2007:54

LICENTIATE THESIS

Navigation Sensors for Mobile Robots

Håkan Fredriksson

Luleå University of Technology
Department of Computer Science and Electrical Engineering
EISLAB

2007:54|ISSN:1402-1757|ISRN:LTU-LIC--07⁄54--SE

Navigation Sensors for Mobile Robots

Håkan Fredriksson

EISLAB
Dept. of Computer Science and Electrical Engineering
Luleå University of Technology
Luleå, Sweden

Supervisor:

Kalevi Hyyppä

To my family...

Abstract

In mobile robot applications navigation systems are of great importance. Automation of mobile robots demands robust navigation systems. This thesis deals with a few solutions on how to help a mobile robot navigate in an environment. Navigation systems using retroreflective beacons and applications relying on laser range finders are two different solutions to aid a mobile robot. The first type of system provides very robust navigation, with the use of extra infrastructure in the environment. The latter has no need for any extra infrastructure as it can use the structure in the surrounding environment. Navigation using dead reckoning and GPS systems are also discussed. Furthermore, some of the mobile robot platforms available at Luleå University of Technology, and their application are presented.

Contents

Preface

My work presented in this theses started during spring 2005 as a research project. At that time I was not sure if I wanted to continue as a PhD student, but as I got familiar with the academic work at the university I realised that this was what I wanted to do. It has been great fun to work in all the projects I have been involved with so far, and I am looking forward with great expectations to the following two years as a PhD student.

I would like to thank all my colleges for creating an inspiring environment at work. Especially, I would like to thank my two roommates Sven Rönnbäck and Tomas Berglund, and my supervisor Kalevi Hyyppä. I would also like to thank Jan van Deventer for helping me to get the position as a PhD student I have today.

My work so far has been more focused on the actual applications than the algorithms behind them, i.e. to solve problems. I am trying to live after my device: *"Don't make things more complicated than they need to be"*. I prefer to do things that can be tested in reality and not only proven in theory.

Part I

Chapter 1

Introduction

In this thesis, sensors and algorithms for automation of mobile robots are discussed. A definition of '*Mobile Robots*' is presented, and some of the most resent mobile robot platforms at Luleå University of Technology are described.

1.1 Mobile robots

The general definition of a '*Robot*' is very wide [1], and very dependent on the person defining it. So I leave for the reader to make his own decision on what is considered to be a robot.

This thesis deals with a small part of the area of mobile robots. The definition of a '*Mobile Robot*' used in the thesis is: *"A vehicle that can move from position A to position B, either remotely controlled or fully autonomously"*.

1.2 Teleoperation and automation

A mobile robot can be anything from just a remotely controlled (teleoperated) robot that has no automatic functions at all, to a fully autonomous robot that performs all its specific tasks without the need of a human operator.

The most simple form of teleoperation of a mobile robot is when the operator is situated at a location where he has full visibility of the vehicle. In this case, no sensors on board the robot are necessary. A typical example of this kind of teleoperation is driving an ordinary remotely controlled toy like an RC-car.

The next step could be to move the operator to another room, mount a camera on board the mobile robot, and let the operator drive the vehicle only with feedback from a monitor. This is a little bit harder for the operator since it may be difficult to keep track of the exact position and heading of the vehicle when looking through the camera-monitor system.

The concept of teleoperation can be evolved to the extent that the operator mainly needs to supervise the robot [2, 3], and only in specific occasions tell the robot what to do. The more functionality that is included in the system, the less is the difference between teleoperation and an entirely autonomous system.

An example application of a teleoperated robot is the automated LHD (Load Haul Dump) vehicles used in parts of the LKAB underground mine in Kiruna, Sweden. These vehicles are used to move the ore from the blast area to the ore pass. Normally, hauling and dumping is done fully autonomously, only the loading is teleoperated by a human.

1.3 Sensor systems on a mobile robot

In order to implement automatic functions on a mobile robot, the robot needs to be equipped with sensors. A mobile robot can be equipped with numerous sensors. This thesis discuss a few of these sensors.

Chapter 2

Navigation systems

In this chapter a few different navigation solutions and some sensors are described. Some applications using the sensors on mobile robots are also discussed.

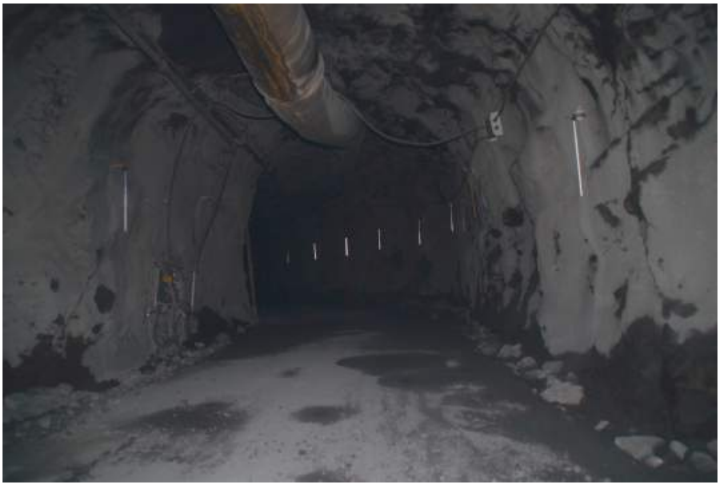

Figure 2.1: Picture from the LKAB underground mine in Kiruna, Sweden. On both sides of the tunnel one can see the retroreflective beacons used by the navigation system on the autonomous LHD vehicles.

2.1 Navigation systems using retroreflective beacons

One way to navigate in a known environment is to place retroreflective beacons on the walls, see Figure 2.1. A sensor on the mobile robot detects those beacons. The beacons are used as fixed references in the navigation system. If the position of every beacon is known, it is possible to calculate the position and heading of the robot.

The basics of beacon navigation

At first, a number of retroreflective beacons are placed in the environment. After that, a map describing the position of every beacon is created. The exact number of beacons and the distance between them depends on the environment. However, the system requires a minimum of three beacons to be visible at all time for navigation.

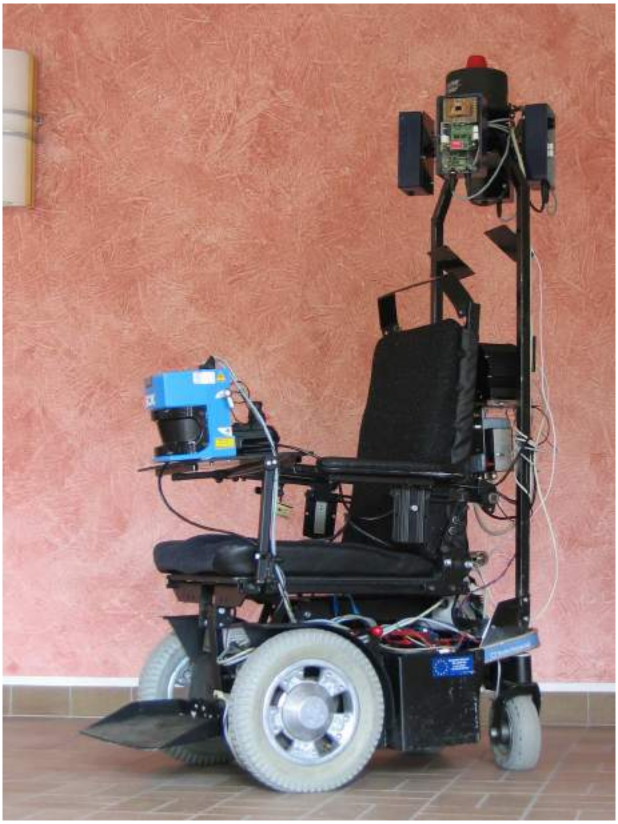

Figure 2.2: The MICA, (Mobile Internet Connected Assistant) wheelchair. An NDC8 navigation system and a camera based navigation system is mounted on top. In the front a SICK LMS200 range measuring laser is seen.

