Risto Kaijaluoto

Precise indoor localization for mobile laser scanner

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 15.5.2015

Thesis supervisor:

Prof. Ville Kyrki

Thesis advisor:

D.Sc. (Tech.) Antero Kukko

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Risto Kaijaluoto

Title: Precise indoor localization for mobile laser scanner

Date: 15.5.2015 Language: English Number of pages: 7+54

Department of Electrical Engineering and Automation

Professorship: Automation technology Code: AS-84

Supervisor: Prof. Ville Kyrki

Advisor: D.Sc. (Tech.) Antero Kukko

Accurate 3D data is of high importance for indoor modelling for various applications in construction, engineering and cultural heritage documentation. Terrestrial Laser Scanning (TLS) is currently the most accurate method to collect such data. Due to its static single view point data collection, excessive time and data redundancy are needed to capture data of good integrity and coverage. With Mobile Laser Scanner (MLS) data acquisition is considerably faster and changing view point reduces occlusions. However, MLS requires an accurate estimate of its location and attitude to project the measurements. The location can be calculated by using a horizontally mounted laser scanner combined with a Simultaneous Localization and Mapping (SLAM) algorithms.

The aim of this thesis is to investigate, what level of trajectory accuracies can be achieved with high quality sensors and freely available state of the art SLAM algorithms and whether the trajectories can be used to project measurements collected with a secondary laser scanner to generate visually aesthetic and good quality three dimensional point clouds.

The performance of three SLAM algorithms were studied on five datasets captured at our office with Slammer platform. The datasets were processed with the algorithms using a number of different parameter combinations and their performance was evaluated by comparing the results against a TLS based reference. A combination of Hector SLAM and Karto showed the best performance and 20 mm root mean squared error (RMSE) levels could be achieved. Analysis of the 3D point cloud produced with the trajectory showed good agreement with the TLS reference with 13 mm planar RMSE. The obtained point cloud is useful for indoor modelling with accuracies close to TLS with vastly faster data collection.

Keywords: SLAM, Indoor localization, Mobile laser scanning, MLS, Point cloud

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Risto Kaijaluoto

Työn nimi: Liikkuvan laserkeilaimen tarkka sisätilapaikannus

Päivämäärä: 15.5.2015 Kieli: Englanti Sivumäärä: 7+54

Sähkötekniikan ja automaation laitos

Professuuri: Automaatiotekniikka Koodi: AS-84

Valvoja: Prof. Ville Kyrki

Ohjaaja: TkT Antero Kukko

Kolmiulotteisille malleille sisätiloista on monenlaisia käyttötarkoituksia aina rakennusten kunnon seuraamisesta virtuaalimaailmojen luomiseen. Mallin pohjaksi tarvitaan tarkat 3D mittaukset ja tarkin tapa kerätä ne on maalaserkeilaus. Maalaserkeilauksen ongelma on staattisista mittauksista johtuvat katvealueet ja näiden minimoiseksi tarvittava suuri määrä aikaavieviä mittauksia. Liikkuvalla laserkeilaimella datan keräys on huomattavasti nopeampaa ja jatkuvasti vaihtuvan mittauspaikan ansiosta katvealueita jää vähemmän. Keilaimen liikerata täytyy kuitenkin tietää tarkasti ja yksi tapa laskea se on toisen, vaakatasossa mittaavan laserkeilaimen ja sen dataa prosessoivan samanaikaisen paikannus ja kartoitus (Simultaneous Localization and Mapping, SLAM) algoritmin avulla. Tämän diplomityön tarkoitus on tutkia millaisiin liikeradan tarkkuuksiin päästään käyttämällä hyvälaatuisia sensoreita ja laadukkaita, avoimen lähdekoodin SLAM algoritmejä ja että pystytäänkö tämän liikeradan avulla muodostamaan hyvännäköinen ja laadukas pistepilvi.

Diplomityössä tutkittiin kolmea SLAM algoritmia viiden Slammerilla kerätyn testiaineiston avulla. Jokaisen testiajon mittaukset prosessointiin useilla SLAM algoritmien parametriyhdistelmillä ja laskettujen liikeratojen hyvyyttä arvioitiin vertaamalla niitä referenssi liikerataan joka oli muodostettu maalaserkeilausten perusteella tuotetun kohdekartan avulla. Parhaaseen tulokseen päästiin Hector SLAM ja Karto algoritmien yhdistelmällä, jolla keskineliövirheen neliöjuuri oli 20 mm tai alle. Verratessa näiden liikeratojen avulla muodostettuja pistepilviä maalaserkeilauksen avulla muodostettuun, tasovirheen keskineliövirheen neliöjuuri oli 13 mm. Vaikka maalaserkeilauksen tarkkuuksiin ei päästy, liikkuvan laserkeilaimen avulla muodostetut pistepilvet ovat riittävän hyviä moniin käyttötarkoituksiin ja ne saatiin mitattua huomattavasti nopeammin.

Avainsanat: Samanaikainen paikannus ja kartoitus, Sisätilapaikannus, Liikkuva laserkeilaus, Pistepilvi

Preface

This research was conducted at Finnish Geospatial Research Institute (FGI) in the Centre of Excellence in Laser Scanning Research and in the Department of Remote Sensing and Photogrammetry. First I would like to thank Prof. Juha Hyyppä for the opportunity to work at FGI on my thesis in the excellent Mobimap and Laser Scanning Research group. I would also like to thank my thesis supervisor Prof. Ville Kyrki from the Intelligent Robotics group of Aalto University for always being ready to help and for the excellent improvement suggestions during the finalizing stages of the thesis. The biggest thanks go to my thesis advisor Dr. Antero Kukko, who always managed to find time to help me even when burdened by his own research. Among many other things, his instructions on how to use the hardware was essential and the help with assessing the quality of point cloud was also invaluable.

Harri Kaartinen and Anttoni Jaakkola from the research group also deserve thanks for always providing a helping hand when needed. I would also like to thank Roope Näsi, Saija Simola and Mika Kekäläinen, who worked on their Master's theses at the same time for peer support.

Finally I would like to thank my family for their enduring support they gave and especially my father Sakari Kaijaluoto for the innumerable suggestions on improving the grammar and language of the thesis.

Masala, 18.5.2015

Risto Kaijaluoto

Contents

Symbols and abbreviations

Abbreviations

GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
MCL	Monte Carlo Localization
MLS	Mobile Laser Scanner/Scanning
RMSE	Root mean squared error
SLAM	Simultaneous Localization and Mapping
SPAN	Synchronized Position Attitude Navigation
TLS	Terrestrial Laser Scanner/Scanning

1 Introduction

The Demand for digital 3D models of building interiors has been growing as the cost of producing them has been decreasing. They can be used for a variety of purposes from creating virtual worlds to monitoring building condition. In building monitoring periodically taken 3D measurements can be used to assess the structural integrity of a building by for example measuring if supporting beams or walls have bulged or moved. The 3D model of a building can also help with renovations planning and can also be used for assessing the result. Virtual models of important cultural and historical sites can be used in marketing for the lesser known ones and for the popular ones it can serve as a way to visit them alone as crowds of tourists can greatly affect the atmosphere. While a virtual visit to a site is not the same as an actual one, having the option enables people from all over the world without the resources or time for actual visit to have the experience.

Currently the most common way to make indoor models is by using Terrestrial Laser Scanning (TLS). In TLS, a 2D scanner scanning vertically is mounted on a tripod and as the scanner is rotated on the tripod, a 3D point cloud of the scanner's surroundings is created. Specific reference targets, which are not moved between the scans and can be seen from multiple scanning locations are used to combine point clouds acquired from different positions to create one representing the whole area of interest.

Scanning large indoor areas by a TLS system is time consuming and requires a large amount of manual work. Taking a scan from one location can take several minutes and as the scanner and, depending on the complexity of the building, the targets too must be moved a number of times, the whole process of scanning a building can take hours. The scanning positions and the locations of the targets used to register the different scans have to be well planned to ensure that all scans can be registered to one cloud and to minimize occlusions rising from the limited number of viewpoints. The number of required reference targets and scanning locations can quickly grow to a large number especially in cluttered spaces with short visibility.

A mobile laser scanner (MLS) consists of a laser scanner taking measurements of the surroundings and sensors which provide orientation and position to the mobile platform. With MLS continuously taking measurements, a large area can be covered quickly and occlusions are much less of a problem. A MLS platform can also easily be augmented with additional sensors such as infrared or hyperspectral cameras to add additional information to the measured points. As the measurements must be continuously projected to the world coordinate system, the trajectory of the platform must be known with high precision at all times for the MLS system to be able to produce point cloud of the environment.

Global Navigation Satellite Systems (GNSS) such as GPS maintained by the United States government or GLONASS maintained by the Russian government provide signals, which enable receivers to pinpoint their location with accuracy in the range of meters. This can be further improved to the range of tens of centimetres with augmentation methods such as differential correction[1]. As the name implies, both systems provide global coverage and when combined with Inertial Measurement

Unit they can provide accurate trajectory. Unfortunately GNSS signals are weak and cannot reliably penetrate building walls and so cannot be relied upon indoors. Even if the signal could be intermittently received, the accuracy is further degraded by multi-path propagation caused by the signal reflections from the multitude of surfaces in buildings [2].

One field receiving much interest is the use of nowadays ubiquitous RF signals (such as WiFi, Bluetooth or FM radio) to help with localization. The device location can be inferred by measuring the received signal strength (RSS) of different signal sources and by comparing them to a database of measurements from known locations[3]. Unfortunately the accuracy possible with such techniques is in the range of meters which is far too inaccurate for any mapping or data collection purposes[3, 4]. Changes in the environment also affect signal propagation and the accuracy can be further lowered if for example a piece of furniture is moved around.

Horizontally mounted laser scanners are widely used for mobile robot localization and mapping purposes as they provide accurate spatial information of the world with little noise[5–7]. When combined with Simultaneous Localization and Mapping (SLAM) algorithms they can provide the trajectory of a platform in unknown environment [5, p.153, p.309]. As a laser scanners do not rely on external infrastructure they work indoors just as well as outdoors.

High localization accuracy could also be achieved with cameras. Cameras providing depth information in addition to colors such as Microsoft Kinect based on structured light[8], Microsoft Kinect v2 based on time-of-flight[8] or stereo cameras are especially suitable for localization as they, like laser scanners provide measurement of spatial relations. While applicable for indoor localization their use require different algorithms and as such they are out of scope for this thesis.

The SLAM algorithms used in robotics are commonly geared towards online use as the position and map provided by them are used for navigating around the environment. In our case, the data is collected by a platform moved by a human so there is no need for real time position information and the SLAM problem can be solved offline using all the collected data. This enables more processing to be conducted to the measurements which should translate to better accuracy. The main research questions this thesis answers are:

- What kind of trajectory accuracies can common SLAM algorithms provide with high quality sensors through offline processing
- Can the produced trajectory be used for projecting data to create a visually aesthetic and good quality point cloud

To study these problems we mounted an IMU and two high quality laser scanners, one horizontal for SLAM and the other tilted for point cloud data collection to a pushable wheeled cart.

The thesis starts with an introduction to the working of sensors, first of a laser scanner (Section 2.1) then of an inertial measurement unit (Section 2.2). Section 3 describes basic principle of SLAM and the three studied SLAM algorithms in their basic form. Detailed explanation of the hardware used is given in Section

4. Section 5 starts with explanation on the basics of Robot Operating System (ROS) framework which is used for data processing. The modifications made on the SLAM algorithms presented next and finally the total pipeline from data collection to finished 3D point cloud. Section 6 describes the experiments made to assess research questions. The Section starts with the explanation of a calibration procedure used to find the transformation from the data collection scanner to the vehicle frame and then proceeds to describe how the reference point cloud and trajectories were created. Next Sections describe the results of the experiments, first for the produced trajectories and then for the produced point cloud. Finally in Section 7 conclusions are made and future directions for research are discussed.

2 Background

To localize itself, platform just like human requires some way to perceive the world around it. Wide variety of sensors with different cons and pros can be utilized to give a robot a sense of its movements and of its surroundings.

2.1 Laser Scanner

Laser scanner (also called LiDAR, scanning laser range finder or just laser range finder) is a device measuring distances. Basic operating principle of the scanner is shown in Figure 1. Commonly the mirror is rapidly rotated to measure distances in a 2D plane. Recently manufactures such as Velodyne [9] have begun to offer laser scanners which have multiple lasers (up to 64 for Velodyne HDL-64E) of which each scans a different plane to enable perceiving in 3D. Laser scanners are widely used in robotics as a measurement device of choice for perceiving the world as they provide accurate spatial information in easily comprehensible format which requires little processing to be usable.

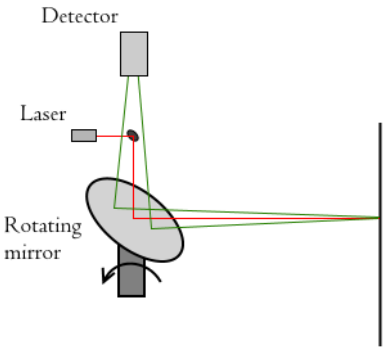


Figure 1: Principle of a 2D planar laser scanner, as the mirror rotates the beam scans a plane. [10, p. 38]

Two methods for measuring the distance travelled by the laser beam to the object are commonly employed, one based on measuring of the time-of-flight of a pulse and one on measuring the phase difference of a modulated continuous wave.

In time-of-flight (pulsed laser) measurement a short laser pulse is emitted and the time Δt between sending and receiving reflected echo is measured. As the speed of light c is constant and known to a high precision, the distance to the target can be calculated with Equation 1, where n is the refractive index of air [10, p. 3].

$$r = \frac{\Delta t \cdot c}{n \cdot 2}. \tag{1}$$

In phase difference measurement system a continuous amplitude modulated beam is transmitted to the target and the phase difference of the transmitted and received waveform is measured. When the modulation frequency f_m is known, the phase

difference $\Delta\varphi$ can be used to calculate the time difference Δt as shown in Equation 2. Δt can then be translated to range r by Equation 1.[10, p. 6]

$$\Delta t = \frac{\Delta\varphi \cdot f_m}{2\pi}. \quad (2)$$

实际范围超过一半波长则来回超过一个波长。

If the actual range is more than half of the wavelength, the actual time difference and therefore range cannot be determined uniquely as only the phase difference can be measured. Multiple modulation frequencies can be used to overcome this range ambiguity [10, p. 6]. Starting from the longest wavelength, the phase difference is used to calculate a coarse range which can then be refined by the phase differences in the shorter wavelengths to obtain a unique and accurate range measurement.

Unlike ideal rays, real life laser beams have some radius which grows by range as the beam diverges [10, p. 12]. This can cause ambiguity to the measurement when the beam hits an edge of an object. Part of the beam is reflected back from the object in the foreground and the rest from the background. If the arrival times of the reflections are averaged the result is a 'ghost point' between the objects with no physical meaning.

The problem with multiple reflections is even more profound with phase difference measurement based laser scanners. When hitting a spatial discontinuity, the measured phase and therefore range depends on the distance between foreground and background objects, their relative reflectance and the proportion of the beam hitting each one. These affect the measurement in an unpredictable way and the resulting range measurement can be anything between the maximum range of the scanner and the distance to the closest object and in some rare cases even smaller [11].

Probabilistic modelling of the measurement process can be used to include noise and other uncertainty always inherent in real life measurements to localization methods. Commonly the measurement can be modelled by a conditional probability density $p(z_t|x_t, m)$ where z_t is the measurement (single sweep of a scanner), x_t is the pose of the scanner and m is the map. The individual measurements of the sweep can be assumed to be independent of each other so the combined probability can be calculated as shown in equation 3 where k denotes the index of the individual measurement.

$$p(z_t|x_t, m) = \prod_{k=1}^{K} p(z_t^k|x_t, m) \quad (3)$$

Even though the independence assumption is often not true in real world scenarios, it simplifies calculations and the resulting algorithm still works [5, p. 149].

Modelling each laser measurement as a beam is an obvious way and is strongly linked to the physics of the device [5, p. 153]. Different error sources are modelled by different distributions which are combined together using predefined weights. Examples of distributions can be seen in Figure 2. The beam model is accurate, but ray casting operation is required to see where each beam would hit from the location x_t to compute z_t^{k*}, which is computationally expensive. Additionally, in complex environments with clutter the model is not smooth. Small changes in the

pose, especially in the heading direction can cause individual measurements and the probability $p(z_t|x_t, m)$ to change radically which makes this model less useful with algorithms trying to optimize the pose [5, p. 168].

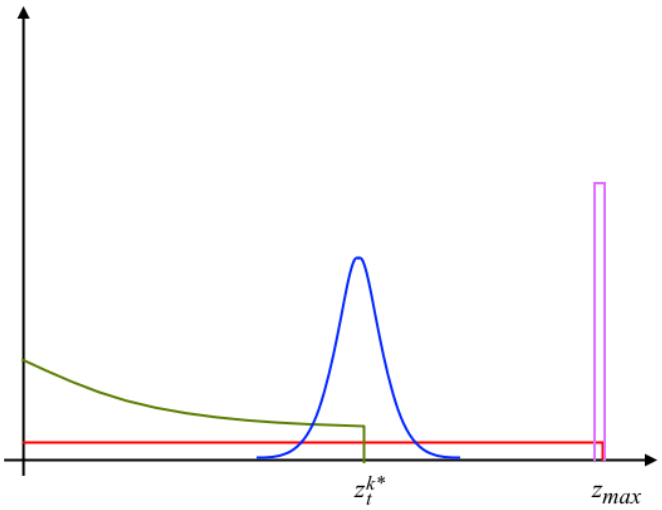


Figure 2: Example of common combination of point density functions used with beam model. z_t^{k*} is the range calculated by ray casting and z_{max} is the maximum range of the sensor. Blue represents the measurement uncertainty in the actual measurement (Gaussian distribution), green represents possible dynamic obstacles in front of the target (Exponential distribution), red represents random noise from unexpected sources and pink represents the possibility of a failed measurement [5, p. 154].

A simpler and computationally more effective option to model the uncertainty is to use likelihood fields [5, p. 169]. In this model only the endpoints of the measurements are considered and the probability can be calculated as a combination of zero mean Gaussian and an uniform probability as shown in Equation 4.

$$p(z_t|x_t, m) = z_{hit} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(dist)^2/2\sigma^2} + z_{random} \cdot p_{random} \tag{4}$$

The *dist* is the euclidean distance between the measured point and the nearest occupied grid location and the Gaussian then models the uncertainty in the measurement accuracy while the p_{random} models the possibility of a random measurement always possible with laser scanners, failed measurements with z_{max} value are ignored altogether.

While the inability to model dynamic obstacles reduces the accuracy when obstacles are present, the smoothness and computational simplicity make this algorithm more useful [5, p. 173]. One clear advantage of this method is the possibility to precompute the likelihood values of a map or a single scan by using a discrete grid, the operation in Equation 4 is then replaced by a simple table lookup [5, p. 172].

2.2 Inertial Measurement Unit

Inertial Measurement Unit (IMU) is a device which measures linear accelerations and angular velocities which are used in inertial navigation to calculate the position, velocity and orientation of the device [12, 13]. An IMU consists of accelerometers measuring linear accelerations and gyroscopes measuring angular velocities. While not directly providing position and orientation information, IMU measurements can be integrated over time to calculate them.

In traditional approach the IMU sensors are mounted on a stabilized platform [13]. The sensor platform is connected to the host vehicle through a set of gimbals which let it rotate freely and the angular velocity measurements are used to drive servo motors on the gimbal axes to keep the platform stable. With stable platform the integration of accelerometer information is straightforward and the attitude can be determined from the gimbal positions. While they provided good performance, the mechanical construct of gimballed system is complicated, and it is expensive to manufacture and difficult to miniaturize [12].

A simpler approach is to mount the sensors rigidly to the host vehicle in a strapdown configuration. In strapdown IMU the mechanical platform stabilization is replaced with a mathematical one by integrating the angular velocity measurements to gain the orientation of the platform and then by projecting the accelerometer measurements accordingly before integrating them. This requires sufficiently high sensor sampling rate to keep the errors caused by integrating discrete samples small. Gyroscopes also need to have much higher dynamic range (4 orders of magnitude) than in stabilized platform IMU as they need to be able to measure angular velocities from the full manoeuvre envelope of the vehicle. In online use strapdown IMU also requires more computing power as the calculations are more complex although with modern computers this is not a big cost driver any longer. [12]

IMU based inertial navigation requires only the measurements from its sensors and is a totally self-contained system requiring no infrastructure. The largest disadvantage of inertial navigation is caused by the need to integrate the measurements, twice for accelerometers and once for the gyroscope. The integrated errors keep accumulating with no limit if no external corrections are available, as the IMU measurements are not error free. The errors in the orientation of the IMU are especially harmful as the orientation is also used in the integration of the accelerometer measurements. Even a tiny orientation error quickly manifests as a large error in position. It is also possible to include magnetometers to an IMU; measurement of the stable Earth magnetic field direction can help to reduce the drift error in orientation [13]. Ferromagnetic materials interfere with the field which complicates the use of these measurements, the interference effect is especially pronounced in buildings as they commonly contain high amount of ferromagnetic materials in the form of, for example, reinforced concrete.

