HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Laboratory of Automation Technology

Seppo Heikkilä

Development of Laser Based Localisation Methods for Personal Navigation System - PeNa

Master's thesis

Supervisor: Aarne Halme
Instructor: Jari Saarinen

Espoo 9th July 2005

Preface

This master's thesis has been carried out at the Automation Technology Laboratory of Helsinki University of Technology during the year 2004.

First I want to thank professor Aarne Halme, head of the laboratory and the supervisor of my master's thesis, for giving me the opportunity to work on this project.

I also thank the European Community for the project funding. This master's thesis was part of the PeLoTe project, which has been supported within the IST-2001-FET framework under project no. 38873.

Many thanks to my instructor Jari Saarinen for clear and direct answers to my questions and for constructive feedback on various stages of the research process. The successful completion of the PeLoTe project and this master's thesis would not have been possible without you.

I want to thank also all the members of the PeLoTe project for cooperation, especially the HUT team members, Mikko Elomaa and Jussi Suomela.

Finally I want to thank the staff of the Automation Technology Laboratory for the inspiring working atmosphere.

Espoo, 9th July 2005

Seppo Heikkilä

HELSINKI UNIVERSITY OF TECHNOLOGY | **ABSTRACT OF THE MASTER'S THESIS**

Author:	Seppo Heikkilä		
Title of the thesis:	Development of Laser Based Localisation Methods for Personal Navigation System - PeNa		
Date:	9th July 2005	**Number of pages:**	91
Department:	Electrical and Communications Engineering		
Professorship:	Automation Technology	**Code:**	AS-84
Supervisor:	Aarne Halme		
Instructor:	Jari Saarinen		

This master's thesis presents a stand-alone system for human indoor localisation, named Personal Navigation system (PeNa). The PeNa system operates using dead reckoning and map based localisation algorithms. The dead reckoning is calculated using stride length measurements and Kalman filtered heading information, which are further enhanced using laser scan matching. A map based localisation algorithm is introduced to provide localisation against absolute reference frame.

Three laser scan based angle correction algorithms are presented: a histogram correlation with sequential lines, a histogram correlation with segmented lines, and a line intersection based algorithm. Also an evidence grid correlation based location correction algorithm and a virtual scan based map localisation algorithm are presented. Furthermore, a simple Simultaneous Localisation and Mapping (SLAM) algorithm, based on incremental map building, is presented.

The PeNa was developed as part of the Building Presence through Localisation for Hybrid Telematic Systems (PeLoTe) project funded by the IST programme of the European Community. The PeLoTe project is shortly introduced in the beginning of this master's thesis.

Keywords: human indoor localisation, laser scan matching, SLAM.

TEKNILLINEN KORKEAKOULU | **DIPLOMITYÖN TIIVISTELMÄ**

Tekijä:	Seppo Heikkilä	
Työn aihe:	Laserpohjaisten paikannusmenetelmien kehittäminen henkilökohtaiseen paikannuslaitteeseen - PeNa	
Päivämäärä:	9. heinäkuuta 2005	**Sivumäärä:** 91
Osasto:	Sähkö- ja tietoliikennetekniikan osasto	
Professuuri:	Automaatiotekniikka	**Koodi:** AS-84
Työn valvoja:	Aarne Halme	
Työn ohjaaja:	Jari Saarinen	

Tämä diplomityö esittelee itsenäisen ihmisen sisätilapaikannusjärjestelmän nimeltään Personal Navigation system (PeNa). PeNa perustuu merkintälasku ja karttapaikannusalgoritmeihin. Merkintälasku perustuu askelpituuden mittaukseen ja Kalman suodatettuun kulmainformaatioon, joita parannetaan edelleen lasermittaussovituksella. Karttapohjainen paikannusalgoritmi esitellään mahdollistamaan paikannus absoluuttisessa koordinaatistossa.

Työ esittelee kolme lasermittaukseen perustuvaa kulmakorjausalgoritmia: perättäisten viivojen histogrammikorrelaatio, ositettujen viivojen histogrammikorrelaatio ja viivojen leikkauspisteisiin perustuva algoritmi. Lisäksi esitellään korrelaatioon perustuva paikankorjausalgoritmi ja virtuaalilasermittauksiin perustuva paikannusalgoritmi. Lopuksi esitellään yksinkertainen vaiheittaiseen kartanmuodostukseen perustuva SLAM-algoritmi.

PeNa kehitettiin osana PeLoTe projektia, joka on Euroopan Unionin IST ohjelman rahoittama projekti. PeLoTe projekti esitellään lyhyesti diplomityön alussa.

Avainsanat: ihmisen sisätilapaikannus, lasermittaussovitus, SLAM.

Contents

Symbols and Abbreviations

$\mathcal{A}_{k,i}[\beta]$	angle difference function
$\mathcal{B}_{n,m}^{k,i}$	matrix element correlation function
$C[k]$	cross-correlation function
$D[k]$, $E[k]$	histogram function
K_{gain}	Kalman filter gain
P	number of points
P^-	a priori angle covariance
P^+	a posteriori angle covariance
Q	variance of gyro measurement
R	variance of compass measurement
z	compass measurement
ω	angular velocity
$\hat{\chi}^-$	a priori angle value estimate
$\hat{\chi}$	a posteriori angle value estimate

AOA Angle of Arrival

CBML Correlation-Based Markov Localisation

CCA Cross-Correlation Algorithm

CI Cell Identification

CLSF Constrained Local Submap Filter

CML Concurrent Mapping and Localisation

CSM Combined Scan Matcher

DGPS Differential GPS

DLL Dynamic Link Library

DP Distributed Particle

DR Dead Reckoning

DRM Dead Reckoning Module

EGNOS European Geostationary Navigation Overlay Service

EKF Extended Kalman Filter

EM Expectation Maximisation

EOTD Enhanced Observed Time Difference

EPFL Swiss Federal Institute of Technology

EU European Union

FCC Federal Communications Commission

FOC Full Operational Capability

GLONASS Global Orbiting Navigation Satellite System

GLUT OpenGL Utility Toolkit

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

GSM Global System for Mobile Communications

GUI Graphical User Interface

HUT Helsinki University of Technology

HSGPS High Sensitivity GPS

ICP Iterative Closest Point

IDC Iterative Dual Correspondence

IDC-S Iterative Dual Correspondence-Sector

IMU Inertial Measurement Unit

IST Information Society Technologies

JNI Java Native Interface

KF Kalman Filter

LANDMARC Location Identification based on Dynamic Active RFID Calibration

LDR Laser Dead Reckoning

LRF Laser Range Finder

LRGC Local Registration / Global Correlation

MEMS Micro-Electro-Mechanical Systems

MCL Monte Carlo Localisation

ML Markov Localisation

MSAS Multi-Functional Satellite Augmentation System

NAVSTAR NAVigation Satellite Timing and Ranging

NTP Network Time Protocol

PAS Personal Assistance System

PDR Pedestrian Dead Reckoning

PeLoTe Building Presence through Localisation for Hybrid Telematic Systems

PeNa Personal Navigation system

PNM Pedestrian Navigation Module

PNS Pedestrian Navigation System

PTS Pedestrian Tracking System

RF Radio Frequency

RFID RF Identification

RMI Remote Method Invocation

RTK Real-Time Kinematic

SBAS Satellite-Based Augmentation System

SIG Special Interest Group

SiLMU Stride Length Measurement Unit

SLAM Simultaneous Localisation and Mapping

SoapBox Sensing, Operating and Activating Peripheral Box

SSF Space Systems Finland

STL Standard Template Library

TDOA Time Difference of Arrival

TOF Time of Flight

UI User Interface

WAAS Wide Area Augmentation System

WLAN Wireless LAN

ZUPT Zero Velocity Update

Chapter 1

Introduction

Imagine you had a device that could always tell your exact location on a map. On request, it could give you a path to the post office, or show a path through all the shelves that you have to visit in a convenient store based on your shopping list. Who could count the number of frustrated hours saved by this kind of device? In fact, our everyday life is a constant struggle with various localisation problems.

Unlike robots, the humans only rarely know their accurate position. Human beings use mostly eyes for localisation which is based on recognition of an object and estimation of the distance to the object. If the sight is reduced, e.g. by darkness or smoke, the accuracy of the localisation suffers significantly. The localisation is also always relative in nature. Only in situations when human can identify a known object, like corridor crossing or stairway, he can know his accurate position relative to that object. Thus the transfer of the position information to another person or to a computer system can be difficult.

The solution for the problem is not necessarily so far ahead. This master's thesis describes a novel localisation system, named as Personal Navigation system (PeNa), capable to operate reliably in indoor environments without any external infrastructure. The emphasis is on the development of laser

based localisation methods. The Personal Navigation system (PeNa) system is built as a part of Building Presence through Localisation for Hybrid Telematic Systems (PeLoTe) project. The core goal of the PeLoTe project is to increase entities' feeling of presence in a demanding rescue environment through accurate localisation and mapping.

The PeNa system has been introduced earlier in [1, 2, 3, 4]. In [1] is presented the development and analysis of the Stride Length Measurement Unit (SiLMU). The [2] presents analysis of the dead reckoning using a SiLMU unit, an inertial measurement unit, and a laser. The [3] presents laser based methods for localisation and mapping, and [4] presents the description of the integrated PeNa system.

1.1 PeLoTe Project

Building Presence through Localisation for Hybrid Telematic Systems (PeLoTe) project is part of the Information Society Technologies (IST) programme funded by the European Community. The project started in October 2002 and finishes in the end of March 2005. The goal of the PeLoTe project is to design novel methods for building presence through integrating and coordinating autonomous collaborative entities in a telematic application, comprising nonliving systems and human actors. Special focus is put on solving the problem of building presence and integration of autonomous entities of different kinds in a teleoperated task. Also novel methods in telemaintenance and diagnostic system are developed.

The objective of the PeLoTe project effort is to investigate, formalise and develop methods for creating a presence feature for a community of living and nonliving entities as humans and autonomous or tele-operated devices are. The research provides theoretical foundations required for advanced building the presence feature, where integration of the independent entities of utterly different nature, that are either (semi)autonomous machines or humans, both

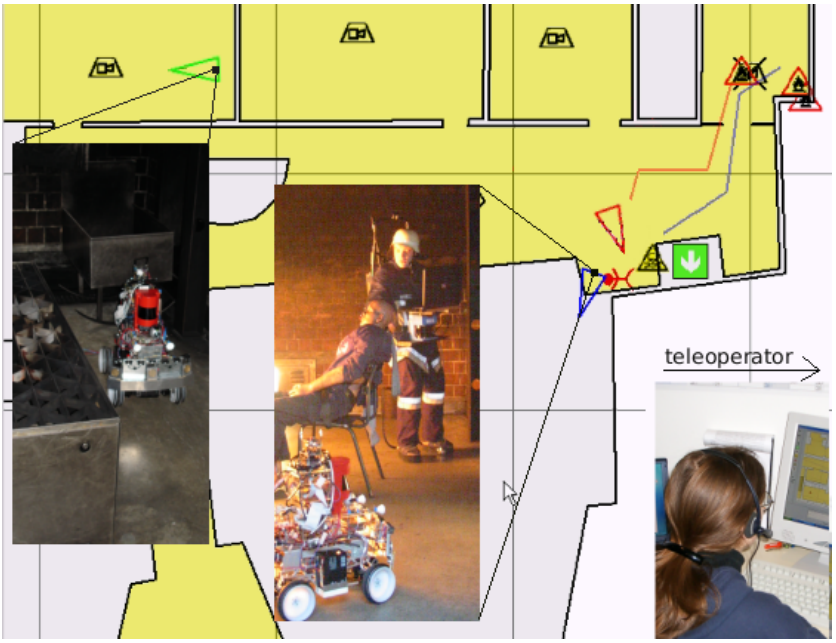

Figure 1.1: PeLoTe use case example. Human and robot entities are searching and mapping the rescue area, while the operator is combining the information and controlling the tasks.

being teleoperated and assisted for co-operation, will be inevitable. The ability to recover the entities' (tele)presence promises a high application potential in industrial telemaintenance and telediagnosis. Telepresence methods interpret remote sensor data characterising the remote work environment to provide an efficient and realistic user interface for human teleoperators.

Key functionalities needed for presence features in PeLoTe system are

- Robust navigation systems and sensor data fusion.
- Robotic autonomy and teleoperation architectures.
- Strategies for cooperation in heterogeneous teams.
- Novel methods for SLAM.
- Robust telecommunication strategies.

Figure 1.2: With help of the PeNa system, the fireman can locate and navigate to the victim found by the robotic entities.

The selected PeLoTe use case example is a fire-rescue task, where the firemen and supporting robots are mapping a simulated rescue area together, as shown in figure 1.1. Both entities execute specialised tasks which are most natural for them. Robots can perform accurate navigation and measurements from the environment even in hostile conditions. Humans are able to give fast verbal descriptions of the situation and conditions. Human senses are also more versatile than robot senses in dynamical complex situations requiring indirect conclusions. Remote human operator supports the exploring entities by teleoperating the robots and supervising the firemen. He also summarises the verbal information given by the firemen to the common environment model, e.g. adds reported victims. These both kinds of entities are expected to verify the operating space in non-standard situations, e.g. in poisonous and low visibility environments, using cooperative activity planning and localisation, see figure 1.2. [5]

1.2 Personal Navigation System

The Personal Navigation system (PeNa) is a human localisation system for indoor environments. It is designed to operate without any external infrastructures. This is especially applicable in large-scale missions, e.g. fire fighting missions, where the infrastructure inside the building can be damaged, or can not be assumed to exists at all. Thus, the PeNa development aims for a robust stand-alone capability, i.e. the system is independent of all other systems, to guarantee a functional system in all indoor environments.

The PeNa can also be described as a wearable multi-sensor system for human indoor localisation. It provides an extent of sensory information for the operating human entity through additional sensors carried along by the PeNa user. This sensor information can be further used by teleoperating entities to gain a better feeling of presence on the location and to the task at hand. In this sense, the PeNa equipment works as an augmented reality device as adding computer generated information to the user environment. This can be seen as "local telepresence", and is especially applicable in situations where human senses are limited, e.g. due to darkness or smoke.

The PeNa system can receive information from several different sources, in addition to its own sensors. It can be supported with radio beacons, and other robots and humans can support PeNa by sharing environment information. The information between entities is transferred through PeLoTe server. If the server connection is not available, the PeNa can still operate using the data acquired by the PeNa sensors.

The PeNa sensors include a 3DM-G orientation sensor (gyro, compass, and accelerometers), a fibre optic gyroscope, a Stride Length Measurement Unit (SiLMU), and a 2D laser scanner. These sensors are mounted on standard hiking backpack along with a power system and laptop computers. All the PeNa system components are shown in figure 1.3.

The PeNa system is designed to be usable without extensive training. The user does not have to care about the sensors or how the localisation is done,

Figure 1.3: The PeNa system in operation. All the main components are pointed with arrows in the picture.

only the localisation result displayed on the map has to be understood by the user. The PeNa equipment does not need any kind of calibration for different users, the only assumptions are the maximum user velocity and the maximum angular velocity which are, consecutively, about $10m/s$ and $270deg/s$. The only assumption about the environment is that there exists static objects, e.g. walls or pillars, which are visible to the sensors for most of the time. As this assumption does not expect any specific objects, or even require them to be visible all the time, it is a minimalistic assumption for an indoor navigation system.

Chapter 2

Related work

In the first part of this chapter several existing systems for human localisation are presented. The emphasis is on the systems that are operating without external infrastructure or use already existing infrastructures, i.e. systems requiring operation environment modifications are not reviewed. The latter part of the chapter presents laser based localisation algorithms. All of these presented localisation algorithms were developed for robots, but are also applicable for human localisation.

2.1 Personal Navigation Systems

Personal Navigation Systems [6, 7], also referred to as Pedestrian Navigation Systems (PNS) [8] or Personal Positioning Systems [9], have been an actively researched topic. The developed navigation systems can be divided into indoor and outdoor systems, and further to infrastructure based systems and stand-alone systems. Most of the previously proposed stand-alone systems use accelerometers for step detection, and a magnetic compass or low cost gyros for heading determination [9]. The infrastructure based navigation systems often use a Global Positioning System (GPS) and rely on dead reckoning in case the GPS signal is unavailable [8].

The reference [10] represents a typical solution for a human navigation system. The system includes a three-dimensional rate and acceleration sensors, a three-axis magnetic sensor, and a digital barometer. The system velocity is estimated using walking frequency calculated through detected accelerations. The step length of the user is computed from the accelerometer's power spectrum, and the system heading information is received from the magnetic sensor. The relative accuracy of the system is from 5% to 10% of the distance travelled.

The Technical Research Center of Finland has developed a distributed location estimation system based on wearable accelerometers and magnetic sensors [11]. The sensors are packed into SoapBoxes [12] which are sensor boxes specially designed for ubiquitous computing. One SoapBox is placed on the user's waist and one, or two, on the legs. The sensor box on the leg is used for step detection and step length estimation. The sensor box fixed on the waist is used for heading determination. A computer is used to check the compass heading during the step and to calculate the position displacement. The system can recognise two special types of user behaviour: "walking along a corridor" and "walking up or down the stairs", which can be used with odometry to localise the entity in a map. The average error of the system was found to be around 5% from the total travelled path in case of level walking. In the presence of magnetic field distortions, which have severe effects on the heading accuracy, the average error was found to be around 9%. Similar systems are also reported in [13] and [14].

The Geodetic Engineering Laboratory of Swiss Federal Institute of Technology (EPFL) and Vectronix AG have developed a Pedestrian Navigation Module (PNM), shown in figure 2.1 [15]. The PNM has a GPS receiver, a digital magnetic compass, a gyroscope, a barometer, and some embedded dead reckoning algorithms. All the sensors are placed in a small box, with operational weight of about 150 grams and size of 74.7 mm x 48.3 mm x 18 mm. The PNM can navigate relatively well without absolute position correction from the GPS, e.g. in indoors, but is inherently faced to lose the

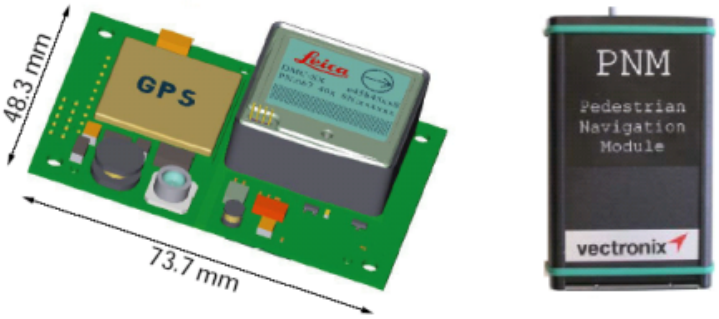

Figure 2.1: Commercial Pedestrian Navigation System (PNM) sold by Vectronix AG. Adopted from [16].

location because of the errors in gyroscope angle integration and step length estimation. The average error is found to be around 2% of the distance travelled even without the GPS support. [16]

One commercially available outdoor navigation system for humans is the POS/LS system of Applanix, shown in figure 2.2. The POS/LS is an approach of GPS/Inertial Measurement Unit (IMU) integration customised for land survey. The POS/LS is a portable backpack system comprising an IMU unit, 3 ring laser gyros, 3 accelerometers, and a Real-Time Kinematic (RTK) GPS. During use, POS/LS continuously monitors position quality in order to request Zero Velocity Update (ZUPT) to limit error propagation. Simultaneously, the POS/LS automatically incorporates available GPS signals to real time position updates. When the GPS is unavailable, POS/LS allows the operator to command manual position updates. The ZUPT aided 3D real time location error of the system is less than 1 metre per 1 kilometre. [17]

The Point Research Corporation has developed a Dead Reckoning Module (DRM) for human positioning, shown in figure 2.3. The DRM module was originally designed for military purposes but is now also available for public, making it one of the few commercially available human dead reckoning

Figure 2.2: The Position and Orientation System for Land Survey (POS/LS) made by Applanix. Adopted from [17].

devices. The DRM measures the displacement of the user from the system initialisation point by measuring the direction and the distance travelled with each footstep. The DRM module can be used with a GPS which makes the outdoor localisation accurate, i.e. within the GPS accuracy. Moreover, the module offers capability to reset the position when accurate position information is available. The accuracy of the module, without the GPS support, is promised to be between 2% to 5% from the distance travelled. [18, 19]

The DRM module heading is provided with a solid-state 3 dimensional compass, which is calibrated with accelerometers to get the horizontal component. The step counting is automated with an electronic pedometer which uses accelerometers to detect the footfalls. The module also includes a barometric altimeter to detect changes in the altitude. The module sends data and is controlled via bidirectional RS-232 serial interface. A second serial port on the module may be connected to a GPS receiver. When GPS is available, the module will blend the GPS localisation data with the Dead

Figure 2.3: Point Research CO's Dead Reckoning Module (DRM).

Reckoning (DR) information using a Kalman filter. The DRM module parameters are shown in the table 2.1. [20]

Table 2.1: Technical parameters of the DRM module.

Parameter	Value
Dead Reckoning Accuracy	2%-5% of distance travelled
Altitude Accuracy	3 metres
GPS Accuracy	5 metres
Temperature	0 to +70 degrees Celsius
Power Required	0.24W (DRM only), 1.24W (DRM+GPS)

The DRM module was tested and found inadequate for PeLoTe purposes, because the DRM module only approximates the steps based on the calibrated step length, without any usable error estimates [2]. This causes that further calculations have to be made based on the worst-case scenario, i.e. the step length and the direction of the step can be anything. This kind of worst-case initial estimate can also be done without the DRM module.

There exists a wide range of location systems that require special infrastructures, e.g. the HiBall Optical tracking system [21], the infrared-based Active

Badge location system [22], the ultrasonic-based Cricket location-support system [23], the Bat Ultrasonic Location System [24] and the RFID-based LANDMARC system [25]. Some systems fuse wearable sensor information with infrastructure based navigation, e.g. Pedestrian Tracking System (PTS) based on Berkeley motes [26] and infrared-based personal navigation system [7], which both have an average localisation error less than 1% of the distance travelled.

The research of infrastructure based localisation systems have been popular because the systems are capable to provide absolute position corrections in a fixed environments, e.g. in warehouses. An extensive comparison of several infrastructure based localisation systems is presented in [27].

2.1.1 Global Navigation Satellite Systems

The Russian Global Orbiting Navigation Satellite System (GLONASS) and the American NAVigation Satellite Timing and Ranging (NAVSTAR) GPS systems have been providing localisation services since early 1980's. A third emerging satellite localisation system is the European Galileo system, which should reach the Full Operational Capability (FOC) in 2008. It will be interoperable with GPS and GLONASS systems. [28]

The NAVSTAR system, usually referred as the GPS system, has been established as the preferred method for outdoor positioning. Currently the GPS allows receivers to localise with about 10 metres accuracy. The accuracy can be further enhanced to about 5 metres with the Differential GPS (DGPS), which uses stationary receivers to calculate the errors in the GPS timing signals and to transmit the error corrections to other receivers. [29] The wide use of GPS receivers have brought the size and price of the receivers so low that it is possible to embed GPS positioning systems into various devices. For example the Benefon company has a GPS based ESC navigation phone available for the consumer markets [30].

