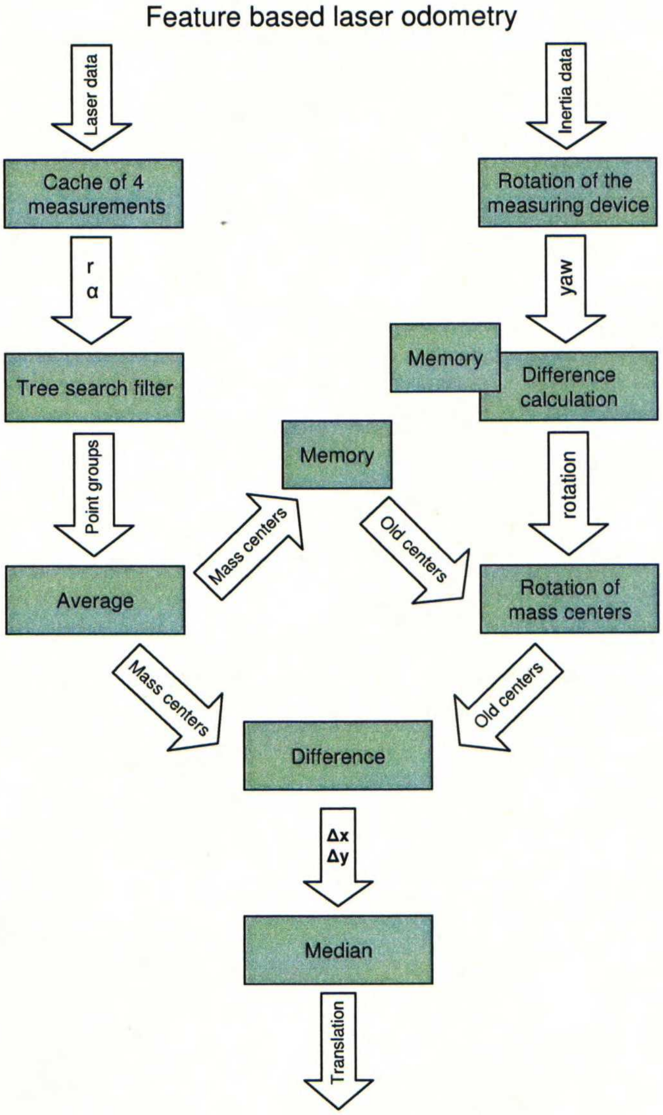

Figure 45: The feature based laser scanner odometry method, where mass centers are used as features and the rotation is gained from the IMU measurements. This system is documented in Section 4.2.3.

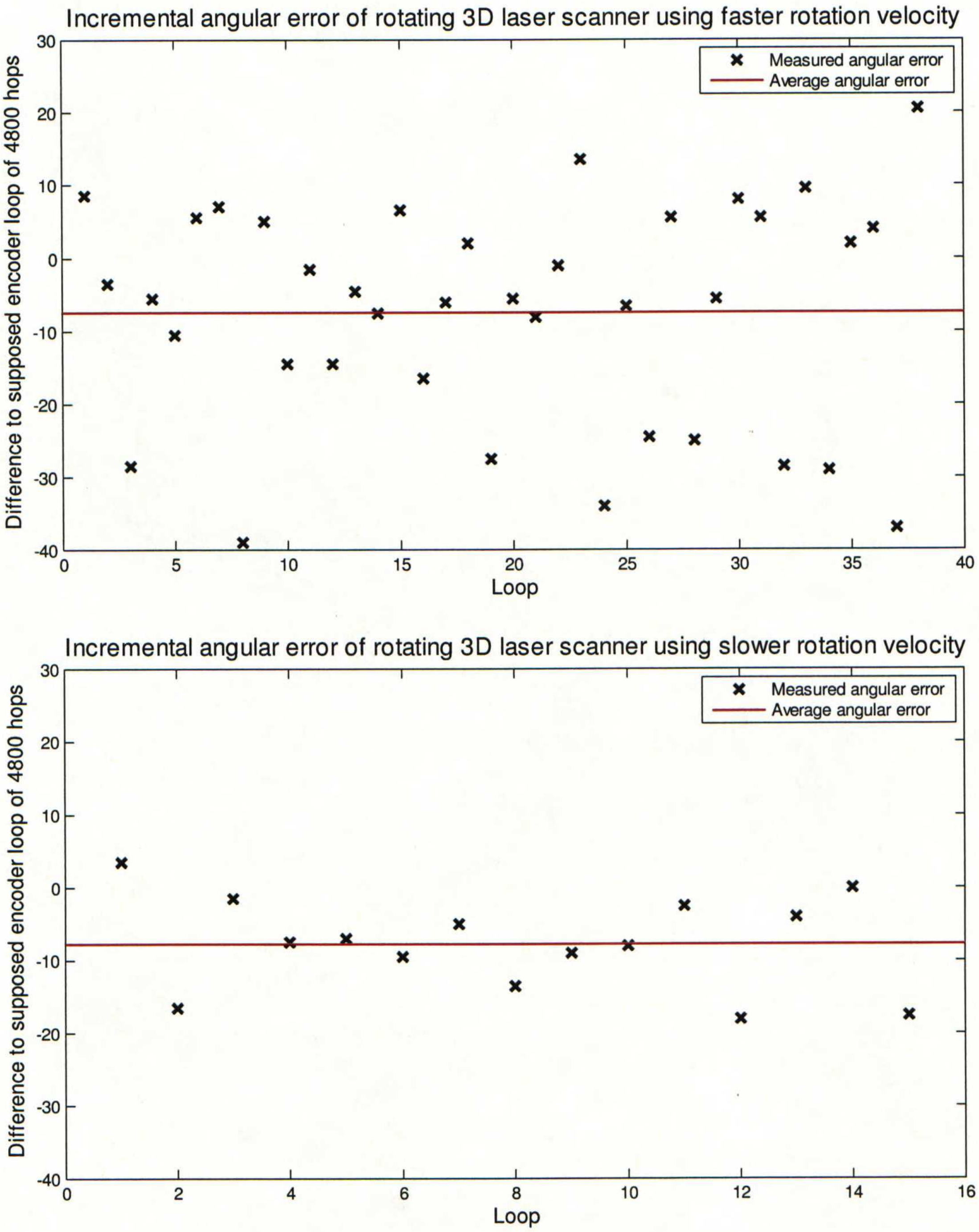

Figure 46: The measured angular error of the rotating 3D laser scanner used in this work. The measuring was done by comparing relative angular encoder measurements to absolute reference measurements. The difference to assumed 4800 encoder ticks is captured when the reference signal is got. This measurement shows the uncertainties in the reference sensor as there is quite large variance in the measurements, especially at the faster measuring mode shown in the upper image. The average shows the statistical error in the incremental encoder as it lacks on average eight ticks at every round. These deficiencies complicated the measuring procedure needed for the 3D data and decreased the measuring accuracy of the scanner.