3 Simultaneous Localization and Mapping Algorithms

In an unknown environment the robot faces two problems, it neither knows its location x_t nor does it know the map m of the area. Inferring these two pieces of information from measurements of robot's internal sensors $u_{1:t}$ (IMU, odometry) and measurements of the world $z_{1:t}$ is commonly called Simultaneous Localization and Mapping (SLAM) [5, p. 309]. A schematic picture of SLAM problem is shown in Figure 3. The problem of estimating the posterior $p(x_{1:t}, m|z_{1:t}, u_{1:t})$ is difficult because there is a circular dependency between the robot needing a map of its surroundings to localize itself and the need to know its location for building the map. Fortunately, this problem can be solved by iteratively localizing the robot pose relative to its starting position and by using this information to construct the map.

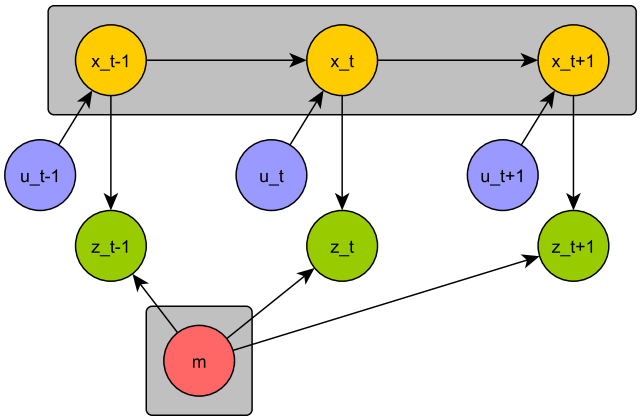


Figure 3: Schematic picture of a full SLAM problem. The aim is to estimate all robot poses and a map (grey background) from the controls and the measurements. When solving an online SLAM problem we are only concerned with the map and the last pose.[5, p. 311]

SLAM algorithms can be divided into a frontend and a backend. The frontend deals with sensory input doing scan matching and calculating spatial relations between poses. The backend on the other hand deals with the collected information and tries to optimize the estimated robot poses to keep the resulting map coherent [14]. In practise a SLAM frontend is enough to solve the SLAM problem but as the small errors in scan matching will inevitably accumulate and reduce the accuracy, in complex environments with longer trajectories and loop closures (returning to an area visited before) a backend can greatly improve the result.

Scan Matching aims to find the rigid-body transformation (translation $\Delta x, \Delta y$ and rotation $\Delta\theta$) between two scans or a map and a scan. If available, the transformation estimate from motion measurements such as odometry can be used as an initial guess which is then refined by a variety of methods. Scan matching is an integral part of SLAM algorithms as (if successful) scan matching provides a much more accurate estimate of the movement between two time steps than internal sensors.

This thesis concentrates on SLAM algorithms working on a 2D plane using volumetric maps typically expressed as an occupancy grid. Occupancy grid is an fine uniform grid covering the whole area of interest. Typically the grid cells can have three values, occupied (by wall for example), free or unknown. Probabilities can also be incorporated by giving each cell a probability of it being occupied which is then updated when new measurements are taken [5, p. 284]. Maps and localization based on landmarks could also be used but with occupancy grids the whole scan data can be used in the localization and mapping process and there is no added complexity of extracting and detecting landmarks.

ROS framework is used for data processing in this thesis which constrained the SLAM algorithms tested to ones available as ROS packages. Machado Santos et al. [15] evaluated different SLAM algorithms available in ROS and the three best performing SLAM algorithms based on their evaluations were selected for tests with high quality sensors in this thesis. An additional motivation for selecting the algorithms was their use of different techniques for solving the SLAM problem.

3.1 Hector SLAM

Hector SLAM algorithm utilizes the full scan rate of modern high frequency laser scanners. By using every subsequent scan the search space for rigid-body transformation between scans stays small and the transformation can be found in real time. The correct transformation is found by optimizing the alignment of scan endpoints with the map learned so far. Gradients of the grid cells are calculated at scan end points and the pose is optimized using Gauss-Newton method [7].

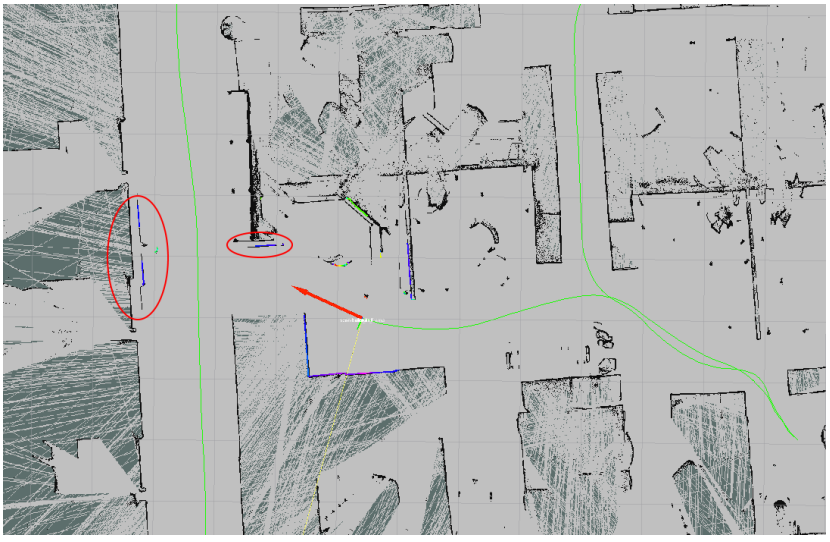


Figure 4: The position estimated by Hector SLAM has diverged from the correct one after running a couple of minutes in an area with short visibility and almost no line of sight to previously mapped area. The current laser scan (colored points) does not match with the walls mapped previously. Red arrow marks the current position and green line marks the trajectory so far.

Hill climbing style optimization methods such as Gauss-Newton are prone to getting stuck at local optima. To ensure convergence to global minimum, the algorithm maintains a pyramid of multiple occupancy grids with each having half the resolution of the preceding one [7]. Scan matching is started with the coarsest resolution occupancy grid and then repeated on finer grids using the result of scan matching done at the previous resolution as the starting guess. Bilinear filtering is employed to reach precision greater than available from discrete sized grid cells [7]. As Hector SLAM only uses the scan endpoints for matching and does not use the structure of a laser scan, the endpoints can be processed in flexible way. This enables Hector SLAM to take into account attitude information from IMU which is used for projecting the laser scans to horizontal plane. This improves matching precision if the laser scan attitude differs from horizontal.

Even though the high quality scan matcher gives the algorithm good accuracy, the position slowly drifts away (Figure **??**) and during the closing of a loop the lack of optimizing backend can show up as a discrete jump in the trajectory when the pose is recovered. If the accumulated error is large enough, the pose might not recover and the errors continue to accumulate while the mapper overwrites previously built map.

3.2 GMapping

GMapping is based on Rao-Blackwellized particle filter which can also be used to solve the SLAM problem. Particle filter is a non-parametric implementation of a Bayesian Filter where, instead of describing the posterior distribution with a parametric function, it is represented by a set of discrete samples (particles) drawn from this distribution [16].

The localization step of GMapping is similar to the Monte Carlo Localization (MCL) method, where large number of particles x_t, each representing a possible pose (state) of the robot at a specific time is created and maintained. The localization is an iterative process which requires a set of particles X_{t-1} (pose hypotheses), control u_t, measurement z_t and a map m as an input. As the first step of each iteration of the filter and for each particle, the probability $p(x_t|x_{t-1}, u_t)$ which is based on the motion model is sampled to get a new pose hypothesis x_t. A weight $w_t = p(z_t|x_t, m)$ is then calculated based on the measurement model and assigned to the pose hypothesis. These new particles and corresponding weights form a temporary set $\bar{X}_t$. To complete the iteration of the filter, resampling of the particles is done. A resampled set X_t is created by drawing with replacement from $\bar{X}_t$ with each particle drawn with probability based on its weight. Now even though the sample set was drawn from just the motion model, the resampling based on the measurement model has changed the distribution to reflect the measurements. [5, p. 250]

During SLAM the algorithm must also estimate the map in addition to the position, each particle also holds a copy of the map generated by it so far. The SLAM problem can be factorized to first estimating the movement and then estimating the map as shown in Equation 5. This type of factoring is called Rao-Blackwellization,

hence the name [5, p. 435][6, 17].

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) = p(m|x_{1:t}, z_{1:t}) \cdot p(x_{1:t}|z_{1:t}, u_{1:t}) \tag{5}$$

As can be seen, the movement estimation step is similar to Monte Carlo localization based on a known map. A straight forward way to solve the SLAM problem would be by just adding a step to each iteration where for each particle the estimated pose is used to integrate the current laser scan to the map learned so far.[5, p. 478] Unfortunately it is inefficient to use as many particles as with MCL because each particle must maintain a map which requires processing power and memory. The ability to use less particles makes the algorithm less robust.

Grisetti et al. [6] proposed an improved method where they use a more accurate x_t proposal sampling process and only resample the particles when needed. The improved proposal sampling method is based on the insight that the information carried by the laser scanner measurement is much more accurate than one available from odometry. It employs scan matching, which uses the position from odometry as the initial guess. The area around the pose acquired by scan matcher is then sampled and evaluated to calculate a Gaussian approximation of the proposal distribution. New pose for the particle is then drawn from this distribution instead of the motion model.

While particle resampling is necessary to replace samples unlikely to represent the correct robot path and map with better ones, it also carries the risk of replacing the correct hypothesis with a worse one as there is random element in the resampling step. To reduce this risk Grisetti et al. only commit the resampling step when the particle set stops approximating the target posterior well enough i.e. when the variance of particle importance weights rises above a predefined level [6] . With these improvements the number of particles needed for consistent map building and localization by a factor of ten as compared to the method employed by Hähnel, Thurn et al. [5, p. 474].

While the algorithm does not have an optimizing backend, the multiple hypotheses of the map that are carried by the particles help maintain a coherent map even with loop closures.

3.3 Karto

Karto [18] uses correlative scan matching algorithm by [19] in its frontend and Sparse Pose Adjustment (SPA) [14] as its backend. The scan matcher requires an estimate of movement as its first guess (from for example odometer, in our case from Hector SLAM) and then performs a search around it at discrete steps over a search window. The likelihoods of different transformations are calculated with the help of a lookup table similar to one used in the likelihood field model explained earlier in Section 2.1. To narrow down the search over the large 3D space (x, y, θ) of possible transformations, the area is first evaluated at a coarser scale against a lookup table with larger cell size to find areas of interest for finer search. The use of multiple resolutions enables the algorithm to run in real time. An additional advantage of the method is that, as a large variety of transformation candidates

are evaluated, their likelihoods can be used to estimate covariance values for the proposed transformation. This value is important for the backend as a weight for the constraint created from the scan match. Covariance values calculated from larger set of transformations are less overconfident than the ones calculated from a smaller set of transformations [19]. The downside of discrete steps is the maximum limit on accuracy they give. This is especially noticeable if every scan from a scanner is input to the algorithm as the displacement between scans can be less than the search resolution and this causes the localization to become unreliable.

The current scan is matched with a fixed size rolling buffer of previous scans. Old scans are searched after each scan match to see if a sufficiently long connected chain falls under predetermined radius from the current location (chains closely connected to the current scan are ignored). If a chain of scans is found a loop closure is attempted by performing scan matching between the current scan and the found chain of previously processed scans, if the scan matching succeeds with match response value larger than predetermined threshold, loop is closed and a constraint between those poses is added to the underlying pose graph.

The SPA based backend of the Karto uses a pose graph structure where scanner poses x_t form the nodes and the edges between them are the scan matching results z_t from the frontend. The measurements always carry some errors from sensor noise etc. and as such, there are no poses which would satisfy all the constraints. Therefore the graph based SLAM problem must be considered as an optimization problem where one seeks to minimize these errors [20]. As the number of nodes grows the accompanying non-linear constraints form a large system of equations which would be difficult to solve if it were not for the fact that as nodes are for the most part only connected to nodes nearby, the system is sparse [5, p. 343][14]. By employing solvers specifically tailored for sparse systems such as sparse Cholesky factorization the optimization can be performed quickly enough to be run in real time [14, 20]. The Karto with its SPA backend has been shown to give excellent results when compared to other algorithms [21].

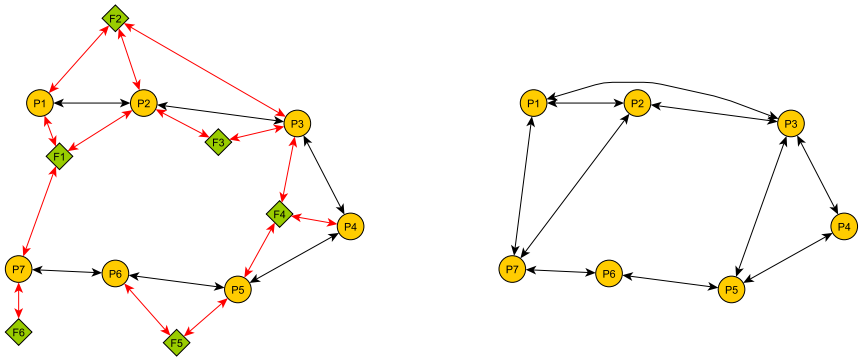


Figure 5: Example graph, poses are represented by circles and world features by diamonds. In the left side, red arrows denote measurements z_t and black ones denote odometry u_t. In the right graph they both are combined to spatial constraints between poses.

An example of a small graph is shown in Figure 5. With maps based on occupancy grids the features and corresponding measurements are replaced with a scan matching operation between the poses where the feature was observed. The pose optimization is sensitive to false constraints caused by a failed data association during loop closure or wrong result from scan matching [22].

4 Hardware Implementation

The Slammer indoor MLS uses the same sensor module that has previously been used outdoors on various scenarios either mounted on a car, all-terrain vehicle or backpack [23, 24]. The sensor module consists of a NovAtel SPAN Flexpak6 GNSS receiver with tactical grade IMU (UIMU-LCI) and two Faro Focus 3D (S120 and X330) high precision laser scanners. For the indoor use the sensors are mounted on a wheeled cart as shown in Figure 6. The S120 laser scanner is mounted horizontally to measure the platform movement, while the X330 in front of the system is tilted 10 degrees from the vertical for producing 3D point cloud of the scene (the angle is adjustable at 10 degree steps for optimizing the configuration). The positions of the laser scanners are interchangeable.

All devices (GNSS antenna, IMU and laser scanners) are connected to the SPAN central unit which is then connected to a tablet computer recording the received data. From the IMU its attitude at 20 Hz and accelerations and angular velocities (corrected for gravity and earth rotation) at 200 Hz are recorded. The laser scanners only send a timing signal to the SPAN system for logging (pulse when a new scan starts). The collected laser measurements are saved on a SD card on each of the scanners. On start up the SPAN system updates its clock and needs a GNSS signal for it. The SPAN system itself could provide a trajectory estimate but this would require frequent GNSS signals which are not available and therefore only the IMU measurements are used.

Table 1: The scanner specifications. Ranging accuracy and noise are at 25 m range, noise minimum with 90% reflectance and maximum with 10 % reflectance.

Scanner	Range	Wavelength	Field of View	Measurement speed
	m	nm	Deg	kpoints/s
S120	120	905	305	122-976
X330	330	1550	300	122-976
Scanner	Rotation Rate	Min. Ang. Res.	Ranging error	Ranging noise
	Hz	mrad	mm	mm
S120	30-97	0.157	±2	0.95-2.2
X330	30-97	0.157	±2	0.3-0.5

The laser scanners are designed for TLS and as such the whole scanner is able rotate around its mount. For our purposes this is not required and the scanners are set to helical scan mode where the scanner itself stays stationary and only the mirror rotates. Their TLS background can also be seen in their settings, rotation speed and the number of points on a single scan can be set by adjusting scan resolution and the degree of measurement averaging. The scanner specifications are listed in Table 1. The scanners are largely similar but X330 as an evolved version of S120 has longer range and lower measurement noise.

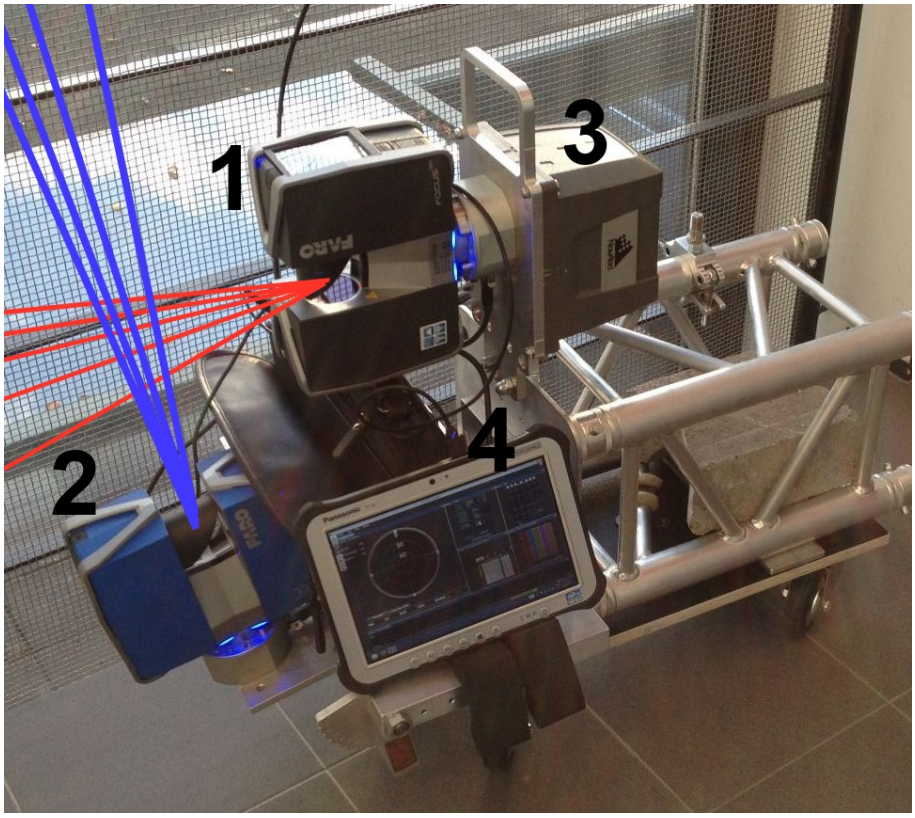


Figure 6: Slammer platform consisting of IMU (NovAtel UIMU LCI, #3 in the figure), horizontally mounted scanner for SLAM (FARO Focus 3DS120, #1), and secondary scanner for 3D point cloud generation (FARO Focus 3DX330 #2). The scanners are interchangeable. The tablet computer (#4) is used for IMU and timing data recording. Lines illustrate the approximate measurement planes of the scanners.

5 Software Implementation

5.1 Robot Operating System

Robot Operating System (ROS) is not an actual operating system like Windows or Linux but a flexible general framework for robotic software. At its core it provides a peer to peer communication layer for a network of software modules which run on a single computer or a cluster of heterogeneous computers [25]. In the framework software modules doing computation are called nodes. Nodes communicate with each other by exchanging messages. The fields of a message are defined using a simple interface definition language and they can contain basic datatypes such as strings, integers or arrays of them or combinations of other message types. It is also easy to create own message types.

Messages can be exchanged in asynchronous manner in publisher subscriber fashion or in synchronous mode as services in client-server mode [25] . In publisher subscriber communication a node (or multiple nodes) can publish messages (e.g. sensor measurements or results of data processing) to a topic specified by a string. Nodes interested in this information can subscribe to the topic to receive the messages. In client-server communication one node acts as a server by advertising a service, which is specified by a name and nodes interested in this service can use it by connecting to it as a client and sending a service call which has two sets of fields similar to messages, one for request and one for response. In both communication modes a ROS master node keeps track of topics/services and corresponding subscribers/clients and publishers/servers to inform nodes where they need to send their information. An example of how nodes are linked together through communication can be seen in Figure 7.

In addition to the communication framework ROS and the accompanying community provide numerous tools and libraries which enable developers to concentrate on their main problem instead of developing supporting software. Commonly used tools include rviz visualization tool and rosbag recording/playback tool. Rviz enables easy visualization of many message types from point clouds to trajectories while rosbag enables one to record messages from topics of interest for later use.

The transformation library tf [26] is an important part of ROS framework as mobile robots commonly have various coordinate frames (for sensors, tools and the robot itself) requiring data to be transformed between them. The tf library listens to messages containing transformations between different coordinate frames from all nodes and forms a tree structure of the frames for easy traversal. The tf node also keeps a history of each transformation between two frames for a predetermined time period. Nodes can then query transformations between any two frames from the tf node at any (current or past) time instant, in case there is no transformation at the queried time instant, one is interpolated from the closest ones.