The main disadvantage of GPS system is the loss of signal due to obstruction

and attenuation, e.g. because of buildings or the human body. The GPS has also relatively poor accuracy in relation to the scale of the locations in building level, e.g. when seeking building entrances, and thus the GPS system applications are mainly developed for outdoors. [31]

For indoor localisation, a GPS signal based NAVIndoor pseudolite system is introduced by the Space Systems Finland (SSF). It uses indoor pseudo satellites to send signals that are usable with standard GPS receivers. It has been showed that the system is capable to sub-decimetre accuracies. [32]

There exists also satellite systems that provide GPS signal corrections. In North America this service is provided by the Wide Area Augmentation System (WAAS), in Asia by the Multi-Functional Satellite Augmentation System (MSAS), and in Europe by European Geostationary Navigation Overlay Service (EGNOS). For example a WAAS-capable receiver has a positioning accuracy of less than three metres for 95% of the time. EGNOS and MSAS-enabled receivers can operate at least three times more accurately than standard GPS receivers, i.e. the localisation accuracy is three metres or less. Both EGNOS and MSAS should be fully operating in 2005. [28]

2.1.2 Navigation Systems Based on Existing Infrastructures

It is always a big advantage if navigation system does not require modifications to the used environment, i.e. the navigation system is stand-alone or utilises existing infrastructures. This section describes how mobile phone, IEEE 802.11 wireless LAN, or Bluetooth networks can be applied to navigation systems. At simplest, all these Radio Frequency (RF)-based networks can use cell-identification for localisation, i.e. entity is localised in the area of the currently used network node.

User localisation in mobile phone networks has already been in use for some time. In 1996, an U.S. Federal Communications Commission (FCC) decision

imposed that the location of mobile phones had to be identified at the cell level. A second added impetus was given by a law passed in the U.S. requiring that from October 2001, it must be possible to localise the emergency calls with an accuracy of 100m (300m) in 67% (95%) of the cases. [27] The EU Commission has passed a similar goal in directive 2002/22/EC (Article 26) in March 2002 [33].

The basic mobile network location services are based on Cell Identification (CI), with operating accuracy of few hundreds of metres in urban areas and several kilometres in rural areas. The accuracy can be further enhanced by using Time of Flight (TOF), Time Difference of Arrival (TDOA), or propagation time measurements. Also by adding more transmitters/receivers to base station, the GSM cell can be divided into sectors that can be used for sector level localisation. Still, a further amended Enhanced Observed Time Difference (EOTD) measurement gives about one hundred metres accuracy, but it requires some hardware changes in the networks and software updates for the handsets. [31, 34] The GSM localisation works also indoors, but the accuracy described above isn't usable for more than pin pointing the building where the user is currently in.

More suitable ready-to-use RF-based systems for indoor localisation are the RADAR system and the Ekahau Positioning Engine, which both use a standard 802.11 network adapter to measure RF signal strengths [35, 36].

The localisation using the RADAR can be based on scene analysis or on multilateration. Scene analysis localisation requires a priori measured signal pattern database from the building, against which the current measurement is compared. The achieved accuracy is 3 metres with 50% probability. The multilateration localisation needs to calculate distances to at least three base stations, and uses triangulation to achieve about 4.3 metres localisation accuracy. The major strength of this system is that it uses the infrastructure provided by general wireless networking. Somewhat drawback is that the statically recorded signal pattern database values may greatly differ in dynamic environments. [35]

The Ekahau Positioning Engine requires a floor map image, which is used for manual signal strength reference point teaching. The calibration is done in several points and the software compares the calibration data with the current data to derive the current user position. The average localisation accuracy is promised to be about 1 metre. [36]

Another established solution for indoor localisation is based on Bluetooth networks. The accuracy of these systems is currently about the same as the Bluetooth cell size, which is normally about 10 metres. A better support for Bluetooth localisation might be available in the future because the Bluetooth Special Interest Group (SIG) is actively developing Bluetooth standards that also involve localisation. [37]

2.2 Laser Based Localisation

Laser based localisation refers to methods that use laser scanner data for localisation. The data can be processed as raw laser scan data or by using more advanced features extracted from the data, e.g lines. The localisation algorithm can be based on sequential scan matching, referred to as laser odometry, or to map matching. The laser odometry is inherently accumulating error to the location, but the map can be used to provide absolute positioning against the map's reference frame.

The first part of this section describes laser scan-to-scan matching methods that can be used to provide the laser odometry. In the second part is described how laser scans can be used with map, and in the last part, the situation where the odometry calculation and map matching are done concurrently, a problem referred to as Simultaneous Localisation and Mapping (SLAM).

Most of the presented localisation algorithms have been developed for robots, but can also be applied to humans. In most of the cases, this is not a completely straightforward procedure. This is because of the different dynamics

of robots and humans. In mobile robots the sensors are usually installed statically on top of the robot, when with humans, the sensors move pretty much arbitrary along with the human body. For example in walking, the heading variations are typically from 20 to 40 degrees/s, but can be up to 200 degrees/s in the worst-case.

2.2.1 Laser Scan-to-scan Matching

Scan-to-scan matching, or simply scan matching, is a problem where translational and rotational parameters are estimated to fit consecutive laser scan measurements against each other. This process gives a set of incremental position and heading changes which can be integrated to the current pose. This kind of position estimation is referred to as "laser odometry" [38]. The laser scan-to-scan matching is a subproblem of the shape registration problem, the task of aligning two similar shapes, and thus there exists numerous algorithms which can be applied to scan matching [39].

To solve the rotation error between two scans, a Cross-Correlation Algorithm (CCA), also known as histogram correlation algorithm, was introduced to scan matching. [40, 41, 42] The Cross-Correlation Algorithm (CCA) calculates a histogram using angles between adjacent scan points for each laser scan. By calculating a cross-correlation over two angle histograms, the place of maximum correlation gives the angle difference between the scans. The algorithm is also roughly invariant against translations, enabling thus the rotation difference to be calculated separately.

The CCA can also be applied for translation corrections [41]. The laser scans have to be rotated so that x- and y-axis are aligned with common directions with the scans, i.e. aligned with walls or other dominant structures. The most common direction is obtained as maximum of the scan's angle histogram. After the rotation, the x- and y-histograms can be calculated separately, which is computationally more effective way to determine the translation than the correlation calculation over the xy-plane. Because x-

and y-coordinates are inherently correlated, the algorithm can not utilise all information contained by the scans, and thus fails especially if no distinct common direction is found.

Iterative Closest Point (ICP) algorithm [43] is a well-established iterative matching algorithm for scan matching. The ICP algorithm finds first the closest point pairs for two scans and then calculates transformation to minimise the distance between these coupled points. This matching is repeated until the convergence is concluded. The assumption is that the transformation between two scans is relatively small to converge. The ICP can also be used to pair a point and a tangent plane to overcome the need for exact point pair correspondence [43]. The algorithm can be made more robust by neglecting outliers and missing features [44]. There exists also ICP modifications that induce probability estimation to the algorithm, e.g. through Extended Kalman Filter (EKF) [45].

Lu and Millos introduced two laser scan matching methods: a tangent-based matching algorithm and a point-to-point least-squares minimisation algorithm called Iterative Dual Correspondence (IDC) algorithm [46, 39]. The tangent-based algorithm defines tangents for two scans and minimises a distance function between them. It can handle large initial pose errors and thus gives a good starting point for the IDC algorithm, which is more accurate but too complex to be used with large search areas. The IDC algorithm is a ICP variant which calculates point-to-point correspondences between the scans that are being aligned by minimising the sum of squares of distances. The point correspondence problem is solved based on two heuristics: the closest point rule and the matching range rule [39].

Iterative Dual Correspondence-Sector (IDC-S) is an IDC variant that is developed for dynamic environments. The algorithm divides the laser scan into several sectors, detects changed sectors, and matches the scans only using unchanged sectors. The IDC-S is shown to recover errors even if the environment changes more than 50%. [47]

In [48] is presented an algorithm, named as Combined Scan Matcher (CSM), which combines the point-to-line matching approach and the IDC method to provide better efficiency and robustness for the scan matching. The results from CSM shows that right combination of different scan matching algorithms can be used to add robustness to the localisation.

2.2.2 Map Based Localisation Using Laser Scans

Map based localisation is a problem of finding the entity's position when given a known map. In our case, the laser scan measurement are used to determine the entity's position on the map. The environment representation is an important factor for the map based localisation. It defines the used information, features usable in localisation and if the localisation uncertainties can be known. In general, map representations can be divided into three categories: geometric, topological and hybrid environment representations [49]. In the case of entity localisation, the geometric representation is usually further divided into spatial occupancy grids and geometric maps [50], as shown in figure 2.4.

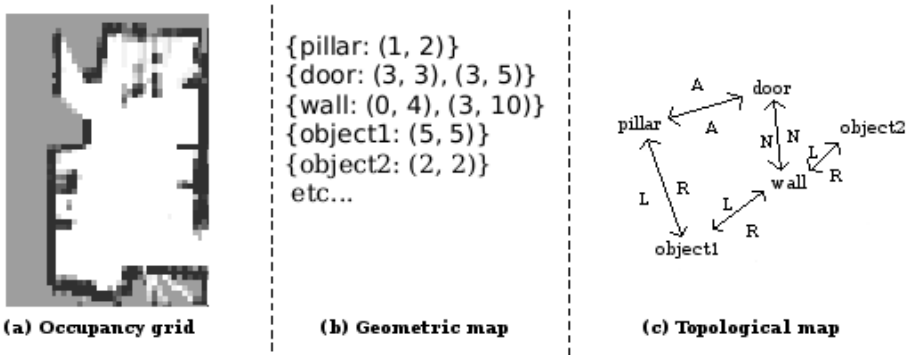

Figure 2.4: Environment representations for entity localisation.

One of the simplest map matching algorithms is the point-to-line matching. The [51] matches laser scan points to a priori CAD line segments based on closest neighbourhood, and then searches for a translation and rotation that

minimises the total squared distance between scan points and their targeted lines.

Occupancy grids [52] and evidence grids [53] represent an effective solution for localisation in small environments. Localisation is accomplished by using the same cross-correlation methods that are used in image matching [54]. The correlation can be calculated between a priori long-term map and short-term perception map by rotating and translating the short-term map over the search area. The localisation error is comparable to the chosen grid resolution. The occupancy grids can be matched directly, i.e. grids to grids [53], or be extracted into features (e.g. line segments) before the matching [55]. Occupancy grids are also popular solution for sensor fusion, because the map can be updated using various different sources including sonar, laser, and proximity sensors [56].

Topological presentation [57] contains no distance information, so the sense of position is maintained in connectivity map which contains all the possible states. For example the office navigation robot Dervish [58] recognised walls, doors, and hallways which it uses to calculate probabilities for possible locations on the map. Also a person driving to some destination uses a similar topological environment presentation. He doesn't know the exact distance where to turn but instead drives along the road until he recognises a place where to turn, e.g. the third street on the left after the hilltop. Similarly Dervish robot uses simple wall-following algorithms to move between the map nodes. [58]

The advantage of topological environment presentation is that the map size is relatively small and it can be easily used for path planning. The need for recognisable objects is instead a quite drawback for generalisation, because it requires the environment to contain some predefined landmarks. This can be in some level neglected if new recognisable objects are generated on the fly, but recognition of these objects can not be guaranteed. Thus the topological environment is mainly used with landmark-based navigation. [59]

Markov Localisation (ML) is a well-known localisation method with both occupancy grids [60] and topological maps [59]. The key idea of ML is to calculate probability distribution over all possible locations in the environment, thus making it also suitable for global localisation. With large state space, e.g. with occupancy grids, this is computationally very time consuming task [61]. Thus there exists several enhanced ML algorithms, such as Correlation-Based Markov Localisation (CBML) [62] and dynamic Markov Localisation [63], that are better suited for real-time applications.

A Monte Carlo Localisation (MCL) algorithm is a particle filter based on Bayesian theory that has effectively been used for a priori map matching [64]. In MCL the pose of the entity is approximated with particles. Each particle represents a position with a probability that tells how probable it is that the entity is in that position. The whole particle cloud is normalised and thus represents the probability distribution of the entity's pose. The MCL is introduced in details e.g. in [65]. MCL is currently very actively studied area especially for the SLAM purposes, which is subject of section 2.2.3.

Another well-suited method for localisation is the Kalman filtering. The Kalman Filter (KF) is a state estimator that first makes a prediction based on the system dynamics and then corrects this prediction using measurements. It provides an estimate of the current location of the entity and gives also a measure of how certain the estimate is. If the noise sources are Gaussian distributed, the KF estimator is optimal with respect to any reasonable measure for optimality. [66]

Stochastic mapping is a special way of organising the states in an Extended Kalman Filter (EKF) for the feature relative navigation. Stochastic mapping assumes a metrical, feature-based environmental representation, in which objects are represented as points in a defined parameter space. A state vector represents estimates of the locations of the entity and the features in the environment, and an associated error covariance matrix represents the uncertainty of these estimates, including the correlations between the entity and feature state estimates. As the entity moves through its environment

taking measurements, the system state vector and covariance matrix are updated using an EKF. [67, 68]

The environment representation categorisation could have been done also on other basis than geometric versus topological. One well known category is the feature-based models [69], which classify the environment geometric features according to sensor responses. The detected geometric features can be for example Hough transformed lines or corridor corners [68]. In simplest, feature based map localisation can be based on sensor trained model which is trained before the run time. From this model, some features can be recognised later and the user location can be recognised. An example of this kind of world modelling and localisation is a wearable computer which operates only with 3D accelerometer, 3D magnetometer, fluorescent light detector, and temperature sensor [70].

The algorithms described in section 2.2.1, can be utilised in map matching by using "virtual scans". Virtual scan is an artificial scan that is extracted from the map in a supposed entity location. By matching this virtual scan against the real laser scan measurements, the relative rotation and translation corrections against the map can be calculated. This algorithm provides accurate localisation if the initial guess is good enough. [38] This approach is presented in this study on section 4.3.

2.2.3 SLAM

Simultaneous Localisation and Mapping (SLAM), also referred to as Concurrent Mapping and Localisation (CML), is the problem of acquiring a map from an unknown environment while simultaneously localising relative to this map [71, 72]. The SLAM problem is considered to be the biggest challenge for developing a fully autonomous robot because the solution would allow robots to operate in a unknown environments without a priori map and without independent position information sources, e.g. GPS.

The idea of SLAM is to get the consistent map by observing static features

from the environment. The entity localisation and map are both exposed to the same sensor measurement noise, and thus the map and the pose errors are inherently correlated. This makes the data association difficult because the map and pose have to be solved simultaneously. The complexity of SLAM problem has been shown to been $O(N^2)$, N being the number of landmarks in the map [38].

Because the SLAM problem is characterised by uncertainty and sensor noise, the probabilistic techniques have been the common approach to the solution. Probabilistic algorithms approach the problem by explicitly modelling different sources of noise and their effects on the measurements. [73] The 2D laser scanner, which is also used in this study, is probably the most accurate and common sensor used with SLAM problem. For this reason there exists many SLAM algorithms that can be used with range finder data. [38]

The dominant solution was established in [71] where a Extended Kalman Filter (EKF) based localisation algorithm is used to incrementally estimate the posteriori distributions of entity's pose and landmark positions. In [72] a proof is presented that a EKF based "full" solution to the SLAM problem truly exists. The solution is full in the sense that it can optimally use the system measurements to estimate location, a property inherited directly from the Kalman filter.

The EKF SLAM fuses the information from odometry sensors and the observation made from the environment. The map features are extracted from observation into a state vector. The solution will converge if the features are detected and associated correctly. Because the size of the state vector is proportional to the map features, and the size of the covariance is the size of the state vector squared, the EKF estimation becomes computationally heavy in large environments. [38]

A SLAM solution with $O(M \log(N))$ complexity (M is number of particles, N is number of features) is described in [74], called FastSLAM. The FastSLAM solution uses a Rao-Blackwellized particle filter to sample the robot poses

and a Kalman filter to track predetermined landmarks. Each particle has N Kalman filters that estimate the landmark positions on the travelled path. The assumption is that the landmark positions are conditionally independent given the robot pose. It has been demonstrated to work with 100000 landmarks, which can not be achieved e.g. with the EKF SLAM [75]. A modified FastSLAM algorithm proposed in [76] adds the most recent measurement in the pose prediction process. The paper proofs that the modified FastSLAM solution converges in one particle case, and can run in real-time with the same accuracy than the EKF SLAM.

An especially challenging SLAM problem is the so called loop-closing problem [77], also known as problem of mapping cyclic environments [78]. While the entity moves through a large loop, the error in position grows, and when it returns to initial position, it is impossible to tell whether this is a new position or already visited position. In [77] the loop-closing problem is solved using visual features along with laser scanner data. The solution uses stored visual features to determine when the loop closing should occur, as shown in figure 2.5.

Another SLAM solution that solves loops is an EM-based SLAM algorithm described in [79]. EM algorithm is a hill-climbing routine which repeats two steps: expectation step (E-step) and maximisation step (M-step). In context of SLAM, these steps correspond roughly to a localisation step and a mapping step. The approach represents the world as a collection of small local maps associated with their locations. The Expectation Maximisation (EM) is applied to align these local maps which are eventually projected into a single global map, as shown in figure 2.6. The problem of this solution is that the alignment phase is so complex that it can not be run in real-time.

The reference [80] represents Local Registration / Global Correlation (LRGC) method for building large cyclic maps in real-time. In the local registration part the scans are integrated into a local map patches that are connect to form a topological network. In case of loop closing, the global correlation part finds topological relationships for the closing poses by correlating most

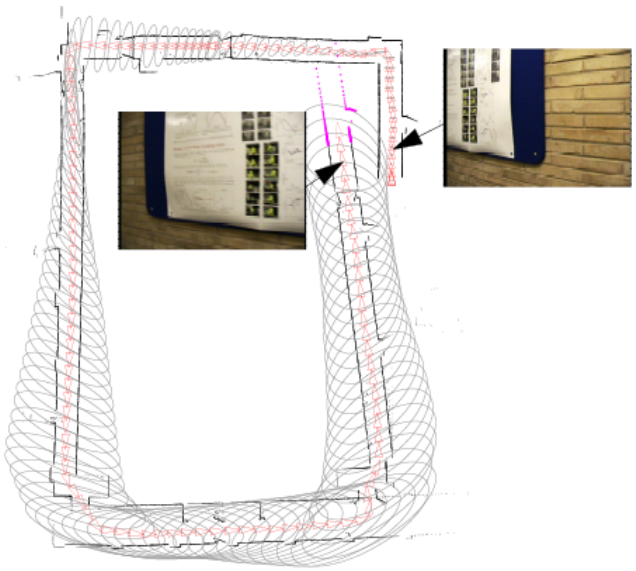

Figure 2.5: SLAM algorithm just before loop closing takes place. The vehicle poses are shown in red and global location uncertainty with grey ellipses. The images are the two cameras views used in the loop-closing process. Adopted from [77].

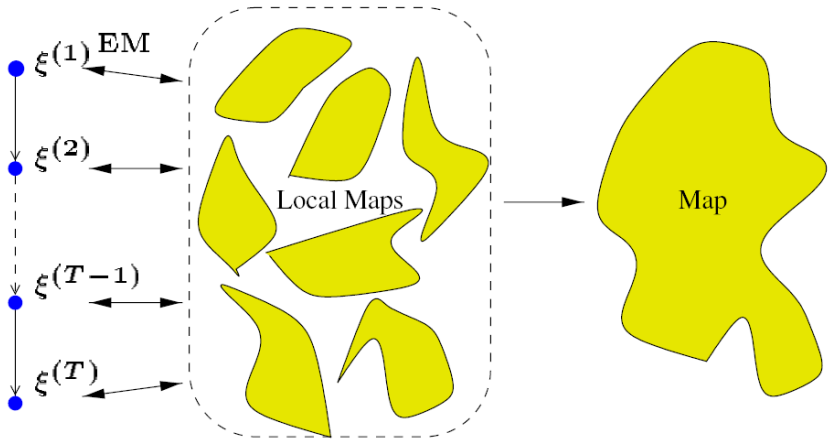

Figure 2.6: Two-layered map representation uses EM algorithm to align small local maps into a single global map. Adopted from [79].

recent local map patches with older portions of the map. If there is a good match then it is likely that the local maps are topologically connected, and

thus the map can be updated according to this. This computationally heavy loop closing operation can be run in background because the correlation map doesn't have to be available immediately. The method was demonstrated to be capable to close approximately 200 metres long loops in less than 10 seconds.

In the reference [81], a particle filter is used to represent both entity pose and also possible map configurations. The algorithm, called DP-SLAM, is based on a new map representation, called Distributed Particle (DP) mapping, which makes possible to maintain several thousands entity poses and candidate maps in real time. In DP mapping each particle maintains a pointer to its parents and also a list of grid squares that it has modified in the map. The DP has only a one occupancy grid map, in which each grid square contains particles that have occupied that grid square. Thus the solution allows each particle to behave like if it would have its own map. The algorithm makes no attempts to detect and smooth the errors when loops are closed, the precision results directly from the robustness of maintaining multiple maps. The algorithm was demonstrated to be able to correct 60 metres long loops without misalignment errors. [81] There exists also a DP-SLAM 2.0 with an improved map representation and a laser penetration model [82].

One serious problem of many SLAM algorithms is the growing size of the global map during long missions. In [83] a Constrained Local Submap Filter (CLSF) is used to generate local sub maps which are then fused periodically into a global map using constrains between common feature estimates. The approach is shown to be effective in reducing the computational complexity while still maintaining the accuracy of global SLAM algorithms operating with single map. Several other efficiency issues and SLAM algorithm criteria are reviewed thoroughly in [54].

Chapter 3

Technical Description of Personal Navigation System (PeNa)

This chapter describes implementation aspects of the PeNa system. The first part of the chapter describes the hardware components of the PeNa system. The latter part concentrates to the PeNa software architecture aspects.

3.1 PeNa Hardware

The PeNa hardware modules are all based on components previously used on robotic applications. The laser scanner, the 3DM-G orientation sensor, the portable computers, and the power and support structure are all off-the-shelf components. The Stride Length Measurement Unit (SiLMU) is the only hardware component that was developed for the project because there were not any such devices commercially available. Based on the related work review on the previous chapter, this is the first time that a laser scanner is used in a human localisation system.

The final version of PeNa is equipped with two laptops: PeNa/PAS laptop and a communication laptop. The PeNa/PAS laptop makes sensor measurements, processes them, and displays the Graphical User Interface (GUI).