NDC8

One navigation system that uses artificial beacons as references for the navigation is the NDC8 system, also known as LazerWay. This system is invented at Luleå University of Technology [4], and is today produced by Danaher Motion [5]. The navigation system is used in a variety of industrial applications, including the automated LHD vehicles in the LKAB underground mine in Kiruna, Sweden.

To detect the artificial beacons the NDC8 system uses a laser scanner. The scanner is visible on top of the wheelchair in Figure 2.2. When a necessary number of beacons are detected, the vehicle position and heading can be found by comparing the beacon positions to the known map. The NDC8 system can estimate a position of a vehicle with an uncertainty of a few centimetres when correctly adjusted.

For navigation while the vehicle is moving, the motion during one laser scan has to be considered. If not the speed of the vehicle is so low that it can be neglected, a complementary dead reckoning process has to be implemented to keep track of the vehicle motion during a laser scan.

CMOS camera based navigation

A CMOS camera based navigation system using the same retroreflective beacons as the NDC8 system has been presented at Luleå University of Technology [6]. The system is intended for short range (less than 20m) indoor use, and consists of four individual camera modules mounted perpendicular to each other, as shown in Figure 2.2. The detection of beacons is done in hardware in each camera module. The output is the angles and distances to the detected beacons.

One advantage when using a camera based system instead of a scanning laser system is the possibility to perform a full 360° beacon detection simultaneously. When doing so, the motion of the vehicle is not important for the navigation process. Hence, no dead reckoning process is needed.

Disadvantages with retroreflective beacon navigation

Since a navigation system using retroreflective beacons requires extra infrastructure in the environment, it is only suitable to use well known environments. The system has no possibility to detect obstacles nor traverse areas without retroreflective beacons.

If the system is to be used in an environment where the driving surface is rough, i.e. when the vehicle and hence also the sensor wiggles, it might be a problem to detect the beacons at large distances. However, this is not as much of a problem for a camera based navigation system, since the cameras have a wide vertical field of view.

2.2 Dead reckoning

By knowing the vehicle heading and speed, it is possible to estimate the vehicle position based on the previously determined position. This kind of navigation is called dead

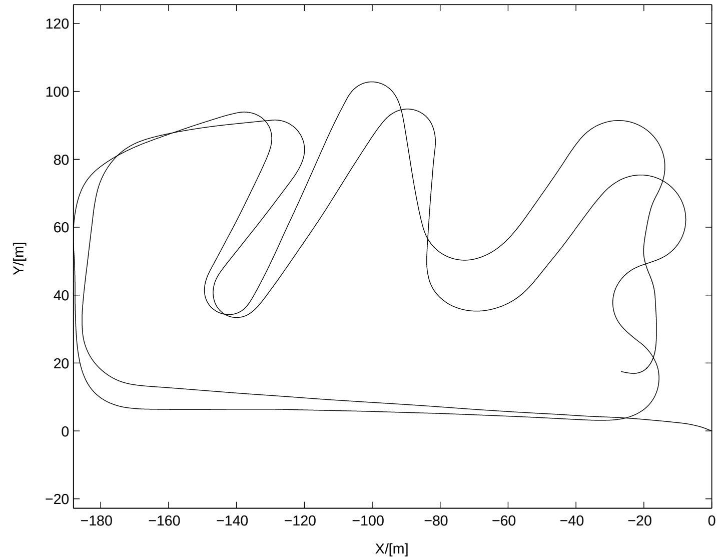

Figure 2.3: Example of dead reckoning navigation while traversal of an outdoor gokart track in Luleå, Sweden, The dead reckoning is based on data from a rate gyro and a wheel encoder. The rate gyro is of same type as those used in ESP (Electronic Stability Program) systems on modern cars. The vehicle has traversed the track twice. Note the difference in the angle of the long straight path of the track. This is due to drift in the rate gyro.

reckoning navigation. Dead reckoning is often used in mobile robot applications.

The navigation process can use different kinds of sensors depending on the application. It can be based solely on IMU[1] data or steering angle and wheel speed. Any combination of sensors that gives the heading and the speed of the vehicle can be used. The resulting navigation solution is very dependent on the accuracy of the sensors. Due to integrated errors and drift in the sensors, overall performance over time is usually bad. An example of dead reckoning navigation based on data from a rate gyro and a wheel encoder is shown in Figure 2.3.

However, dead reckoning is very useful for short term navigation, as well as a complement to other navigation systems.

2.3 Outdoor navigation, GNSS systems

The most common Global Navigation Satellite System (GNSS) is probably NAVSTAR GPS, maintained in the USA. There are other systems using satellites as references in the navigation process. Galileo is one GNSS system under development by the European Union (EU). Furthermore, Russia has a system under restoration called GLONASS.

[1] Inertial Measurement Unit, consisting of rate gyros and accelerometers

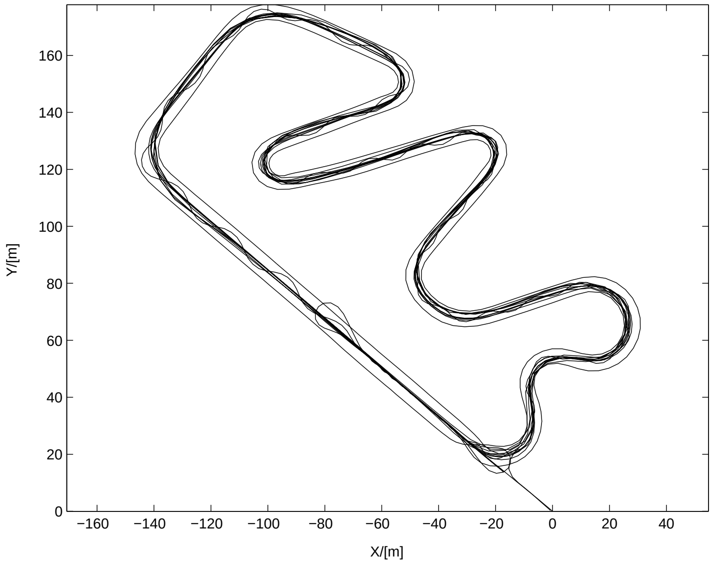

Figure 2.4: Plot of recorded GPS positions during 11 laps around the same outdoor gokart track as shown in Figure 2.3. The GPS data has been post-processed and fused with information from three different reference base stations and data from a IMU. Estimated absolute position uncertainty is in the order of a few decimetres.