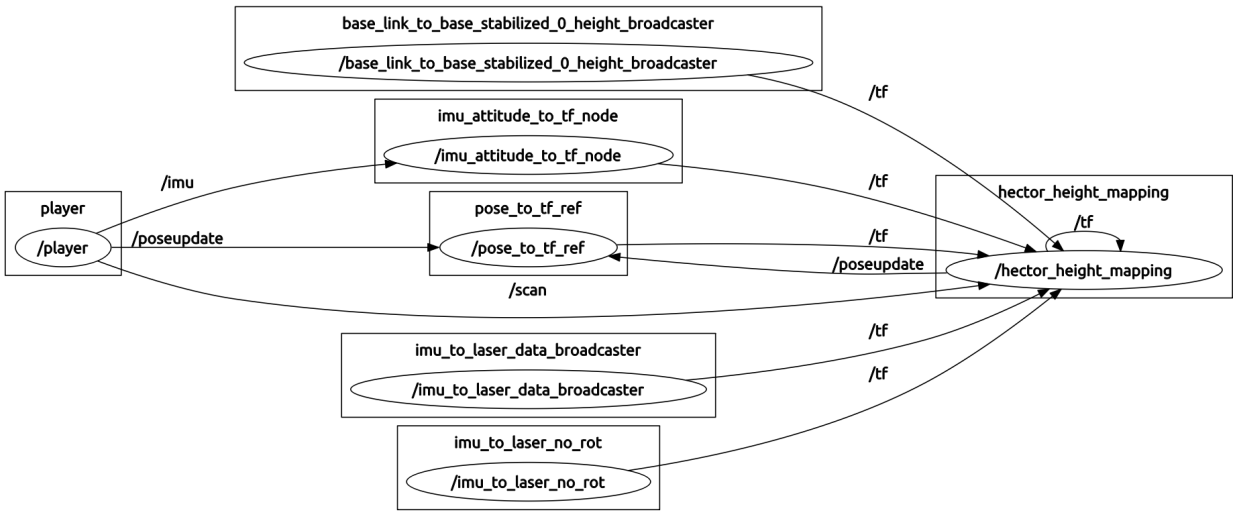


Figure 7: Example node graph of Hector SLAM running on data played from bag file. Boxes denote nodes and arrows between them denote messaging between them with the names above arrows denoting topic names. The communication consists mostly of nodes passing transformations between them.

5.2 Modifications to the SLAM algorithms

Two modifications to the SLAM algorithms were made, one for Hector SLAM and one for Karto. Both are optional features which can be disabled.

The movement of the scanner during single scan sweep causes distortion to the scan adding error to scan matching, an example of this effect is shown in Figure 8. If the movement is known, it can be compensated by transforming each measurement of the scan according to the scanner pose at the time of measurement. To improve the performance of the algorithm we added a processing step before matching to mitigate this effect. The IMU deployed is of sufficient quality and the IMU measurements could be used to estimate this movement but to keep the estimates from drifting away, fusion of the IMU and SLAM estimates with help of for example extended Kalman filter (EKF) would be required. An EKF implementation provided by the makers of Hector SLAM was tested but no set of parameters resulting in convergent behaviour were able to be found.

As an intermediate measure only the rotational component of the movement is considered. This is more straightforward as the IMU provides angular velocity measurements and orientation estimates based on them. Even though the absolute heading (rotation around z-axis) value based on the IMU measurements drifts away from the truth, it can be used to estimate the relative rotation over short periods of time accurately. In this work Hector SLAM algorithm was augmented with this feature. Currently only the scanner start and end poses are queried from the tf node while the intermediate poses corresponding to each single scan point measurement are linearly interpolated . Karto and GMapping algorithms make assumptions of undistorted scan when doing scan matching, which makes implementing movement

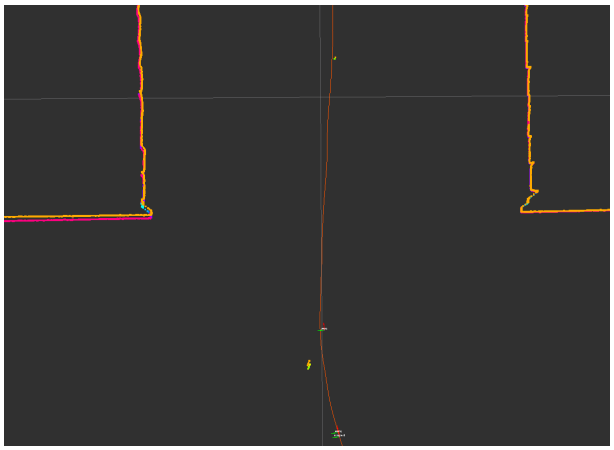


Figure 8: Effect of movement on a scan. Yellow coloured points denote the scan transformed to compensate for movement during the scan. Other coloured points represent the scan without any processing. As the platform moves forward the measurements captured on latter part of the scan appear closer than where they should be. A previously calculated trajectory was used to give an estimate of the movement. The point size is 5 mm.

compensation scheme implemented for Hector SLAM more difficult. It is a task of future experimentation to be fully exploited.

The Karto library is mainly aimed for mobile robots and online use, but as we do the processing offline with all of the data available some changes can be made. In addition to using parameters which provide higher accuracy and are computationally too heavy for online computation, a modification to the loop closing function was made. In its original form a loop is closed immediately when a match between current scan and a chain of nearby processed scans gives a response value greater than a predetermined threshold. As processing is done offline, there is no immediate need to close the loops and an approach called delayed loop closure is used instead. With delayed loop closure when an acceptable loop closure is found, instead of closing it the scan and corresponding scan chain are saved and the processing of subsequent scans is continued. Loop closure attempts with saved scan chain are continued during the processing of the subsequent scans and if a higher response value is attained, the saved scan is updated with the scan with better match. This is continued as long as loop closure is attempted with a member of the saved scan chain. After no more updates to the saved scan and corresponding chain are attempted the loop is closed.

This is advantageous as a better match should lead to smaller error and the match generally gets better as the scanning locations get closer together. In the original form, if the threshold is small a suboptimal loop closure is made when the current location is still far away from the chain of scans but if the threshold is too high, some not as good but still useful loop closures might be missed altogether. With the delayed loop closure approach more loop closures can be made while ensuring that they are as good as possible.

5.3 Processing pipeline

Robot Operating System (ROS) [27] framework is used for data processing . It provides a straightforward way for testing, comparing and combining different algorithms as all of the algorithms can be used within the common framework. The pipeline from collected data to 3D point cloud of an area can be seen in Figure 9.

The data preprocessing starts by transforming the IMU measurements and timestamps logged in a propriety binary format to ASCII text. The transformation is done using a NovAtel convert4 tool. Next the ASCII file is parsed with a Matlab script to separate the different message types and messages from different devices to separate files. The text files and recorded laser scan data are then combined to form IMU and LaserScan ROS messages.

The IMU messages contain attitude, accelerations and angular velocities of the system. As attitude is recorded at lower frequency (20 Hz vs. 200 Hz) than accelerations and angular velocities, the attitude information for messages without one is integrated from the angular velocities. For laser scan messages the propriety Faro format is read with the help of Faro SDK and then combined with a time stamp to form a ROS message. The IMU and laser scan messages are then combined to a single ROS bag file. The laser scans from the scanners are saved separately for three separate topics in total. ROS natively provides a method to play the messages

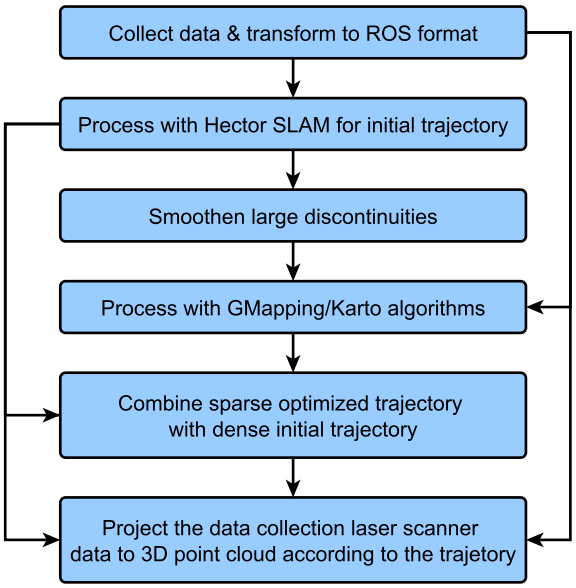


Figure 9: Pipeline from data collection to finalized point cloud. Boxes are processing steps and arrows denote data flows

in a bag file for the algorithms but it is mainly aimed to simulate real time use. Only the speed of the playback of the saved messages can be controlled but there is no way to force every message to be processed. If internal buffers get filled with messages faster than they can be processed the oldest ones are simply ignored. To ensure every message gets processed and that there is no delay between the process-

ing of subsequent messages, the SLAM algorithms were extended with a specific bag handler class which processes the messages in right temporal order with no waiting times. To enable the laser scan messages to be correctly transformed to world coordinates, pose and attitude related messages with timestamps later than current laser scan timestamp must have been processed first.

With the data in ROS formats the actual processing is started by running the laser scan and IMU data through Hector SLAM algorithm to compute an initial trajectory estimate. The Hector SLAM is the only algorithm studied which works without an initial guess of movement from odometry. The GMapping and Karto algorithms require an initial guess and as there are no wheel encoders to provide odometry in Slammer platform, the Hector trajectory is used instead.

The Hector SLAM naturally provides dense trajectory and in environments with good overall visibility of the area, i.e. no loop closures after long periods exploring new areas, the algorithm could provide a sufficiently accurate trajectory. While the lack of a backend causes the algorithm to slowly drift away from the correct solution in more complex environments as shown in Figure 4, the accurate matching provides locally very good estimates of the path taken and as such can be used to generate a sort of accurate fake odometry which can then be employed by the other algorithms.

The Hector SLAM projects the measured points into a XY plane based on the measurements of the IMU. Even though the sensors are mounted on a wheeled platform and only driven on a flat floor, taking into account the small constant offset in attitude and the small fluctuations (less than one degree) caused by low quality wheels does slightly improve SLAM performance.

The large discontinuities caused by loop closures in the Hector trajectory are smoothed away before its use in other algorithms. This is accomplished by checking each transformation between two poses and by comparing it to the preceding ones. If there is a large change in the transformation length or direction from the preceding ones, the transformation is replaced by an average of the preceding transformations.

The smoothed trajectory is then used as input to Karto and GMapping and processed. To speed up the processing only a subset of the scans are processed by Karto and GMapping, in our case with frequency of 3 Hz. Both algorithms also support the ability to process a new scan only if the scanner has moved predetermined amount since last scan used. Time based subset approach was selected as it enables the algorithms to process scans also when not moving. When stationary the scans are undistorted and should provide the best matches.

After processing with Karto or GMapping the produced sparse trajectory is combined with the dense smoothed Hector trajectory. This is accomplished by performing an affine transformation to each subtrajectory which aligns the subtrajectory endpoints with the sparse trajectory. Headings of the poses of subtrajectories are interpolated from the sparse trajectory.

Finally the combined (or in the case of Hector the non smoothed one) trajectory is used to transform the scans made by the tilted data collection laser scanner to form 3D point cloud of the whole area. The scans are transformed point by point based on the position from SLAM and attitude from IMU, the transformation applied to each individual measurement is linearly interpolated from the transformations at

the beginning and end of the whole scan.

6 Experiments

To test our approach we made 5 test runs in the FGI office, which has both long corridors and cluttered spaces with limited visibility. A map of the area with labels is shown in Figure 10. The test runs were made in two sets, three with faster movement speed and Faro S120 scanner as the horizontal SLAM scanner and two with slower movement speed and the newer Faro X330 as the horizontal scanner. Details of the datasets are shown in Table 2. While each test run had different route they all started and ended in the top part of the upper corridor and consisted of one visit to the lower corridor and some exploration of the library.

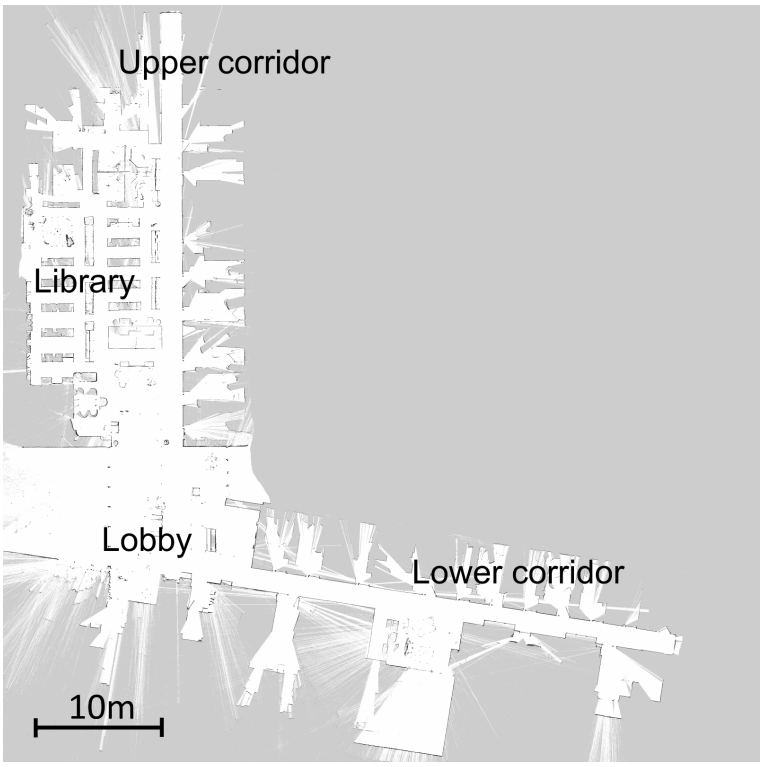


Figure 10: Map of the floor used for experiments with labels of different areas

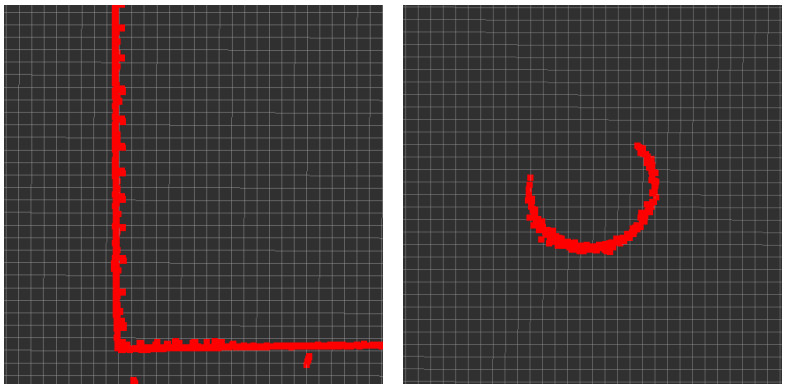
All tests were made with the same IMU logging settings, acceleration and angular velocity measurements were logged at 200 Hz and attitude calculated by SPAN at 20 Hz. Horizontal laser scanner maximum range was set to 41 meters, which is close to the the length of the longest corridor and enables the scanner to detect everything in its field of view.

6.1 Reference

Faro X330 laser scanner was used in TLS mode to generate a point cloud of the study area for geometric reference. Target spheres with 199 mm and 145 mm diameters were placed around the area and used for registering. Spheres were selected

Table 2: Test runs starting with X were made with X330 scanner as the horizontal scanner and ones starting with S with S120. Length is the length of the trajectory and speed is the average movement speed.

	Data acquisition settings and trajectory length				
Test	Rot. Rate	Ang. Res.	p/scan	Length	Speed
	Hz	mrad		m	m/s
X1	59.7	0.767	6828	256.9	0.27
X2	59.7	0.767	6828	265.6	0.25
S1	47.7	0.613	8536	268.5	0.52
S2	47.7	0.613	8536	312.1	0.52
S3	47.7	0.613	8536	232.6	0.49



(a) (b)

Figure 11: Details from the horizontal slice of the TLS scan used to generate the reference trajectory. The TLS scan points (red, point size 5 mm) do not align perfectly with each other which is an evidence of registration errors between different TLS scans. Grid cell size is 10 mm.

over paper or paddle targets as they provide superior registration accuracy [28]. Altogether 31 TLS scans were taken, which were then registered together into a single point cloud with the help of Faro Scene software. The mean standard deviation of the detected sphere locations from different scan locations was 4.7 mm on the registered point cloud. Visual inspection of a horizontal slice of the point cloud was performed and misalignments of similar, around 5 mm magnitude were detected as seen in Figure 11.

To obtain reference trajectories for the test runs, a 50 mm thick slice around the elevation of the horizontal laser scanner from the floor level (about 50 cm) was extracted from the reference point cloud and the scans collected during each of the test runs were matched to it with the Hector SLAM algorithm. The use of Hector SLAM instead of using Monte Carlo Localization or just matching the scans to the reference map is advantageous as there are places (under desks etc.),

which were occluded and not seen in the TLS based reference map. The SLAM algorithm can generate a map for those locations too and use it for localization. The parameters of the Hector SLAM were set to only update the underlying map with very low probability and so the original geometry of the TLS based reference map was preserved and used for matching when available.

After being generated by Hector SLAM the reference trajectories were interactively inspected by checking that each scan correctly matches the slice of the reference point cloud. A trajectory modifier tool, which enables easy inspection and modification if necessary, was created and used for this task. This inspection was necessary as the matching by Hector SLAM occasionally fails and the trajectory can deviate more than 50 mm from the correct one. Precise manual correction was extremely difficult with the scans distorted by movement and some small (under 10 mm) errors remain in the reference trajectories.

6.2 Calibration

The IMU and laser scanners all produce data in their own coordinate frames, to be able to combine measurements from them, the transformations between the coordinate frames must be known. An accurate estimate of the transformation between the horizontal scanner and IMU had been acquired during earlier use of the sensor module and there was no need for any further work with that transformation. The physical configuration for the second data collection laser scanner had not been used earlier and the transformation had to be found.

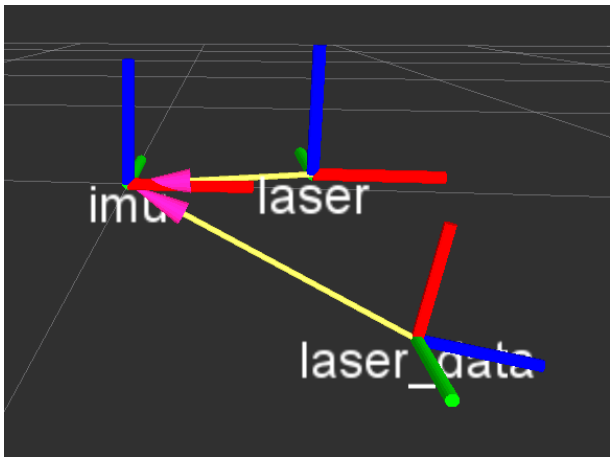


Figure 12: Coordinate frames of the Slammer platform, x axis is red, y is green and z is blue

As a first step, the offsets between the IMU and the data laser scanner were measured with a tape measure. A simple calibration routine was then devised to refine the tape measurements to an estimate closer to the real one. The routine utilizes the platforms ability to localize itself accurately and is based on the insight that if same the object is observed multiple times by the data collection scanner, the measurements will only align with each other when the transformation is correct.

A short section of a corridor was repeatedly measured with the MLS platform by moving in both directions and the trajectory was calculated in same manner as used to create reference trajectories in Section 6.1. A point cloud was then created and three perpendicular planes were used as the objects (two walls and a floor) to measure the quality of a transformation. A subset of points near the real location of a plane was selected for processing. Random Sample Consensus (RANSAC) [29] algorithm was used for detecting the plane and then the average distance from each point of the selected subset to the plane was calculated. This was repeated for all planes and the average distances from each plane were then averaged for transformation quality metric. Point Cloud Library (PCL) [30] was used to process the point cloud.

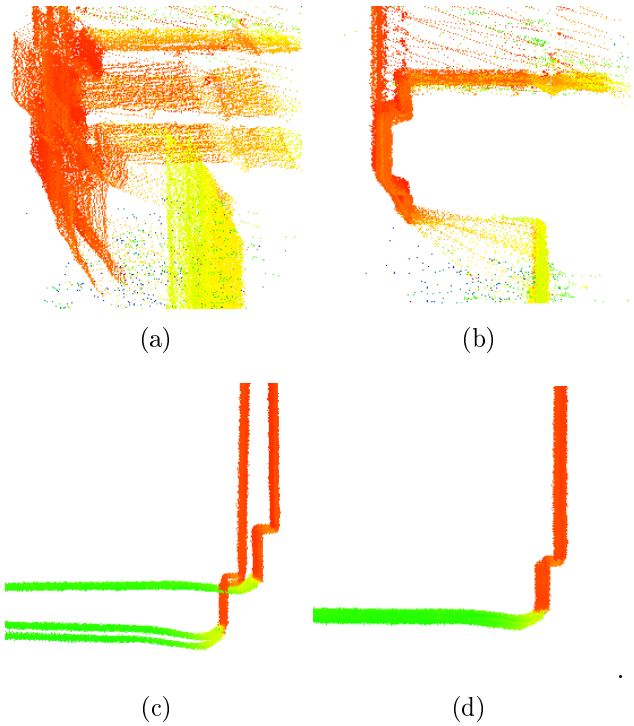


Figure 13: Result of the IMU → data collection scanner calibration. The left side is produced based on the initial measured estimates and the right side shows the situation after running the calibration routine. The object in top is a door frame seen from above and that in bottom is a side view of floor, trim and wall.