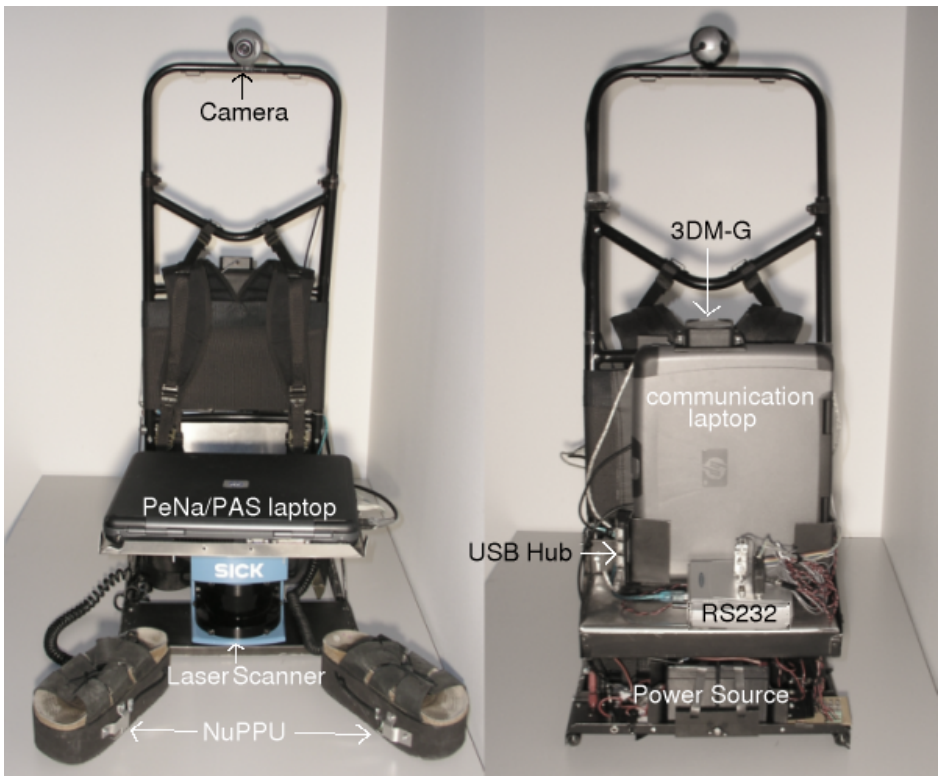

Figure 3.1: Front and back view of the Personal Navigation System.

The PeNa/PAS laptop requires two RS-232 ports, one for SiLMU and one for 3DM-G module, and one high speed RS-422 connection for the laser. The serial ports can be usually arranged to laptops with a suitable combination of USB-hubs and USB-serial-adapters, as shown in figure 3.1. The communication laptop is added to guarantee that the actual PeNa/PAS calculations are not affected by the resource peaks caused by WLAN connection roaming, video capturing, or audio connection.

3.1.1 Power and Support Structure

Supporting structure, on which the PeNa system is built, is a traditional hiking backpack which is stripped from unnecessary components. Some additional aluminium rods are added to support the power source, laptop, and

the laser. The backpack is needed because of the relatively large total weight of 20 kilogrammes.

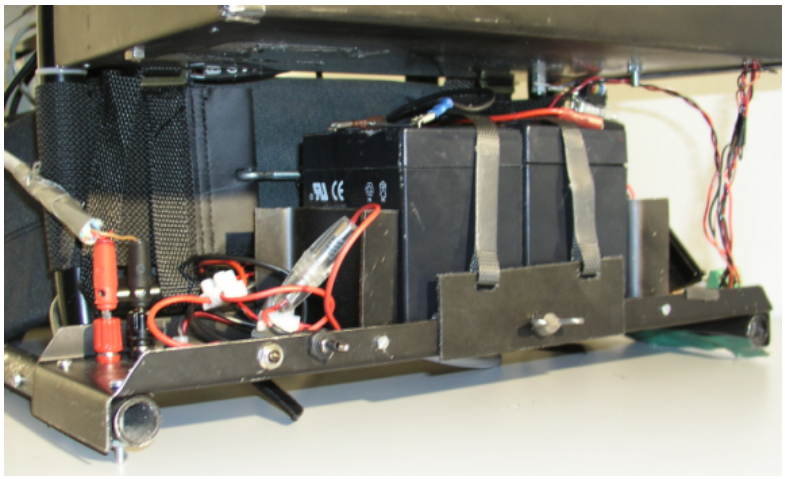

Figure 3.2: The power subsystem. Batteries, charging plug, and diode are marked in the picture. On the left side of the batteries, not visible in the picture, are 24VDC to 12VDC converter, 12VDC to 5VDC regulator and three fuses.

The power subsystem, shown in figure 3.2, provides the power for the PeNa sensors. It consists of two 12VDC 4Ah lead-acid batteries connected in serial to provide a 24VDC. The 24VDC voltage is regulated to 5VDC and for the SiLMU and to 12VDC for the gyro. Only the laser uses the unregulated supply voltage of 24VDC. The system is equipped also with a charging plug for recharging the batteries. The charging line includes a diode to prevent the reverse current in case of reversing the charging voltage. The system should be charged with 29VDC and with no more than 1A of current. The system has a main fuse to protect the system in case of short circuit.

3.1.2 3DM-G Inertial Measurement Unit

MicroStrain Inc.'s 3DM-G module, shown in figure 3.3, is a 3-axis orientation sensor combining 3 angular rate gyros, 3 orthogonal MEMS accelerometers,

Figure 3.3: The 3DM-G orientation sensor provides 360 degrees range over pitch, roll, and yaw axis.

and 3 vector magnetometers to provide a 360 degrees angle range over pitch, roll, and yaw axis. The 3DM-G measurements output is in global coordinate system with respect to the Earth's coordinate system, i.e. its z-axis is pointing down through the centre of the Earth, its x-axis is pointing north and its y-axis is pointing east. [84]

The data is received through 38.4 kbps RS-232 connection with an average data refresh rate of 100Hz. The data is acquired in polling mode, but continuous mode would have been equally applicable. The technical parameters of the 3DM-G are presented in table 3.1.

Also Hitachi Fiber Optic Gyroscope HOFG-X was tested with the PeNa system. The gyroscope measures the angular velocity around horizontal axis, i.e. the yaw angle, without trying to compensate the angle around any other axis. The data measurement rate is approximately 20Hz, which is more or less invariant to used gyroscope settings. Minimum detectable rotation rate is 0.05degrees/second and maximum recommended angular velocity is round 60 degrees/s. Due to this low maximum angular velocity, the optical gyroscope was not used in the final PeNa setup. It was noticed that humans can rotate at least 180 degrees/s, which is reached with the 3DM-G sensor.

Table 3.1: The technical parameters of the 3DM-G sensor. Global accuracy is the typical error for an arbitrary angular orientation in world reference frame and repeatability is the error in relative local reference frame. In practise, these present the accuracy of compass and gyroscope components of the 3DM-G sensor.

Parameter	Value
Operating voltage	5.2 - 12VDC
Supply current	around 90mA
Max angular velocity	300 degrees/s
Global Accuracy	+- 5 degrees
Repeatability	0.10 degrees
Gyro Bias Stability	0.7 degrees/s

3.1.3 Laser Scanner

The used laser scanner is a SICK LMS200 which measures distances in 2D plane. The device is set to measure 180 degrees with 0.5 degrees steps to 80 metres distances with the resolution of 1 centimetre. In this mode the sensors response time is 26ms. A RS-422 serial high speed connection provides 37.5Hz measurement frequency in continuous mode, which is more than enough to track the human movements. The laser is a class A1 laser product, and thus completely eye safe. The power consumption is approximately 20W and the weight of the laser is 4.5 kilogrammes.

The LMS200 laser scanner is mainly intended to be used in good visibility conditions. The range of the sensor is dropped rapidly when visibility is getting worse, e.g. with visibility of 50 metres the maximum sensor range is approximately 20 metres. The PeNa was tested in Würzburg firehouse with smoke, and the results was that laser could not see pass the smoke for more than few tens of centimetres.

Figure 3.4: SICK LMS200 provides 361 distance measurements from 180 degrees angle range.

3.1.4 Stride Length Measurement Unit

An accelerometer based stride length measurement unit was developed to estimate the human gait. The sensor is based on two orthogonal VTI's Micro-Electro-Mechanical Systems (MEMS) accelerometers which can provide accelerations in the xy-plane. The accuracy is only indicative because the earth gravity component and the angle of the leg are not compensated during the walk.

Also an ultrasonic based stride length unit was tested with PeNa system, presented in [1]. It worked better than the accelerometer based unit, but caused conflicts with ultrasonic beacons used in the PeLoTe project. An advantage of the accelerometer based unit, compared to the ultrasonic version, is that it is capable to distinguish if the human is walking onwards, backwards, or sidewards.

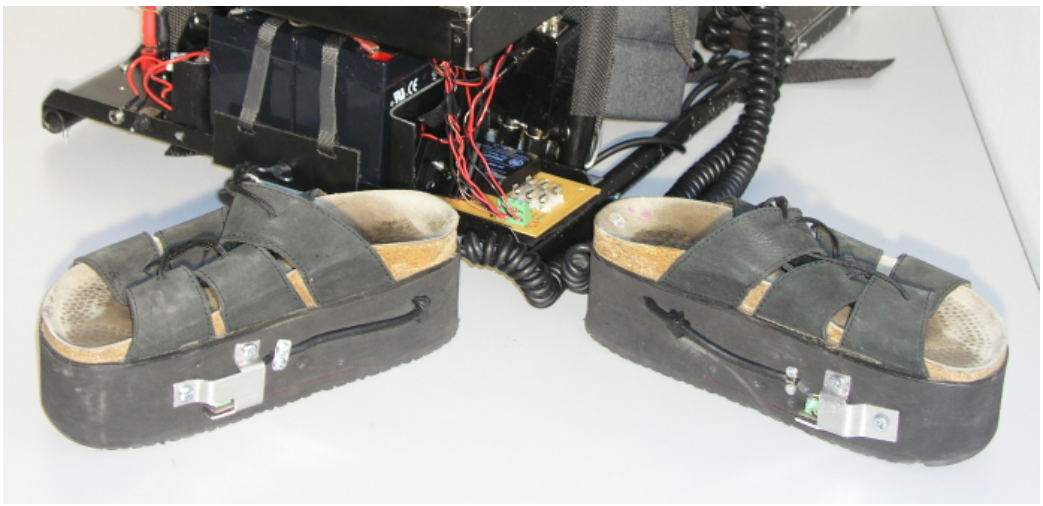

Figure 3.5: The Stride Length Measurement Unit (SiLMU). The stride length is estimated using two orthogonal accelerometers. The raw accelerometer data is read through RS-232 connection to laptop for the actual stride length calculations.

3.2 Software Architecture Description

The PeNa software architecture can be divided into two parts: a hardware oriented platform dependent C/C++ part and a platform independent Java part. The C/C++ part contains device driver codes, measurement calculations, and sensor fusion codes. The C/C++ code is compiled into a shared Dynamic Link Library (DLL), which is a shared library of executable functions used by Microsoft Windows applications. The main program is run with Java, which uses the DLL-libraries through a Java Native Interface (JNI). In PeLoTe project this main program is Personal Assistance System (PAS) system which provides GUI to the PeNa system, and sends the received data through Remote Method Invocation (RMI) to the PeLoTe server, see figure 3.6. The Java version used in the project is "Java 2 Platform version 1.4.2 API".

The software is written specially for the PeNa sensors, but can be easily

adopted to use other sensors, with only modifying the driver for them. For example compass module is just a wrapper module that converts the function requests to match the currently defined compass hardware.

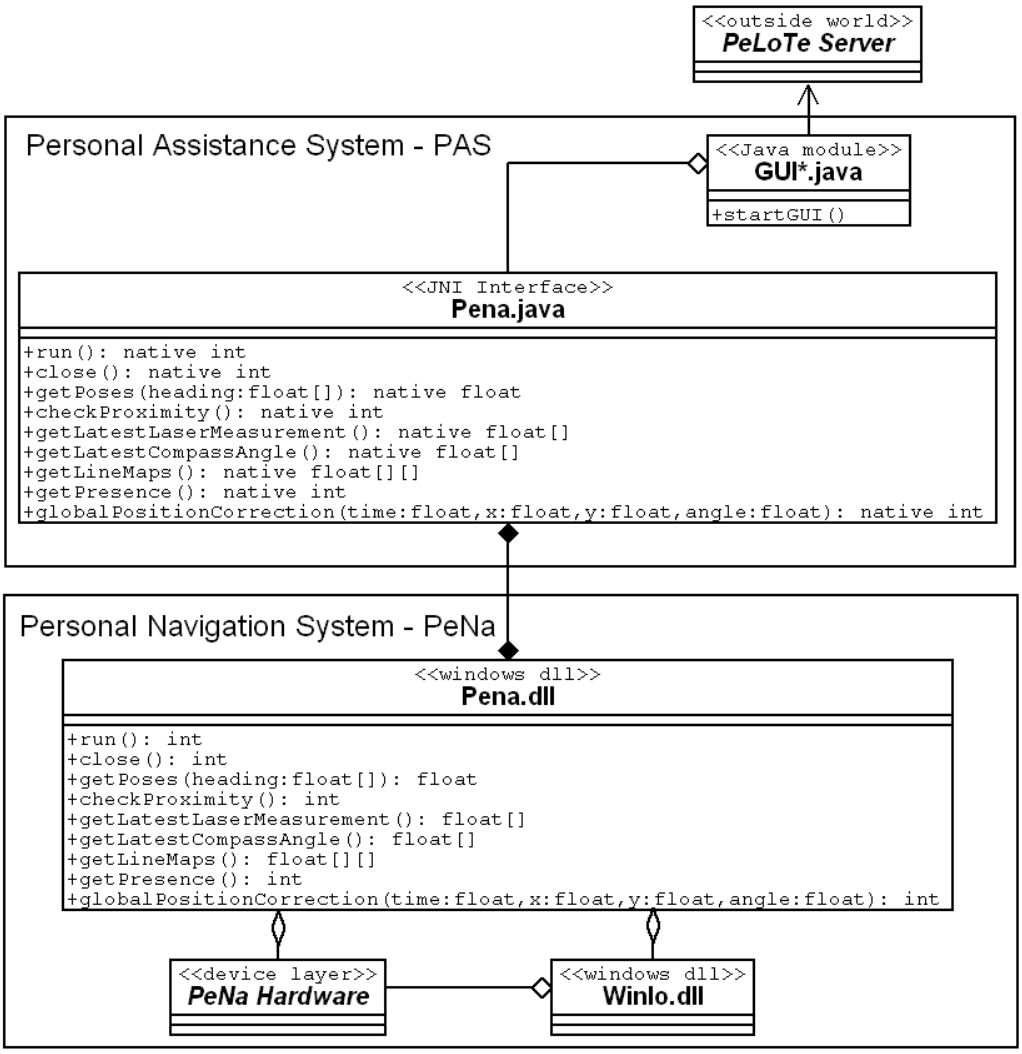

Figure 3.6: The PeNa software system architecture in the PeLoTe system.

3.2.1 PeNa library

The core of the PeNa software code is written with C/C++ and compiled using Microsoft Visual C++ 6.0 compiler, which contains some non-standard implementations from things like threads, serial ports, and sleep functions. This makes some parts of the implementation platform dependent. Modules using RS-232 serial connections are designed so that all the platform dependent code is written to the serialdrv.cpp, and thus only that file would have to be replaced when changing the platform.

The Pena.dll, shown in figure A.1, is started from the Java environment through JNI interface with Pena.Java class run() function. This starts all the necessary device drivers and calculation threads in the C/C++ side. The threading is necessary because several tasks have to be executed simultaneously and with different priorities. For example all device drivers are run in separate high priority threads to ensure that all data from sensors is read to the memory.

The main PeNa modules are: compass, optical_gyro, kalmanfilter, nuppu and laser_odometry. Tracking module controls the execution of these modules through mutexes and semaphores. Kalman filter is not running in it's own thread because it is purely computational module. It takes compass and gyro data as input, and returns the Kalman filtered data as output.

Gyro, Compass and SiLMU module

Compass module is a wrapper module, i.e. it converts the function calls to the format understood by the selected compass driver. Two compass drivers were implemented: Dead Reckoning Module (DRM) and 3DM-G module. The selection of which one to use is done in the compass header file. Gyroscope module has similar structure with compass. Both request measurements continuously, and push the results in a delivery buffer from where other modules can fetch them as needed.

Nuppu is driver of the SiLMU module. It has similar structure with Gyro and Compass, using separate thread to make measurements through RS-232 port. The SiLMU module driver implementation is written in nuppu.cpp/h and computation algorithms are written in nuppulation.cpp/h. The received data is processed using functions defined in nuppulation.cpp. The nuppulation.cpp is not included in figure A.1 because it is concealed from the other modules by the nuppu.cpp.

Laser module

The laser module is the most complex module in PeNa. The laser scanner gives measurements with up to 37.5Hz frequency, and between measurements, it makes the heavy scan-to-scan matching and map correlation computation tasks required. Data is transferred in and out from the laserOdometry-Interface using Standard Template Library (STL) List class and mutexes. Laser dead reckoning calculations and map correlation are written in the laserOdometry.cpp. The laserOdometry is actually the central thread for all laser related calculations, as can be seen from the figure A.2.

The laser functionality is separated into several modules which are sealed behind the laserOdometryInterface class. The laser communication module is the lms200, which continuously reads measurements from the laser and pushes them on request to the user. One of the biggest modules is the linelib module, which contains for example all the line handling functions. The final PeNa software has two laser odometry calculation and map localisation implementations. Both of the laser odometry algorithms are correlation based algorithms, as the one described later in the chapter 4. The map localisation algorithms are a virtual laser scan matching and a MCL-based map matching algorithm.

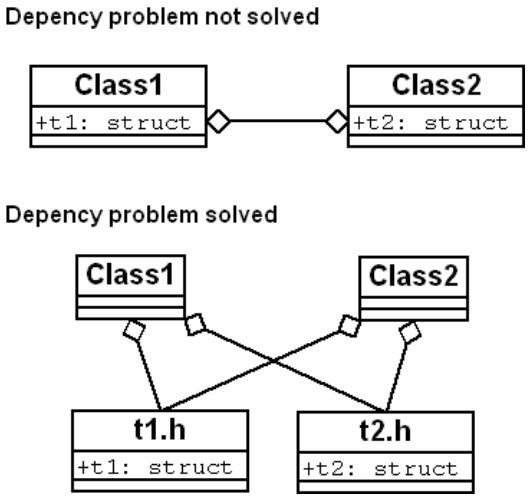

Figure 3.7: Dependency problem. Both classes need t1 and t2 structures which causes dependency conflict. The problem is solved by separating the conflicting structures into separate files.

Other modules

A error_handler module is designed to be used for error and debug printing. From the error_handler.h file can be chosen which messages are printed and where, i.e. in the file or on the screen. Also few common functions like converting parameters is defined here. Serialdrv module is a common driver for handling serial connections.

TPose class was made to deal with problem of module dependency problems, i.e. several modules need structures which have to be available separately or the module dependencies would conflict with each other. The situation is shown in figure 3.7. This solution could have been used more widely with structs that are needed by several separated modules. The problem is very hard to deal otherwise when number of code modules increases.

A timestamp class is needed to convert time between the Java time and the native C/C++ time. The native code in Pena.dll use timestamps given by clock() which is milliseconds from the start of the program. Java instead, uses number of milliseconds since 01.01.1970. This Java timestamp is valid

timestamp in PeLoTe system if the system clocks are synchronised, e.g. using the Network Time Protocol (NTP) protocol.

There is also currently unused code module called Java_query_module.cpp/h, which can make arbitrary type of Java method calls. The calls are saved into buffer which is later processed in the Pena.cpp class. The called Java methods have to be implemented in Pena.Java file. If the method does not exists in Java side, then the call does not do anything. These Java calls were designed to be non-blocking so no value is returned by the Java methods. The Java method call requires the method name, method signature, and Java parameter types. The possibility to misuse the parameters is significant, so it is practical to write wrapper functions for each Java method that also checks if the given parameters are valid. The Java_query_module was tested with external PeLoTe beacon module, which provides co-operative localisation capabilities between human and robotic entities.

3.2.2 PeNa User Interfaces

Several different User Interface (UI)s were developed for the PeNa system in different parts of the project. The first ones were developed for real-time algorithm debugging and to see what properties should be included in the final PeLoTe GUI. The simplest version is a text interface which only prints the current PeNa position. Similar simplistic test interfaces can be found for all the PeNa modules.

First GUI version was a 3D OpenGL GUI application which only displays the latest laser scan, as shown in figure 3.8. The program was written in C using OpenGL Utility Toolkit (GLUT) Win32 OpenGL library. The 3D presentation is very realistic when there is only empty walls in the environment, e.g. in hallways, but most of the rooms look a bit confusing in the GUI. This is because some of the walls seem to appear and disappear when the laser scanner only now and then hits some of the furniture. The laser scanner can not see any objects laying on the floor, and thus can mislead the PeNa user to

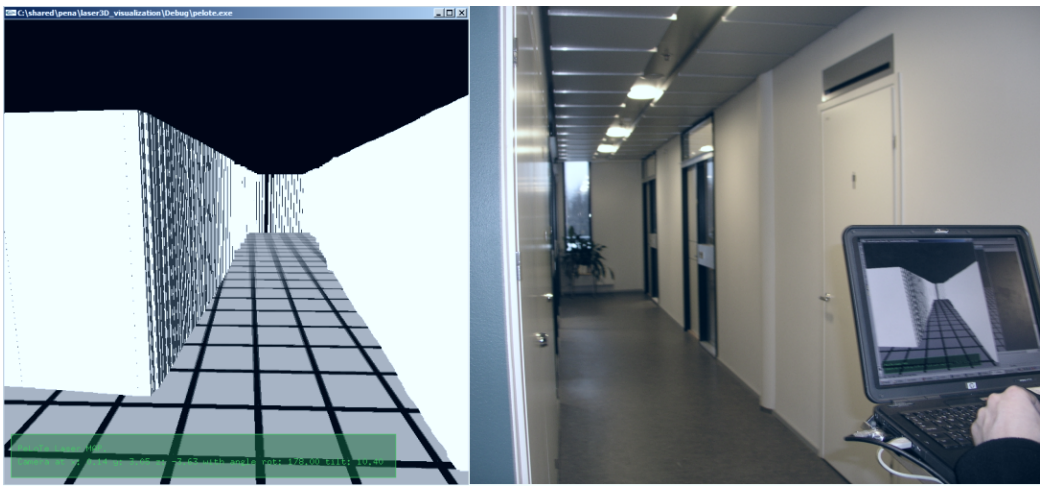

Figure 3.8: The PeNa GUI which displays the laser scan information into a 3D presentation.

move a bit too recklessly. The 3D GUI relies mostly on OpenGL supported functions and thus requires a laptop with OpenGL acceleration card.

The second GUI was Java Swing based PeLoTe GUI prototype, where the laser scan matching results could be seen in real-time, as shown in figure 3.9. It displays the latest laser scan in a current estimated global position. Although the GUI was very simplistic, it already had some performance problems when few hundreds of laser scans needed to be updated on the screen. The GUI used Java canvas class which is fast to draw but hard to keep updated with the latest scans.