A NAVSTAR GPS receiver estimates its position by measuring the distance to three or more medium Earth orbit satellites [7]. The uncertainty of the position estimate is, for a standard civilian GPS receiver, in the order of ±10m. It is possible to increase the accuracy for the position estimate in several different ways. One way is to use differential GPS [8]. Even better is to fuse GPS data with IMU information [9].

To test the performance of a NovAtel SPAN system, consisting of a GPS receiver[2] and an IMU package[3], we mounted the equipment on a car and drove several laps on a gokart track. The collected data from the SPAN system where post-processed and fused with GPS data from three different reference base stations. A plot of the resulting position estimate is shown in Figure 2.4.

2.4 Range measuring lasers: sensing the vehicle surrounding

This section describes some applications using scanning laser range finders and important considerations when using this type of sensor.

The sensor can be used for both navigation and sensing of the robot surroundings

[2] NovAtel GPS receiver ProPak-G2plus

[3] NovAtel IMU HG1700-AG58

[10, 11, 12, 13]. One advantage with this type of sensor is the possibility to combine obstacle avoidance and navigation using only one sensor.

Scanning range measuring lasers

A laser range finder, also called a laser scanner, is an environmental sensor. The laser measures distances to objects in its environment. A common laser scanner is the SICK LMS200 [14]. This laser scanner measure distances to objects in a 2D plane with an 180° field of view. Two LMS200 laser scanners are visible, mounted on top of a car in Figure 3.1.

2.5 Creating 3D data from 2D laser measurements

It is possible to produce 3D environments of 2D laser measurements by moving or rotating the laser. In order to do so, it is important to know the orientation (roll/pitch/yaw), and position (x,y,z), of the laser for every single laser measurement.

We have performed tests with laser scanners mounted on a roof rack on a car. When the laser is mounted with a tilt angle, it is possible to recreate a 3D environment as the vehicle moves.

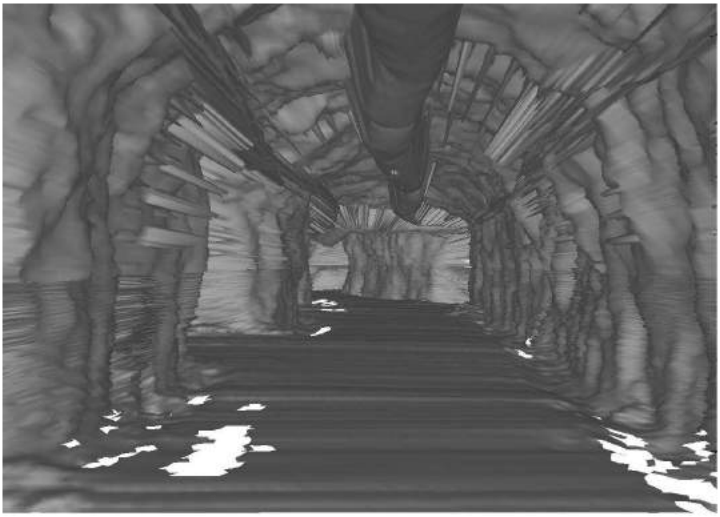

Figure 2.5: Image in 3D of a part of a tunnel in the LKAB underground mine in Kiruna, Sweden. The image is created by fusion of data from two LMS200 range measuring lasers and an NDC8 navigation system.

Underground mine

We have created a 3D image of one of the tunnels in the LKAB underground mine in Kiruna, Sweden, see Figure 2.5. The image was created by fusion of data from two LMS200 range measuring lasers and one NDC8 navigation system.

The 3D environment was created from sensor data collected while passing the tunnel one time with the equipment mounted on a car. All consecutive laser scans and vehicle positions and orientations at the time for every scan where logged. The laser scans where then rotated and translated into a global coordinate system and plotted as a surface.

During the test in the underground mine, we only had information about the vehicle position and heading in 2D. Hence, the vehicle was assumed to always be on the same height. We also assumed that the pitch and roll angles of the laser where constant. In reality they are not, due to motions in the vehicle while driving. Still, this kind of 3D representation of the collected data provides a powerful visualisation of the environment.

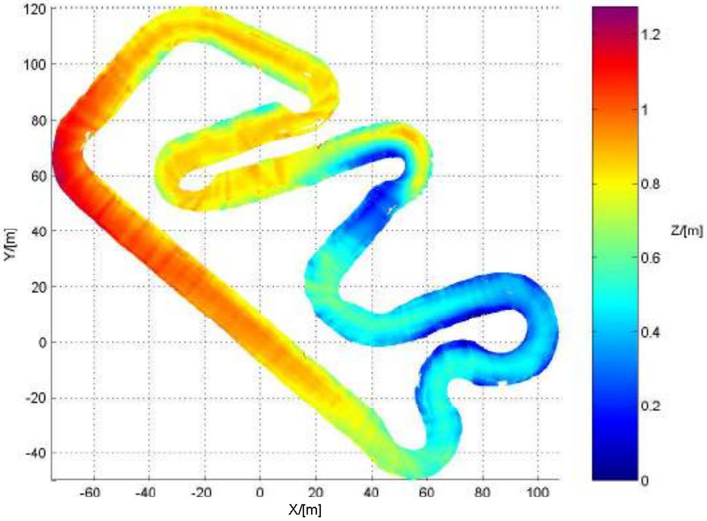

Figure 2.6: Color coded image of the same outdoor gokart track as shown in Figure 2.3. The color shading represent the height information of the track. The image is created by fusion of data from an LMS200 laser scanner and pose (position and orientation) information gained from a NovAtel SPAN GPS system.

Gokart track

A 3D environment describing an outdoor gokart track is shown in Figure 2.6. The image was created by fusion of data from a laser scanner and the pose (position and orientation) information gained from a NovAtel SPAN GPS system. To enhance the 3D information in the figure, the height information is color coded.

The measurements at the gokart track where made as a test of the performance of the NovAtel SPAN GPS system. The collected data from the SPAN system where post-processed and fused with GPS data from three different reference base stations.

The gokart track is suitable for such tests since it is sealed off from the public, the surface of the track is smooth, and the track contains tight corners that make a car lurch. One interesting part of the gokart track is the velodrome, a steeply banked corner in the middle of the track.

So far, the tests has shown promising results. The post-processed data from the NovAtel SPAN system tells us that the total height difference on the whole track is less than 1.2m. Furthermore, the velodrome is clearly visible in the color coded plot. Both the small height difference on the whole track, and the steeply banked velodrome, requires a high quality navigation system to be measured correctly.

Chapter 3

Our Test Vehicles

This chapter describes some of the different mobile robot platforms available at Luleå University of Technology (LTU), Sweden.

We have the experience of connecting and mounting sensors and equipment on several different platforms. In most applications we use the same software, originally developed for the MICA WheelChair [15], and we only change the sensor setup to fit the purpose of the specific test.

We have support for a variety of sensors in our software, several different laser scanners, rate gyros, IMUs, wheel encoders, etc.

3.1 MICA wheelchair

The MICA (Mobile Internet Connected Assistant) wheelchair shown in Figure 2.2, is one of the experimental platforms available at LTU. It is equipped with different types of sensors and an embedded PC which is connected to the Internet via WLAN. The wheelchair can be remotely controlled, and sensor readings can be requested from the onboard mounted sensors over the Internet [15].