A supervised brute force approach was used to find the correct transformation. Limits and step sizes for all of the 6 parameters were set and all possible combinations were evaluated. The correct transformation was found by starting around the initial parameters with a large step size and progressively moving towards smaller step size around good transformations. While some automated optimization scheme could have been implemented and would have required no supervision and manual setting of the limits between changing step size to find the correct transformation, this simple approach was sufficient for this one time use. The influence of calibration can be seen in Figure 13. The transformation found is shown in Table 3. The

smallest step size used was 0.5 mm for translations and 5 mrad for rotations which give the upper limit for calibration accuracy. The actual accuracy is less than this because the accuracy of localization also limits the transformation accuracy available with this calibration approach as errors in localization translate to errors in the point cloud.

Table 3: IMU → data collection laser scanner transformation. Measured values are based on tape measurements and estimated rotations while calibrated values are acquired with the help of a calibration routine.

	Linear offsets			Rotations		
	x	y	z	roll	pitch	yaw
	mm	mm	mm	rad	rad	rad
Measured	472	3	-247	3.142	-1.396	0.0
Calibrated	462	6	-250.5	3.237	-1.409	-0.1

6.3 Evaluation of trajectories and effects of parameters

The Rawseeds benchmarking toolkit [31] was used with some modifications to compare the results of different algorithms. It provides tools to calculate Absolute Trajectory Error (ATE) and has functionality to draw figures of trajectories and errors.

Absolute Trajectory Error (ATE) is calculated as shown in Equation 6 where E_i^{trans} is the translational error of the ith pose in XY plane. The poses calculated with SLAM algorithms are compared with poses from the reference trajectory.

$$RMSE_{ATE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i^{trans})^2} \tag{6}$$

Equation 7 shows the calculation of the the rotational component of the error where E^{rot} is the difference in the heading angle between poses of the computed trajectory and corresponding reference one.

$$RMSE_{rot} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i^{rot})^2} \tag{7}$$

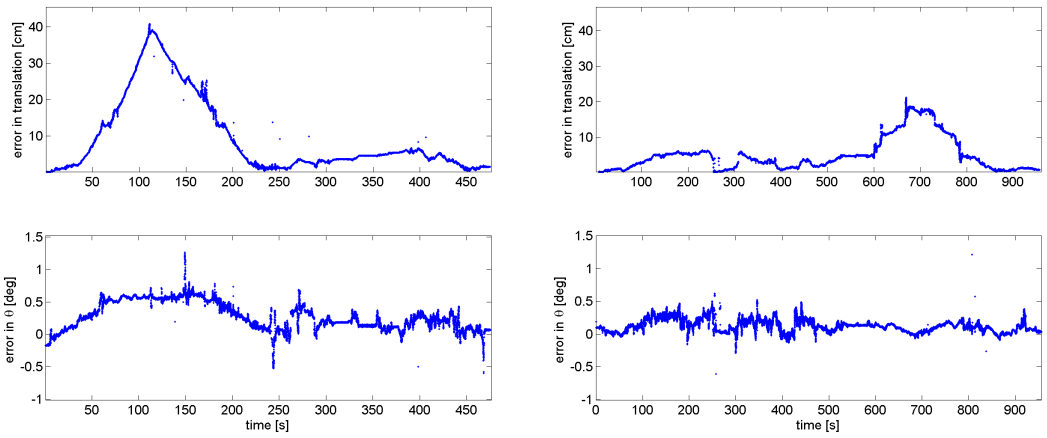
Absolute Trajectory Error is very sensitive to small errors in the beginning (especially in rotation) as they cause the rest of the trajectory to diverge from the correct one. The trajectories were tied to the corresponding reference ones and through them to the TLS point cloud by setting their starting pose to the reference trajectory starting pose. The results of first scan matches of SLAM processing have large effect on the precise orientation of the map produced and the random noise always present in the scan matching cause the orientation of the map and trajectory

to also fluctuate slightly. To reduce this effect, all trajectories were rotated in a way which gives best alignment with the reference based on the first 10 meters of the trajectory.

In Sections 6.3.1 to 6.3.3 the influence of several parameters of the algorithms is studied and the quality of trajectories produced with good parameter combinations is assessed. Parameters to be inspected were selected based on tests with a wider variety of parameters on a short section of test run S1 or because of their likely importance (for example map resolution).

6.3.1 Hector SLAM

Hector SLAM was tested with three occupancy grid maximum resolutions (1 cm, 2 cm and 4 cm), two occupancy grid pyramid level heights (smaller value giving 8 cm maximum grid cell size and larger giving 16 cm grid cell size) and with and without rotation compensation for a total of 12 parameter combinations. Average, minimum and maximum RMSE errors for translation and rotation for the different parameter combinations are listed in Table 4. The errors of the slower and faster movement test runs with X330 scanner are listed separately as they differed greatly from each other. Full results for each separate test run are listed in Appendix A.



(a) (b)

Figure 14: Errors as time series of two runs with Hector SLAM. Diagrams on the left side are from test run S3H241 and those on the right side are from X1H241. The large increase in the error is caused by visiting the lower corridor.

The bulk of the trajectory error for each test run was caused by a slight error in the estimated orientation when visiting the over 40 m long lower corridor of the building. This can be seen as the large rise in errors in Figure 14, while the magnitude of error differs, all test runs showed similar tendency. The corridor is quite feature poor and as such difficult for scan matching and as extra challenge the lobby which connects the two corridors contains glass and metal surfaces on multiple sides which further deteriorates scan matching performance in the crucial

Table 4: ATE metrics and heading standard deviation with 100 pose moving window for the test runs with Hector SLAM. The first column denotes the test run and parameter combination. The first character marks the test run type, S for Faro S120 and X for X330, H stands for Hector SLAM and the last three digits indicate the parameters used. The first one tells the occupancy grid resolution in centimetres, the second tells the number of levels in the occupancy grid pyramid and the last one tells whether rotation compensation was used (1) or not (0).

	Position RMSE			Heading RMSE			Heading
Test	Mean	Min	Max	Mean	Min	Max	SD_{100}
	mm	mm	mm	deg	deg	deg	mdeg
SH150	137.8	106.3	171.6	0.46	0.43	0.49	85.7
SH151	151.1	109.3	176.1	0.56	0.45	0.62	66.0
SH140	135.3	104.6	155.7	0.46	0.42	0.48	84.8
SH141	133.0	98.3	155.5	0.47	0.41	0.52	57.5
SH240	118.0	89.9	146.8	0.40	0.33	0.48	86.2
SH241	115.6	71.0	140.8	0.37	0.33	0.43	64.3
SH230	118.8	90.5	151.3	0.45	0.33	0.64	86.1
SH231	194.3	68.8	376.7	0.59	0.32	0.96	61.7
SH430	92.1	47.6	146.6	0.27	0.21	0.34	83.7
SH431	122.8	50.6	233.1	0.30	0.16	0.54	78.8
SH420	818.7	46.5	2341.4	0.47	0.22	0.92	88.4
SH421	485.8	55.0	1330.9	0.37	0.16	0.69	70.4
XH150	35.3	23.5	47.1	0.08	0.06	0.10	21.4
XH151	40.0	25.5	54.5	0.11	0.09	0.13	17.8
XH140	28.8	26.7	31.0	0.09	0.08	0.11	21.5
XH141	30.0	25.8	34.2	0.11	0.09	0.13	18.0
XH240	51.0	36.8	65.2	0.14	0.14	0.15	26.3
XH241	45.9	24.7	67.2	0.13	0.10	0.15	22.0
XH230	43.1	29.8	56.4	0.15	0.13	0.16	27.2
XH231	38.2	36.2	40.1	0.13	0.13	0.13	21.5
XH430	69.0	59.4	78.7	0.23	0.23	0.24	39.0
XH431	82.1	67.9	96.3	0.32	0.22	0.42	39.3
XH420	65.4	53.6	77.3	0.24	0.23	0.25	40.5
XH421	67.4	44.0	90.8	0.34	0.19	0.49	35.5

time when mapping of the lower corridor starts. Spikes in the error during visit to the lower corridor are caused by the position sliding along the length of the feature poor corridor. This is the cause for The high range of the scanner would enable it to see the ends of the corridor all the time which would add objects perpendicular to the length of the corridor to reduce the sliding tendency but unfortunately the corridor ends are mostly glass which cannot be reliably detected. With 40 m long corridor a 0.15 degree error in the orientation when entering and starting to map

the corridor would result in 10 cm position error at the opposite end of the corridor.

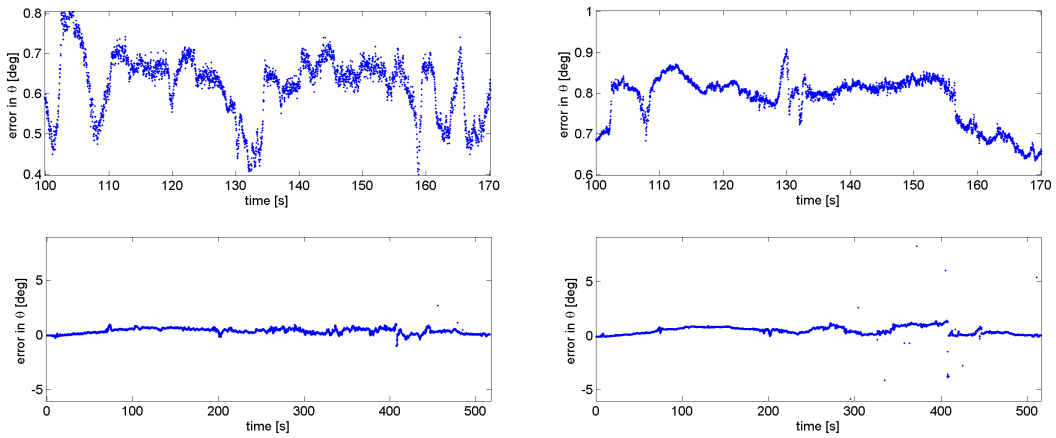
The maximum RMSE errors for SH231, SH420 and SH421 are all caused by the SLAM algorithm being unable to recover correctly from erroneous scan match, instead it starts to build new map slowly overwriting the old correct one. It is no coincidence that all these failures happened with the lower height setting of the occupancy grid pyramid. Additional large cell sized occupancy grids add a layer of robustness to the scan matching but this comes with a price. In the remaining 27 test runs with no major matching failure having the additional occupancy grid with 16 cm cell size increases the RMSE value in 14 cases while the opposite is true in only 5 cases with the difference in the remaining 8 runs being below 5%. The coarsest grid is used for scan matching first and the coarsenesses of it can cause the first level match to be further from the correct one than the scan matching starting pose.

Most noticeable thing about the results is the large difference in position accuracy between the test runs with different laser scanners and movement speed. While the X330 is upgraded version of S120 and provides better performance, the difference in accuracies is just too great to be explained by the similar but moderately updated scanner. The more likely culprit for majority of the increased accuracy is the almost halved movement speed of the X1 and X2 trajectories. Slower movement speed and higher scanner rotation rate help the scan matching in two ways, less movement leads to less of distortion to the scans while the transformation between two subsequent scans also stays smaller and easier to find as the search starts from the previous scan location.

The effect of the highest occupancy grid resolution setting on errors can also first seem counter intuitive, for XH test runs, higher resolution leads to lower errors while the opposite is true for SH tests. Again this is likely caused by the difference in movement and scanner rotation speed as the slight differences in the precision and noise handling of the laser scanners is not large enough to warrant such a drastic difference in SLAM behaviour. The faster movement affects both the map building and scan matching. Larger cell size can help mask the effect of movement related scan distortion during both scan matching and map building. With smaller cell size the inaccuracies in the scan matching cause the points to fall easier to different cells making the actual location of the walls less certain, with larger cell sizes the points can fall to the correct cells even when they are distorted or misplaced by localization errors.

The effect of slower movement speed on trajectory accuracy could imply that rotation compensation could well improve performance but the results evidently show that this is not the case. Out of the 30 test run results, the compensated trajectories were better in 11 cases, worse in 15 cases and in 4 runs there was no major difference (less than 5% difference from the non compensated one). In total the effect on trajectory accuracy in position and heading tilts to the negative and it clearly does not provide the improvement which was hoped from it.

Closer inspection of trajectory error time series with and without compensation revealed that while the compensation scheme has little effect on the absolute accuracy, it greatly reduces noise in the heading estimates as can be seen in Figure



(a) (b)

Figure 15: Effect of rotation compensation on Hector SLAM trajectory heading error, left side is without compensation and right with. Lower ones show the total trajectory while the upper ones show zoomed in detail. Rotation compensation increases smoothness while also adding single large errors. Left is from test run S1H150 and right from S1H151.

15. To better quantify this effect, moving standard deviation with 100 pose wide window was calculated for all trajectories and the result can be seen in Table 4. No such smoothing effect was observed for the position. As a down side the rotation compensation also increases both amount and magnitude of relatively large single errors in the heading estimates which is evident in the Figure 15 and in the large maximum heading errors which can be seen in Appendix A. The reason for these outliers could be in the SPAN central unit which occasionally drops measurements from the IMU. These gaps in the orientation record could result in bad estimates for the rotation as the tf system of ROS interpolates the orientation values from the messages with closest timestamps.

6.3.2 GMapping

Twelve parameter combinations were tested on GMapping. Three map resolutions (1 cm, 3 cm and 5 cm), two sigma values (0.01 and 0.05) which are used by the greedy endpoint matcher to smooth the map and two step sizes (0.04 and 0.02, angular and linear, meters for linear and radians for angular) used by the gradient descent based scan matcher pose optimizer. During match optimization the step size is halved when better matches cannot be found using the current step size. The halving is continued as long as the results improve or at least for a predetermined number of times.

Smoothed Hector trajectories with parameter combination HS241 were used as the initial trajectory providing laser odometry to the GMapping algorithm. The random nature of the sampling done by GMapping makes each run different. Each trajectory and parameter combination was processed three times to reduce the ef-

Table 5: ATE metrics for the test runs with GMapping. The first column denotes the test run and parameter combination. The first character marks the test run type, S for Faro S120 and X for X330, H stands for Hector trajectory and G stands for GMapping. The last three digits code for parameters used. The first one states the map resolution, the second represents the linear and angular optimization step size and the last one marks the sigma of scan end point matcher. All actual parameters are one hundredth of the given numbers. The parameter combination with lowest error for XG and SG trajectories is made bold.

	Position RMSE				Heading RMSE			
Test	Mean	SD	Min	Max	Mean	SD	Min	Max
	mm	mm	mm	mm	deg	deg	deg	deg
SG121	48.5	22.2	34.6	105.2	0.24	0.05	0.19	0.36
SG141	**31.9**	**5.5**	**25.5**	**39.8**	**0.21**	**0.02**	**0.19**	**0.24**
SG125	38.9	9.9	24.5	53.5	0.22	0.03	0.19	0.26
SG145	43.0	13.4	23.3	65.7	0.23	0.03	0.19	0.27
SG321	115.2	26.0	78.6	151.5	0.36	0.05	0.27	0.41
SG341	124.8	44.3	76.4	200.0	0.38	0.09	0.26	0.50
SG325	191.8	55.1	127.4	289.6	0.42	0.07	0.32	0.53
SG345	197.2	28.7	161.2	262.5	0.44	0.09	0.32	0.58
SG521	116.7	31.8	72.7	149.9	0.41	0.06	0.35	0.51
SG541	113.3	29.0	69.4	146.7	0.40	0.03	0.34	0.45
SG525	261.6	78.3	184.9	393.6	0.60	0.15	0.46	0.86
SG545	270.7	65.0	188.1	392.4	0.65	0.17	0.42	0.96
Mean	129.5	34.1	88.9	187.5	0.38	0.07	0.29	0.49
XG121	52.8	8.3	42.6	65.9	0.17	0.03	0.13	0.20
XG141	48.4	8.4	36.6	59.2	0.17	0.01	0.14	0.18
XG125	54.3	16.7	29.6	77.8	0.17	0.02	0.15	0.20
XG145	54.2	11.8	38.7	68.4	0.16	0.02	0.13	0.18
XG321	35.1	12.1	22.5	52.3	0.16	0.02	0.14	0.19
XG341	39.8	16.8	21.0	60.4	0.17	0.01	0.15	0.19
XG325	154.3	38.4	103.9	205.0	0.46	0.26	0.27	0.91
XG345	167.8	53.1	118.4	233.8	0.37	0.06	0.33	0.48
XG521	27.8	10.4	16.1	38.7	0.16	0.02	0.14	0.18
XG541	**25.2**	**9.6**	**14.4**	**35.7**	**0.15**	**0.01**	**0.14**	**0.16**
XG525	119.4	27.8	92.3	156.9	0.28	0.03	0.25	0.34
XG545	126.9	41.5	96.8	208.7	0.32	0.05	0.26	0.39
Mean	75.5	21.2	52.7	105.2	0.23	0.04	0.19	0.30

fect of randomness. In total 15 test runs for each parameter combination were done and the results were averaged for the analysing of the performance of the different parameter combinations. GMapping has a hard coded maximum size of 2048 measurements per scan and as scans in these test runs had considerably more mea-

surements, scans were subsampled to fit in to the limits of GMapping. The number of particles used was 100.

Table 6: ATE metrics for the test runs with GMapping for selected parameter combinations. The first column denotes the test run and parameter combination. The first character marks the test run type, S for Faro S120 and X for X330, H stands for Hector SLAM trajectory which after smoothing is used as initial trajectory and G stands for GMapping. The last three digits code for parameters used. The first one states the map resolution, the second gives the linear and angular optimization step size and the last one marks the sigma of scan end point matcher. The parameter combination with lowest error for each test run is made bold.

	Position RMSE				Heading RMSE			
Test	Mean	SD	Min	Max	Mean	SD	Min	Max
	mm	mm	mm	mm	deg	deg	deg	deg
S1H241	135.1	0.0	135.1	135.1	0.43	0.0	0.43	0.43
S1G121	45.5	4.5	41.1	50.2	0.19	0.002	0.19	0.19
S1G141	**34.6**	**0.9**	**34.0**	**35.7**	**0.19**	**0.005**	**0.19**	**0.20**
S1G521	132.7	21.7	107.7	146.7	0.46	0.066	0.39	0.51
S1G541	123.1	14.3	108.3	136.9	0.42	0.026	0.40	0.45
S2H241	71.0	0.0	71.0	71.0	0.33	0.0	0.33	0.33
S2G121	36.1	1.1	35.0	37.2	0.24	0.006	0.24	0.25
S2G141	**35.4**	**5.6**	**29.1**	**39.8**	**0.23**	**0.002**	**0.23**	**0.24**
S2G521	77.7	4.6	72.7	81.6	0.36	0.010	0.35	0.37
S2G541	78.7	14.4	69.4	95.2	0.37	0.026	0.34	0.39
S3H241	140.8	0.0	140.8	140.8	0.36	0.0	0.36	0.36
S3G121	64.0	36.7	34.6	105.2	0.28	0.067	0.23	0.36
S3G141	**25.8**	**0.4**	**25.5**	**26.2**	**0.21**	**0.006**	**0.21**	**0.22**
S3G521	139.8	9.2	132.0	149.9	0.41	0.024	0.39	0.43
S3G541	138.0	9.0	128.7	146.7	0.40	0.022	0.38	0.42
X1H241	67.2	0.0	67.2	67.2	0.15	0.00	0.15	0.15
X1G121	54.2	10.2	46.7	65.9	0.15	0.022	0.13	0.17
X1G141	54.4	6.4	47.1	59.2	0.16	0.017	0.14	0.18
X1G521	37.2	1.3	36.3	38.7	0.17	0.008	0.16	0.18
X1G541	**33.8**	**1.7**	**32.5**	**35.7**	**0.16**	**0.003**	**0.16**	**0.16**
X2H241	24.7	0.0	24.7	24.7	0.10	0.0	0.10	0.10
X2G121	51.4	7.9	42.6	58.0	0.19	0.007	0.18	0.20
X2G141	42.4	5.2	36.6	46.7	0.17	0.004	0.17	0.17
X2G521	18.5	2.1	16.1	20.2	0.14	0.002	0.14	0.14
X2G541	**16.6**	**2.0**	**14.4**	**18.2**	**0.14**	**0.001**	**0.14**	**0.14**

Average, standard deviation and minimum and maximum RMSE errors for translation and rotation for the different parameter combinations are listed in Table 5. Test run wise results can be found in Appendix B. Again as with Hector, the slower

movement X330 trajectories show vastly smaller errors over the faster S120 trajectories. Out of the parameters, the effect of sigma value on trajectories is the clearest; the larger value markedly increasing the RMSE values with larger map cell sizes. Smoothing makes finding the correct transformation for scan matching without getting stuck in local optima easier but, as a drawback, it decreases achievable accuracy. When GMapping matches scans to the map, larger cell size already makes the response smoother and doing too much further smoothing is clearly detrimental to the SLAM result. Different map resolutions require different sigma values.