The second GUI was used to test the RMI connection to the PeLoTe server which enabled use of several GUIs, e.g. one for the PeNa user and one for the teleoperator. The communication with server was very minimal, only position and laser scanners were saved to the server. The main purpose was to test that the designed software architecture is feasible for the PeLoTe system. Only problems encountered were with the laboratory firewalls which did not support the wide port range used by the RMI connection. This was

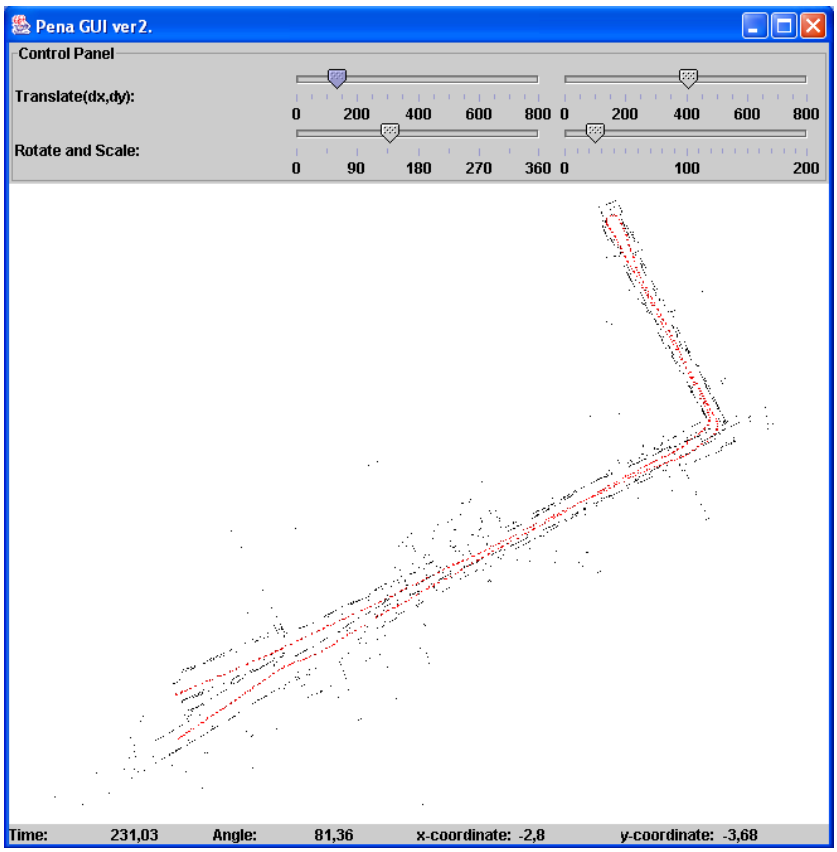

Figure 3.9: The PeNa GUI developed to display the laser scan matching results in real-time.

solved using RMI http-tunnelling, which is known to be valid solution for tunnelling through corporate network firewalls.

The most advanced GUI developed for the PeNa system is the one used in the final PeLoTe project demonstration, called Personal Assistance System (PAS). It was developed by Frauke Driewer, and it is capable to display all the PeNa data in a layered map, as shown in figure 3.10. The user can enable and disable displayable objects to get just the right level of information required. The PAS GUI communicates through RMI connection with the PeLoTe server, which makes possible for example to display data from other entities and to get updates into the map.

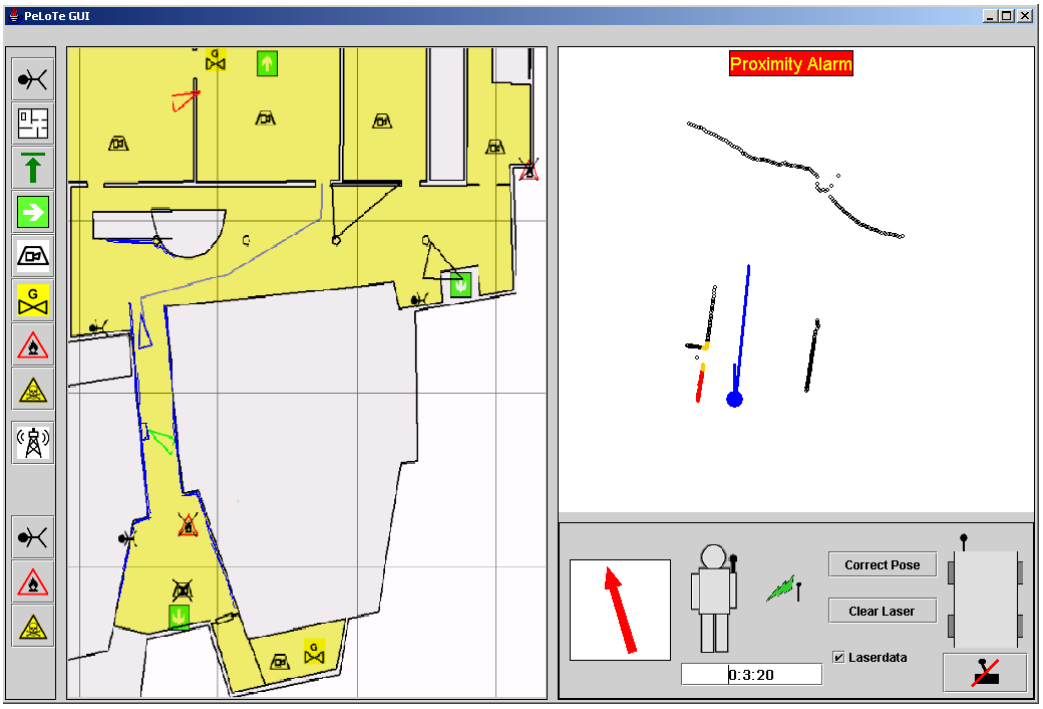

Figure 3.10: PAS is PeNa GUI used with Pelote project. It displays all PeNa data and also information about other entities.

The PAS GUI has two screens, global map view and local laser scanner view. Path is displayed in both views, but only to the next check point in the laser scanner view. The global view should be used to get view of the whole rescue situation, like locations of other entities, fires, and possible victims. Laser view provides a local navigation view, practical for dodging objects and navigating to the next path checkpoint.

Chapter 4

PeNa Localisation Algorithms

This chapter describes the algorithms developed for PeNa localisation purposes. The traditional dead reckoning algorithms, i.e. localisation based on inertial sensors, are first described in the section 4.1. In the section 4.2 is presented algorithms that use laser measurements to enhance the dead reckoning results. The section 4.3 contains algorithms that use a priori maps to localise the PeNa entity. Last in the section 4.4 is described the Simultaneous Localisation and Mapping (SLAM) algorithms.

4.1 Dead Reckoning

The simplest localisation in PeNa is based on dead reckoning, often also named as odometry or inertial navigation. It refers to process where the location is calculated based on angle and distance estimates provided by the inertial sensors. In the PeNa, this dead reckoning is calculated using the SiLMU stride length measurements and heading estimates given by compass and gyro modules.

Indoor heading estimation with compass proved to be difficult because of numerous electrical gadgets are creating disturbing magnetic fields. Compass heading can thus be only used as a long time averaging drift compensator.

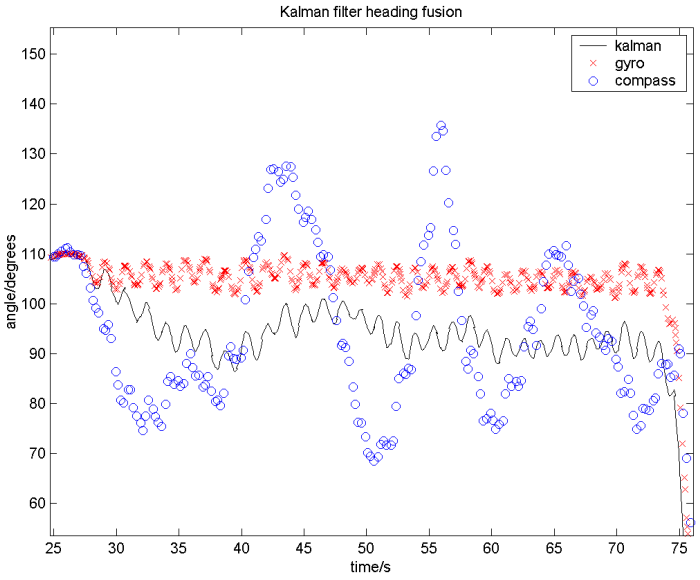

Figure 4.1: Gyroscope and compass angle fusion with Kalman filter.

The problem with the gyroscope heading is the drift, which is for example with our optical gyroscope about 2 degrees per minute. The problem is solved by fusing compass and gyroscope measurements with Kalman filter, as shown in figure 4.1.

The Kalman filter is configured so that it follows rapidly gyroscope values and slowly drifts towards the compass heading value. This way the PeNa heading is both absolute and capable to accurate rapid movements. Gyroscope drifting is not still efficiently compensated by the compass, and thus, the Kalman angle is never very accurate. This causes the heading to accumulate errors to the position calculated from the SiLMU measurements.

The initial Kalman estimate can be set to be the first compass angle measurement which means that the angle is started approximately in the global magnetic reference frame. Between compass measurements the system heading is predicted using the gyroscope measurements. The update for the Kalman prediction part, shown in equation (4.1), is the time difference multiplied with angular velocity. The Kalman filter measurement update part is

the Kalman gain multiplied with measurement residual, see equation (4.3).

The Kalman filter a priori estimation part, i.e. the prediction update equation, is

$$\hat{\chi}_{k+1}^{-} = \hat{\chi}_k + \triangle t \cdot \omega_{gyro} \tag{4.1}$$

$$P^{-} = P^{+} + Q \cdot \omega_{gyro} \tag{4.2}$$

where ω_{gyro} is gyroscope's angular velocity, $\hat{\chi}^{-}$ is estimated a priori angle value, $\hat{\chi}$ is a posteriori angle value estimation, P is the covariance of the angle estimate, and Q is the variance of the gyro measurement.

Kalman filter a posteriori estimation part, i.e. the measurement update equation, is

$$\hat{\chi}_k = \hat{\chi}_k^{-} + K_{gain}(z_k - \hat{\chi}_k^{-}) \tag{4.3}$$

$$K_{gain} = \frac{P^{-}}{S} \tag{4.4}$$

$$S = P^{-} + R \tag{4.5}$$

$$P^{+} = P^{-} - K_{gain} \cdot S \cdot K_{gain} \tag{4.6}$$

where z_k is compass measurement, R is the compass variance, and K_{gain} is Kalman filter gain. Since the only estimated state variable is the angle, all the variables are scalars.

When using the optical gyroscope as gyro and 3DM-G as compass, the selected angle variance values were $0.5\frac{rad^2}{s^2} \approx 40\frac{degrees^2}{s^2}$ for compass and $1.2 \cdot 10^{-4}\frac{rad^2}{s^2} \approx 0.6\frac{degrees^2}{s^2}$ for gyroscope. The filter behaviour with these setting can be seen in figure 4.1. The Kalman filter output follows very rapidly the changes in gyroscope values, and only very slowly drifts towards the compass value, which is induced to significant fluctuations. In the final

experiment the compass was so keen for random fluctuations that it was not used at all. Only the 3DM-G was used as a gyroscope. This was done by setting a very low reliability value for the compass measurements of the Kalman filter.

The SiLMU module provides estimation of the current stride length. This is combined with Kalman filtered heading estimate into a position vector estimate. The last complete step is held as a reference level and the upper body location is approximated with half of the perpendicular distance to the other foot. The idea is that the heading is fixed when the human has both legs on the floor, i.e. when the human has taken a step. The new position estimate can be calculated by finding the nearest SiLMU measurement for each Kalman angle by using associated time stamps. If new step was taken, the position is updated permanently. Otherwise only a temporary estimation for the position is calculated, as shown in figure 4.2.

The accelerometer-based SiLMU estimates the step length only from the leg decelerations. Because the acceleration part does provide good stride length estimates, the estimation between complete steps is predicted based on the last step. This works fine as long as the walk is somewhat steady, and in any case, it was found to give good enough initial estimates for further localisation. An example of dead reckoning data is shown in figure 4.3.

4.2 Laser Dead Reckoning

The term laser dead reckoning, or laser based dead reckoning, is used because the position calculation is based on integration of elementary movements between scans, i.e. position is updated based on sequential heading and distance values. This introduces similar noise to the localisation as with the normal dead reckoning. The laser dead reckoning can be thought to solve two distinct problems, heading error correction and translation error correction, which can be solved either separately or simultaneously. The heading error

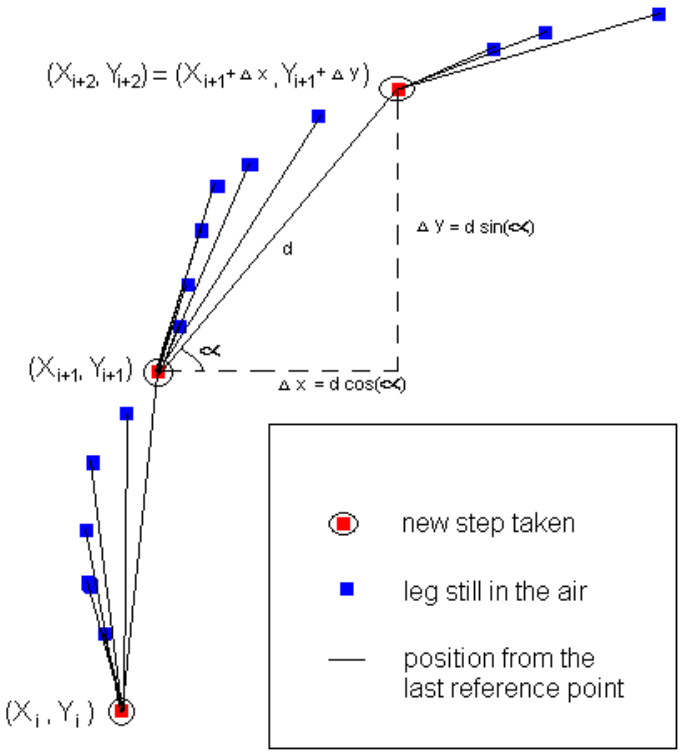

Figure 4.2: Location vector estimation using SiLMU stride length measurements and Kalman filtered angle measurements.

correction calculates the angle between two scans, and the translation error correction solves the translated distance between two scans.

If the problem is solved in two parts, the complexity is sum of one dimensional and two dimensional searches, i.e. complexity is $O(N^2 + N) \approx O(N^2)$. Simultaneous solving of these problems requires search in three dimensions and thus the complexity is $O(N^3)$. The advance of solving angle and translation simultaneously is that the correlation between the variables is maintained. This means that the final solution is really the best correlating answer regards to angle, x- and y-directions. Angle correction algorithms usually include some translation information, e.g. relative position of points, which maintains some level of correlation. Instead x- and y-directions are so strongly correlated that the separate solution is not the best correlating solution in

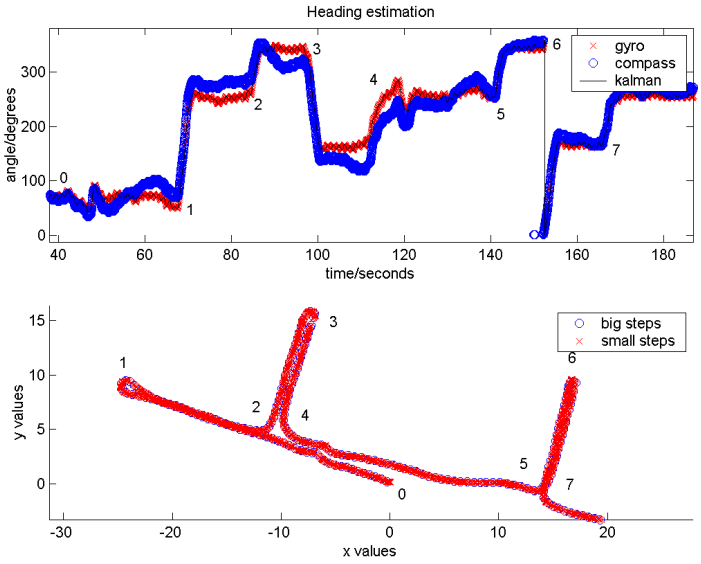

Figure 4.3: The relationship between Kalman filtered angle data and the locations calculated using SiLMU step lengths.

the xy-plane.

The laser scanner's limited measurement resolution and algorithm approximations, e.g. value discretisation, cause that errors are always present in the laser scan matching. When position is estimated between sequential scans, this error accumulates into a significant level. One way to reduce this error is to use static reference scans, shown in figure 4.4b. This means that a one scan is kept static until there is enough information between that and the current measurement. In many cases, the reference scan holds enough information for many successive measurements. Especially when localised entity is not moving, the reference does not have to be changed at all. Thus the error is not accumulating while the reference scan is static. This process can be thought to be a special case of the SLAM: the map is always re-initialised when reference scan is changed. [38]

More generally, the position calculation between several laser scan measurements can be made more error tolerant with redundant calculations, as shown

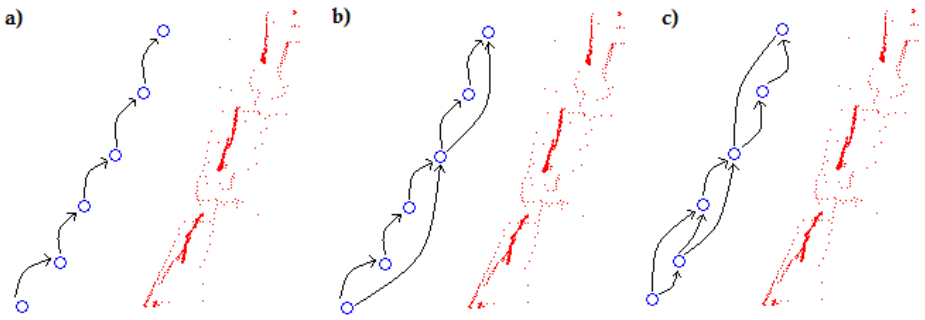

Figure 4.4: Integrating the position from laser scan measurements. The position calculated by a) integrating sequential positions, b) using reference scans, and c) making redundant calculations.

in figure 4.4c. For example, the estimation can be calculated between the reference scan and the current measurement, and between the current measurement and some older measurement. This way there is at least two paths from reference scan position to current position: the path between the reference scan and the current scan, and the integrated path based on consecutive scans. Both paths should give the same position estimate, and if not, there is an error between some pair of matched scans. This erroneous pair can be detected by determining which part of the path is inconsistent with the other calculated paths. [38]

The movement between laser scans can be large, especially when the localised entity is a human. This is a problem because large search area is computationally very expensive and is more prone to find ambiguous matches, e.g. a match in a square room with over 90 degrees search area gives several equally good solutions. An initial position estimate, e.g. from the dead reckoning, can be used to limit the search area significantly.

The initial estimate can also be used to detect unsuccessful matches using the redundancy idea: the initial estimate and the laser scan matching should give approximately the same result, and if not, somewhere is an error. Use of initial estimate is examined using an opening door problem, which is an especially difficult task for the scan matching algorithms. In the opening

door problem, the human is standing very close to the door, so that the door is almost the only visible object for the laser. When the door is opened, the situation looks for the laser like the user would be moving away from the door, although in reality, the observer is standing still and the door is moving. The opening door problem is shown in the figure 4.5. The resulted error in the heading is approximately 45 degrees.

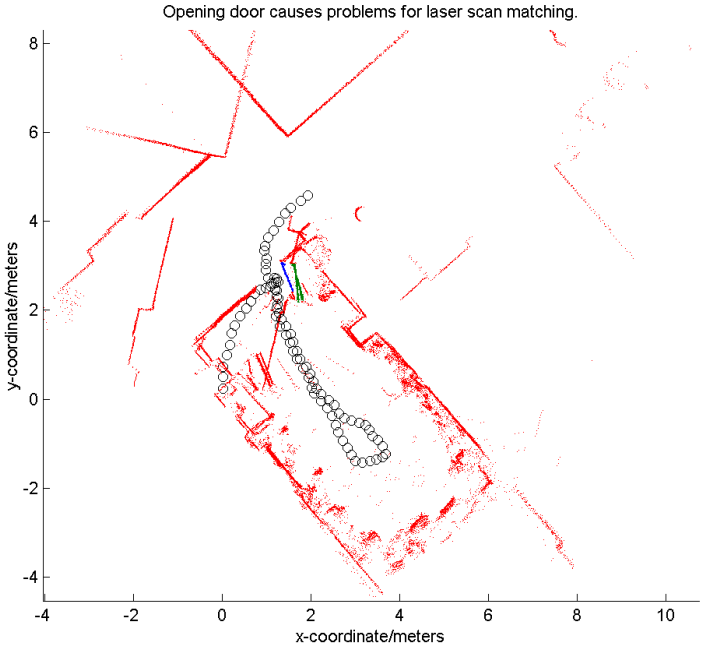

Figure 4.5: An opening door causes severe problems for the laser scan matching, because most of the laser scan points are moving along with the opening door. The door before entering the room is shown in green and after the visit in blue.

The door opening problem could be solved if the dead reckoning heading would always provide accurate initial guesses for the matching. In figure 4.6, the location is gradually lost when relying only on the dead reckoning information. Because of the errors in the dead reckoning heading, the laser scan matching search area has to be at least 5 degrees between two scans. This is actually more than the specified gyroscope angle accuracy. This can

be explained with the measurement synchronisation problems, i.e. the laser and the gyroscope measurements are not being executed exactly at the same time.

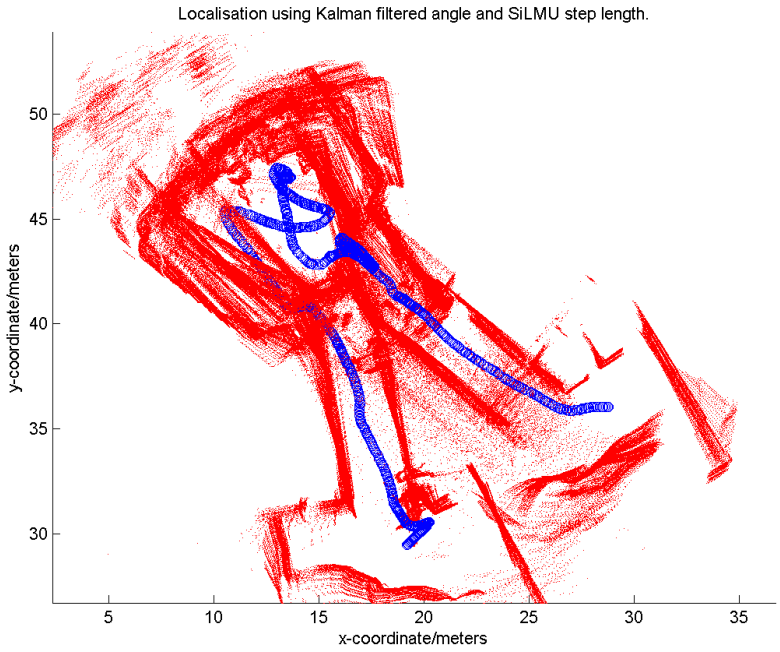

Figure 4.6: Inertial navigation using SiLMU step length measurements and Kalman filtered angle. The path starting and ending place are approximately in the same place.

4.2.1 Angle Histogram Correlation with Sequential Lines

This algorithm corrects the angle between two laser scans by using the laser scans' angle histograms, i.e. using a representation of the frequency distribution of data values. . If the consecutive scans represent the same environment, the histogram of the reference scan and the latest measurement are the same. If the scans are taken from different angles, then the histogram in the other scan is only shifted. The histogram calculation relies on the fact

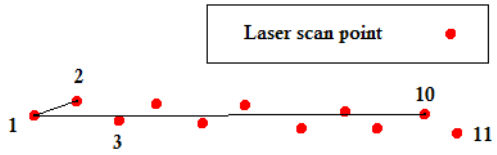

Figure 4.7: Angle histogram calculation using consecutive points. Lines between closes points do not describe the objects very well.

that the points received from laser range finder are ordered, i.e. one distance measurement after every 0.5 degrees is received.