It is possible to access live sensor measurements and send back control commands to the wheelchair using Matlab. This makes the platform very useful in robotics research and education at Luleå University of Technology. Researchers and students are able to develop different algorithms in Matlab and use the wheelchair for live testing.

3.2 Measurements with ordinary cars

We have performed tests with sensors and other necessary equipment mounted on ordinary cars. Pictures from two of those setups are shown in Figure 3.1 and 3.2. During these test we where not interested in closing the loop, i.e. use the sensor information for controlling the vehicle. The purpose of the tests was to collect data for post-processing. However, it was possible to look on live sensor data while driving.

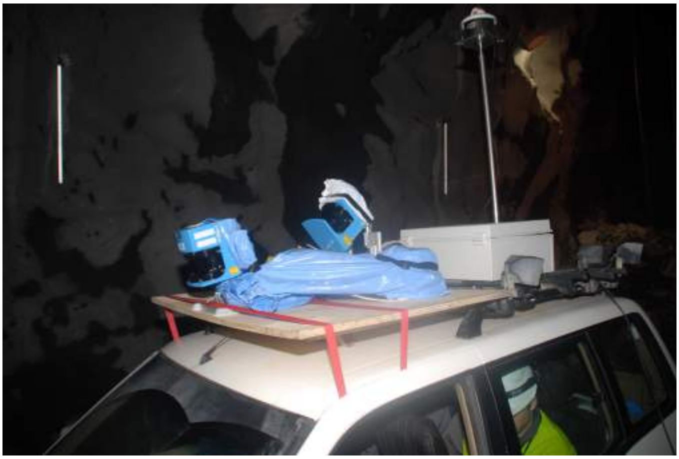

Figure 3.1: Test setup used when collecting measurements in the LKAB underground mine in Kiruna, Sweden. For navigation, we used an NDC8 system. The navigation laser is visible on top of the long stick on the rear part of the car. Two retroreflective beacons are also visible on the wall in the background. As environmental sensors we used two SICK LMS200 laser scanners. One of the lasers is tilted looking down on the ground and one is tilted up towards the ceiling.

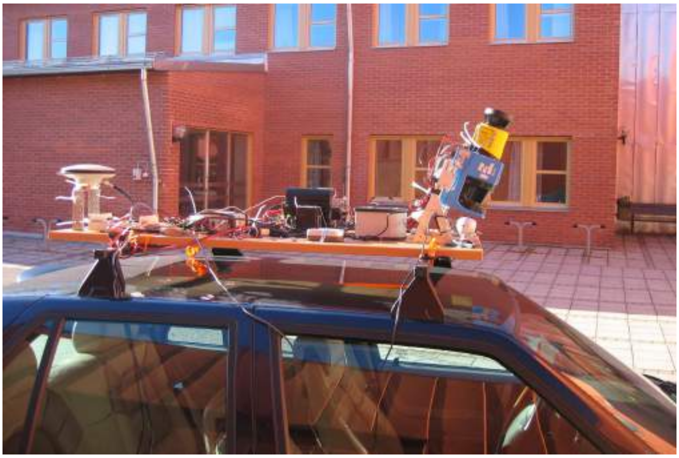

Figure 3.2: Test setup used during the measurements at a gokart track in Luleå, Sweden. All sensors and all other necessary equipment are placed on the plate on the roof rack. For navigation during the test, we used a NovAtel SPAN system consisting of a GPS receiver with an external IMU. Two different range measuring lasers - one SICK LMS200 and one SICK S300 - are mounted on top of each other, both looking forward and tilted down to see the ground.

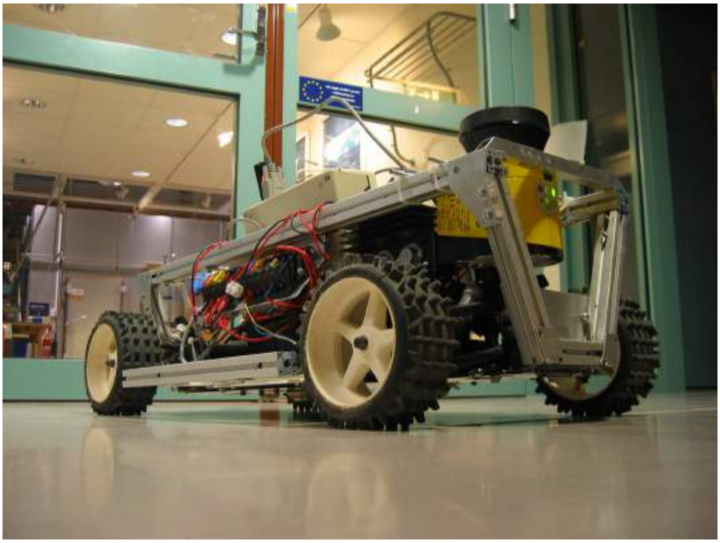

Figure 3.3: MiniLusar is a scale 1:5 RC-car. It was originally petrol driven, but it has been converted to electrical drive to enable indoor use. It is, apart from batteries and necessary servos for controlling it, equipped with a rate gyro, a laser scanner, and a computer with WLAN access.

3.3 MiniLusar

The small car called MiniLusar is a scale 1:5 RC-car, see Figure 3.3. It was originally petrol driven, but it has been converted to electrical drive to enable indoor use. It is, apart from batteries and necessary servos for controlling it, equipped with a rate gyro, a laser scanner, and a computer with WLAN access.

A recent use of the autonomous vehicle MiniLusar was in the RobotDay competition [16], arranged by the company SICK GmbH [17]. The basic rule in the competition was to let an autonomous vehicle drive a track as fast as possible. Any sensor technology whatsoever was allowed. For this race SICK donated one S300 laser scanner to each of the teams. The laser scanner can be seen in the front part of the vehicle in Figure 3.3.

In the RobotDay competition the track was defined by cones and other objects. Our vehicle was planned to use the circle sector expansion (CSE) method described in Paper D, to avoid obstacles and to find its way along the track. The CSE method is suitable for guidance in that type of environment. Unfortunately, MiniLusar did not complete the competition due to a malfunction in its hardware.

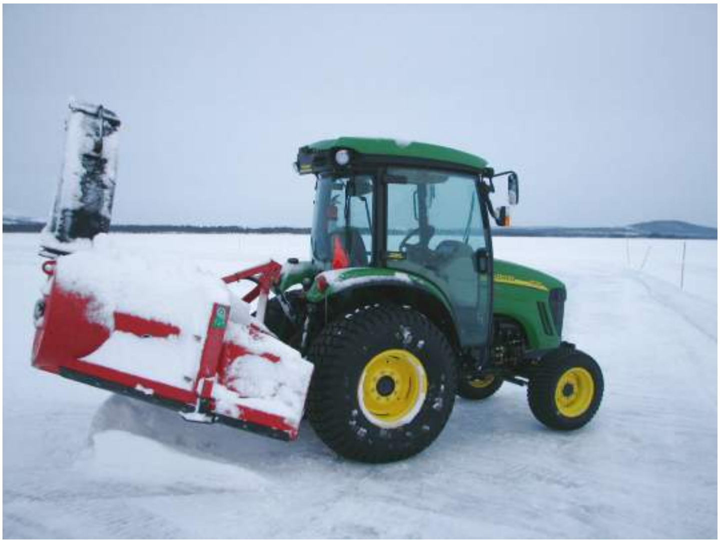

Figure 3.4: IceMaker I, a tractor with mounted snow blower in its natural frozen lake environment. Its purpose is to remove snow. The tractor is part of a research project at Luleå University of Technology.