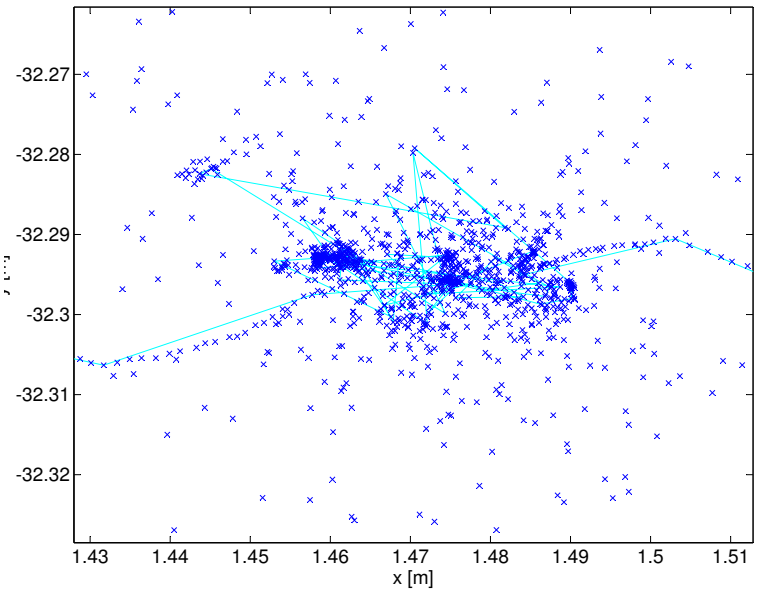
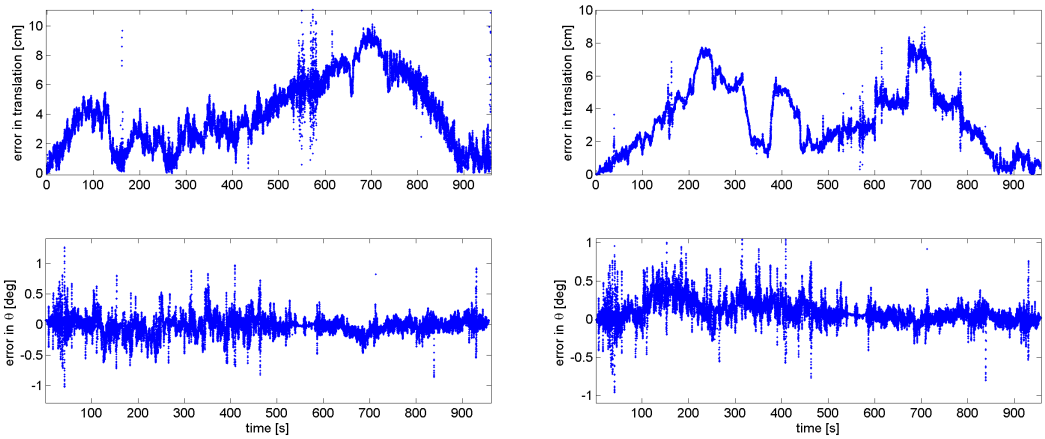


Figure 16: The effect of staying stationary for GMapping. The position starts to oscillate with large magnitude when compared to the other errors in the trajectory. The crosses connected by lines represent poses from GMapping while the other crosses are from the dense initial trajectory.

Visual inspection of the trajectories reveals that GMapping does not handle updates well while staying stationary with the higher resolution settings. With the higher resolution settings the relatively large scan match optimization step size could cause the trajectory to become noisy and when the position is not changing the location starts to oscillate with growing amplitude. The largest errors in trajectories are caused by this behaviour. This can be seen in Figure 16. The effect of this behaviour on trajectory error is further amplified during post processing when the dense Hector trajectory is combined with the sparse one from GMapping. Transforming the subtrajectories to fit the gaps between random and large movements when there is almost no movement causes the points to spread widely around the pose where scanner was actually stopped. This is a problem mostly for the X330 trajectories as there were fewer stops and they were shorter on the S120 trajectories. The longer the scanner stays stationary the larger the oscillation becomes.

In S test runs higher map resolution leads to lower errors while in the X test runs the opposite is true. A major reason for this is the smoother trajectory with 5 cm

cell size as seen in Figure 17. While the XG5 trajectories still achieve smaller errors than XG1 trajectories, the difference is less pronounced than could be inferred from the RMSE values as they are polluted by position oscillating. More sigma values should be tried to find the one fitting with each map resolution level to get a better idea on map resolution effect on trajectory quality. In principle higher resolution should provide better performance as there is more information available for scan matching.



(a) (b)

Figure 17: Left is from test run X1G121 with 1 cm resolution and right from X1G521 with 5 cm resolution. The 5 cm cell size reduces the random errors while not moving which causes the X1G521 to have smaller RMSE values while the actual errors in trajectory are not so different.

The lack of smoothness in the trajectory is evident when looking at error time series in Figure 17. With 1 cm resolution the error oscillates constantly in the centimetre range and while the trajectory with 5 cm resolution is smoother there are many large jumps. When compared to Hector SLAM results the GMapping heading estimate is also more noisy (Figure 17 vs. Figure 14). This is caused by the combined effect of interpolating the heading values for the subtrajectories between GMapping poses and from the lack of smoothness in the overall GMapping trajectory.

The results in Table 6 reveal that larger scan matcher optimization step sizes increases the accuracy of trajectories. While both values are large enough to account for the errors in the initial guess from Hector, the scan matching results are also used to calculate a model from which the next set of particles are drawn and sampling a larger area improves the performance. Smaller optimization step sizes should also be tested with the higher map resolution as they could eliminate the oscillation in the trajectories.

In general GMapping with its multiple map hypothesis seems to be able to improve Hector results but unfortunately at least with current parameters the trajectory is far from smooth which is a serious problem whe it is used to produce a

3D point cloud of a building.

6.3.3 Karto

Karto with its explicit loop closure functionality seemed like the most promising approach and to get a better idea of its performance it was tested with a total of 48 setting combinations. Two scan matcher resolutions in the higher end of usable resolutions, 5 mm and 2.5 mm and four (0.1, 0.3, 0.6 and 0.9) values for a multiplier which penalizes scan matches differing from the initial guess given by odometry. These parameters were selected to gain some insight of the quality of Karto's scan matcher. Giving a larger value to the parameter which penalizes transformations differing from the initial guess should force it to follow the initial trajectory more closely. Three loop closure minimum response values, 0.8, 0.7 and 0.6 for coarse matching and correspondingly 0.7, 0.6 and 0.5 for fine matching were tested. Both the default loop closure method and the delayed one were tested.

For test runs X1, X2 and S2 the parameters were additionally evaluated with initial Hector trajectory calculated without rotation compensation (only with delayed loop closing). As the only advantage of the rotation compensation schemes was found to be reduced jitter in the heading value (Section 3.1) which is currently ignored as the heading values between Karto matches are interpolated, no attempt was made to analyse the effect of initial trajectory rotation compensation on Karto. These trajectories were used as additional samples when testing different parameter combinations.

Hector trajectories with parameter combinations H241 and H240 were smoothed and used as a laser based odometry for the algorithm. Results of the test runs can be seen through Tables 7 to 9. Full results for every parameter combination and test run can be found in Appendix C. Again the XK datasets give vastly lower errors when compared to the SK datasets for same reasons as explained in Section 3.1. The effect of scan matcher resolution can be seen in Tables 7 and 8 where mean RMSE errors have been calculated for both scan resolutions. In SK sets the 0.0025 resolution is clearly better but the situation is more even in the XK sets. Closer inspection of the test runs X1 and X2 reveal that also in the X2 test run the 0.0025 resolution gives better results and that it is only the X1 test run where coarser resolution gives better scores (and with a wide margin). In X1 all the 6 parameter combination which led to worse RMSE value than the initial trajectory were with 0.0025 resolution while in the other test runs the worst RMSE values were mostly with the 0.005 resolution. The better performance with higher resolution was expected as higher resolution enables the scan matcher to find matches closer to the truth and this should improve the trajectory. The reason for this not applying for X1 trajectory is mystery.

The influence of scan matching distance penalty multiplier and loop closure response threshold on the RMSE value was analysed by calculating correlation coefficients between the parameters and the RMSE value. No correlation was found for the penalty multiplier and only a weak one for the response threshold. In delayed loop closing datasets there was a weak positive correlation of 0.099 and with normal

Table 7: ATE metrics and the number of loop closures for the test runs with Karto. The RMSE values are calculated from 7 datasets for S and 6 datasets for the X tests. The first column denotes the test run and parameter combination. The first character marks the test run type, S for Faro S120 and X for X330, K stands for Karto and the last three digits indicate the parameter values used. The first number denotes the coarse loop closure minimum response value. The second tells the scan match resolution, 5 for 5 mm and 2 for 2.5 mm and the last one marks the scan matching penalty multiplier.

	Position RMSE				Heading RMSE				Loops
Test	Mean	SD	Min	Max	Mean	SD	Min	Max	
	mm	mm	mm	mm	deg	deg	deg	deg	N
SK621	49.1	19.9	15.6	78.2	0.26	0.05	0.20	0.32	14.9
SK721	64.1	30.5	18.4	117.5	0.29	0.07	0.21	0.41	10.9
SK821	56.7	49.9	17.6	164.3	0.30	0.11	0.20	0.53	6.6
SK623	47.7	28.3	17.5	99.9	0.28	0.04	0.21	0.35	15.3
SK723	67.9	51.3	22.3	166.2	0.32	0.12	0.21	0.57	11.6
SK823	40.3	20.2	16.9	79.4	0.28	0.04	0.19	0.31	7.1
SK626	49.0	32.0	16.0	111.9	0.28	0.05	0.22	0.37	15.3
SK726	51.6	38.9	28.6	138.1	0.28	0.08	0.22	0.46	11.3
SK826	66.4	19.8	37.8	88.5	0.31	0.04	0.25	0.36	6.4
SK629	44.0	20.1	25.8	71.6	0.27	0.05	0.22	0.33	15.6
SK729	44.7	17.5	18.8	60.7	0.27	0.03	0.23	0.30	10.9
SK829	57.5	22.1	40.9	98.7	0.30	0.05	0.23	0.39	6.4
Mean	53.2	29.2	23.0	106.3	0.29	0.06	0.22	0.39	11.0
SK651	58.3	41.2	29.8	123.6	0.26	0.06	0.21	0.38	14.6
SK751	66.5	23.8	36.0	103.3	0.28	0.05	0.22	0.35	10.4
SK851	68.3	29.4	34.8	112.4	0.27	0.05	0.20	0.35	7.1
SK653	76.3	42.3	31.3	143.5	0.31	0.07	0.22	0.43	15.6
SK753	101.1	45.9	49.2	142.8	0.34	0.07	0.25	0.41	12.7
SK853	611.3	1340.7	68.1	3650.6	1.59	3.37	0.27	9.23	5.6
SK656	121.5	96.2	34.0	255.1	0.57	0.49	0.25	1.28	16.1
SK756	133.8	93.2	21.9	269.3	0.61	0.50	0.25	1.40	13.1
SK856	144.3	85.6	51.0	254.5	0.62	0.46	0.25	1.28	8.4
SK659	70.2	28.0	36.7	116.7	0.28	0.05	0.19	0.36	14.7
SK759	71.8	40.6	21.3	119.7	0.29	0.06	0.18	0.35	11.6
SK859	83.5	46.8	20.0	158.4	0.31	0.07	0.21	0.42	6.4
Mean	133.9	159.5	36.2	454.2	0.48	0.44	0.23	1.35	11.4

loop closing behaviour the correlation was weakly negative ,-0.1. While weak they do make sense as the delayed loop closure would benefit from lower threshold as more (still good) loop closures could be made and for the normal loop closing if loops can be found, higher threshold ensures good quality loop closures. The effect

Table 8: ATE metrics and the number of loop closures for the test runs with Karto. The RMSE values are calculated from 7 datasets for S and 6 datasets for the X tests. The first column denotes the test run and parameter combination. The first character marks the test run type, S for Faro S120 and X for X330, K stands for Karto and the last three digits indicate the parameter values used. The first number denotes the coarse loop closure minimum response value. The second tells the scan match resolution, 5 for 5 mm and 2 for 2.5 mm and the last one marks the scan matching penalty multiplier.

	Position RMSE				Heading RMSE				Loops
Test	Mean	SD	Min	Max	Mean	SD	Min	Max	
	mm	mm	mm	mm	deg	deg	deg	deg	N
XK621	28.1	21.5	12.3	69.5	0.14	0.04	0.11	0.23	16.3
XK721	21.3	7.2	15.8	35.3	0.13	0.03	0.10	0.18	15.3
XK821	27.5	8.5	12.8	35.7	0.15	0.03	0.12	0.20	9.7
XK623	21.0	7.3	15.2	35.1	0.13	0.02	0.11	0.16	16.8
XK723	24.1	11.4	15.0	44.8	0.13	0.02	0.12	0.18	14.8
XK823	35.9	17.6	15.4	64.5	0.15	0.04	0.12	0.22	9.7
XK626	34.2	25.9	13.6	71.6	0.16	0.07	0.12	0.28	16.7
XK726	31.1	17.7	19.3	66.0	0.15	0.05	0.11	0.24	14.3
XK826	29.6	9.1	20.5	43.2	0.15	0.02	0.12	0.19	9.2
XK629	28.8	10.1	20.6	47.9	0.14	0.02	0.11	0.17	16.8
XK729	34.9	14.5	19.0	60.5	0.15	0.03	0.13	0.22	14.3
XK829	26.8	10.0	12.1	41.7	0.14	0.02	0.12	0.18	9.5
Mean	28.6	13.4	16.0	51.3	0.14	0.03	0.12	0.20	13.6
XK651	28.5	8.9	16.8	42.3	0.13	0.02	0.11	0.16	16.2
XK751	24.5	7.9	14.8	35.2	0.13	0.02	0.11	0.15	14.8
XK851	20.1	4.8	13.6	26.0	0.13	0.02	0.11	0.16	8.2
XK653	32.6	6.3	25.5	43.0	0.13	0.02	0.11	0.15	16.7
XK753	26.2	10.7	15.1	42.9	0.13	0.03	0.11	0.19	14.7
XK853	22.1	4.6	16.6	28.1	0.13	0.02	0.11	0.16	7.7
XK656	26.4	15.4	10.7	54.9	0.14	0.04	0.11	0.22	17.2
XK756	27.5	7.6	16.5	40.4	0.13	0.03	0.11	0.19	14.2
XK856	27.1	4.2	23.5	34.0	0.14	0.03	0.11	0.19	7.8
XK659	27.2	7.9	19.5	40.2	0.14	0.03	0.11	0.18	16.8
XK759	31.3	13.3	15.9	49.3	0.14	0.04	0.10	0.22	13.3
XK859	20.2	2.6	17.3	23.6	0.13	0.03	0.11	0.17	7.8
Mean	26.1	7.8	17.1	38.3	0.13	0.03	0.11	0.18	12.9

of matching penalty multiplier were found to be negligible, the use of intermediate values from 0.3 to 0.6 seemed to lead to some increase in error but as a whole they do not seem like very important parameters.

The performance of the two loop closing strategies were compared on each test

run and the results are listed in Table 9. Two cases for the both closing strategies are listed, one with the preferred response threshold (0.6 for delayed, 0.8 for normal), and one averaged over all response threshold values. The resolution used was 0.0025 as it gives better results on average and eliminates many parameter combinations which resulted in worse solution than initial trajectory.

Selecting the response threshold accordingly gives improvement in a majority of cases. The effect is especially clear for the delayed loop closure where using the lower threshold gives smaller average and smallest minimum error on all test runs. With suitable response thresholds the delayed loop closing gives better results than the normal loop closing on 4 out of 5 test runs and while the improvement gain is not very large or clear in the S sets the result still imply that delayed loop closing is the preferred method to be used.

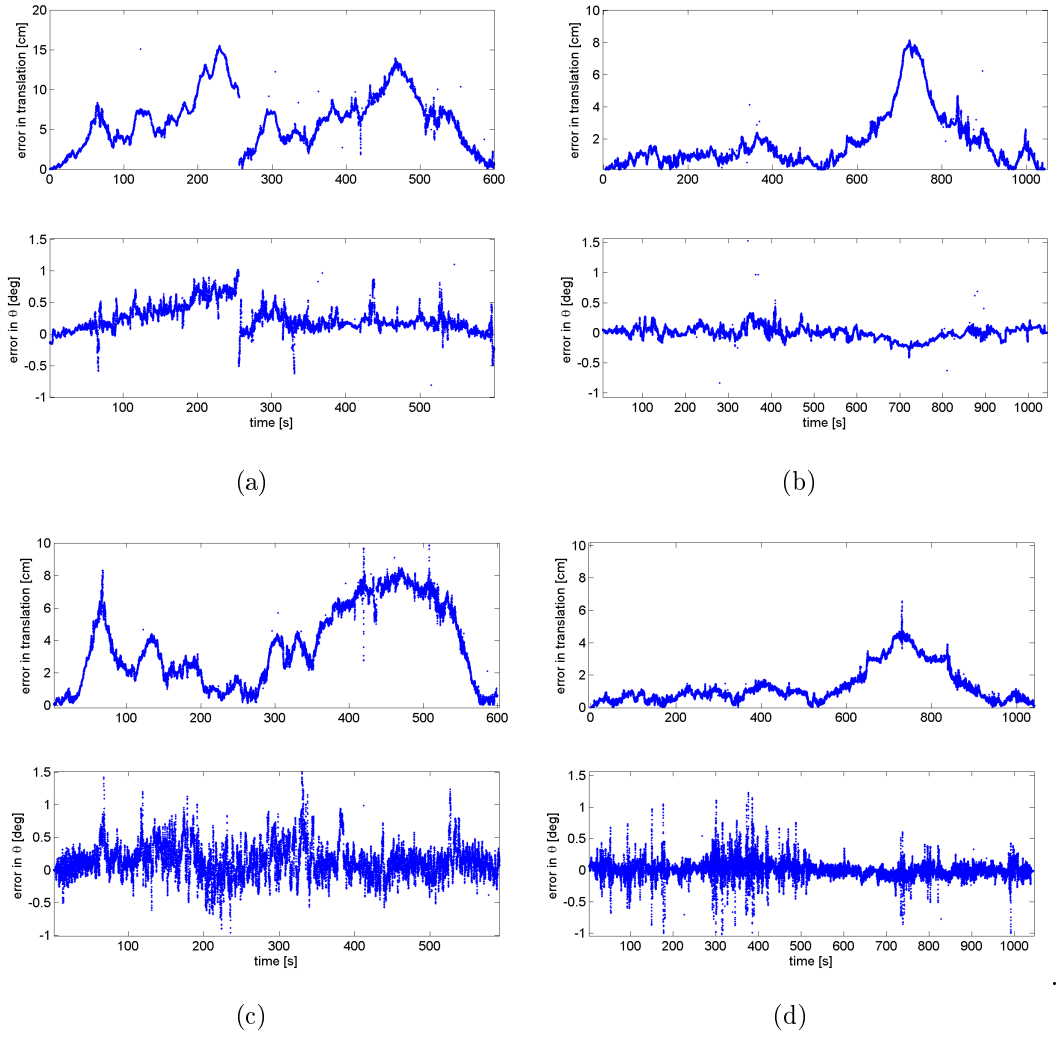


Figure 18: Errors from test runs S2K6211 and X2K6211 and corresponding initial Hector trajectories S2H241 and X2H241 as time series.

With the higher In all of the test runs Karto provides clear improvement to the

Table 9: ATE metrics of the different loop closing strategies. The first column denotes the test run and parameter combination. The first character marks the test run type, S for Faro S120 and X for X330, K stands for Karto and the last three digits indicate the parameter values used. Accuracy of Delayed loop closing is compared to the normal one. RMSE values are calculated from 8 datasets for KD62, 4 for KO82, 24 for KD2 and 12 for KO2. Last number tells the scan match resolution, 5 for 5mm and 2 for 2.5mm. The parameter combination with lowest error for each test run is made bold.