The angle histogram is calculated by computing the angle between adjacent scan points. The original idea of the angle histogram is to calculate the angle between two consecutive scan points. This however does not provide accurate angle histogram, due to the sensor uncertainty. For example in figure 4.7, the angle between points 1 and 2 is completely different than the angle between point 1 and 10. The solution is to calculate the histogram for points separated by some fixed number of consecutive points, as illustrated in figure 4.8. The number of points used for angle calculations is an adjustable parameter of the algorithm.

Figure 4.9 shows a typical angle histogram for a straight corridor. There is clear maximum in 90 degrees angle, which implies to the corridor walls. Angle histogram for a office room is shown in figure 4.10, with wider range of angles, i.e. lines with several different orientations is found. The highest peaks are again from the room walls.

The algorithm steps are:

1. Calculate angles from the laser scan for the angle histogram using some fixed spacing between scan points, e.g. with step size 15, as shown in figure 4.8.

$$\alpha_i = \tan(\frac{y_i - y_{i-step}}{x_i - x_{i-step}}) \tag{4.7}$$

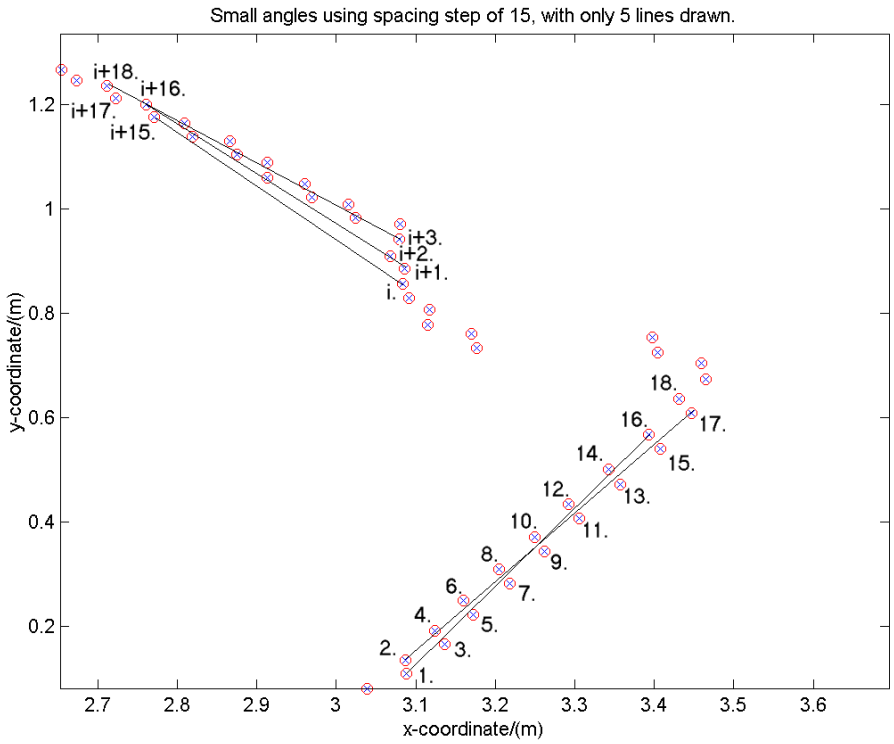

Figure 4.8: The angle between every 15th point is calculated.

2. Discretise the calculated angles ($alpha_i$) and calculate the angle frequency count, i.e. the angle histogram.

3. Calculate cross-correlation (C) for the angle histograms (D and E) over the wanted search area, e.g. 45 degrees search area with 0.5 degree step means that $k \in [-90, 90]$ and thus $C[k]$ has to be calculated in 181 points. Cross-correlation function is

$$C[k] = \sum_{i=0}^{N} D[i] \cdot E[i+k] \tag{4.8}$$

where N is the number of possible discrete values in the histogram, e.g with 0.5 degrees discretisation step the $N = \frac{360}{0.5} = 720$. The cross-correlation (C) plotted with all different shifts (k) can be seen in figure 4.11.

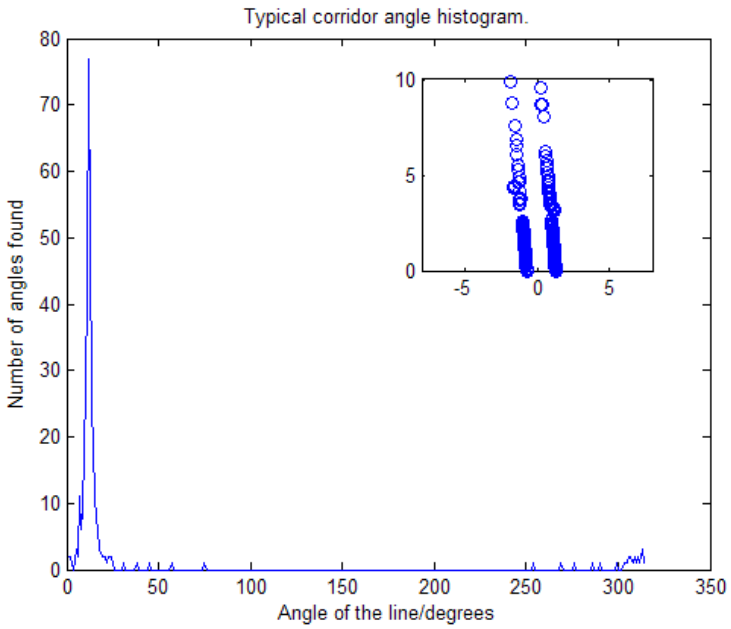

Figure 4.9: Angle histogram of a typical corridor. The constructed lines are all going parallel to the corridor walls so only one peak is found.

4. Save all local maximum cross-correlations ($max(C[k])$) and the corresponding angles.

5. Sort these cross-correlations and return wanted number of best correlating angles.

The algorithm gives good estimates especially when there are long lines in the scans, e.g. walls. The step size has to be large enough to smooth the measurement inaccuracies but small enough to be able to see some details. A step size around 30 was found to be the most reliable when using laser scans with 361 points, see figure 4.12.

The algorithm can be made more robust by using different step sizes in the angle estimation. The result is a set of different angle candidates for which the translation correction can be calculated. The histogram can be also used to give an approximation of the estimation error. For example in figure 4.13,

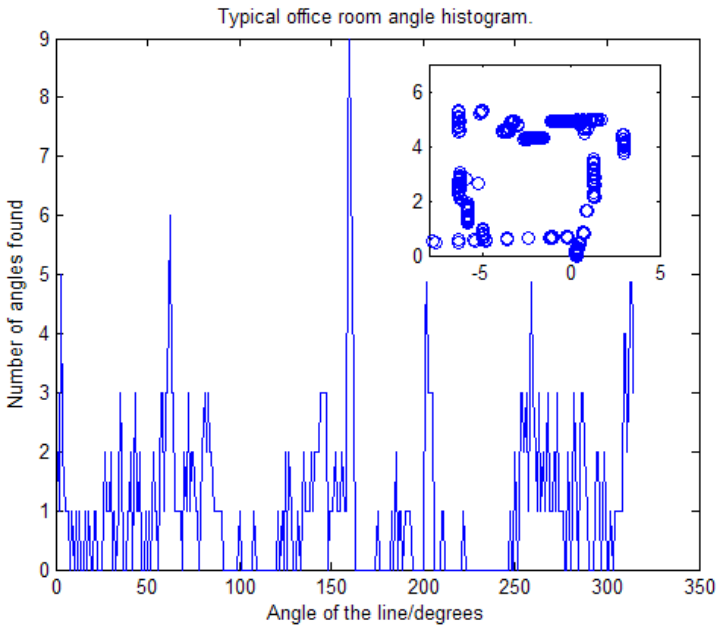

Figure 4.10: Angle histogram of a typical office room. The walls can be found from the histogram about 90 degrees apart from each other.

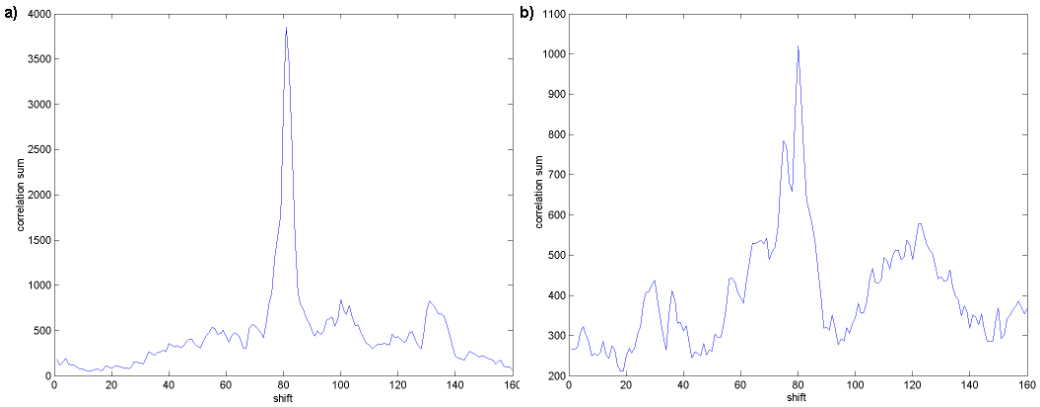

Figure 4.11: Cross-correlation sums of two scan pairs. In a) the cross-correlating histograms are from a corridor, and in b) from an office room. The best angle correction is the shift with highest cross-correlation value.

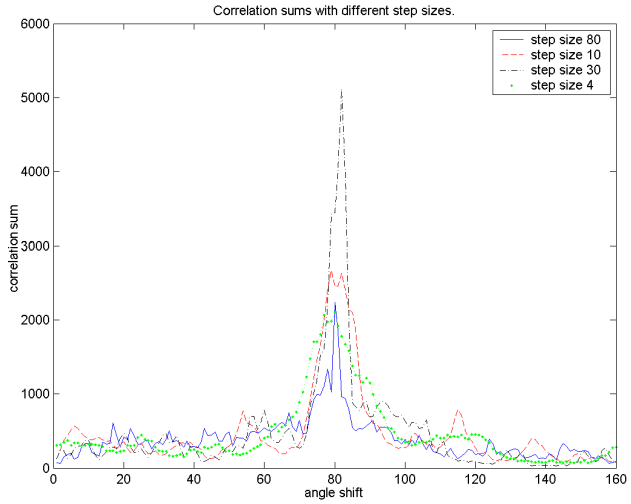

Figure 4.12: Angle cross-correlations with different angle step sizes. Step size 30 is found to be most suitable in many cases.

the maximum of the cross-correlation slightly differs between different step sizes, but the correct value is likely to be between 75 and 85 degrees. The information about the possible estimation error is important for a priori map localisation and SLAM algorithms.

The presented algorithm is basically the CCA algorithm described in the beginning of the section 2.2.1. The only difference is that the angle histogram was not calculated using lines connecting sequential points, but instead using lines connecting points separated by some fixed number of points. This algorithm was simple to implement, but it is still robust. The algorithm tolerates large orientation differences between the scans. Furthermore, the implementation only uses raw data as an input and does not make any assumptions of the environment.

With 20 different steps and 6 local maximums computed for each, the algorithm takes about 60ms to execute with 1GHz Pentium III. Use of three different steps is usually enough to return an accurate correction, which takes about 10ms to calculate.

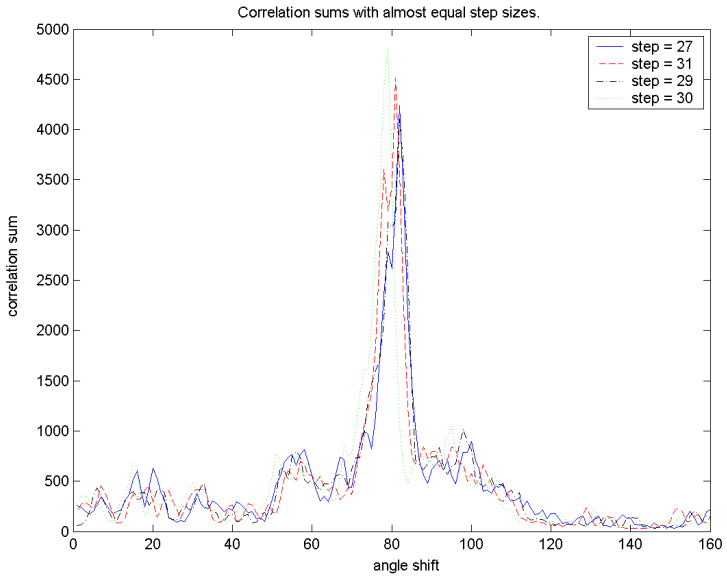

Figure 4.13: Angle cross-correlation with slightly different angle step sizes.

4.2.2 Angle Histogram Correlation with Segmented Lines

This algorithm, as the sequential lines algorithm described above, calculates the angle difference for two scans using histogram correlation. Main difference is that now the histogram is calculated directly from segmented lines. Scans are extracted into lines, so goodness of this algorithm depends on the goodness of the line extraction. If the lines are extracted successfully in a polygonal environment, the lines represent the environment very effectively. But if the environment is not polygonal, the lines are not actually representing any actual objects. This means that algorithm will probably fail because the line representations of sequential scans can be very different.

Benefit of using lines for histogram calculation is that the line angles are sharp, and the calculation is computationally more effective than if making the angle histogram directly from raw data. The lines should also be weighted according to their relevance. If significant number of points was used to construct a line, then it is also representing some distinct object that should be noted in the algorithm.

The segmented lines angle correlation algorithm steps

1. Extract scans into lines, e.g. using split and merge algorithm and least squares fitting.
2. Calculate angle for each line. Angles are between $[0, \pi]$.
3. The histogram weight for a specific angle is the number of points used to construct the line, e.g. the line in figure 4.7 would have weight of 15.
4. Go through angle correction search area using some discrete step size, e.g. 45 degrees in both directions with 0.5 degrees step size.
5. Calculate correlation on each discrete step (β). Correlation ($\mathcal{C}$) is calculated by multiplying the number of points in each line with angle difference function ($\mathcal{A}$). The equations are

$$\mathcal{C}[\beta] = \sum_{i=0}^{N_{scan2}} \sum_{k=0}^{N_{scan1}} \cdot P_i^{scan1} \cdot P_k^{scan2} \cdot \mathcal{A}_{k,i}[\beta] \tag{4.9}$$

$$\mathcal{A}_{k,i}[\beta] = \begin{cases} \frac{\alpha_l - (\alpha_{i,k} + \beta)}{\alpha_l} & , \alpha + \beta \leq \alpha_l \\ 0 & , \alpha + \beta > \alpha_l \end{cases} \tag{4.10}$$

   where P_i^{scan} is the number of points in the i:th line of the scan, and α is the angle between i:th and k:th lines in the compared scans. The $\mathcal{A}_{k,i}$ value is close to one if the lines are close to each other with the given offset (β).
6. Save all local correlation maximums and corresponding angles.
7. Sort the correlation maximums, e.g. with quicksort, and return wanted number of best correlating angles.

The segmented lines algorithm angle correction is compared with sequential lines algorithm in figure 4.14. Both algorithms find the maximums approximately from the same places. A more complex room navigation mission with both angle correction algorithms is shown in appendix B.

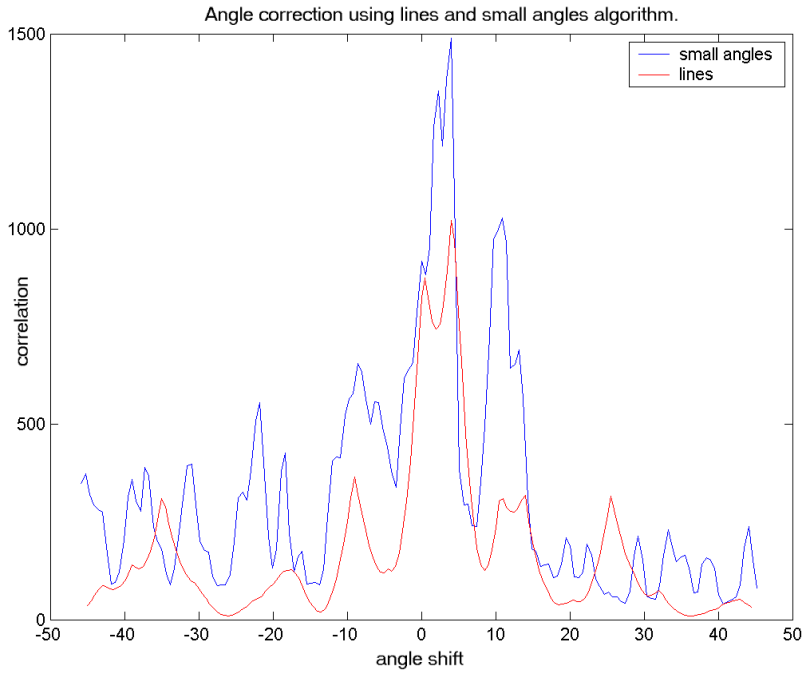

Figure 4.14: Comparison of angle correction using sequential (blue) and segmented lines (red) algorithms in a office room. Both algorithms find the maximums approximately from the same places.

With 44 lines on each scan and 18 best correlating angles returned, the algorithm takes less than 50ms to complete. The time is still roughly 50ms with 1Ghz Pentium III laptop if split and merge line extraction is included.

4.2.3 Angle Correction Using Line Intersections

The line intersection angle correction algorithm calculates angle difference for two laser scans using extracted lines, just like the segmented lines algorithm. This algorithm differs from the above described histogram algorithms

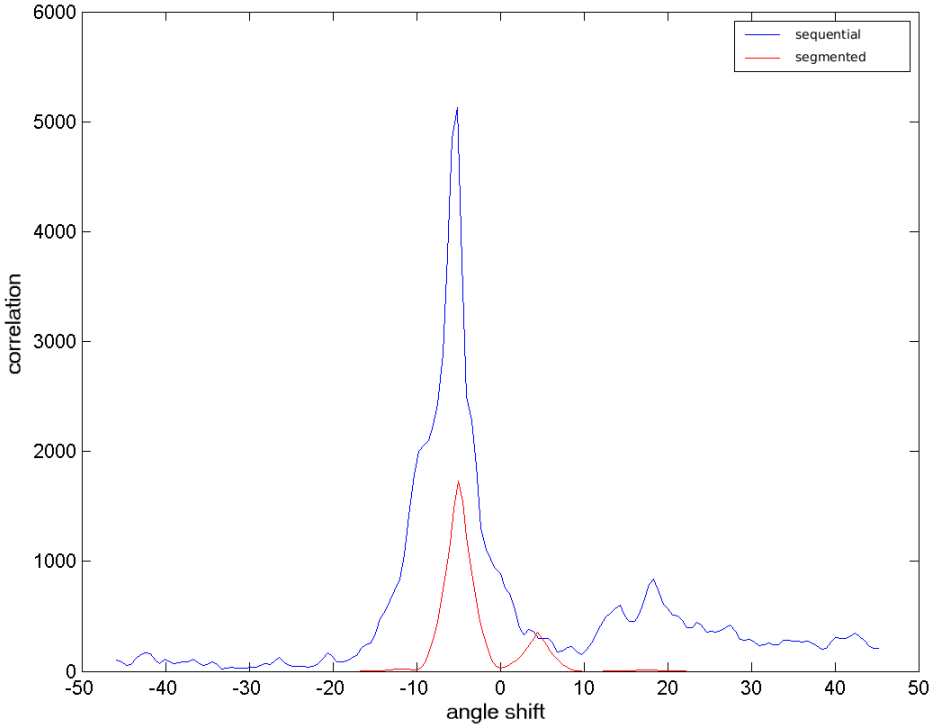

Figure 4.15: Both sequential lines (blue) and segmented lines (red) algorithm give relatively distinct estimates for the angle correction in a corridor.

in a way that it has a data association phase, where the maximum number of corresponding lines is detected between scans. The correction can be then calculated from the angle differences between corresponding lines. The algorithm starts by calculating the angle differences between all lines in one scan. Again, if two consecutive scans represent the same static, or near static, environment the angle differences are the same in both scans. The algorithm searches the corresponding lines between scans by maximising the number of similar angle differences.

The algorithm steps are

1. Extract scans into lines, e.g. with split and merge and least squares fitting algorithms.

2. Calculate angles between the intersecting lines within one laser scan.

3. Save the angles into a matrix. For example element in the i:th row and k:th column, is the angle between i:th and k:th line on that laser scan. Thus one row in the matrix contains all the intersection angles for one line. This is also illustrated in figure 4.16 using simple three lines example. All the intersection angle values are converted between $[-\pi, \pi]$. One matrix is constructed for each scan.

4. Compare scan matrixes by finding the most similar matrix rows, i.e. most similar lines. Similarity is determined by number of corresponding angles, e.g. if 4 out of 5 elements in the rows are about the same, then there is four lines that intersected in the same angle and only one in different angle. This means that the algorithm is actually searching for corresponding lines from the scans, i.e. line pairs.

5. Calculate correlation ($\mathcal{C}$) between rows by weighting with the element correlations ($\mathcal{B}_{n,m}^{k,i}$), see equation 4.11. Small $\mathcal{C}$ value indicates good correlation between rows, i.e. rows are likely to represent the same line. The $\mathcal{C}_{k,i}$ is correlation between k:th line in scan1 and i:th line in scan2. $\mathcal{B}_{n,m}^{k,i}$ is element correlation between n:th element of k:th line and m:th element of i:th line in the matrix.

$$\mathcal{C}_{k,i} = \sum_{n=0}^{N} min(\mathcal{B}_{n,m}^{k,i}) \tag{4.11}$$

$$\mathcal{B}_{n,m}^{k,i} = \begin{cases} (\frac{\Delta\alpha_n^{scan1} - \Delta\alpha_m^{scan2}}{\alpha_{limit}})^2 & , \Delta\alpha_n^{scan1} - \Delta\alpha_m^{scan2} \leq \alpha_{limit} \\ 1 & , \Delta\alpha_n^{scan1} - \Delta\alpha_m^{scan2} > \alpha_{limit} \end{cases} \tag{4.12}$$

A line is associated with the line which gives the lowest correlation. The lines which are consisting of most points can choose their pairs first.

6. Calculate angle differences for the associated line pairs. For example, if the first row from the first scan is line in 34 degrees angle and the matching row from the second scan is in 45 degrees angle, the comparison results a angle correction of $45 - 34 = 11$ degrees.

7. Determine the correction angle using "voting criteria". Criteria affecting the decision are, in order of importance

   (a) There is one angle correction that is suggested more often.

   (b) Rows that suggested the angle had same number of points.

   (c) Correlation value given for the rows was high.

   For example if the first criterion is true, then second and third are not even evaluated.

Again the algorithm works well if the line extraction algorithm works well. In environment with distinct straight features, e.g. in a corridor, the algorithm gives reliable and accurate corrections. The algorithm is not very suitable to find real line pairs because it does not take into account the topological relationships, i.e. the left and the right walls in a straight corridor are usually seen as identical objects. If the algorithm could find the real line pairs, then it would be also possible to calculate the translation correction.