3.4 IceMaker I

A tractor called IceMaker I, is shown in Figure 3.4. It is part of a research project at Luleå University of Technology. The tractor is owned by the company IceMakers, located in Arjeplog in the northern part of Sweden. IceMakers is one of the leading ice track makers in the world [18], and provides winter test tracks for car manufacturers from all over the world. Today, IceMakers uses the tractor every year for early preparation of their ice tracks located on frozen lakes.

IceMaker I is a John Deere 4720 compact tractor [19]. It is lightweight, (less than 2ton). Therefore, it can be used on thin ice without breaking through. But, not to risk any persons life, there are high safety margins on the thickness of the ice.

The basic idea behind the research project is that if IceMaker I can drive autonomously, the preparation of the tracks can start earlier without risking any persons life. Hence, the winter test season for the cars can start earlier.

So far, IceMaker I has been successfully remotely controlled over WLAN.

Chapter 4

Summary of Contributions

In this chapter, brief descriptions of the papers included in this thesis are presented. Furthermore, my personal contribution in all the included papers are pointed out.

4.1 Paper A

Multi source flash system for retroreflective beacon detection in CMOS cameras

Authors: Håkan Fredriksson and Kalevi Hyyppä

The paper describes the design considerations that has to be taken care of when designing a flash system for retroreflective beacon detection.

The main work behind this paper is done by me, with the support from Kalevi Hyyppä.

4.2 Paper B

snowBOTs: A Mobile Robot on Snow Covered Ice

Authors: Håkan Fredriksson, Sven Rönnbäck, Tomas Berglund, Åke Wernersson and Kalevi Hyyppä

In the paper, an novel algorithm for snow edge detection is introduced and described. The algorithm was tested with success in a closed loop system with a robotised wheelchair as test platform.

All authors contributed to the general ideas behind the paper. I wrote the implementation, made the tests, and wrote the main part of the paper.

4.3 Paper C

Circle Sector Expansions for On-Line Exploration

Authors: Sven Rönnbäck, Tomas Berglund, Håkan Fredriksson and Kalevi Hyyppä

In the paper, a novel and effective method denoted Circle Sector Expansion (CSE) is presented. The method is used to generate reduced Voronoi diagrams from range laser data. In turn, these diagrams can be used to compute possible paths for a vehicle.

My contribution to this paper is discussions and writing.

4.4 Paper D

Signature Graphs for Effective Localization

Authors: Sven Rönnbäck, Tomas Berglund, Håkan Fredriksson and Kalevi Hyyppä

The Signature Graph is introduced as a way for a robot to be able to localise itself in the environment. A signature is a compact description of the free space in the environment. No effort to model any specific obstacles or beacons is done. Only the free space is of interest.

My contribution to this paper is discussions and writing.

References

[1] "Robot definition." http://en.wikipedia.org/wiki/Robot, Okt 2007.

[2] T. Högström, J. Nygårds, J. Forsberg, and Å. Wernersson, "Telecommands for remotely operated vehicles," *IFAC and Intelligent Autonomous Vehicles*, 1995.

[3] J. Forsberg, U. Larsson, and Å. Wernersson, "Tele-commands for mobile robot navigation using range measurements," *Paper in PhD thesis: Mobile Robot Navigation Using Non-Contact Sensors, Johan Forsberg, Luleå University of Technology*, 1998.

[4] K. Hyyppä, *On a laser anglemeter for mobile robot navigation.* PhD thesis, Luleå University of Technology, Sweden, Apr 1993.

[5] "Danaher motion." http://www.danahermotion.com, Sept 2007.

[6] M. Evensson, A. Marklund, K. Kozmin, and K. Åhsberg, "Ett kamerabaserat navigeringssystem," Master's thesis, Luleå University of Technology, Sweden, 2002.

[7] "Navstar gps." http://en.wikipedia.org/wiki/gps, Okt 2007.

[8] G. Morgan-Owen and G. Johnston, "Differential gps positioning," *Electronics & Communication Engineering Journal*, vol. 7, Feb 1995.

[9] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, June 1999.

[10] E. S. Duff, J. M. Roberts, and P. I. Corke, "Automation of an underground mining vehicle using reactive navigation and opportunistic localization," in *Australasian Conference on Robotics and Automation*, 2002.

[11] R. Madhavan, M. Dissanayake, and H. Durrant-Whyte, "Autonomous underground navigation of an lhd using a combined icp-ekf approach," in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3703–3708, 1998.

[12] J. Larsson, *Reactive navigation of an autonomous vehicle in underground mines.* Licentiate thesis, Örebro Universitet, Sweden, 2007.

[13] S. Rönnback, *On Methods for Assistive Mobile Robots.* PhD thesis, Luleå University of Technology, Sweden, 2006.

[14] SICK, "Sick and lms200 and laser measurement system." http://www.sick.com, Dec 2003.

[15] S. Rönnbäck, *On a Matlab/Java testbed for mobile robots.* Licentiate thesis, Luleå University of Technology, Sweden, Dec 2002.

[16] S. Rönnbäck, H. Fredriksson, D. Rosendahl, K. Hyyppä, and Å. Wernersson, "An autonomous vehicle for a robotday," in *Mekatronikmöte 2007*, 2007.

[17] "Sick." http://www.sick.com, Okt 2007.

[18] "Icemakers." http://www.icemakers.se, Okt 2007.

[19] "John deere." http://www.deere.com, Okt 2007.

Part II

Paper A

Multi source flash system for retroreflective beacon detection in CMOS cameras

Authors:
Håkan Fredriksson, Kalevi Hyyppä

Reformatted version of paper originally published in:
To be submitted

Multi source flash system for retroreflective beacon detection in CMOS cameras

Håkan Fredriksson, Kalevi Hyyppä

Abstract

We present a method for improving a flash system for retroreflective beacon detection in CMOS cameras. Generally, flash systems are designed in a manner that make them suited for beacon detection in a small range interval. We strive to increase the flash system range interval such that the beacon detection need not be limited to small specific range. Using several LEDs at different distances from the optical axis of the camera, the received optical power will stay at an almost constant level. Hence, we increase the usable flash range. Underlying theory and formulae are presented. An improved LED flash system was built considering the presented method. Simulations show that the usable flash range of the improved system can be almost doubled compared to a general flash system. Tests were performed indicating that the presented method works according to theory and simulations.

1 Introduction

In this paper we present a method to design a flash system consisting of several infrared light emitting diodes. The purpose of the flash is to illuminate retroreflective beacons for detection in a CMOS camera.