	Position RMSE				Heading RMSE			
Test	Mean	SD	Min	Max	Mean	SD	Min	Max
	mm	mm	mm	mm	deg	deg	deg	deg
S1H241	135.1	0.0	135.1	135.1	0.43	0.0	0.43	0.43
S1KD62	46.5	13.6	27.6	59.01	0.26	0.03	0.22	0.29
S1KO82	**31.8**	**10.6**	**16.9**	**40.90**	**0.22**	**0.03**	**0.19**	**0.25**
S1KD2	69.8	40.4	27.6	166.21	0.31	0.10	0.22	0.57
S1KO2	59.3	36.7	16.9	148.37	0.28	0.07	0.19	0.43
S2H241	71.0	0.0	71.0	71.0	0.33	0.0	0.33	0.33
S2KD62	**47.1**	**17.1**	**31.8**	**71.59**	**0.29**	**0.03**	**0.27**	**0.33**
S2KO82	50.5	26.5	26.9	88.49	0.31	0.04	0.28	0.36
S2KD2	69.2	38.5	31.8	164.35	0.33	0.07	0.27	0.53
S2KO2	59.5	23.4	26.9	109.95	0.31	0.03	0.27	0.36
S3H241	140.8	0.0	140.8	140.8	0.36	0.0	0.36	0.36
S3KD62	**46.4**	**36.9**	**15.6**	**99.94**	**0.27**	**0.06**	**0.20**	**0.35**
S3KO82	50.0	22.1	26.1	77.39	0.28	0.03	0.25	0.32
S3KD2	61.3	47.3	15.6	205.45	0.28	0.05	0.20	0.36
S3KO2	49.0	48.9	16.0	205.45	0.26	0.05	0.21	0.36
X1H241	67.2	0.0	67.2	67.2	0.15	0.00	0.15	0.15
X1KD62	**20.5**	**4.1**	**16.8**	**26.29**	**0.12**	**0.01**	**0.11**	**0.12**
X1KO82	42.6	15.9	27.7	64.45	0.17	0.03	0.14	0.22
X1KD2	29.5	9.1	16.8	44.35	0.14	0.03	0.11	0.20
X1KO2	43.8	18.4	16.7	69.50	0.17	0.04	0.10	0.24
X2H241	24.7	0.0	24.7	24.7	0.10	0.0	0.10	0.10
X2KD62	**16.1**	**3.7**	**12.3**	**21.19**	**0.12**	**0.00**	**0.12**	**0.12**
X2KO82	21.7	7.0	12.1	28.07	0.13	0.01	0.12	0.14
X2KD2	20.6	6.6	12.3	32.11	0.12	0.01	0.10	0.14
X2KO2	22.4	8.6	12.1	47.92	0.13	0.01	0.10	0.17

trajectories over Hector with the delayed loop closing with low loop closure threshold providing the best results in 4 out of 5 test runs as seen in Table 9. While there is variation in the results and sometimes Karto fails giving out a worse trajectory, in almost every case there is significant improvement in the trajectory.

6.4 Discussion on used algorithms

No single SLAM algorithm was able to provide a trajectory filling the requirements for point cloud creation of being dense, smooth and globally consistent. Hector SLAM fills the first two requirements but it has no functionality to reduce the errors in trajectory caused by accumulating drift except for discrete jumps. On the other hand GMapping and Karto both have problems if the displacement between scans is too small and can not provide reliable trajectory if they need to process every subsequent scan.

Combining the dense Hector SLAM trajectory with GMapping or Karto provides a solution which can produce a trajectory meeting the requirements. By using the trajectory from Hector as the initial guess, the Karto library is working as a kind of ad-hoc backend for the Hector SLAM. This overcomes the problems caused by the lack of backend in Hector while enabling Karto to benefit from Hector's high quality scan matcher. While not containing an optimizing backend, the multiple hypothesis of GMapping provides similar capability.

The large errors caused by the localization diverging with GMapping when not moving made the RMSE values less valuable for estimating the actual trajectory accuracy. The inspection of error time series and trajectories made it clear that Karto provided trajectories which were more coherent and closer to the reference. This inspection also revealed that with Karto the smoothness of the Hector trajectory is better preserved than with GMapping. Of course it must be taken in to account that more time was spent finding good parameters for Karto and Hector than for GMapping. With better parameter values the jitter in the GMapping trajectory might disappear altogether.

GMapping processing also contains a random element in both the resample and the motion model sampling steps, which adds an additional source of uncertainty to the process. Getting a different trajectory every time the same dataset is processed is not desirable but the differences are small and total failure is unlikely. It should be noted that Karto too fails occasionally, either through the error metric of optimization becoming Not A Number (NaN) or with loop closures associating wrong scans together. This can result in a worse trajectory than the initial trajectory given by Hector SLAM without the trajectory being noticeable wrong.

An attempt was also made to process the data again with Hector SLAM. The smoothed trajectory from the first run was used to transform the scans to compensate for the distortions caused by movement. Unfortunately this just resulted in larger errors to the trajectory. One likely cause for this behaviour is the small errors in the initial trajectory, which get magnified. If the movement estimate used for laser scan transformation is incorrect, the scan can get even more distorted which results in even larger error in the scan matching. The smoothing used was quite conservative in changing the trajectory and only removed large jumps. More aggressive smoothing might provide better performance.

For Karto one obvious source of improvement would be the increasing of scan matching resolution, but unfortunately it makes the algorithm unstable. With the highest setting used so far (2.5 mm) the scan matching occasionally fails with rota-

tional component becoming NaN, which causes the algorithm to crash. Setting the resolution to 2 mm made this happen noticeably more often and values less than 2 mm caused Karto to crash always. The algorithm should be studied in detail to see if the source of the crashes could be identified and fixed. This

The developed delayed loop closure behaviour should also be studied further. The results in Section 3.3 show that, while in general delaying the closing of a loop gives improved performance, with some settings it actually worsens the situation. In theory the same loops are closed with the both methods the only difference being that with delayed closing the matches are better. The reason why this still gives worse trajectories should be investigated.

Significant accuracy gains could also be achieved by improving the combining of dense and sparse trajectories. An obvious first step would be the use of real heading values for the subtrajectories instead of the interpolated ones. Another error source is that when a subtrajectory between two optimized poses is transformed to fit between the endpoints, it is scaled by the same amount in both X and Y directions. This scaling can magnify small random jitter in the localization to centimetre range when the scanner is almost stationary. This effect could be reduced by scaling only along the line between the endpoints.

6.5 Evaluation of point cloud

X1KD621 and X2KD621 trajectories processed by Karto and X2H241 trajectory processed by Hector SLAM were selected for creating point clouds from data scanner scans and for the evaluation of the full pipeline (an smoothed version of the X2H241 trajectory was used as initial trajectory for X2KD621). The selected trajectories were among the most accurate ones with least error. The trajectory error metrics can be seen in Table 10. GMapping trajectories were not used due to lack of smoothness. The computed trajectories were combined with the data scanner to generate 3D point clouds of the FGI study area. The point cloud data was scanned with 30 Hz scan frequency and 244,000 points per second with maximum range of 6.0 m. The point clouds were validated for geometric accuracy against the TLS reference point cloud also used for the SLAM trajectory evaluation. An example of a point cloud data collected in X2KD621 is shown in Figure 19. The point cloud in X1 includes 31 million points, while the X2 cloud has about 40 million points in total.

The geometric error of the point clouds was analysed on seven pylons in the study area. Some of them are visible in the figure 20. The centre of the pylon was measured from the TLS and Slammer point cloud interactively. The resulting errors are seen in Table 11. The maximum errors for the both cases were found in the lower corridor. The error grows steadily starting from the lobby as could be expected based on the trajectory errors.

To get the scale of the error over the whole building the lengths of the main corridors were measured. From the TLS cloud the upper corridor length was determined to be 43.498 m, X1KD621 shows 43.492 m, and X2KD621 43.501 m length correspondingly. The lower corridor length from the TLS cloud was 44.039 m, while it was found to be 44.056 m for the X1KD621 data, and 44.068 m for the X2KD621.

Table 10: ATE errors for the selected trajectories and initial trajectories used for Karto.

Test	Position error			Heading error		
	RMSE	SD	Max	RMSE	SD	Max
	mm	mm	mm	deg	deg	deg
X1HS241	44.3	24.3	113.7	0.15	0.04	1.21
X1KD6211	16.8	12.3	63.1	0.11	0.06	2.09
X2H241	24.7	17.5	81.5	0.10	0.07	1.62
X2HS241	25.9	13.3	64.0	0.10	0.04	1.62
X2KD6211	12.3	7.7	35.4	0.12	0.07	1.41

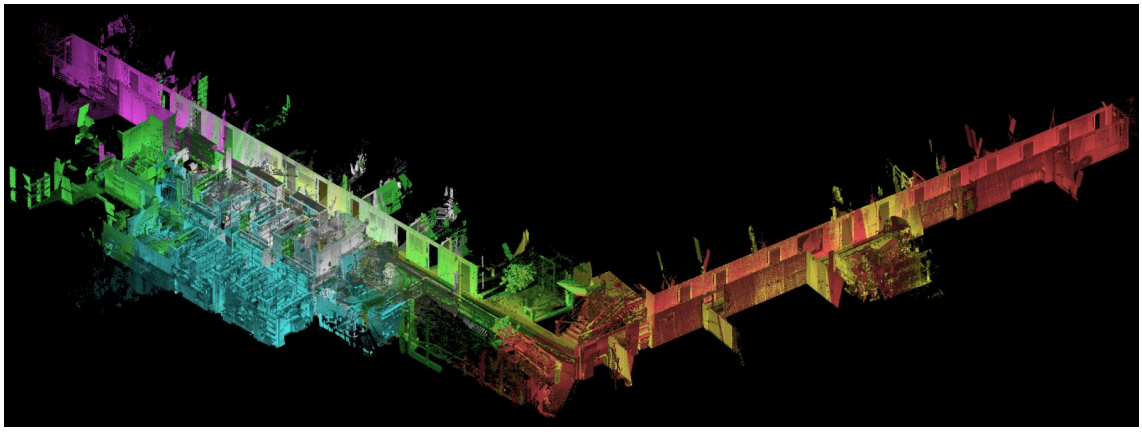


Figure 19: 3D point cloud of the whole floor collected with the Slammer system from test X2KD621. Point coloring is based on the laser intensity and data file number to illustrate the progress of the data acquisition.

Table 11: Planar errors in the point cloud measured on 7 pylons. N is the number of locations compared.

	X1KD621	X2KD621	X2H241
	mm	mm	mm
RMSE	13.1	12.8	16.2
SD	7.3	12.3	9.8
Minimum	3.6	0.1	1.7
Maximum	31.4	47.0	28.7
N	18	19	19

The errors are of an order of magnitude of 0.01-0.1% of the total length which is extremely small. However, the horizontal location error for the data point cloud varies from place to place. This variation is summarized in Table 11.

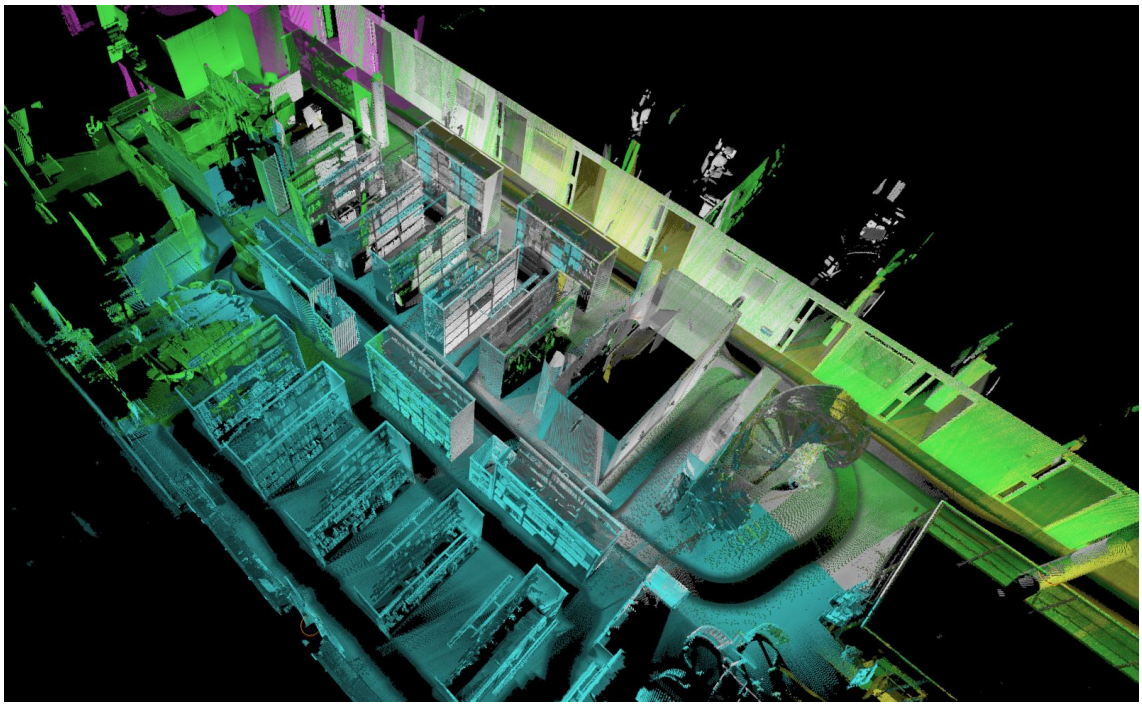
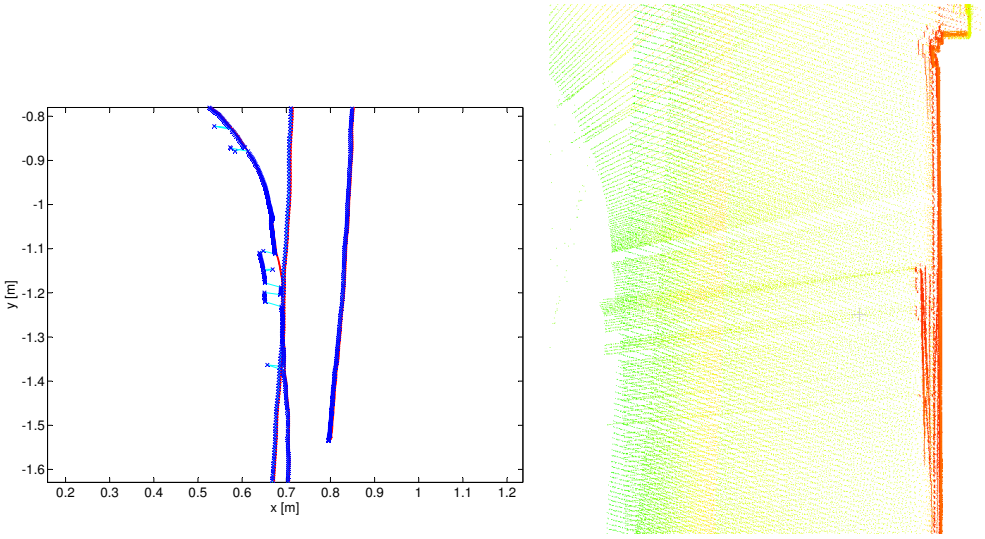


Figure 20: 3D point cloud of the library area from test run X2H241. The precision and accuracy of the data is well represented, regardless of the complexity of the scene. Point colouring is based on the laser intensity and data file number to illustrate the progress of the data acquisition.

Figure 21: Details of the building wall from X1KD621, doors and interior features captured by Slammer. Points are colored by laser intensity.

The X2H241 trajectory had an accuracy comparable to that of Karto trajectories and the point cloud produced showed no great errors and actually had the smallest maximum error as seen in Table 11. This does not mean that Hector trajectories in general are suitable for generating point clouds as X2H241 is exceptional among

the Hector trajectories with its small deviation from the reference and lack of large jumps.



(a) (b)

Figure 22: The effect of returning to previously mapped area after error in the position has accumulated for extensive time period. The left side shows the trajectory (red is the reference) and the right side the resulting point cloud of test run X1H241.

In the X2 test run the library was explored by progressively moving to the closest unexplored corridor starting from the upper corridor. This suits well with Hector SLAM as it minimizes the lengths of stints exploring new areas. In the X1 test run the library was explored in an opposite way by starting with long loop around the furthest corridors. The effect can be seen in trajectory X1H241, part of which is shown in Figure 22. Just before entering the area shown in the Figure, the position had jumped to one matching the area mapped in the beginning but the map used by the algorithm still contains parts mapped before the jump and the position starts oscillating between the two competing map representations. This manifests itself as ghost walls in the produced point cloud.

7 Conclusions and Future Work

In this study we investigated the use and efficiency of high-end laser scanners and offline processing with SLAM algorithms to compute trajectories of a MLS platform and used them to generate 3D point clouds indoors in the absence of GNSS signals. The trajectory estimation was initially carried out using Hector SLAM with its high quality scan matcher to provide an initial trajectory for Karto and GMapping algorithms. While Hector alone can provide good quality trajectories and GMapping can somewhat improve them, Karto with its explicit loop closure behaviour and a backend for optimizing poses to reduce cumulating scan matching errors turned out to be a superior choice.

For investigating the use of Slammer and combination of different SLAM algorithms, five different trajectories were collected and processed using different parameter sets. The resulting trajectories were evaluated against a TLS reference. The results indicate that the scanner version used for SLAM might have a role in the performance of the algorithms, but most probably the differences are mostly due to increased scan frequency and slower data acquisition. The analysis of the 3D point cloud generated from the secondary scanner data based on the obtained SLAM trajectories shows good agreement with the TLS reference. The best point cloud generated showed an accuracy of 13 mm mean and 30 mm maximum error for horizontal position. The dimensions of the building show under 0.1% error.

The wide variability on the trajectory errors revealed by different parameter and test run combinations highlighted the need to be diligent when benchmarking SLAM algorithms. A large variety of latent variables and processes work together and affect the results from SLAM algorithms in seemingly random ways to making the process in effect to a stochastic one. Without a sample size large enough the results obtained are more or less useless for creating any general conclusions. For example, if only one test run with many parameter sets or one set of parameters with many test runs were used, the conclusions on performance of different SLAM algorithms, parameter choices or data collection practises might have been totally different. One example from the current work is the usefulness of using IMU to compensate for scanner rotation during movement. It was first evaluated using only some of the test runs with H24 parameters. In those runs the compensation scheme seemed to provide a noticeable improvement to Hector trajectories requiring only relatively simple modifications to the algorithm. Actually this improvement didn't materialize with the majority of other parameter combinations.

While the point cloud quality with 4.7 mm RMSE achievable by TLS scanning might not be reached, the centimetre range localization accuracy provided by the combination of Karto and Hector SLAM is enough to produce coherent and visually pleasing point clouds with vastly faster data collection. The layout of the building mapped with long dead-end corridors and many glass surfaces in crucial places is also especially difficult for SLAM. Tests in other buildings might compare more favourably with TLS.

Strict global accuracy of the model in relation to reality is important for cases such as building quality monitoring, but for many other uses small errors in the

position and orientations of walls and other structures can be tolerated as long as they are locally consistent and there are no ghosting objects. If used, for example, to generate a virtual world, the direction of a dead end corridor being off by even a few degrees is almost impossible to notice without extensive and close inspection. It is sufficient if the resulting 3D point cloud "looks right" for a human eye.

These results are also only preliminary and can be expected to be improved by further modifications to the SLAM algorithms and by integration of the IMU measurements more robustly to compensate for movement during scan and to constraint scan matching. Processing the scans in inverted order could also be attempted, by merging the result with results of normal processing some improvements could be made.

Other SLAM algorithms could also be tested, for example, Stoyanov et al. [32] have created SLAM algorithm based on Normal Distributions Transform which is available also as a ROS package. Unfortunately their approach does not contain explicit loop closure handling and they recognized it as a problem. Without loop closure handling mechanism it likely can not match the excellent performance of Karto in larger environments. Other SLAM algorithms with loop closure handling mechanism and optimizing backend should be searched for.

The huge impact of movement speed on trajectory accuracy also points to the importance of data collection practises; improvements in that area could further increase the accuracy. For example would slower movement provide even higher accuracy, could moving back and forth between the starting areas of a corridors produce a more robust estimate of their relative orientation, or during movement in long corridors, could orienting the platform in a way which would enable the laser scanner to view the both ends and one full wall continuously lead to improved results. Mounting a second, cheaper laser scanner horizontally should also be tried to assess, how much of the good performance is due to the expensive laser scanner and what is the contribution of the SLAM algorithms.

The SPAN system can occasionally (when passing a window) acquire coordinates from satellites and these could be used in the future to tie the measurements to a global coordinate system. This could also be realized by pushing the platform outdoors when possible.

It is also possible to add synchronous RGB, infrared or hyperspectral cameras to the platform using the modular sensor suite of Slammer. One future option to be explored is to set the data collection laser scanner to TLS mode where it rotates continually around its mount. This would reduce occlusions even further but as a down side the vehicle would block part of the measurements and the effective measurement rate would be reduced. The use of a wheeled cart also limits the use to areas with flat floors. A backpack based measurement platform would provide improved mobility and is one direction for future research. In backpack the platform would not be any longer tied to the floor and the third dimension would had to be taken into account, which would greatly complicate the SLAM problem.

References

[1] Moiz Chasmai, Arun Barde, Gagandeep Purohit, and Anjaneya Sharma. Accuracy enhancement techniques for global navigation satellite systems and its military ground based navigation applications. *International Journal*, 3(1), 2014.

[2] Gérard Lachapelle. Gnss indoor location technologies. *Positioning*, 1(08), 2004.

[3] Rainer Mautz. Overview of current indoor positioning systems. *Geodezija ir kartografija*, 35(1):18–22, 2009.

[4] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 305–316. ACM, 2012.

[5] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[6] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.