The algorithm is compared with sequential histogram correlation algorithm in the figure 4.17. The scan is segmented to lines, and displayed before and after the correction. The relevant difference is that line intersection algorithm corrects the angle based on best correlated lines, i.e. without discrete accuracy.

4.2.4 Location Correction with Evidence Grid Correlation

Location correction with evidence grid correlation returns the best correlating shift in x- and y-directions for two scans. Correction is done only for

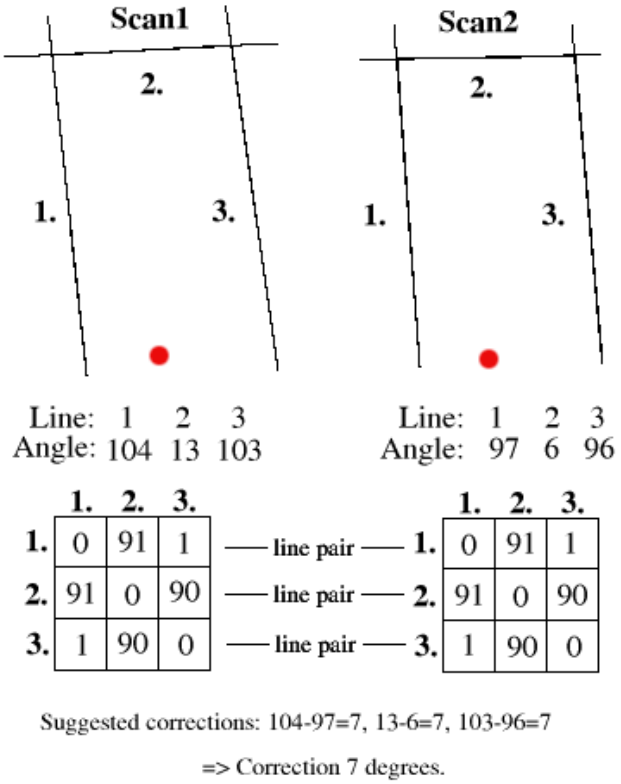

Figure 4.16: Line intersection angles are written in the matrix and corresponding lines are searched by comparing matrix rows. Note that 1. and 3. row are very similar and thus are only barely paired correctly.

the translation and it is assumed that the angle deviation between scans is already corrected.

The algorithm steps are

1. Scan coordinates are discretised in the search area (between some minimum and maximum coordinates), e.g. $x \in [-10, 10]$ and $y \in [0, 15]$.

2. Coordinates are sorted according to y-values using quicksort algorithm to speed up the search. The values in the y-direction are usually more spread and thus finding the right y-value narrows the distance search more than if searching in x-direction. This is because walls seen by the

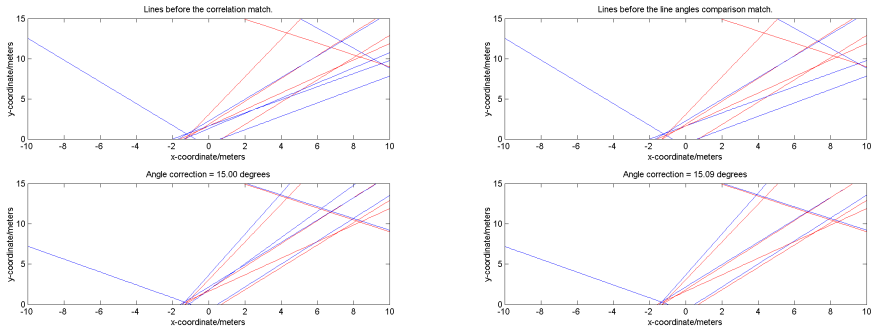

Figure 4.17: Sequential lines algorithm (left) and the line intersection based algorithm (right). Lines before and after line intersection correction.

laser scanner are usually parallel to direction of forward movement, i.e. in the direction of the y-axis.

3. Select the x and y search area, and the search step size. For example search area of 20cm times 20cm with 1cm step size could be selected.

4. For every discrete point from the first scan, search the nearest discrete point from the second scan. This requires a lot of computation power, although the quicksorting was done to ease this task. The quicksort limits the correlation algorithm search area into a smaller portion of the array, i.e. if we know that y-values are between 1.2 and 1.6 then we can jump directly to the right part of the sorted array. An index is used to keep a track with the search area in the sorted array, so that the next nearest point search can continue from where the last one ended. In the worst-case, all the y-values are within the search area which means that all the 361 scan points have to be compared with all 361 points of the other scan. In the best case the values are distributed so that search area is only one point, e.g. when the scans are almost identical.

5. The nearest point is estimated to be hit (0) or miss (1), or something between those two, e.g. 0.45 is a partial hit. The hit is defined as

follows:

$$Hit(d) = \begin{cases} \frac{d}{D} & , d <= D \\ 1 & , d > D \end{cases} \tag{4.13}$$

where d is distance from point and D is minimum distance which is counted to be as hit.

6. Sum all the hit values for each scan to calculate correlation value for the matched scans. Small correlation value indicates good match, i.e. largest number of hits found.

7. Return wanted number of best matching angles with associated correlation value. The returned correlation values can be used to estimate the goodness of the match.

The matching algorithm has difficulties when there are no details against which to match. For example a straight corridor with no objects and with an upward stairs at the end is very difficult. The best results are gained when translations between several scans are calculated to verify the match, as described earlier in this chapter.

Because the complexity of this algorithm is $O(N^2)$, it is not possible to use very large search areas in real-time. This problem can be partially solved by using iterative calculations. This means that we first calculate a solution with large step size and search area, which is then used to limit the next search area with smaller step size.

An example of evidence grid correlation result is shown in figure 4.18. The figure shows correlation values in each discrete correlation point in the xy-plane. A low value means a good correlation. The evidence distribution in the figure is typical for a corridor, because the scan has a good match in x-direction due to corridor walls, but in the y-direction the correlation forms practically a valley of uncertainty.

The algorithm returns also an ellipse estimate fitted to the correlation grid if requested, as shown in figure 4.19. The uncertainty estimation is impor-

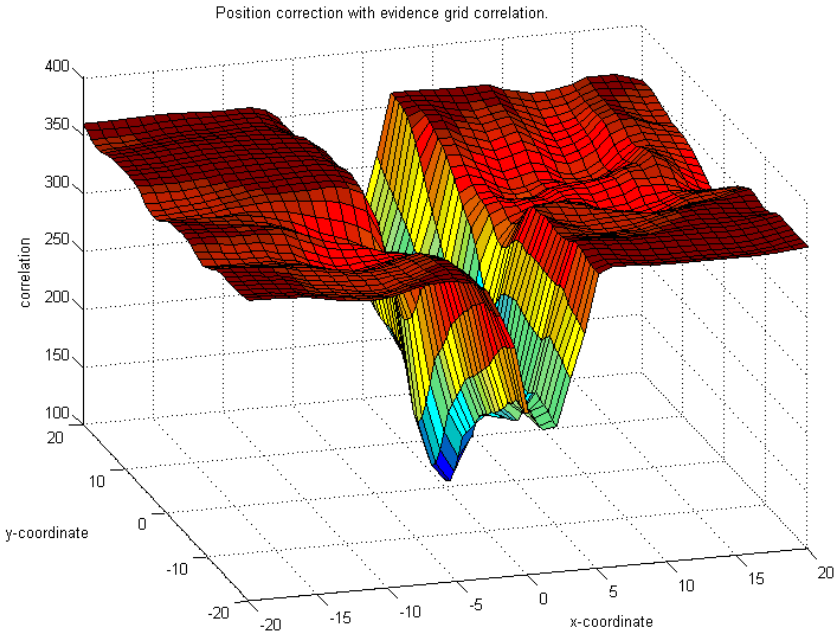

Figure 4.18: Evidence grid correlation. The solution is quite limited in x-direction, but the y-direction looks more like a canyon without any distinct minimums.

tant when using the method for a priori localisation or SLAM. In the figure the position uncertainty grows in direction of the corridor while travelling through it. If the uncertainty of current location is known, then e.g. a map localisation algorithm is able to localise the entity in a corridor intersection where more features is available for the match.

4.3 Localisation with A Priori Map

The laser odometry algorithms can also be used for a priori map localisation. The idea is to use the dead reckoning estimates to determine the current position in the map. In this estimated map place a virtual scan is calculated from a priori given lines map. The distances of the virtual scan measure-

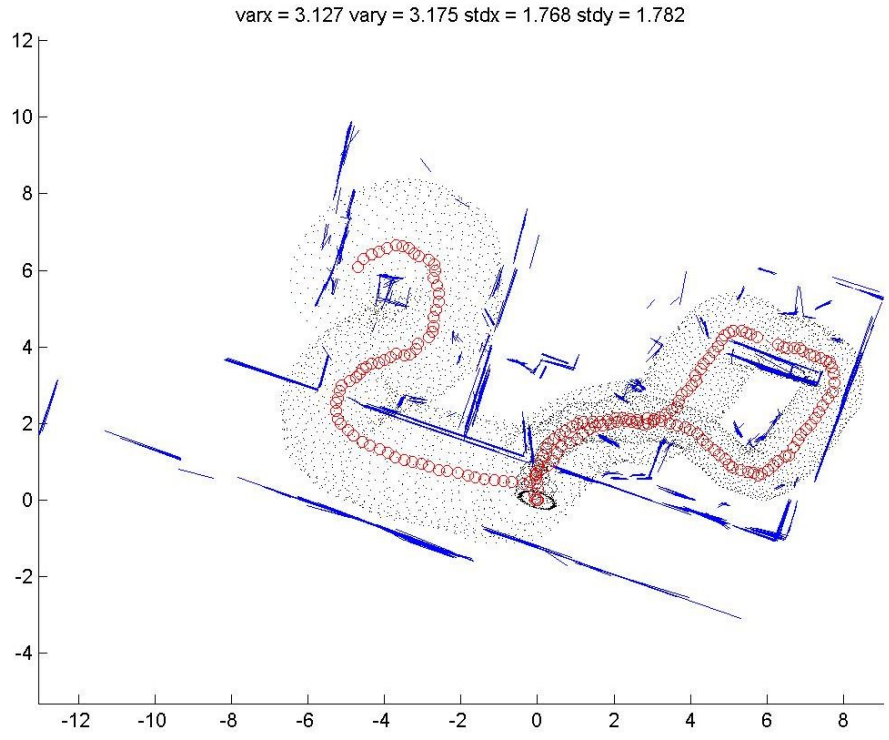

Figure 4.19: The ellipses fitted to evidence grid correlation result can be used to estimate the accumulating error in the walked path. In the room environment the distribution is almost a circle.

ment are obtained by finding the closest intersections between the map lines and the line drawn from the estimated position to the scan direction, as shown in figure 4.20. This virtual scan is finally matched against the current measurement which returns a correction to the place on the map.

The virtual scan matching is done every time the reference scan is changed by the Laser Dead Reckoning (LDR) algorithm. If the correlation value between the virtual scan and the measurement is good, then the position is corrected on the map. If not, the algorithm uses the LDR localisation results. If the measurement is noisy, the algorithm does not try to correct the position, and thus allows the entity to enter outside the map. In the PeLoTe project it is expected that some kind of a priori map of the building exists, but most

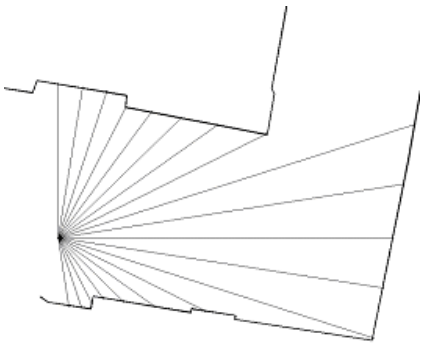

Figure 4.20: An example of virtual scan.

probably this map is somehow incorrect. This means that we can not rely blindly on the virtual scan matching results, and thus we have to be able to estimate if the map correction is usable or not.

The algorithm does not support global localisation, and therefore, the initial position should be known. Also, if the position error grows too large, so that there is not enough overlap between the virtual scan and the measurement, the algorithm cannot recover. This could happen if a too long distance is travelled outside from the map.

4.4 SLAM Algorithm

One SLAM algorithm was also implemented to test the PeNa without a priori map, and especially, to operate with partial maps. The SLAM algorithm checks if the number of outliers, i.e. points not found when taking virtual scan, is above some threshold. If the value is below the threshold, the current map is then good enough and no additional map structure is needed. If there is more outliers than the threshold, the current scan is extracted into lines and placed on the map.

The advantage of this kind of SLAM algorithm, compared to the dead reckoning, is that the position error does not accumulate if the entity stays in the mapped area. This is because the SLAM algorithm can correct the position

on the map few times before it has to add there new lines. Also with small side paths, e.g. when visiting rooms, the position error does not accumulate as it would with dead reckoning.

Although, the SLAM map does not necessarily look like the real environment map, it reflects the environment as the sensors sees it. This means that the SLAM map is suitable to keep the entity localised within the map, although, it might be more or less useless to other entities.

One severe problem of this kind of incremental map building algorithm is that it can not cope from most of the loop closing problems. This is because during the loop the position uncertainty is growing and when closing the loop, there is a cap between starting and closing positions, as shown in figure 4.21. After this the map will be inconsistent for even to the user entity.

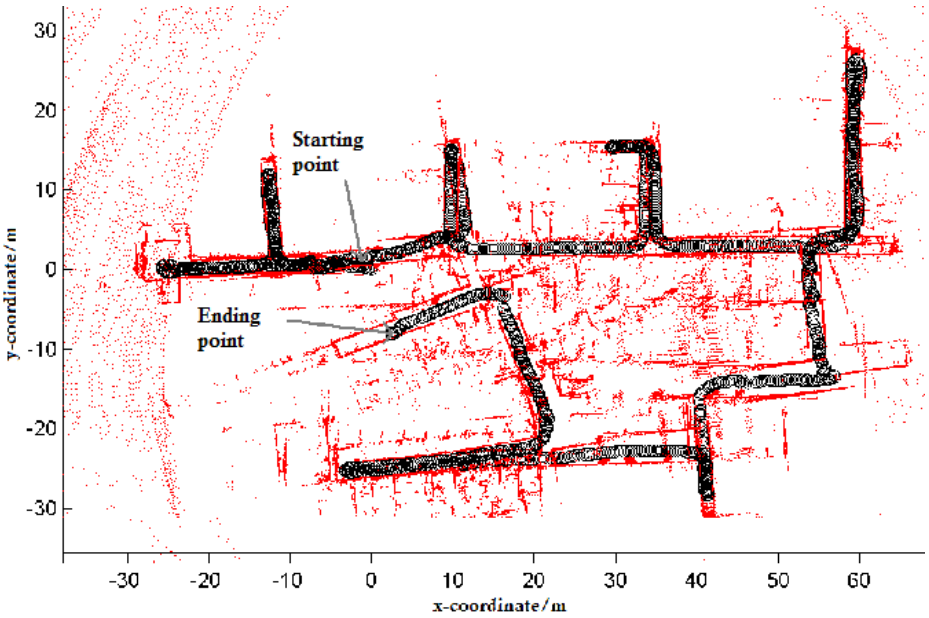

Figure 4.21: Loop closing problem. The starting and closing position are not aligned.

Chapter 5

Results

This chapter presents the PeNa system test results. The emphasis is on the PeNa localisation capabilities, but also suitability for a real world application is demonstrated. The first part of this chapter presents PeNa localisation results achieved using algorithms developed and described in this study. The last part of this chapter presents test results from the PeLoTe project demonstration with PeNa configuration which also includes algorithms not developed in this study.

It should be noted that it took very long to determine the PeNa localisation algorithm parameters, compared for example to robot algorithm parameter calibration. This is because the prediction of human behaviour requires the algorithms to be very robust for various situations that are hard to notice before the actual user tests. For example variations in the pitch and roll when picking objects, and different walking styles, required modifications to the algorithms to add robustness. The modified parameters are the algorithm search areas, the algorithm step sizes, the scan matching acceptance criteria, and the system real-time performance.

5.1 Localisation and SLAM Tests

Localisation is an essential part of the PeNa functionality, and also a major part of this study. For this reason, four localisation tests with fully integrated PeNa system are presented: a simple back and forth walk in a straight corridor, a corridor walk with two room visits, a walk through all automation laboratory corridors, and a laboratory walk with 23 laboratory room visits. The tests were selected so that they would cover most of the typical indoor situations, i.e. various kinds of corridor and room environments. One test is also presented to demonstrate the PeNa SLAM capabilities.

The used PeNa configuration is based on algorithms which are all described in the chapter 4, i.e. developed in this study. The used algorithm settings are listed in details in table 5.1. The Dead Reckoning (DR) functionality uses the SiLMU step length measurements and the Kalman filtered heading measurements, as described in the section 4.1. The Laser Dead Reckoning (LDR) is calculated using the sequential lines histogram correlation algorithm and the evidence grid correlation algorithm. These algorithms are described, respectively, in the section 4.2.1 and 4.2.4. The map based localisation is done using the algorithm described in section 4.3.

All the tests are analysed by comparing some calculated key figures. Three path length estimates are calculated using the DR, the LDR, and the map matching results. The most useful variable in the result evaluation is the final location error, which is defined as the distance between the starting and ending points of the walk. Also the relative location error, the distance error divided with real path length, is calculated. All four tests were conducted three times to gain higher reliability to the results.

In the table 5.2 is presented the results from the straight corridor walk test. The walked path was 55 metres back and forth in a straight corridor, i.e. 110 metres in total. The test walks are illustrated in figure C.1. The DR result is quite noisy on all walks and the localisation result is significantly enhanced with the LDR. The LDR contains still errors especially on the heading, which

Table 5.1: The PeNa algorithm settings used in the tests.

Parameter	Value
SiLMU initial guess	Yes, used.
LDR location search area	0.32m
LDR location search step size	0.02m
LDR angle search area	$0.5rad \approx 28.6°$
LDR angle search area step size	$0.01rad \approx 0.6°$
Map location search area	0.64m
Map location search step size	0.02m
Map angle search area	$0.5rad \approx 28.6°$
Map angle search area step size	$0.01rad \approx 0.6°$

is further corrected by the map matching algorithm.

During the first and the third walk the map matching made a false match in the halfway of the corridor, where is essentially least features against which to match. The error can be seen as a smaller map length, and of course, from the difference of the starting and ending points of the path. This kind of error can be due to a wrong initial step length estimation given by the DR.

The lack of features was a serious problem for the map matching also during several other tests. A straight corridor, without doors or columns, is impossible task for scan correlation algorithms because there is nothing against which to correlate. One solution is to estimate the location match error, e.g. with an ellipse, and calculate the correction later with larger search area when more details are present. This situation is shown in figure 4.19. The problem with this kind of estimation is that the computational complexity changes along with the environment making thus the resource allocation very difficult.

The second test was a shorter corridor walk with two room visits. The results are shown in table 5.3. The DR localisation is again pretty inaccurate, but it

Table 5.2: The straight corridor walk results. The walks are shown in figure C.1.

Parameter	Walk A	Walk B	Walk C
DR path length	95m	97m	111m
LDR path length	109m	110m	109m
Map path length	108m	111m	109m
Time Elapsed	125s	108s	118s
DR location error	2.7m	4.9m	1.5m
Relative DR error	2.9%	4.6%	1.3%
LDR location error	4.6m	5.4m	4.9m
Relative LDR error	3.2%	5.0%	4.4%
Map location error	2.8m	0.8m	1.5m
Relative map error	2.5%	≈0%	1.3%

is corrected relatively well by the LDR. The LDR results are so good that it is easy task for the map matching to localise the entity. The most significant difference compared to the first test is that the environment contains many features against which to match. In this kind of environment the role of DR is not so important because LDR and map matching perform very well. The second test results are visualised in figure C.2.

In the third test all the Automation laboratory corridors were walked through, with a total path length about 290 metres. The results are shown in table 5.4. The LDR is again able to correct the noisy DR results, and in all cases, significantly decrease the amount of localisation error. There exists again the same long corridors with poor level of details against which to match, but because several smaller corridors are visited relatively often, the error in the map matching does not accumulate to level from which it could not recover. The third test walks are visualised in figure C.3.

The last test was the most extensive one, containing 23 room visits. The results are shown in the table 5.5. The DR performs relatively well compared

Table 5.3: The corridor walks with two room visits, see figure C.2.

Parameter	Walk A	Walk B	Walk C
DR path length	84m	89m	84m
LDR path length	79m	76m	77m
Map path length	88m	79m	80m
Time Elapsed	120s	111s	129s
DR location error	5.8m	7.5m	8.9m
Relative DR error	6.6%	9.4%	11.1%
LDR location error	1.9m	1.8m	0.6m
Relative LDR error	2.2%	2.3%	0.8%
Map location error	< 0.5m	< 0.5m	< 0.5m
Relative map error	$\approx$0%	$\approx$0%	$\approx$0%

Table 5.4: The walk through all automation laboratory corridors, see figure C.3.

Parameter	Walk A	Walk B	Walk C
DR path length	250m	213m	292m
LDR path length	285m	285m	283m
Map path length	292m	289m	291m
Time Elapsed	286s	296s	283s
DR location error	7.8m	9.2m	16.6.m
Relative DR error	2.7%	3.1%	5.7%
LDR location error	1.6m	5.7m	3.9m
Relative LDR error	0.6%	1.9%	1.3%
Map location error	< 0.5m	< 0.5m	< 0.5m
Relative map error	$\approx$0%	$\approx$0%	$\approx$0%

to LDR. This is mostly due to the compass measurements that provide absolute reference frame for the angle. This means that the error might be large in small scale because the relative low accuracy of compass, but in the long run, the compass points in average to the right direction, and thus the error does not accumulate as fast as it would if based only on the relative accuracy. This is thanks to Kalman filter, which can combine the good gyroscope relative accuracy with absolute angle accuracy of the compass.

Table 5.5: The laboratory walk with 23 rooms visited, see figure C.4.

Parameter	Walk A	Walk B	Walk C
DR path length	309m	299m	291m
LDR path length	309m	299m	291m
Map path length	309m	299m	291m
Time Elapsed	428s	387s	408s
DR location error	7.88m	4.21m	0.71m
Relative DR error	2.6%	1.1%	0.2%
LDR location error	12.80m	28.47m	7.30m
Relative LDR error	4.1%	7.3%	2.5%
Map location error	0.15m	0.10m	0.10m
Relative map error	≈0%	≈0%	≈0%

The LDR low performance can be also explained with errors made by the angle correction algorithm. One badly matched angle messes the heading permanently because Kalman filtered compass estimate is used only for the initial guess that is relative to reference scan. Many of the visited rooms do not contain distinct walls, or they have symmetrical objects, which means difficulties for the angle matching algorithm. Nevertheless, the LDR accuracy is more than enough for map matching, which performs almost perfectly with all walks in this test. The map matching performs well again because there are enough details against which to match. The last test is visualised in figure C.4.

A simple SLAM test was also done by using algorithm described in the section 4.4. The SLAM test included most of the Automation laboratory corridors and also one room. The test results are shown in the table 5.7 and in figure 5.1. The accuracy of the map is very good, but because no line merging was done, the number of lines (281) is relatively high compared to the level of detail.

The SLAM algorithm works pretty reliably in non-cyclical environments, and is capable to localise using the built map even if the map would not be geometrically perfect. Few tests were done with about 100 metre loops, but because the map is build incrementally and there is no algorithm to handle the loop closing, the loop fails to close and produced an inconsistent map in most of the cases.