The camera/flash system is part of a navigation system that uses retroreflective beacons as fixed reference points. Figure 1 shows two beacons at different distances. The camera/flash system estimates the distances and headings to the beacons and uses that information as input to a navigation process. Navigation systems that use this type of beacons as references have been on the market for several years; NDC8, also known as LazerWay, developed by one of the authors [1] and today produced by Danaher Motion is one of those systems. Common for the present systems is that they use a scanning laser for detection of beacons. We are working on a camera based system with no moving parts.

1.1 Properties of retroreflective surfaces

A retroreflective surface has the property that it reflects most of the incoming light within a very narrow angle right back to the source, with only a small dependency of incident angle. In contrast with a ordinary bright surface like a paper sheet, which has a very diffuse reflection, or a mirror, which has a specular reflection [2].

In the area of road safety and road markers, work has been done to calculate, simulate, and measure retroreflective properties of different retroreflective materials [3, 4, 5, 6].

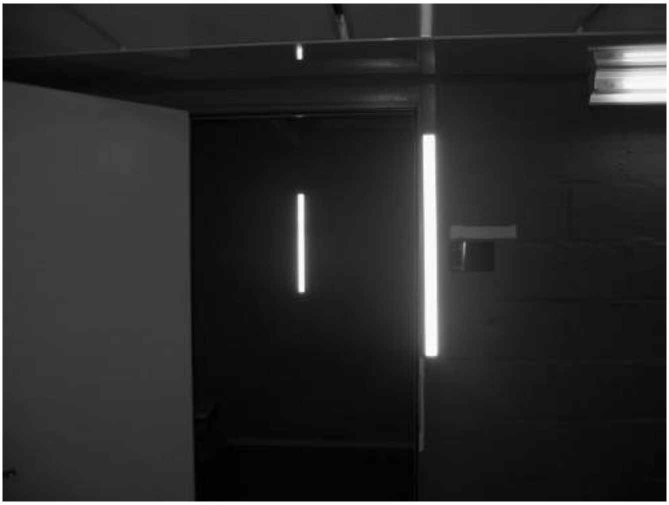

Figure 1: In the figure two retroreflective beacons at different distances are visible, the one to the left is at app. 6m and the one to the right is at app. 3m distance. The width of the beacons is 36mm and the height is 750mm. The picture is taken with an ordinary digital camera with a built in flash.

1.2 Detecting retroreflective surfaces

The retroreflective properties make it possible to reduce the interference from other surrounding light sources by using a strong flash mounted close to the optical axis of the camera. Then retroreflective surfaces will appear much brighter than non retroreflective surfaces. How much depends on the properties of the retroreflective material, and the power of the surrounding light sources.

When designing a flash system for retroreflective beacon detection in a CMOS camera there are mainly two important criterions that has to be fulfilled. The first criteria is the requirement to detect beacons at large distances. To improve the detection distance one needs to increase the power of the flash.

The second criteria is to avoid blooming around bright objects. If the flash is too strong the beacons will appear too bright for the camera chip and cause the individual pixels to saturate and bleed into surrounding pixels. With the camera and lens parameters fixed, the only way to avoid this problem is to limit the power of the flash.

One way to solve this contradiction is to utilise the flash design described in this paper. By constructing a flash system consisting of several LEDs at different distances from the optical axis of the camera the received optical power can be held at an almost constant level, and hence increase the usable flash range.

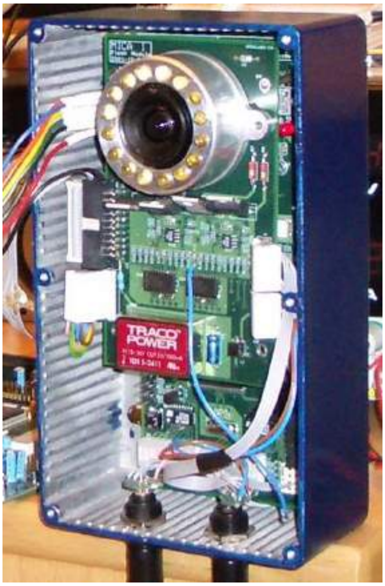

Figure 2: An ordinary LED flash consisting of 16 Light Emitting Diodes mounted in a circle around the camera lens. All LEDs are placed at equal distance from the optical axis of the camera. The flash is mounted on a CMOS camera based navigation system prototype.

2 Flash system equations

In this section we present the equations necessary for designing an LED flash system for retroreflective beacon detection with a CMOS camera. All the calculations and descriptions are based on the assumption that the camera and the flash are mounted with the optical axis in the horizontal plane. The flash is supposed to illuminate beacons as the ones shown in Figure 1. The beacons are assumed to be in the vertical middle of the image, and they are expected to be found anywhere in the horizontal plane. Hence the important field of view lies in the horizontal plane. Though the vertical field of view is not forgotten, it is not of great importance in our application.

In the end of this section we will present a formula for calculating the maximum received optical power by a receiver situated close to the light source, as a function of distance and angle to the beacon. In our system the receiver is one single pixel in the CMOS camera chip. The formula takes into account all the individual LEDs relationships to the receiver.

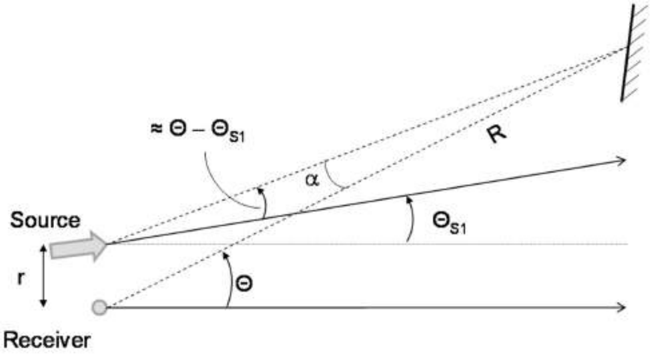

Figure 3: Source, reflective beacon, and receiver constellation for a flash consisting of one LED. The distance R is actually much greater than r, but the figure is scaled to enhance the angle α.

2.1 Optical power from Light Emitting Diodes

The radiant intensity I_S, from a single LED is dependent on the emitting angle θ, where $\theta = 0$ is along the optical axis of the LED. This dependency is assumed to be symmetric around the optical axis, and is in these calculations approximated with a Gaussian distribution,

$$I_S(\theta) = I_{S0} e^{-2(\frac{\theta}{\theta_{S0}})^2}. \tag{1}$$

The parameter θ_{S0} is the angle where the intensity has dropped to $\frac{I_{S0}}{e^2}$, and I_{S0} is the on axes radiant intensity produced by the diode.

To improve the horizontal emitting angle of a complete flash with n diodes, the individual diodes can be slightly tilted in the horizontal plane by the parameter θ_{Si}, and hence the total radiant intensity in the horizontal plane $I_{Stot}(\theta)$ for the whole flash becomes

$$I_{Stot}(\theta) = I_{S1} e^{-2(\frac{\theta - \theta_{S1}}{\theta_{S10}})^2} + ... + I_{Sn} e^{-2(\frac{\theta - \theta_{Sn}}{\theta_{Sn0}})^2}, \tag{2}$$

where I_{Si} and θ_{Si0} are the parameters for the diode i. This expression is valid under the assumption that r is much smaller than R, see Figure 3. If the diodes are placed symmetrically around the receiver, the optical axis of the complete flash can be considered to be the same as for the receiver.