[7] Stefan Kohlbrecher, Johannes Meyer, Oskar von Stryk, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[8] Daniel Lau. The Science Behind Kinects or Kinect 1.0 versus 2.0, 2013. http://www.gamasutra.com/blogs/DanielLau/20131127/205820/The_Science_Behind_Kinects_or_Kinect_10_versus_20.php. Accessed 16.5.2015.

[9] Velodyne. Velodyne HDL-64E, 2014. http://www.velodynelidar.com/lidar/hdlproducts/hdl64e.aspx. Accessed 16.5.2015.

[10] George. Vosselman and Hans-Gerd Maas. *Airborne and Terrestrial Laser Scanning*. Whittles Publishing, 2010. ISBN 9781904445876.

[11] Martial Hebert and Eric Krotkov. 3-d measurements from imaging laser radars: How good are they? In *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, pages 359–364. IEEE, 1991.

[12] AD King. Inertial navigation-forty years of evolution. *GEC review*, 13(3): 140–149, 1998.

[13] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007.

[14] Kurt Konolige, Giorgio Grisetti, Rainer Kummerle, Wolfram Burgard, Benson Limketkai, and Regis Vincent. Efficient sparse pose adjustment for 2d mapping. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 22–29. IEEE, 2010.

[15] Joao Machado Santos, David Portugal, and Rui P Rocha. An evaluation of 2d slam techniques available in robot operating system. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–6. IEEE, 2013.

[16] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[17] Kevin Murphy. Bayesian map learning in dynamic environments.

[18] SRI International. Karto, 2015. http://www.kartorobotics.com/. Accessed 16.5.2015.

[19] Edwin Olson. Real-time correlative scan matching. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4387–4393. IEEE, 2009.

[20] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31–43, 2010.

[21] Regis Vincent, Benson Limketkai, and Michael Eriksen. Comparison of indoor robot localization techniques in the absence of gps. In *SPIE Defense, Security, and Sensing*, pages 76641Z–76641Z. International Society for Optics and Photonics, 2010.

[22] Niko Sunderhauf and Peter Protzel. Switchable constraints for robust pose graph slam. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1879–1884. IEEE, 2012.

[23] Xinlian Liang, Juha Hyyppa, Antero Kukko, Harri Kaartinen, Anttoni Jaakkola, and Xiaowei Yu. The use of a mobile laser scanning system for mapping large forest plots. *Geoscience and Remote Sensing Letters, IEEE*, 11 (9):1504–1508, 2014.

[24] Xinlian Liang, Juha Hyyppa, Antero Kukko, Harri Kaartinen, Anttoni Jaakkola, and Xiaowei Yu. The use of a mobile laser scanning system for mapping large forest plots. *Geoscience and Remote Sensing Letters, IEEE*, 11 (9):1504–1508, 2014.

[25] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[26] Tully Foote. tf: The transform library. In *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, Open-Source Software workshop, pages 1–6, April 2013. doi: 10.1109/TePRA.2013.6556373.

[27] ROS. Robotic Operating System, 2015. http://www.ros.org. Accessed 16.5.2015.

[28] Burcin Becerik-Gerber, Farrokh Jazizadeh, Geoffrey Kavulya, and Gulben Calis. Assessment of target types and layouts in 3d laser scanning for registration accuracy. *Automation in Construction*, 20(5):649–658, 2011.

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

[30] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[31] Andrea Bonarini, Wolfram Burgard, Giulio Fontana, Matteo Matteucci, Domenico Giorgio Sorrenti, and Juan Domingo Tardos. Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets. In *In proceedings of IROS'06 Workshop on Benchmarks in Robotics Research*, 2006.

[32] Todor Stoyanov, Jari Saarinen, Henrik Andreasson, and Achim Lilienthal. Normal distributions transform occupancy map fusion: Simultaneous mapping and tracking in large scale dynamic environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4702–4708. IEEE, 2013.

A Hector errors

Table A1: ATE errors for the test runs with Hector SLAM. The first two characters mark the test run, H stands for Hector SLAM and the last three digits code for parameter values used. The first one states the occupancy grid resolution in centimetres, the second one tells the number of levels in the occupancy grid pyramid and the last one tells whether rotation compensation was used (1) or not (0).

	Position error			Heading error		
Test	RMSE	SD	Max	RMSE	SD	Max
	mm	mm	mm	deg	deg	deg
S1H150	135.6	97.0	398.8	0.43	0.22	2.71
S1H151	167.9	118.4	481.4	0.62	0.38	8.29
S1H140	145.5	107.3	436.4	0.47	0.24	2.68
S1H141	155.5	114.1	479.0	0.52	0.29	5.99
S1H240	146.8	102.1	410.1	0.48	0.24	2.72
S1H241	135.1	96.4	387.8	0.43	0.25	7.78
S1H230	151.3	99.1	393.5	0.64	0.38	2.72
S1H231	137.3	94.7	393.8	0.50	0.29	5.28
S1H430	82.1	40.0	308.4	0.27	0.17	2.56
S1H431	233.1	124.6	743.5	0.54	0.36	7.74
S1H420	68.1	30.0	148.6	0.27	0.17	2.51
S1H421	71.6	31.8	153.8	0.26	0.17	5.22
S2H150	106.3	62.5	285.6	0.49	0.27	1.60
S2H151	109.3	64.2	247.2	0.60	0.38	6.00
S2H140	104.6	61.0	279.5	0.48	0.27	1.60
S2H141	98.3	53.9	238.9	0.47	0.28	4.85
S2H240	89.9	46.7	227.1	0.38	0.22	1.58
S2H241	71.0	34.6	154.8	0.33	0.21	6.16
S2H230	90.5	47.0	224.1	0.38	0.21	1.58
S2H231	68.8	33.3	147.7	0.32	0.20	4.63
S2H430	47.6	18.1	108.4	0.21	0.14	1.36
S2H431	50.6	19.1	143.7	0.16	0.12	6.85
S2H420	46.5	17.0	97.3	0.22	0.14	1.44
S2H421	55.0	23.5	129.4	0.16	0.11	5.54
S3H150	171.6	129.9	467.7	0.46	0.23	1.69
S3H151	176.1	134.2	491.8	0.45	0.25	7.45
S3H140	155.7	119.9	442.0	0.42	0.23	1.69
S3H141	145.3	105.1	400.0	0.41	0.21	6.49
S3H240	117.2	85.7	332.2	0.33	0.18	1.61
S3H241	140.8	105.1	408.2	0.36	0.21	7.83

Table A2: ATE errors for the test runs with Hector SLAM. The first two characters mark the test run, H stands for Hector SLAM and the last three digits code for parameter values used. The first one states the occupancy grid resolution in centimetres, the second one tells the number of levels in the occupancy grid pyramid and the last one tells whether rotation compensation was used (1) or not (0).

	Position error			Heading error		
Test	RMSE	SD	Max	RMSE	SD	Max
	mm	mm	mm	deg	deg	deg
S3H230	114.6	83.3	318.2	0.33	0.19	1.63
S3H231	376.7	187.7	781.7	0.96	0.41	5.55
S3H430	146.6	103.6	500.3	0.34	0.19	1.75
S3H431	84.6	56.8	324.9	0.21	0.14	9.00
S3H420	2341.4	1315.3	3203.3	0.92	0.39	1.83
S3H421	1330.9	765.6	1904.9	0.69	0.37	2.91
X1H150	47.1	37.3	198.7	0.06	0.05	2.04
X1H151	54.5	42.9	197.8	0.09	0.05	1.07
X1H140	31.0	22.2	87.3	0.08	0.05	2.05
X1H141	34.2	22.6	90.8	0.09	0.05	1.06
X1H240	65.2	46.7	244.7	0.15	0.09	2.12
X1H241	67.2	46.7	211.3	0.15	0.09	1.22
X1H230	56.4	36.2	147.3	0.16	0.10	2.11
X1H231	40.1	24.8	108.8	0.13	0.09	1.12
X1H430	78.7	52.9	221.1	0.24	0.13	1.81
X1H431	96.3	69.6	248.3	0.42	0.32	2.29
X1H420	53.6	30.8	134.7	0.23	0.14	1.86
X1H421	90.8	63.7	248.7	0.49	0.37	1.46
X2H150	23.5	16.2	88.8	0.10	0.07	1.08
X2H151	25.5	17.7	132.6	0.13	0.09	1.91
X2H140	26.7	19.5	89.5	0.11	0.08	1.12
X2H141	25.8	17.8	85.0	0.13	0.09	1.88
X2H240	36.8	28.8	122.4	0.14	0.09	1.18
X2H241	24.7	17.5	81.5	0.10	0.07	1.62
X2H230	29.8	22.4	99.7	0.13	0.08	1.14
X2H231	36.2	28.6	121.5	0.13	0.08	1.62
X2H430	59.4	42.3	172.4	0.23	0.13	1.21
X2H431	67.9	49.1	188.6	0.22	0.12	1.58
X2H420	77.3	58.8	229.2	0.25	0.13	1.33
X2H421	44.0	26.8	107.0	0.19	0.12	1.57

B GMapping errors

Table A3: ATE errors for the test runs averaged from three distinct GMapping runs. The first two characters mark the test run, HS stands for smoothed Hector trajectory which is used as an initial guess while G stands for GMapping. The last three digits code for parameter values used. The first one states the map resolution, the second one represents the linear and angular optimization step size and the last one marks the sigma of scan end point matcher. The actual parameters are one hundredth of these numbers.

	Position error			Heading error		
Test	RMSE	SD	Max	RMSE	SD	Max
	mm	mm	mm	deg	deg	deg
S1HS241	148.4	79.5	356.8	0.43	0.11	7.78
S1G121	45.5	21.7	96.7	0.19	0.13	2.34
S1G141	34.6	15.4	73.2	0.19	0.13	2.30
S1G125	32.4	15.2	72.2	0.19	0.13	2.33
S1G145	45.2	18.1	90.2	0.20	0.13	2.40
S1G321	115.0	63.3	299.1	0.34	0.19	2.54
S1G341	126.1	54.5	263.0	0.35	0.18	2.57
S1G325	253.9	85.6	425.3	0.43	0.25	2.48
S1G345	223.0	85.4	415.6	0.46	0.25	2.64
S1G521	132.7	77.8	339.6	0.46	0.22	2.67
S1G541	123.1	73.8	317.7	0.42	0.22	2.60
S1G525	316.9	113.6	625.9	0.62	0.32	2.71
S1G545	329.4	135.4	691.6	0.74	0.37	2.82
S2HS241	109.9	57.0	241.1	0.34	0.11	6.15
S2G121	36.1	17.1	90.8	0.24	0.16	1.66
S2G141	35.4	18.5	94.3	0.23	0.15	1.69
S2G125	44.3	19.7	108.1	0.24	0.16	1.75
S2G145	41.0	18.3	93.8	0.24	0.16	1.62
S2G321	102.2	47.9	242.0	0.37	0.22	1.90
S2G341	97.2	40.2	216.8	0.36	0.21	1.82
S2G325	175.6	54.3	309.2	0.47	0.33	3.41
S2G345	194.0	77.8	427.5	0.53	0.30	2.47
S2G521	77.7	35.6	187.4	0.36	0.21	1.79
S2G541	78.7	39.2	197.5	0.37	0.21	1.80
S2G525	203.6	79.9	474.8	0.57	0.35	3.60
S2G545	225.9	95.7	522.6	0.56	0.30	2.50

Table A4: ATE errors for the test runs averaged from three distinct GMapping runs for S3 and two for X1 and X2. The first two characters mark the test run, HS stands for smoothed Hector trajectory which is used as an initial guess while G stands for GMapping. The last three digits code for parameter values used. The first one states the map resolution, the second one represents the linear and angular optimization step size and the last one marks the sigma of scan end point matcher. The actual parameters are one hundredth of these numbers.

	Position error			Heading error		
Test	RMSE	SD	Max	RMSE	SD	Max
	mm	mm	mm	deg	deg	deg
S3HS241	148.4	79.5	356.8	0.43	0.11	7.78
S3G121	64.0	47.0	194.2	0.28	0.18	1.47
S3G141	25.8	13.3	80.1	0.21	0.14	1.23
S3G125	40.0	26.1	124.9	0.24	0.16	1.33
S3G145	43.0	28.5	127.9	0.24	0.16	1.32
S3G321	128.4	96.2	337.9	0.36	0.21	1.59
S3G341	151.3	109.3	397.8	0.43	0.24	1.69
S3G325	146.1	81.2	345.2	0.35	0.21	1.58
S3G345	174.5	91.5	358.4	0.34	0.21	1.77
S3G521	139.8	107.5	393.0	0.41	0.25	1.75
S3G541	138.0	105.7	387.0	0.40	0.24	1.74
S3G525	264.3	172.7	655.6	0.62	0.35	2.10
S3G545	256.7	178.9	689.0	0.64	0.38	2.12
X1HS241	44.3	24.3	113.7	0.15	0.04	1.21
X1G121	54.2	29.2	527.1	0.15	0.11	2.02
X1G141	54.4	28.8	452.2	0.16	0.11	2.06
X1G125	62.4	30.7	561.1	0.16	0.11	2.03
X1G145	62.8	33.6	573.6	0.14	0.10	2.12
X1G321	45.6	25.5	284.7	0.15	0.10	2.14
X1G341	54.0	32.4	243.9	0.17	0.11	2.17
X1G325	183.6	92.2	867.7	0.61	0.47	2.78
X1G345	215.2	117.7	1827.9	0.40	0.29	2.57
X1G521	37.2	18.6	90.3	0.17	0.12	2.11
X1G541	33.8	19.0	88.8	0.16	0.12	2.13
X1G525	143.5	99.5	489.2	0.26	0.17	2.22
X1G545	149.5	92.8	442.4	0.31	0.20	2.25
X2HS241	25.9	13.3	64.0	0.10	0.04	1.62
X2G121	51.4	36.5	199.8	0.19	0.13	1.69
X2G141	42.4	28.5	190.1	0.17	0.12	1.77
X2G125	46.3	31.8	231.7	0.18	0.13	1.82
X2G145	45.7	30.7	257.1	0.18	0.13	1.75
X2G321	24.7	16.4	126.3	0.18	0.14	1.68
X2G341	25.7	15.3	348.0	0.18	0.14	1.69
X2G325	125.0	75.5	544.8	0.32	0.26	3.02
X2G345	120.4	67.8	623.3	0.34	0.27	2.85
X2G521	18.5	12.4	69.9	0.14	0.11	1.71
X2G541	16.6	10.3	68.8	0.14	0.11	1.73
X2G525	95.4	53.3	319.3	0.30	0.24	2.93
X2G545	104.2	53.6	492.3	0.33	0.26	2.58

C Karto errors

Table A5: ATE errors and the amount of loop closures for the test runs with Karto. The first two characters mark the test run, HS stands for smoothed Hector SLAM (the initial trajectory used for the Karto trajectories below it) and K stands for Karto. The next four characters code for parameter values used. The fourth character tells whether original (O) or delayed (D)loop closing method was used, the fifth character denotes the coarse loop closure minimum response values (the actual value is 1/10th and fine response minimum is 0.1 smaller than the corresponding coarse threshold). The sixth charactes tells the scan match resolution, 5 for 5mm and 2 for 2.5mm and the next one marks the scan matching penalty multiplier for matches differing from the odometry estimate. The last number tells whether the initial trajectory had rotation compensation (1) or not (0).

	Position error			Heading error			Loops
Test	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
S1HS241	148.4	79.5	356.8	0.43	0.11	7.78	0
S1KO6211	49.5	36.3	135.8	0.25	0.12	2.47	10
S1KO6511	30.2	18.3	89.5	0.21	0.12	2.45	15
S1KO7211	79.3	66.2	225.7	0.23	0.11	2.34	7
S1KO7511	80.3	52.4	181.0	0.31	0.14	2.69	8
S1KO8211	31.5	17.7	83.9	0.20	0.11	2.43	5
S1KO8511	55.1	31.9	136.1	0.20	0.12	2.54	4
S1KO6231	63.3	44.5	179.6	0.30	0.14	2.46	8
S1KO6531	48.3	35.4	156.4	0.23	0.12	2.51	16
S1KO7231	71.0	52.6	204.5	0.30	0.14	2.51	7
S1KO7531	49.2	33.6	155.9	0.26	0.13	2.48	11
S1KO8231	16.9	8.7	50.9	0.19	0.11	2.31	5
S1KO8531	154.8	115.5	427.4	0.37	0.14	2.22	3
S1KO6261	111.9	86.8	345.2	0.37	0.14	2.46	11
S1KO6561	255.1	135.0	584.7	1.28	0.17	4.79	19
S1KO7261	33.9	23.2	119.2	0.22	0.12	2.39	6
S1KO7561	269.3	147.1	634.2	1.40	0.17	5.09	14
S1KO8261	37.8	19.3	89.1	0.25	0.13	2.35	5
S1KO8561	254.5	135.1	582.8	1.28	0.17	4.79	10
S1KO6291	47.9	31.7	142.0	0.27	0.14	2.44	10
S1KO6591	36.7	22.5	90.9	0.19	0.11	2.36	14
S1KO7291	18.8	9.7	59.8	0.23	0.12	2.38	6
S1KO7591	25.8	15.6	70.0	0.25	0.12	2.44	7
S1KO8291	40.9	26.4	119.6	0.23	0.12	2.52	3
S1KO8591	31.5	21.4	87.8	0.23	0.13	2.52	3

Table A6: ATE errors and the number of loop closures for the test runs with Karto. Full parameter explanation in Table A5 caption.

Test	Position error			Heading error			Loops
	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
S1KD6211	52.6	40.4	144.6	0.24	0.12	2.36	11
S1KD6511	31.0	15.8	86.5	0.22	0.12	2.45	14
S1KD7211	60.5	40.5	157.1	0.26	0.13	2.50	6
S1KD7511	56.9	37.7	144.4	0.24	0.13	2.58	8
S1KD8211	63.0	37.2	139.4	0.27	0.13	2.61	4
S1KD8511	42.7	24.0	107.1	0.22	0.12	2.43	4
S1KD6231	59.0	41.7	180.3	0.28	0.13	2.41	10
S1KD6531	31.3	19.5	102.6	0.22	0.12	2.39	14
S1KD7231	166.2	108.2	435.1	0.57	0.15	2.67	7
S1KD7531	59.1	44.9	187.1	0.25	0.13	2.45	10
S1KD8231	79.4	53.3	212.3	0.31	0.14	2.65	5
S1KD8531	154.8	115.5	427.4	0.37	0.14	2.22	3
S1KD6261	46.9	31.7	147.5	0.29	0.14	2.48	11
S1KD6561	254.3	134.6	582.0	1.28	0.17	4.79	20
S1KD7261	28.6	21.1	103.0	0.24	0.12	2.38	6
S1KD7561	254.3	134.6	580.1	1.28	0.17	4.79	13
S1KD8261	59.8	38.5	131.5	0.30	0.13	2.58	4
S1KD8561	254.5	135.1	582.8	1.28	0.17	4.79	10
S1KD6291	27.6	15.1	75.5	0.22	0.12	2.40	10
S1KD6591	81.7	56.0	194.0	0.29	0.14	2.56	14
S1KD7291	56.7	36.0	152.1	0.26	0.13	2.51	7
S1KD7591	21.3	14.0	68.5	0.18	0.11	2.39	7
S1KD8291	45.1	30.3	139.1	0.24	0.12	2.46	6
S1KD8591	20.0	12.0	59.5	0.21	0.12	2.44	4
S2HS241	109.9	57.0	241.1	0.34	0.11	6.15	0
S2KO6211	78.2	46.1	161.7	0.32	0.16	1.83	19
S2KO6511	123.6	77.5	256.7	0.38	0.18	1.88	13
S2KO7211	59.4	45.0	198.9	0.28	0.14	1.73	15
S2KO7511	103.3	65.4	225.5	0.35	0.17	1.84	11
S2KO8211	26.9	16.2	70.3	0.28	0.15	1.62	7
S2KO8511	103.4	62.3	213.0	0.35	0.18	1.83	11
S2KO6231	31.8	18.5	86.2	0.27	0.15	1.72	22
S2KO6531	62.8	35.5	144.3	0.29	0.16	1.79	19
S2KO7231	36.7	21.8	95.0	0.28	0.16	1.68	19
S2KO7531	142.8	86.7	283.8	0.40	0.18	1.98	14
S2KO8231	43.5	25.6	102.5	0.30	0.16	1.76	11
S2KO8531	88.7	53.1	183.9	0.32	0.17	1.82	10
S2KO6261	65.4	52.0	224.5	0.31	0.15	1.72	20
S2KO6561	46.4	26.2	111.2	0.25	0.15	1.77	17
S2KO7261	47.4	33.0	129.5	0.28	0.15	1.70	16
S2KO7561	90.7	57.2	201.3	0.32	0.17	1.92	14
S2KO8261	88.5	62.4	227.8	0.36	0.17	2.00	9
S2KO8561	81.7	46.0	165.1	0.33	0.17	1.87	10

Table A7: ATE errors and the number of loop closures for test runs with Karto. See Table A5 caption for full parameter explanation.