Table 5.6: The laboratory corridor walk with the SLAM enabled PeNa.

Parameter	Value
Total Path	248m
Time Elapsed	196s
End-Start Error	0.1m
Relative Error	≈0%
Heading error	≈0%

5.2 Case Experiment - Fire Rescue Scenario

The PeNa was tested with the PeLoTe system during the final experiment in Würzburg between 20-21 November, 2004. In the experiment the fireman had to search through a rescue area and rescue all human victims to the nearest exit. The rescue area was assumed to be mostly dark because of the smoke, which was simulated using a blanket. Thus the fireman had to rely on the sense of touch, and in PeLoTe teams' case, to the information provided by the PeNa equipment.

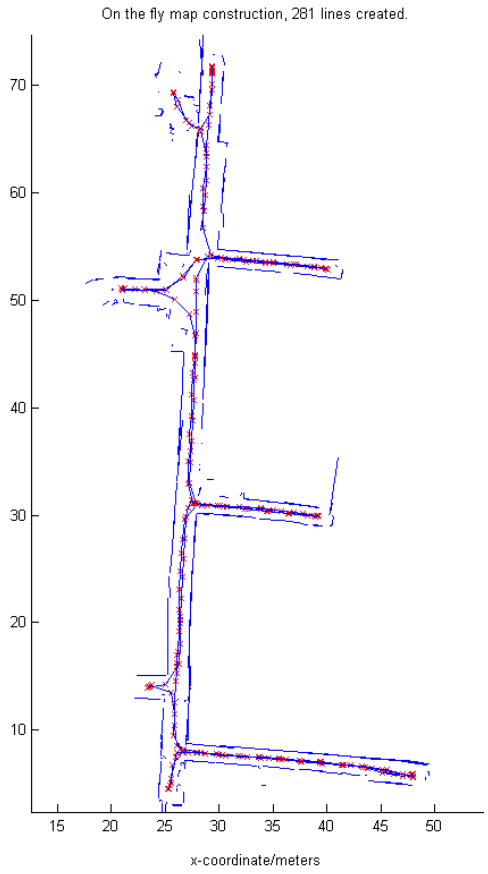

Figure 5.1: The SLAM enabled PeNa was used to map most of the corridors.

The experiment results are shown in table 5.7. In half of the teams using the PeNa system, the position was not lost at all during the whole experiment, see e.g. figure 5.2, but there was also three PeNa users that had to correct their position on the map during the experiment. The errors occurred in places where the environment correspondence with map was especially weak. This happened typically when some walls were missing from the map or when some dynamical objects, e.g. victims, stayed tens of seconds in front of the sensors.

Table 5.7: The final experiment in Würzburg between 20-21.11.2004. 6 teams carried out the same rescue mission.

Team	Path length	Total time	Position corrections
1	313m	1440s	2
2	280m	1320s	0
3	264m	1500s	0
4	246m	1380s	0
5	320m	1740s	3
6	204m	1680s	1

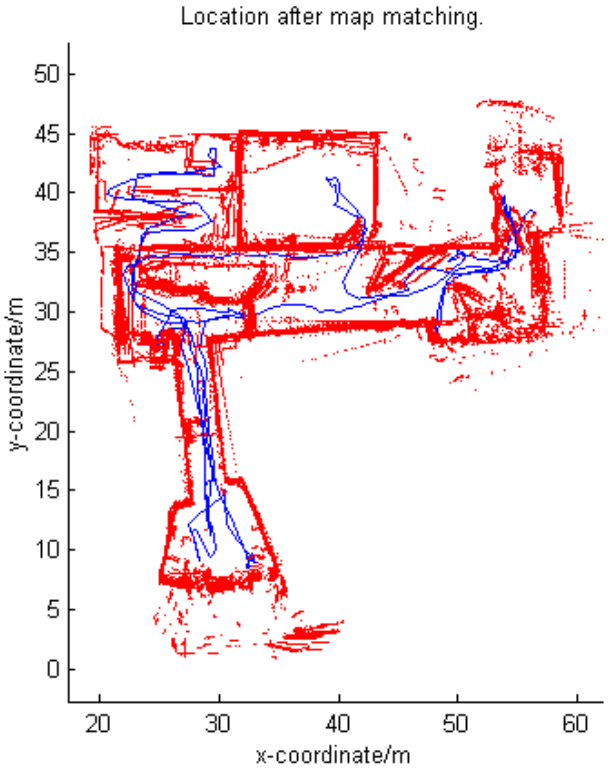

Figure 5.2: The localisation data seen by the 2nd PeLoTe team during the final experiment in Würzburg.

The fire-rescue mission was also carried out by six teams without the PeLoTe

equipment. The most visible difference between the traditional and the PeLoTe teams was that the traditional teams navigated using walls, while the PeNa users navigated only using the PeNa sensor information. Most of the PeNa users commented that the laser scanner was found to be useful for local area navigation. This showed clearly that the PeNa is capable to provide useful information, which can make the rescue situation easier.

The fire-rescue scenario was also an end user test. None of the PeNa users had used the equipment before, but with 20 minutes training, the users were able to navigate in dark areas and find victims with better accuracy than the traditional teams. This demonstrated clearly that a high technology equipment does not necessarily mean extensive training and usability problems. The PeLoTe final experiment and end user test results are described in details in [85].

Chapter 6

Summary and Conclusions

This master's thesis presented a novel human indoor localisation system capable to operate without any external infrastructures. The localisation system, named as Personal Navigation system (PeNa), is based on dead reckoning and map localisation algorithms developed in this master's thesis. The average dead reckoning error was about 5% of the distance travelled which provides almost an error-free localisation when associated with a priori map.

In the first part of this master's thesis the existing human localisation systems were reviewed. Based on the review, the PeNa system is the first developed laser scanner based human indoor localisation system. The PeNa system was demonstrated to be also robust enough for rescue environment localisation when using a priori map, which is also unique property among the reviewed systems. A detailed description of the PeNa software and hardware architecture was presented after the review of related work.

The latter part of this master's thesis presented the developed indoor dead reckoning and map localisation algorithms. Three angle correction algorithms were presented: a histogram correlation with sequential lines, a histogram correlation with segmented lines, and a line intersection based algorithm. Also an evidence grid correlation based location correction algorithm and a virtual scan based map correction algorithm were presented.

The performances of these angle correction algorithms were found to be almost identical, but the sequential lines algorithm was selected because it was the only one not requiring any line extraction from the scans, i.e. it could use raw laser data, and could provide reliable estimates of the angle correction accuracy. The evidence grid correlation can also return matching accuracy estimate which is essential when the algorithm results are further used, e.g. by the map matching.

Although the PeNa system was successfully tested by the only briefly trained end users in the fire-rescue scenario, the size of the PeNa equipment would have to be smaller in size and in weight for the system to be practical for real use. The laser scanner and the power systems present the biggest challenges, but even with today's technologies, the system could be fit to about helmet size. The price would still be pretty high, i.e. several thousands of euros.

The further development of the PeNa system could be continued in similar open source projects that are now used to develop the robot control software and algorithms, e.g. EU initiated open source robot control software project Orocos [86] and the robot/sensor device interface project Player [87]. The algorithms and sensors with robots and humans are very similar, and so the development could be started based on the existing robot projects. In any case, accurate human indoor localisation, just like intelligent service robots, is only a matter of time.

References

[1] Mikko Elomaa. Ultrasonic sensor systems for human localisation. Master's thesis, Helsinki University of Technology, Automation Technology Laboratory, March 2004.

[2] Saarinen Jari, Mazl Roman, Ernest Petr, Suomela Jussi, and Preucil Libor. Sensors and methods for human dead reckoning. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, Amsterdam, Netherlands, March 10-13 2004.

[3] Jari Saarinen, Roman Mazl, Miroslav Kulich, Jussi Suomela, Libor Preucil, and Aarne Halme. Methods for personal localisation and mapping. In *Proceedings of the 5th IFAC symposium on Intelligent Autonomous Vehicles, IAV2004*, Lisbon, Portugal, July 5 - 7, 2004.

[4] Jari Saarinen, Jussi Suomela, Seppo Heikkilä, Mikko Elomaa, and Aarne Halme. Personal navigation system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai International Center, Sendai, Japan, September 28 - October 2, 2004.

[5] Information Society Technologies (IST). *IST homepage and project descriptions*. http://www.cordis.lu/ist/home.html Last visited 17.02.2005.

[6] P.Y. Gillieron and B. Merminod. Personal navigation system for indoor applications. In *11th International Association of Institutes of Navigation (IAIN) World Congress*, October 21 - 24, 2003.

[7] Seon-Woo Lee and Kenji Mase. A personal indoor navigation system using wearable sensors. In *Proceedings of the ISMR 2001 (2nd International Symposium on Mixed Reality)*, pages 147–148, Yokohama, Japan, March 14-5 2001.

[8] R. Jirawimut, M. A. Shah, P. Ptasinski, F. Cecelja, and W. Balachandran. Integrated dgps and dead reckoning for a pedestrian navigation system in signal blocked environments. In *ION GPS 2000, 13th International Technical Meeting of the Satellite Division of the Institute of Navigation,Session D4 - GPS - Reference and Information Fusion*, September 19-22, 2000.

[9] H. Aoki, B. Schiele, and A. Pentland. Realtime personal positioning system for a wearable computer. In *Digest of Papers. The Third International Symposium on Wearable Computers*, pages 37–43, 1999.

[10] Yusuke Konishi and Ryosuke Shibasaki. Development of an autonomous personal positioning system. In *Proceedings of the Asia GIS 2001*, 2001.

[11] Elena Vildjiounaite, Esko-Juhani Malm, Jouni Kaartinen, and Petteri Alahuhta. Location estimation indoors by means of small computing power devices, accelerometers, magnetic sensors, and map knowledge. In *Proceedings of the First International Conference on Pervasive Computing*, pages 211–224. Springer-Verlag, 2002.

[12] Esa Tuulari and Arto Ylisaukko-oja. Soapbox: A platform for ubiquitous computing research and applications. In *Proceedings of the First International Conference on Pervasive Computing*, pages 125–138. Springer-Verlag, 2002.

[13] Jussi Collin, Oleg Mezentsev, and Gèrard Lachapelle. Indoor positioning system using accelerometry and high accuracy heading sensors. In *Proceedings of GPS 2003 (Session C3, Portland, OR, 9-12 September)*, 2003.

[14] C. Randell, C. Djiallis, and H. Muller. Personal position measurement using dead reckoning. In *Proceedings of Seventh IEEE International Symposium on Wearable Computers*, pages 166–173, 2003.

[15] Vectronix. *Pedestrian Navigation Module - Reliable Navigation GPS-Denied Environments.* http://www.vectronix.ch/files/PNM_en.pdf Last visited 08.12.2004.

[16] Q. Ladetto, J. van Seeters, S. Sokolowski, Z. Sagan, and B. Merminod. Digital magnetic compass and gyroscope for dismounted soldier position and navigation, October 2002.

[17] Applanix. *POS/LS, Position and Orientation System for Land Survey.* http://www.applanix.com/pdf/POS_LS_Technical_Note.pdf Last visited 08.12.2004.

[18] C. T. Judd. A personal dead reckoning module. Technical report, in Institute of Navigationś ION 97 Conference, Cambridge, October 1997.

[19] R. B. Marth Sr, R. Levi, I. N. Durboraw III, and K. Beam. The integrated navigation capability for the force xxi land warrior. In *Position Location and Navigation Symposium, IEEE 1998*, pages 193–200, 1998.

[20] Point Research Corporation. *Brochure about the Dead Reckoning Module.* http://www.pointresearch.com/PDFs/DRM3.pdf Last visited 06.12.2004.

[21] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, London, December 1999.

[22] Roy Want, Andy Hopper, Veronica Falco, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[23] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM Press, 2000.

[24] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 4(5):42–47, 1997.

[25] L. M. Ni, Yunhao Liu, Yiu Cho Lau, and A. P. Patil. Landmarc: indoor location sensing using active rfid. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 407–415, 2003.

[26] Lei Fang, Panos J. Antsaklis, Luis Montestruque, Brett McMickell, Michael Lemmon, Yashan Sun, Hui Fang, Ioannis Koutroulis, Martin Haenggi, Min Xie, and Xiaojuan Xie. A wireless dead reckoning pedestrian tracking system. Technical report, June 2004. Department of Electrical Engineering, University of Notre Dame, IN 46556, USA.

[27] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[28] D. Broughton. Gnss - a users' perspective in 2010. In *Proceedings Elmar 2004. 46th International Symposium if Electronics in Marine, 2004*, pages 1–4, 2004.

[29] J. A. Farrell and T. Givargis. Experimental differential gps reference station evaluation. In *Proceedings of the 1999 American Control Conference*, volume 5, pages 3645–3649 vol.5, 1999.

[30] Benefon Oyj. *Benefon Esc! personal navigation phone manual.* http://www.benefon.com/pdf/manuals/esc/nt/english.pdf Last visited 05.02.2005.

[31] Statistics Finland and Antti Raimo. Personal navigation navi programme 2000-2002 in finland. In *Proceedings of Conference of European Statisticians*.

[32] Space Systems Finland. *NAVIndoor pseudolite technology*. http://www.ssf.fi/products/data/attachments/NAVIndoor.pdf Last visited 05.02.2005.

[33] The European Parliament and The Council of the European Union. Directive 2002/22/ec of the european parliament and of the council. *Official Journal of the European Communities*, 7th March 2002.

[34] C. Drane, M. Macnaughtan, and C. Scott. Positioning gsm telephones. *Communications Magazine, IEEE*, 36(4):46–54, 59, 1998.

[35] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, volume 2, pages 775–784 vol.2, 2000.

[36] Ekahau. *The Ekahau Positioning Engine datasheet*. http://www.ekahau.com/pdf/EPE_3.0_datasheet.pdf Last visited 05.02.2005.

[37] K. Thapa and S. Case. Indoor positioning service for bluetooth ad hoc networks. In *Midwest Instruction and Computing Symposium(MICS)*, 2003.

[38] UWUERZ Certicon ARS Automation Technology Laboratory of Helsinki University of Technology (HUT), CTU. The research analysis for methods and algorithms for personal map building and localisation. Technical report. IST-2001-38873, PeLoTe project deliverable 2.3a. May 2004.

[39] Feng Lu. *Shape Registration Using Optimization For Mobile Robot Navigation.* PhD thesis, Department of Computer Science, University of Toronto, 1995.

[40] Weiss and E. V. Puttkamer. A map based on laser scans without geometric interpretation. *Intelligent Autonomous Systems - 4 (IAS-4), Karlsruhe, Germany*, March 27-30, 1995.

[41] G. Weiss, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS 1994)*, volume 1, pages 595–601 vol.1, 1994.

[42] R. Mazl and L. Preucil. Building a 2d environment map from laser range-finder data. In *Proceedings of the IEEE Intelligent Vehicles Symposium, 2000. IV 2000.*, pages 290–295, 2000.

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

[44] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2):119–152, 1994.

[45] R. Madhavan, M. W. M. G. Dissanayake, and H. F. Durrant-Whyte. Autonomous underground navigation of an lhd using a combined icp-ekf approach. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3703–3708 vol.4, 1998.

[46] Feng Lu and E. E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, pages 935–938, 1994.

[47] O. Bengtsson and A.J Baerveldt. Localization in changing environments by matching laser range scans. In *Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot 1999)*, pages 169–176, 1999.

[48] J. S Gutmann and C. Schlegel. Amos: comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robot*, pages 61–67, 1996.

[49] Iwan Ulrich and Illah R. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1023–1029, 2000.

[50] Michael Jenkin and Gregory Dudek. Computational principals of mobile robotics. *Cambridge University Press, Cambridge, England*, pages 214–227, 2000.

[51] I. J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *Robotics and Automation, IEEE Transactions on*, 7(2):193–204, 1991.

[52] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121, 1985.

[53] A. C. Schultz and W. Adams. Continuous localization using evidence grids. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 2833–2839 vol.4, 1998.

[54] Tim Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, Australian Center for Field Robotics, Department of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, August 2002.

[55] B. Schiele and J. L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1628–1634 vol.2, 1994.

[56] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI Mag.*, 9(2):61–74, 1988.

[57] B. Kuipers and Y.T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 1991.

[58] Illah Nourbakhsh. Dervish: an office-navigating robot. *Artificial intelligence and mobile robots: case studies of successful robot systems*, pages 73–90, 1998.

[59] Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1080–1087, 1995.

[60] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proc. of the National Conference on Artificial Intelligence*, 1998.

[61] Dieter Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation.* PhD thesis, Institute of Computer Science III, University of Bonn, Germany, December 1998.

[62] Kurt Konolige and Ken Chou. Markov localization using correlation. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1154–1159. Morgan Kaufmann Publishers Inc., 1999.

[63] W. Burgard, A. Derr, D. Fox, and A. B. Cremers. Integrating global position estimation and position tracking for mobile robots: the dynamic

markov localization approach. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 730–735 vol.2, 1998.

[64] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328 vol.2, 1999.

[65] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dallaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[66] Rudy Negenborn. Robot localization and kalman filters. Master's thesis, Institute of Information and Computing Sciences, Utrecht University, Germany, September 1, 2003.

[67] Hans Jacob S. Feder, John J. Leonard, and Christopher M. Smith. Adaptive mobile robot navigation and mapping. In *International Journal of Robotics Research, Special Issue on Field and Service Robotics*, July 1999.

[68] Tardos J., Neira J., Newman P., and Leonard J. Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research*, 21(4):311–330, 2002.

[69] E. G. Araujo and R. A. Grupen. Feature detection and identification using a sonar-array. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1584–1589 vol.2, 1998.

[70] Andrew R. Golding and Neal Lesh. Indoor navigation using a diverse set of cheap, wearable sensors. In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, pages 29–36. IEEE Computer Society, 1999.

[71] Randall Smith, Matthew Self, and Peter Cheeseman. A stochastic map for uncertain spatial relationships. In *on The fourth international symposium*, pages 467–474. MIT Press, 1988.

[72] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.

[73] S. Thrun. *Robotic mapping: A Survey.* Morgan Kaufmann Publishers, San Francisco, pp. 1-35., 2003.

[74] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: a factored solution to the simultaneous localization and mapping problem. In *Eighteenth national conference on Artificial intelligence*, pages 593–598. American Association for Artificial Intelligence, 2002.

[75] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2003)*, volume 2, pages 1985–1991 vol.2, 2003.

[76] M Montemerlo, S Thrun, D Koller, and B Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 2003.

[77] Paul Newman and Kin Ho. Slam - loop closing with visually salient features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 18-22 April, 2005.

[78] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Mach. Learn.*, 31(1-3):29–53, 1998.

[79] Wolfram Burgard, Dieter Fox, Hauke Jans, Christian Matenar, and Sebastian Thrun. Sonar-based mapping of large-scale mobile robot environments using em. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 67–76. Morgan Kaufmann Publishers Inc., 1999.

[80] J. S Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 1999)*, pages 318–325, 1999.

[81] A. I. Eliazar and R. Parr. Dp-slam: Fast, robust simulataneous localization and mapping without predetermined landmarks. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, 2003.

[82] A. I. Eliazar and R. Parr. Dp-slam 2.0. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2004)*, volume 2, pages 1314–1320 Vol.2, 2004.

[83] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2002)*, volume 1, pages 406–411 vol.1, 2002.

[84] MicroStrain. *3DM-G User Manual.* http://www.microstrain.com/usermanuals/3DMG_usermanual.pdf Last visited 12.12.2004.

[85] HUT Certicon ARS Julius Maximilian University of Würzburg (UWUERZ), CTU. End-user testing. Technical report. IST-2001-38873, PeLoTe project deliverable 6.2b. February 2005.

[86] Orocos project. *Free Software project for robot control.* http://www.orocos.org/ Last visited 17.02.2005.

[87] The Player/Stage Project. *Open Source development tools for robot and sensor applications.* http://playerstage.sourceforge.net/ Last visited 17.02.2005.

[88] R. Stirling, J. Collin, K. Fyfe, and G. Lachapelle. An innovative shoe-mounted pedestrian navigation system. In *Proceedings of GNSS 2003, The European navigation Conference Session F3 on Pedestrian Navigation*, page 15 pages, 22-25 April 2003.

[89] R. Jirawimut, S. Prakoonwit, F. Cecelja, and W. Balachandran. In *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, IMTC 2002.*

[90] Allen Ka Lun Miu. Design and implementation of an indoor mobile navigation system. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2002.

Appendix A

PeNa Software Class Diagrams

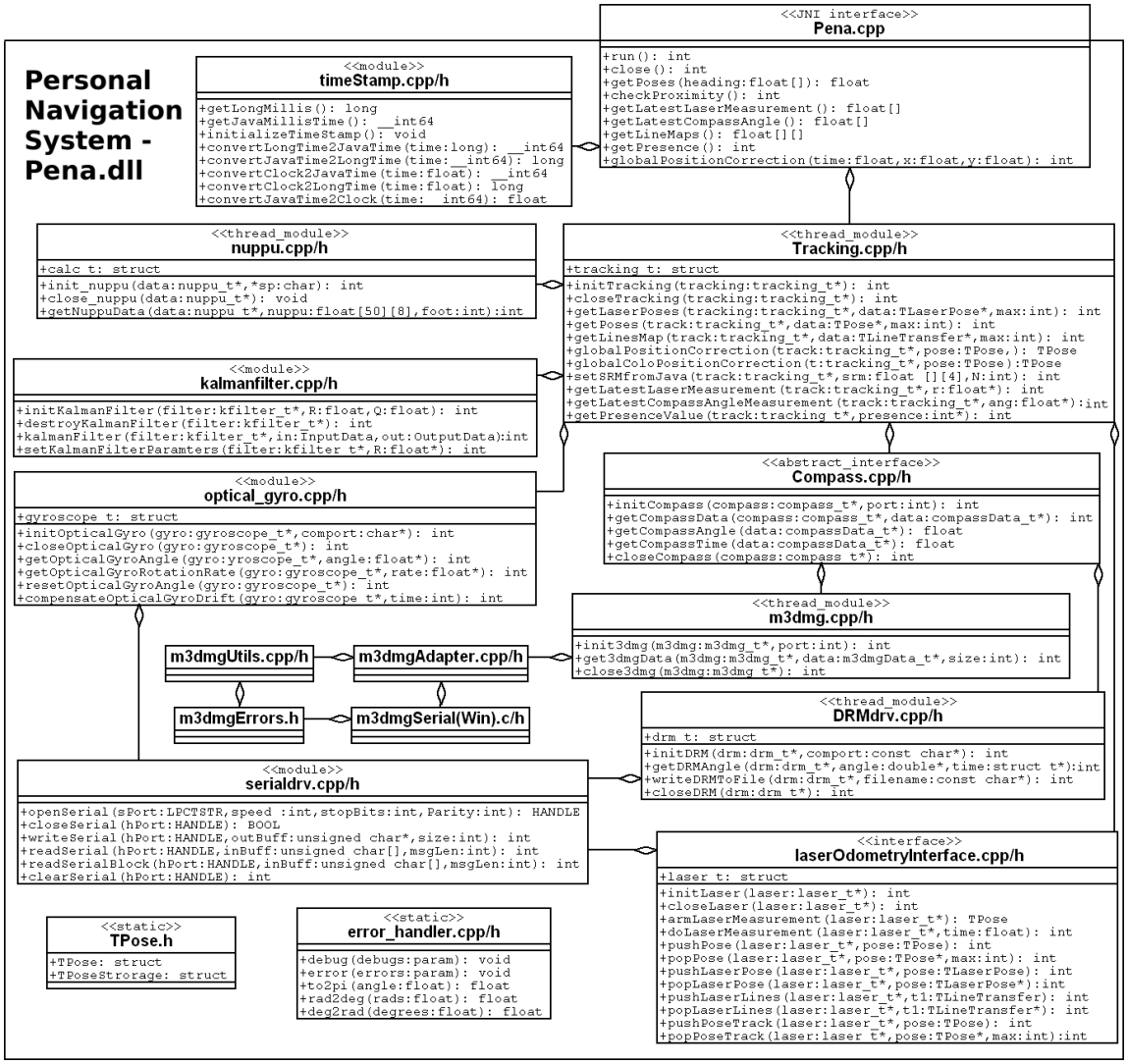

Figure A.1: The C/C++ code modules contained by the Pena.dll. Note that LaserOdometryInterface.cpp/h contains the laser scan matching and mapping modules. This module is shown in details in figure A.2.