2.2 BRDF of retroreflective beacons

The irradiance E_B, received by the beacon can according to [7] be expressed with

$$E_B = \frac{I_S}{R^2}, \tag{3}$$

where R is the distance between the source and the beacon. This expression is valid under the assumption that the incoming radiant intensity I_S is constant over the whole reflector, and the source is considered to be a point source.

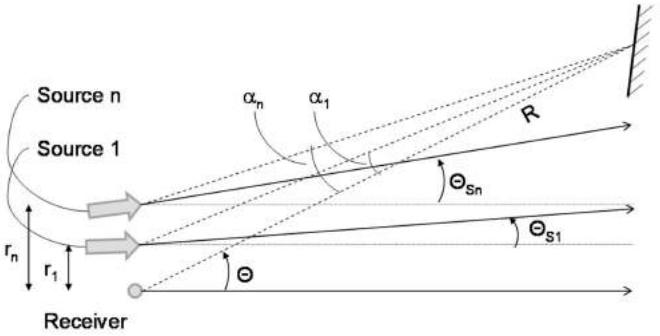

Figure 4: Source, reflective beacon, and receiver constellation with multiple sources. In a real application the diodes are placed symmetrically around the receiver so that the optical axis of the flash and the receiver can be considered to be the same, though the optical axis of the individual LEDs are not.

The Bidirectional Reflectance Distribution Function (BRDF) is used to calculate the reflected radiance from the beacon towards the receiver. The general definition of the BRDF is defined as the ratio of differential radiance to differential irradiance

$$f_B(\theta_i, \phi_i; \theta_e, \phi_e) = \frac{\delta L_B(\theta_e, \phi_e)}{\delta E_B(\theta_i, \phi_i)}, \tag{4}$$

where (θ_i, ϕ_i) and (θ_e, ϕ_e) are the directions of the incoming and exiting light respectively [8].

The reflected radiance L_B, from the beacon towards the receiver, can when the source is considered to be a point source according to [1] be written in the form

$$L_B = E_B f_B(\theta_i, \phi_i; \theta_e, \phi_e). \tag{5}$$

We ignore the small influence the incident angle to the beacon gives on the BRDF function since this mainly will give a damping on the reflected radiance. We also assume that the BRDF of a retroreflective beacon is symmetric around the incident angle, and only dependent on the angle α between the source and the detector direction, see Figure 3.

In [1] the BRDF is assumed to have a Gaussian distribution and the expression

$$f_B = \frac{2\eta_B}{\pi\alpha_{B0}^2} e^{-2(\frac{\alpha}{\alpha_{B0}})^2} \tag{6}$$

is presented. The beacon specific parameters η_B and α_{B0} are representing the efficiency and distribution of the reflected light respectively.

An approximation of the angle α between the source LED and receiver optics seen from the beacon can be calculated with

$$\alpha = arctan(\frac{rcos(\theta)}{R}) \approx \frac{rcos(\theta)}{R}, \tag{7}$$

where r is the small distance between the source and the receiver and R is the distance to the beacon, see Figure 3. This expression is valid when the LEDs are situated in the horizontal plane of the receiver. If the LEDs are placed elsewhere the impact of the parameter $cos(\theta)$ is reduced.

When combining (6) and (7) it becomes clear that the BRDF is dependent on the distance r between the source and receiver, and the distance R to the beacon. Since the distance r is fixed in a given flash constellation, the BRDF can be seen as a function of R and θ, $f_B(R, \theta)$. On the receiver optical axis, i.e. for $\theta = 0$, the BRDF is only a function of R, $f_B(R)$.

In a multiple source configuration, see Figure 4, where the distance r_i between the source and the receiver differs for different sources, the BRDF has to be calculated for every unique source,

$$f_{Bi} = \frac{2\eta_B}{\pi\alpha_{B0}^2} e^{-2(\frac{r_i}{\alpha_{B0}R})^2}. \tag{8}$$

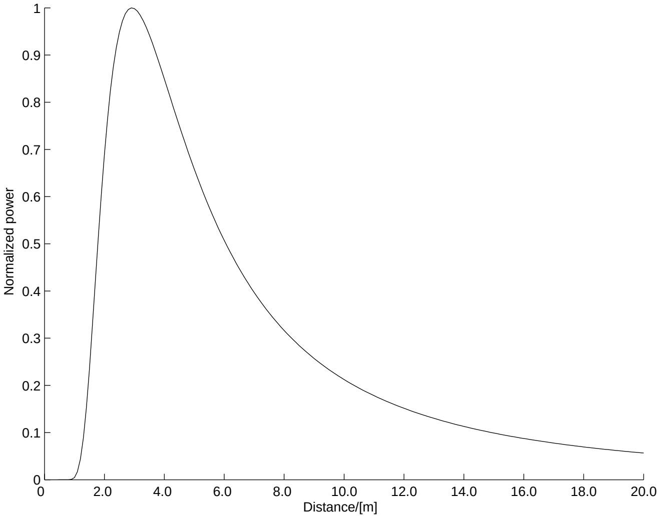

Figure 5: Plot of normalised received power ϕ_D, as function of distance to a beacon along the optical axis of the camera when using one single light source. The distance r, between the source and the optical axis of the receiver is set to 18mm. The received power has dropped to half at 1.8m and 6m, and the power peak is at 2.9m.

2.3 Optical power at the receiver

The optical receiver in our system is a CMOS camera chip with a lens system. Since we are interested in avoiding blooming in the camera chip we have to calculate the maximum received optical power in every pixel of the chip. The calculation is done

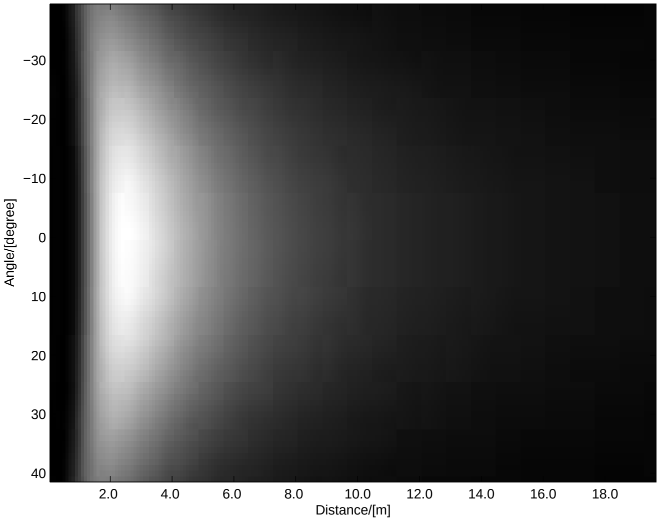

Figure 6: Optical power received by one pixel in the camera chip as function of distance and angle to the reflective beacon for a single source LED flash. Brighter areas represent more power.

under the assumption that the beacon is so close to the camera that the width of the beacon is visible in at least one whole pixel.