	Position error			Heading error			Loops
Test	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
S2KO6291	71.6	54.8	252.0	0.33	0.16	1.73	22
S2KO6591	58.9	36.4	126.3	0.28	0.16	1.76	17
S2KO7291	57.3	35.8	136.0	0.30	0.16	1.73	15
S2KO7591	89.9	53.7	185.6	0.32	0.17	1.94	12
S2KO8291	43.1	28.8	99.4	0.30	0.16	1.74	10
S2KO8591	100.8	59.6	211.3	0.35	0.18	1.91	9
S2KD6211	43.6	24.5	104.3	0.29	0.16	1.84	19
S2KD6511	29.8	18.3	79.1	0.24	0.14	1.71	18
S2KD7211	117.5	103.5	428.0	0.41	0.15	1.68	15
S2KD7511	83.4	50.1	163.9	0.34	0.17	1.87	13
S2KD8211	164.3	143.6	605.9	0.53	0.17	1.61	8
S2KD8511	60.4	39.9	140.7	0.27	0.15	1.71	10
S2KD6231	31.8	18.5	86.2	0.27	0.15	1.72	22
S2KD6531	62.8	35.5	144.3	0.29	0.16	1.79	19
S2KD7231	36.7	21.8	95.0	0.28	0.16	1.68	19
S2KD7531	142.8	86.7	283.8	0.40	0.18	1.98	14
S2KD8231	43.5	25.6	102.5	0.30	0.16	1.76	11
S2KD8531	88.7	53.1	183.9	0.32	0.17	1.82	10
S2KD6261	41.6	29.8	144.1	0.29	0.16	1.66	18
S2KD6561	46.4	26.2	111.2	0.25	0.15	1.77	17
S2KD7261	47.4	33.0	129.5	0.28	0.15	1.70	16
S2KD7561	90.7	57.2	201.3	0.32	0.17	1.92	14
S2KD8261	88.5	62.4	227.8	0.36	0.17	2.00	9
S2KD8561	81.7	46.0	165.1	0.33	0.17	1.87	10
S2KD6291	71.6	54.8	252.0	0.33	0.16	1.73	22
S2KD6591	58.9	36.4	126.3	0.28	0.16	1.76	17
S2KD7291	57.3	35.8	136.0	0.30	0.16	1.73	15
S2KD7591	89.9	53.7	185.6	0.32	0.17	1.94	12
S2KD8291	43.1	28.8	99.4	0.30	0.16	1.74	10
S2KD8591	100.8	59.6	211.3	0.35	0.18	1.91	9
S2HS240	77.5	37.9	190.5	0.38	0.15	1.58	0
S2KD6210	65.5	48.2	204.4	0.31	0.16	1.66	20
S2KD6510	46.2	36.9	160.5	0.27	0.15	1.59	22
S2KD7210	68.1	53.9	284.4	0.36	0.17	1.64	17
S2KD7510	43.3	32.8	172.7	0.29	0.15	1.68	14
S2KD8210	17.6	9.9	53.4	0.28	0.16	1.66	10
S2KD8510	34.8	19.8	91.0	0.27	0.15	1.64	12
S2KD6230	30.9	20.7	72.9	0.26	0.15	1.51	21
S2KD6530	56.8	38.0	175.6	0.30	0.16	1.65	20
S2KD7230	103.3	90.2	373.1	0.38	0.15	1.55	14
S2KD7530	49.8	37.3	134.8	0.28	0.15	1.60	19
S2KD8230	28.5	18.6	62.6	0.28	0.15	1.69	8
S2KD8530	3650.6	2834.4	13814.5	9.23	0.98	17.80	3

Table A8: ATE errors and the number of loop closures for test runs with Karto. See Table A5 caption for full parameter explanation.

Test	Position error			Heading error			Loops
	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
S2KD6260	23.5	13.5	63.3	0.25	0.15	1.53	22
S2KD6560	34.0	22.3	88.9	0.25	0.14	1.55	20
S2KD7260	138.1	118.1	498.9	0.46	0.16	1.58	17
S2KD7560	21.9	12.4	71.3	0.25	0.15	1.58	20
S2KD8260	52.6	40.5	211.1	0.34	0.17	1.58	9
S2KD8560	183.6	120.6	437.4	0.53	0.20	1.83	9
S2KD6290	31.3	22.9	98.8	0.25	0.14	1.47	22
S2KD6590	46.3	29.1	122.0	0.26	0.15	1.60	20
S2KD7290	37.3	24.4	86.4	0.26	0.15	1.57	18
S2KD7590	44.8	28.3	107.5	0.27	0.15	1.65	21
S2KD8290	98.7	78.6	287.1	0.39	0.17	1.68	9
S2KD8590	95.9	58.9	185.5	0.35	0.17	1.71	11
S3HS241	148.4	79.5	356.8	0.43	0.11	7.78	0
S3KO6211	49.5	36.3	135.8	0.25	0.12	2.47	10
S3KO6511	30.2	18.3	89.5	0.21	0.12	2.45	15
S3KO7211	79.3	66.2	225.7	0.23	0.11	2.34	7
S3KO7511	80.3	52.4	181.0	0.31	0.14	2.69	8
S3KO8211	31.5	17.7	83.9	0.20	0.11	2.43	5
S3KO8511	55.1	31.9	136.1	0.20	0.12	2.54	4
S3KO6231	63.3	44.5	179.6	0.30	0.14	2.46	8
S3KO6531	48.3	35.4	156.4	0.23	0.12	2.51	16
S3KO7231	71.0	52.6	204.5	0.30	0.14	2.51	7
S3KO7531	49.2	33.6	155.9	0.26	0.13	2.48	11
S3KO8231	16.9	8.7	50.9	0.19	0.11	2.31	5
S3KO8531	154.8	115.5	427.4	0.37	0.14	2.22	3
S3KO6261	111.9	86.8	345.2	0.37	0.14	2.46	11
S3KO6561	255.1	135.0	584.7	1.28	0.17	4.79	19
S3KO7261	33.9	23.2	119.2	0.22	0.12	2.39	6
S3KO7561	269.3	147.1	634.2	1.40	0.17	5.09	14
S3KO8261	37.8	19.3	89.1	0.25	0.13	2.35	5
S3KO8561	254.5	135.1	582.8	1.28	0.17	4.79	10
S3KO6291	47.9	31.7	142.0	0.27	0.14	2.44	10
S3KO6591	36.7	22.5	90.9	0.19	0.11	2.36	14
S3KO7291	18.8	9.7	59.8	0.23	0.12	2.38	6
S3KO7591	25.8	15.6	70.0	0.25	0.12	2.44	7
S3KO8291	40.9	26.4	119.6	0.23	0.12	2.52	3
S3KO8591	31.5	21.4	87.8	0.23	0.13	2.52	3
S3KD6211	52.6	40.4	144.6	0.24	0.12	2.36	11
S3KD6511	31.0	15.8	86.5	0.22	0.12	2.45	14
S3KD7211	60.5	40.5	157.1	0.26	0.13	2.50	6
S3KD7511	56.9	37.7	144.4	0.24	0.13	2.58	8
S3KD8211	63.0	37.2	139.4	0.27	0.13	2.61	4
S3KD8511	42.7	24.0	107.1	0.22	0.12	2.43	4

Table A9: ATE errors and the number of loop closures for test runs with Karto. See Table A5 caption for full parameter explanation.

Test	Position error			Heading error			Loops
	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
S3KD6231	59.0	41.7	180.3	0.28	0.13	2.41	10
S3KD6531	31.3	19.5	102.6	0.22	0.12	2.39	14
S3KD7231	166.2	108.2	435.1	0.57	0.15	2.67	7
S3KD7531	59.1	44.9	187.1	0.25	0.13	2.45	10
S3KD8231	79.4	53.3	212.3	0.31	0.14	2.65	5
S3KD8531	154.8	115.5	427.4	0.37	0.14	2.22	3
S3KD6261	46.9	31.7	147.5	0.29	0.14	2.48	11
S3KD6561	254.3	134.6	582.0	1.28	0.17	4.79	20
S3KD7261	28.6	21.1	103.0	0.24	0.12	2.38	6
S3KD7561	254.3	134.6	580.1	1.28	0.17	4.79	13
S3KD8261	59.8	38.5	131.5	0.30	0.13	2.58	4
S3KD8561	254.5	135.1	582.8	1.28	0.17	4.79	10
S3KD6291	27.6	15.1	75.5	0.22	0.12	2.40	10
S3KD6591	81.7	56.0	194.0	0.29	0.14	2.56	14
S3KD7291	56.7	36.0	152.1	0.26	0.13	2.51	7
S3KD7591	21.3	14.0	68.5	0.18	0.11	2.39	7
S3KD8291	45.1	30.3	139.1	0.24	0.12	2.46	6
S3KD8591	20.0	12.0	59.5	0.21	0.12	2.44	4
X1HS241	44.3	24.3	113.7	0.15	0.04	1.21	0
X1KO6211	69.5	60.2	241.8	0.23	0.07	2.03	18
X1KO6511	25.1	12.7	73.3	0.12	0.06	2.06	15
X1KO7211	18.3	11.6	65.2	0.10	0.06	2.06	13
X1KO7511	14.8	8.3	49.1	0.11	0.06	2.11	14
X1KO8211	35.0	28.0	127.7	0.15	0.06	2.13	10
X1KO8511	16.1	7.4	58.4	0.11	0.06	2.00	9
X1KO6231	16.7	10.2	56.0	0.11	0.06	2.18	20
X1KO6531	27.9	17.2	85.7	0.12	0.06	2.14	16
X1KO7231	44.8	36.3	163.7	0.18	0.06	2.08	13
X1KO7531	15.1	8.2	60.6	0.11	0.06	2.09	15
X1KO8231	64.5	56.1	226.0	0.22	0.06	2.06	9
X1KO8531	25.2	13.0	74.1	0.12	0.06	2.21	9
X1KO6261	62.9	53.9	211.7	0.21	0.06	2.10	19
X1KO6561	19.0	8.4	59.7	0.12	0.06	2.06	16
X1KO7261	66.0	55.4	236.5	0.24	0.07	2.13	13
X1KO7561	26.1	18.7	79.0	0.12	0.06	2.07	14
X1KO8261	43.2	35.1	151.5	0.19	0.07	2.13	10
X1KO8561	34.0	21.4	125.6	0.13	0.06	2.06	9
X1KO6291	26.0	18.0	93.0	0.15	0.07	2.13	18
X1KO6591	19.5	8.8	59.7	0.12	0.06	2.09	15
X1KO7291	60.5	52.1	217.8	0.22	0.06	2.09	13
X1KO7591	18.2	12.0	56.0	0.11	0.06	2.11	15
X1KO8291	27.7	19.7	100.0	0.14	0.06	2.13	9
X1KO8591	19.7	8.7	62.3	0.11	0.06	2.09	10

Table A10: ATE errors and the number of loop closures for test runs with Karto. See Table A5 caption for full parameter explanation.

Test	Position error			Heading error			Loops
	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
X1KD6211	16.8	12.3	63.1	0.11	0.06	2.09	14
X1KD6511	16.8	9.8	54.9	0.11	0.06	2.11	17
X1KD7211	35.3	27.4	130.9	0.18	0.07	2.12	16
X1KD7511	21.2	11.2	77.9	0.11	0.06	2.04	14
X1KD8211	35.7	20.9	100.2	0.20	0.07	2.17	10
X1KD8511	20.6	7.8	53.6	0.11	0.06	2.12	8
X1KD6231	19.5	12.6	78.0	0.12	0.06	2.12	15
X1KD6531	33.1	19.7	85.5	0.14	0.07	2.21	18
X1KD7231	27.9	22.2	89.9	0.13	0.06	2.09	16
X1KD7531	25.1	17.3	74.7	0.11	0.06	2.13	17
X1KD8231	40.3	30.7	139.9	0.17	0.07	2.20	9
X1KD8531	18.3	9.4	62.0	0.11	0.06	2.10	10
X1KD6261	19.3	12.0	64.1	0.12	0.06	2.09	16
X1KD6561	30.0	15.2	85.5	0.12	0.06	2.21	19
X1KD7261	19.3	13.7	69.3	0.11	0.06	2.09	16
X1KD7561	16.5	9.0	65.4	0.11	0.06	2.06	14
X1KD8261	33.8	25.5	113.4	0.16	0.07	2.19	10
X1KD8561	25.2	14.2	73.6	0.12	0.07	2.15	8
X1KD6291	26.3	16.6	81.9	0.11	0.06	2.09	14
X1KD6591	19.5	8.3	59.6	0.12	0.06	2.09	18
X1KD7291	27.8	21.5	100.4	0.15	0.06	2.13	15
X1KD7591	15.9	9.2	57.1	0.10	0.06	2.03	15
X1KD8291	41.7	31.2	137.1	0.18	0.07	2.13	10
X1KD8591	18.7	10.3	67.7	0.11	0.06	2.12	8
X1HS240	73.9	41.6	161.8	0.15	0.04	2.12	0
X1KD6210	33.6	28.6	113.7	0.15	0.06	2.06	16
X1KD6510	24.6	17.6	109.9	0.11	0.06	1.97	16
X1KD7210	15.8	10.5	66.7	0.12	0.06	2.11	15
X1KD7510	35.2	25.8	136.6	0.12	0.07	2.12	18
X1KD8210	25.5	17.4	76.0	0.15	0.07	2.17	10
X1KD8510	24.9	17.7	108.5	0.11	0.06	2.12	10
X1KD6230	35.1	28.4	122.1	0.16	0.07	2.16	15
X1KD6530	35.8	25.3	157.8	0.11	0.06	2.13	17
X1KD7230	18.1	13.6	63.1	0.12	0.06	2.11	15
X1KD7530	20.8	12.4	58.1	0.12	0.06	2.08	16
X1KD8230	23.2	15.8	69.9	0.12	0.07	2.16	10
X1KD8530	16.6	8.7	57.8	0.11	0.06	2.10	6
X1KD6260	13.6	8.9	52.6	0.12	0.06	2.10	15
X1KD6560	10.7	4.8	44.8	0.11	0.06	2.12	19
X1KD7260	23.9	17.8	83.0	0.13	0.06	2.13	15
X1KD7560	27.5	18.1	73.1	0.12	0.06	2.16	16
X1KD8260	20.5	15.2	72.5	0.14	0.07	2.12	9
X1KD8560	24.5	14.4	65.2	0.11	0.06	2.15	11

Table A11: ATE errors and the number of loop closures for test runs with Karto. See Table A5 caption for full parameter explanation.

Test	Position error			Heading error			Loops
	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
X1KD6290	30.8	24.7	105.1	0.14	0.06	2.15	15
X1KD6590	25.9	17.0	72.7	0.11	0.06	2.12	19
X1KD7290	31.7	26.0	109.2	0.14	0.06	2.08	16
X1KD7590	43.0	32.6	123.3	0.14	0.07	2.17	15
X1KD8290	20.7	13.1	62.1	0.14	0.07	2.12	11
X1KD8590	17.3	8.3	74.7	0.11	0.06	2.12	9
X2HS241	25.9	13.3	64.0	0.10	0.04	1.62	0
X2KO6211	16.7	10.0	55.2	0.12	0.07	1.29	15
X2KO6511	27.5	16.5	66.1	0.14	0.07	1.29	15
X2KO7211	19.3	14.3	72.6	0.13	0.07	1.36	13
X2KO7511	21.5	10.3	50.2	0.13	0.07	1.29	11
X2KO8211	25.2	17.6	60.2	0.14	0.07	1.27	7
X2KO8511	13.6	8.3	50.5	0.13	0.07	1.33	6
X2KO6231	17.2	11.2	65.6	0.12	0.07	1.28	15
X2KO6531	25.5	13.8	67.2	0.14	0.07	1.33	15
X2KO7231	15.0	10.1	44.7	0.12	0.07	1.32	13
X2KO7531	18.2	8.6	40.2	0.14	0.07	1.35	12
X2KO8231	28.1	20.6	88.2	0.13	0.07	1.46	8
X2KO8531	19.6	11.4	64.0	0.16	0.08	1.29	6
X2KO6261	22.0	14.7	71.9	0.14	0.07	1.45	14
X2KO6561	23.8	11.7	48.2	0.13	0.07	1.29	15
X2KO7261	20.0	14.1	64.4	0.13	0.07	1.40	12
X2KO7561	40.4	25.6	115.4	0.19	0.08	1.29	11
X2KO8261	21.3	15.5	54.3	0.12	0.07	1.37	8
X2KO8561	23.5	14.5	71.8	0.15	0.08	1.29	6
X2KO6291	47.9	36.1	143.4	0.17	0.08	1.28	15
X2KO6591	40.2	26.2	114.7	0.18	0.08	1.28	15
X2KO7291	19.0	12.7	76.5	0.13	0.07	1.32	13
X2KO7591	49.3	34.9	159.4	0.22	0.08	1.30	11
X2KO8291	12.1	7.8	37.8	0.12	0.07	1.37	8
X2KO8591	19.0	11.4	59.6	0.15	0.08	1.30	6
X2KD6211	12.3	7.7	35.4	0.12	0.07	1.41	17
X2KD6511	34.5	23.7	87.0	0.15	0.08	1.28	16
X2KD7211	17.0	12.0	60.8	0.13	0.07	1.30	17
X2KD7511	21.3	14.6	58.7	0.14	0.08	1.34	16
X2KD8211	12.8	7.9	51.6	0.12	0.07	1.36	8
X2KD8511	19.3	11.5	58.7	0.14	0.07	1.33	6
X2KD6231	15.2	10.8	48.0	0.12	0.07	1.38	18
X2KD6531	43.0	32.7	113.5	0.15	0.08	1.34	16
X2KD7231	15.1	10.9	50.9	0.12	0.07	1.29	16
X2KD7531	42.9	31.6	138.8	0.19	0.08	1.34	15
X2KD8231	15.4	10.0	54.9	0.13	0.07	1.41	11
X2KD8531	28.1	17.3	65.0	0.14	0.08	1.31	8

Table A12: ATE errors and the number of loop closures for test runs with Karto. See Table A5 caption for full parameter explanation.

Test	Position error			Heading error			Loops
	RMSE	SD	Max	RMSE	SD	Max	
	mm	mm	mm	deg	deg	deg	N
X2KD6261	15.6	11.6	46.1	0.12	0.07	1.35	17
X2KD6561	54.9	44.5	190.5	0.22	0.08	1.34	16
X2KD7261	32.1	25.4	95.0	0.13	0.07	1.36	13
X2KD7561	28.2	18.0	65.1	0.14	0.07	1.33	14
X2KD8261	23.8	18.1	79.5	0.13	0.07	1.34	10
X2KD8561	30.6	21.5	107.4	0.19	0.08	1.27	6
X2KD6291	21.2	16.7	80.7	0.12	0.07	1.36	19
X2KD6591	26.0	16.4	67.2	0.15	0.08	1.24	16
X2KD7291	29.0	20.8	97.5	0.14	0.07	1.31	14
X2KD7591	32.5	22.2	77.3	0.14	0.08	1.31	11
X2KD8291	26.8	19.9	80.8	0.13	0.07	1.33	9
X2KD8591	23.3	15.8	83.2	0.17	0.08	1.27	6
X2HS240	32.8	25.1	113.5	0.14	0.04	1.18	0
X2KD6210	20.0	12.6	58.1	0.13	0.07	1.31	18
X2KD6510	42.3	30.8	98.5	0.16	0.08	1.32	18
X2KD7210	22.2	13.4	61.8	0.12	0.07	1.32	18
X2KD7510	33.0	21.5	73.8	0.15	0.08	1.28	16
X2KD8210	30.6	21.9	71.8	0.13	0.07	1.53	13
X2KD8510	26.0	17.8	73.9	0.16	0.08	1.28	10
X2KD6230	22.3	13.5	57.7	0.14	0.08	1.26	18
X2KD6530	30.4	19.7	68.6	0.14	0.07	1.39	18
X2KD7230	23.9	16.4	79.8	0.13	0.07	1.32	16
X2KD7530	35.2	24.2	82.3	0.14	0.08	1.29	13
X2KD8230	43.8	30.5	118.7	0.14	0.08	1.35	11
X2KD8530	24.8	14.4	55.2	0.14	0.08	1.34	7
X2KD6260	71.6	60.7	261.1	0.28	0.08	1.72	19
X2KD6560	19.8	10.7	54.7	0.14	0.07	1.39	18
X2KD7260	25.4	17.1	74.7	0.13	0.07	1.28	17
X2KD7560	26.3	16.6	60.6	0.14	0.08	1.29	16
X2KD8260	34.8	25.4	94.7	0.14	0.08	1.49	8
X2KD8560	25.1	15.9	59.5	0.14	0.08	1.34	7
X2KD6290	20.6	12.3	62.9	0.13	0.07	1.32	20
X2KD6590	31.9	25.2	109.6	0.16	0.07	1.37	18
X2KD7290	41.3	28.9	113.7	0.14	0.08	1.30	15
X2KD7590	28.7	18.1	66.2	0.14	0.08	1.30	13
X2KD8290	31.8	22.4	88.2	0.13	0.07	1.42	10
X2KD8590	23.6	13.0	55.2	0.14	0.08	1.34	8