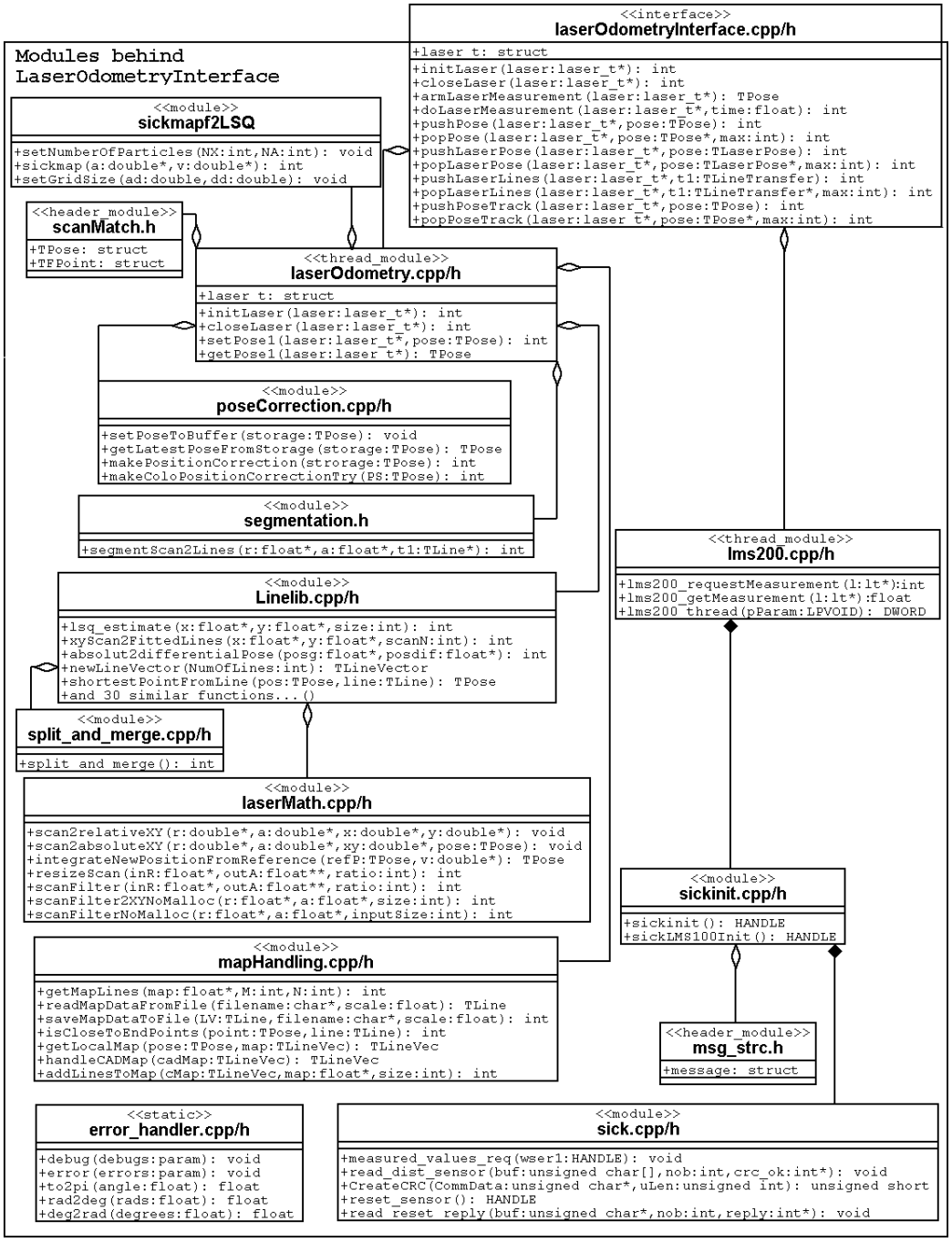

Figure A.2: The laserOdometryInterface class contains the laser drivers, the laser dead reckoning algorithms and the map matching algorithms.

Appendix B

Angle Correction Algorithm Comparison

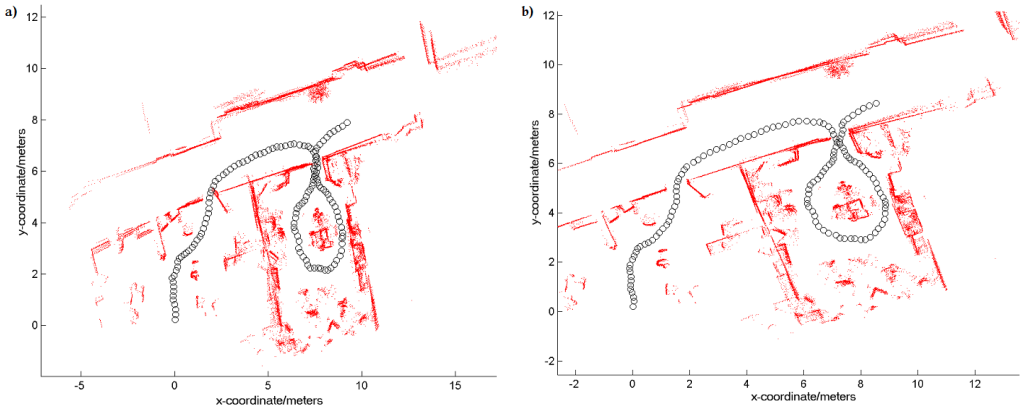

Figure B.1: The a) segmented lines and b) sequential lines algorithms can both maintain the angle in a office room equally well.

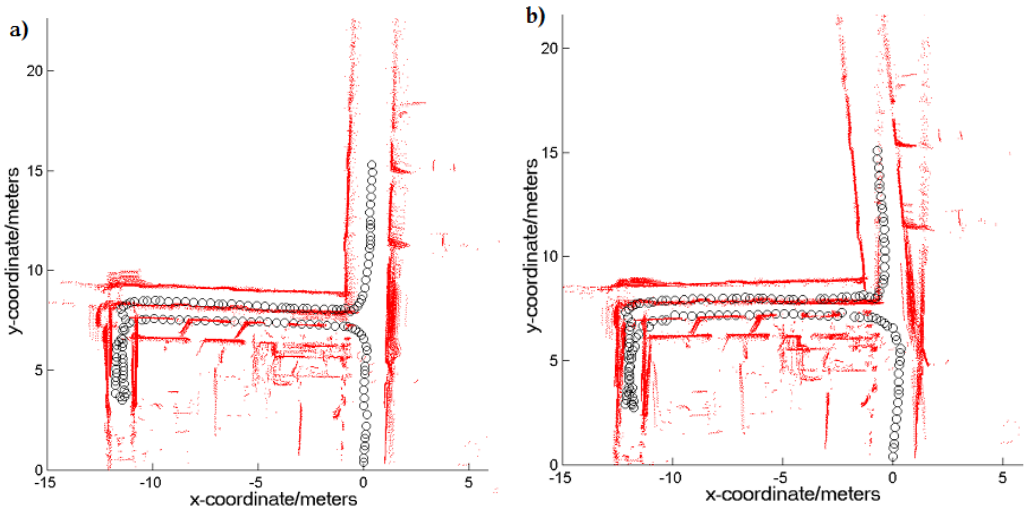

Figure B.2: The a) segmented lines and b) sequential lines algorithms in a corridor environment.

Appendix C

PeNa results from Automation Laboratory walks

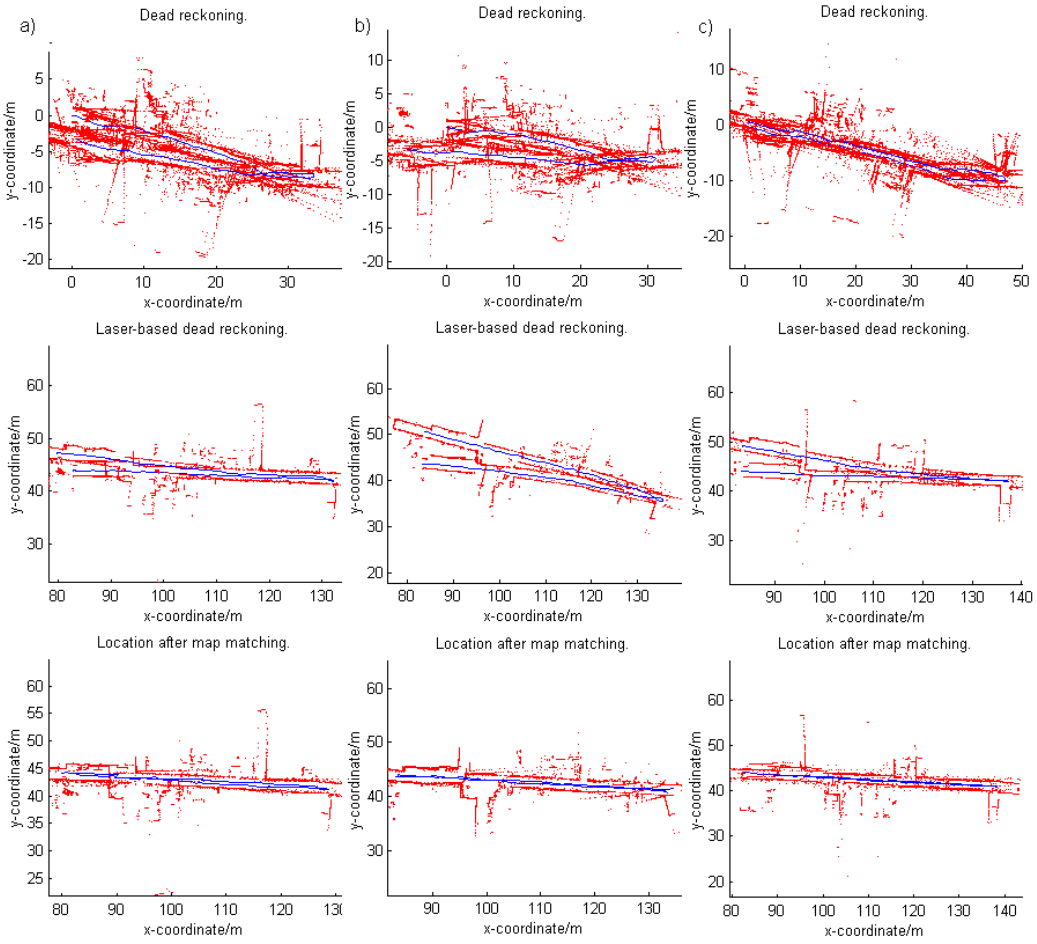

Figure C.1: Back and forth walks in a straight corridor.

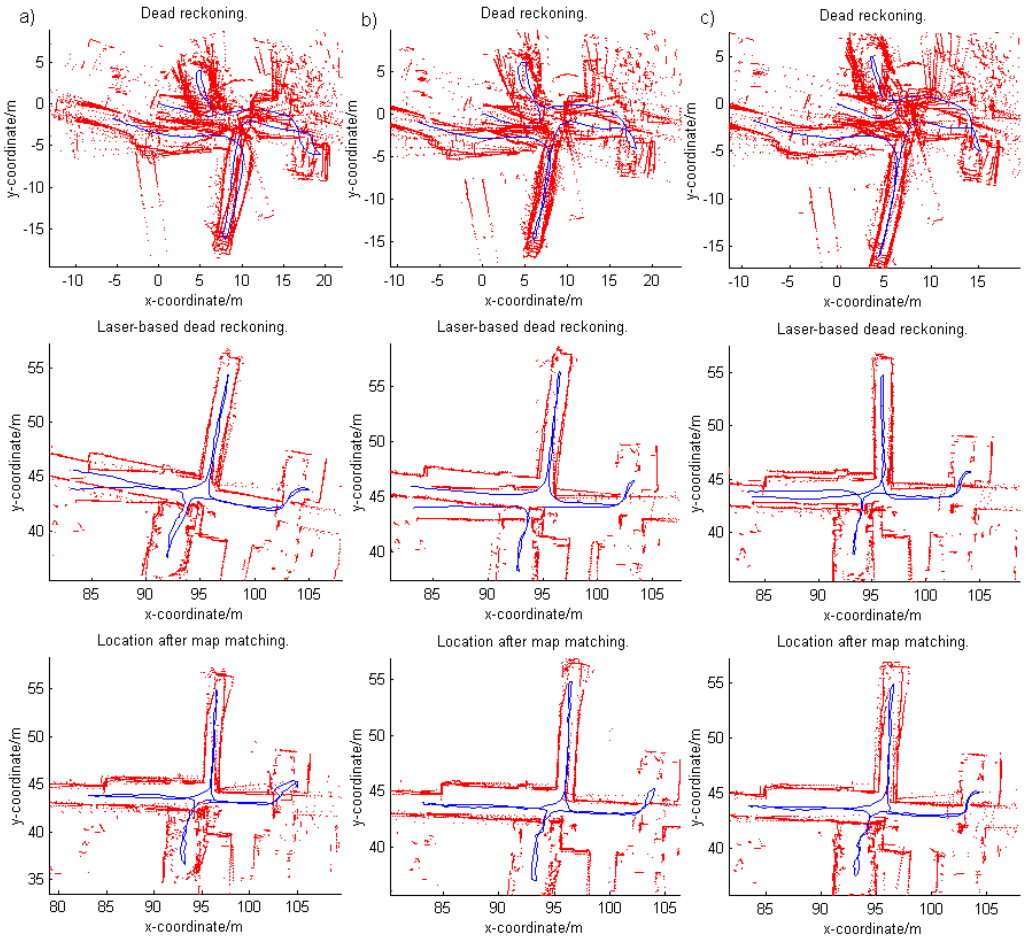

Figure C.2: Corridor walks with two room visits.

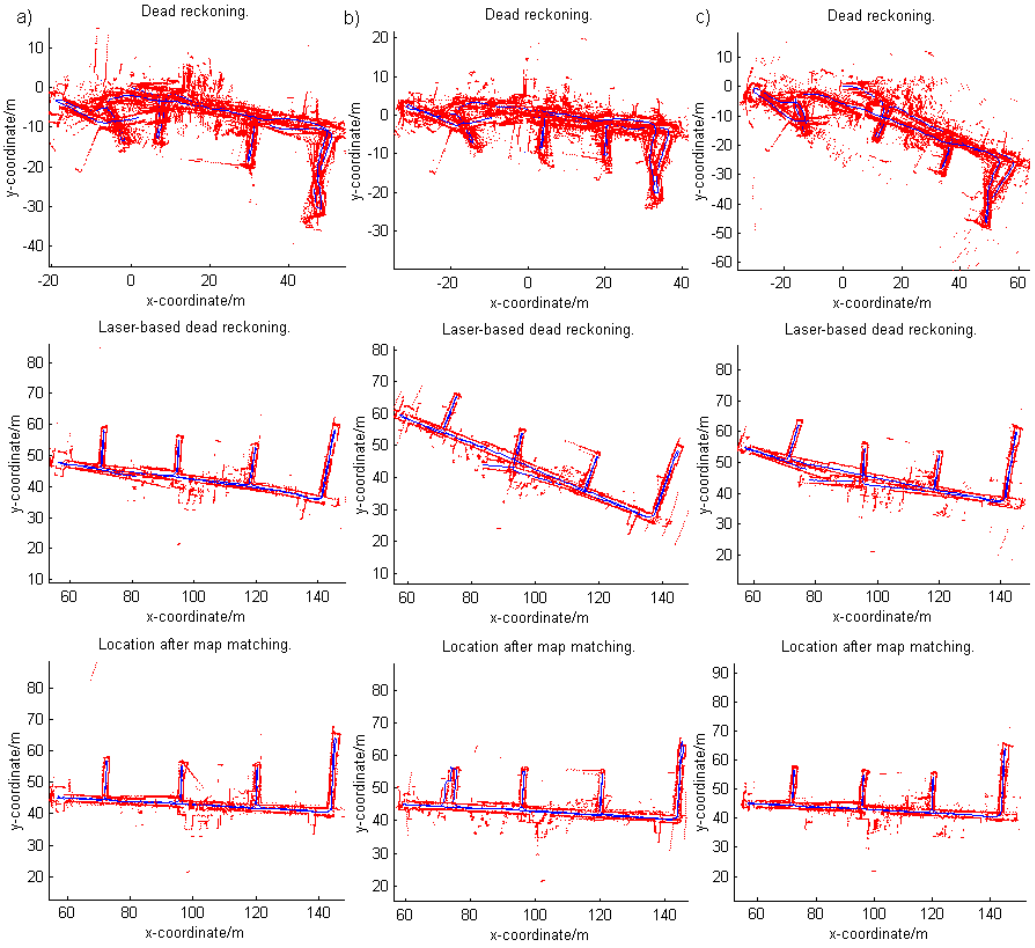

Figure C.3: Walks through all automation laboratory corridors.

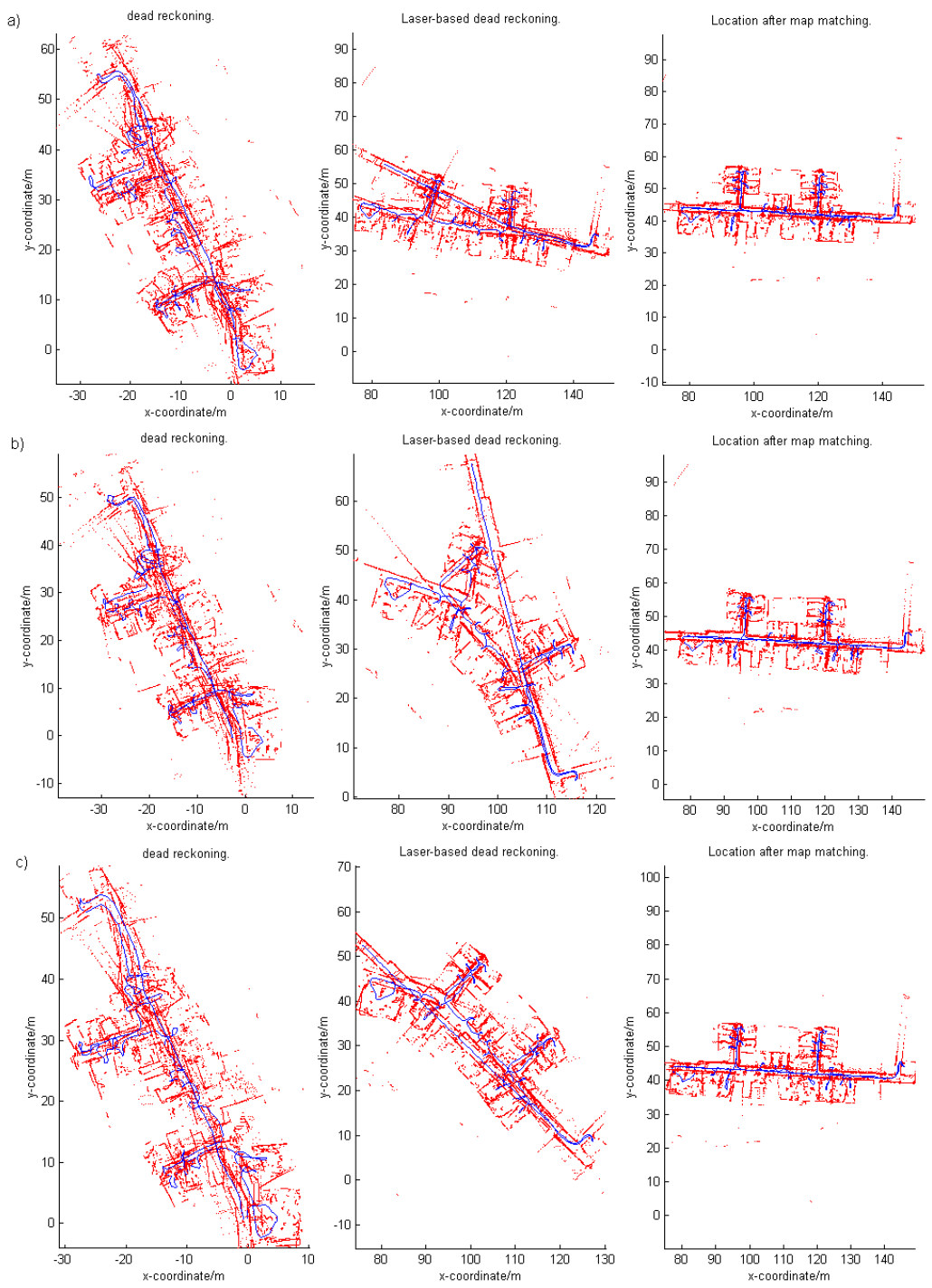

Figure C.4: Automation laboratory walks with 23 room visits.

Appendix D

Personal Navigation System Comparison

Table D.1: Comparison of sensor subsystems used in different localisation systems. A/P is short for Accelerometers/Pedometers.

Paper	Gyro	Compass	A/P	Map	(D)GPS	Beacons	Range scan	year
PeNa	X	X	X	X	-	-	laser	2004
[26]	-	X	X	-	-	X	-	2004
[6]	X	X	-	X	X	-	-	2003
[8]	-	X	X	-	X	-	-	2003
[88]	-	X	X	-	-	-	-	2003
[14]	X	X	X	X	-	-	-	2003
[13]	X	-	X	-	-	-	-	2003
[25]	-	-	-	-	-	X	-	2003
[89]	-	X	X	X	-	-	camera	2002
[68]	-	-	-	X	-	-	sonar	2002
[90]	-	X	-	-	-	X	-	2002
[16]	X	X	X	-	X	-	-	2002
[11]	-	X	X	-	-	-	-	2002
[10]	X	X	X	X	-	-	-	2001
[7]	-	X	X	-	-	X	-	2001
[23]	-	-	-	-	-	X	-	2000
[35]	-	-	-	-	-	X	-	2000
[9]	-	-	-	-	-	X	-	1999
[70]	-	X	X	-	-	-	-	1999
[21]	-	-	-	-	-	X	-	1999
[19]	X	-	X	-	X	-	-	1998
[24]	-	-	-	-	-	X	-	1997
[22]	-	-	-	-	-	X	-	1992