The optical power ϕ_D reaching the detector, i.e. one pixel in the camera chip, can according to [7] be calculated with

$$\phi_D = K_C L_B, \tag{9}$$

where K_C contains the parameters of the detector and the lens. The necessary parameters are the area of the detector A_D, the transmittance of the lens T_L, the diameter of the lens D_L, and finally the focal distance f_L. To calculate K_C we use the formula

$$K_C = \pi A_D T_L N\!A^2, \tag{10}$$

where $N\!A$ is the numerical aperture for a lens working in air,

$$N\!A = \frac{D_L/2}{\sqrt{{f_L}^2 + (D_L/2)^2}}. \tag{11}$$

Writing the full expression of ϕ for $\theta = 0$, along the optical axis, yields

$$\phi_D = \left(\frac{2\eta_B K_C I_{S0}}{\pi \alpha_{B0}^2} \right) \frac{1}{R^2} e^{-2(\frac{r}{\alpha_{B0} R})^2}. \tag{12}$$

A normalised plot of this function is shown in Figure 5. Here it is clearly seen that the constellation with one single source gives a high peak at a certain distance. The distance to the peak is determined by the source detector distance r and the distribution of the reflected light, α_{B0} from the beacon.

The optical power received by the detector when using several different sources can be added,

$$\phi_{Dtot} = \phi_{D1} + \phi_{D2} + ... + \phi_{Dn}. \tag{13}$$

The total received power can then be written as a function of θ and R,

$$\phi_{Dtot}(\theta, R) = \frac{K_C}{R^2}\bigg(I_{S1}(\theta)f_{B1}(R) + ... + I_{Sn}(\theta)f_{Bn}(R)\bigg). \tag{14}$$

Figure 6 shows the received power as function of θ and R for a single source configuration.

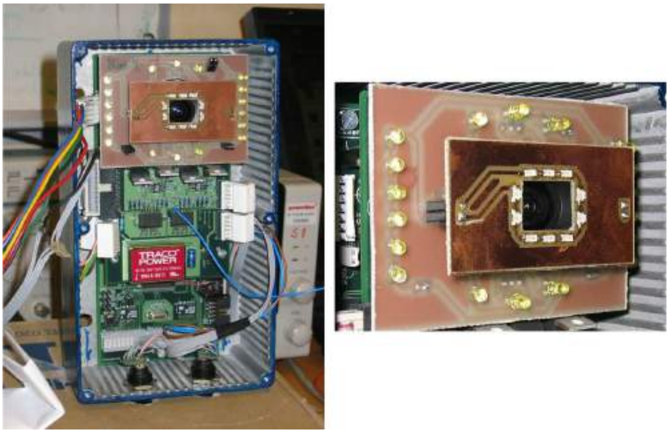

Figure 7: Two views of the improved LED flash which contains 28 diodes of three different types. The different LEDs are placed on different distances from the optical axis of the camera to increase the usable flash range. The flash is mounted on a CMOS camera based navigation system prototype.

3 Implementation

3.1 Design considerations for our system

When designing a flash system with LEDs one important aspect is the field of view of the camera. This sets the preferred emitting angle on the chosen diodes. The horizontal field of view for the camera in our system is less than 60°, and the vertical field of view is less than 45°. For our specific system the beacons vertical position is known to be in the middle of the picture, therefor the vertical emitting angular range is not that important.

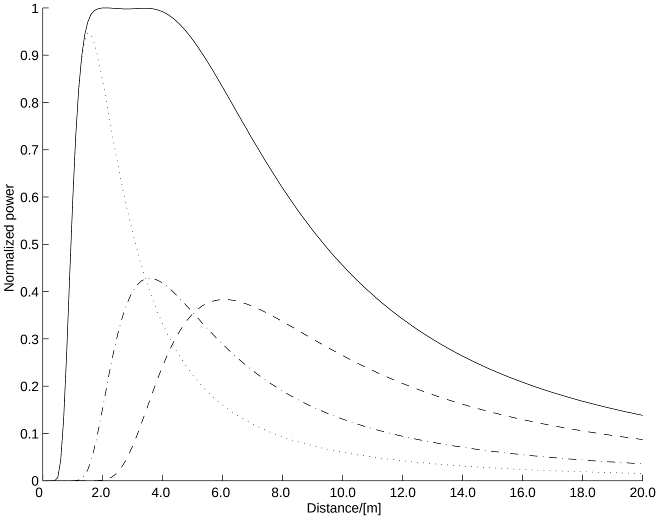

Figure 8: Plot of normalised received power ϕ_{Dtot} for the improved LED flash seen in Figure 2 as a function of the distance to a beacon on the optical axis of the camera. The three smaller curves represent the power ϕ_D for the three different types of diodes, and the solid line is the sum of the three.

The number of diodes that can be used are restricted due to limitations in the power supply. In a future version we might rebuild the electronics to support a larger amount of diodes.

The beacon specific parameters α_{B0} and η_B are also important to know. Especially the distribution α_{B0} is crucial to know since this parameter affects how far away from the receiver the LEDs should be placed to get optimal performance. In this calculation we assume $\alpha_{B0} = 0.5°$ and $\eta_B = 0.7$.

3.2 Ordinary LED flash

A common way to build a LED flash is to place all the diodes in a circle around the lens. Figure 2 shows an example of this type of configuration, 16 diodes are mounted on a circle with a radius of 18mm around the detector lens. With this configuration all the diodes get the same distance to the optical axis of the camera, and can be treated as a single point source.

The calculations are made based on 16 diodes of type HIR204/HO from the manufacturer Everlight Electronics CO, all mounted with their optical axes parallel to the receiver optical axis. Figure 5 and 6 shows plots on how the received power is dependent on the distance and view angle to the beacon. From these plots it can be seen that this type of flash has a fairly good emitting angular range, but there is a high peak in the received power at a fixed distance.

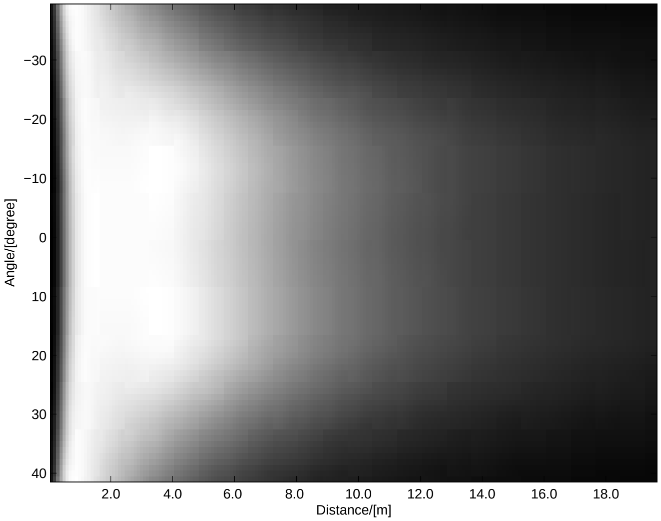

Figure 9: Received power as function of distance and angle to the beacon for the improved flash. Brighter areas represent more power.

3.3 Improved LED flash

Our improved LED flash is shown in Figure 7. It is divided into three different diode setups, that consist of three different types of diodes. Each setup of diodes is placed at a certain distance from the optical axis of the camera to give different work range for the different setups. In the calculations of the received power, the placements of all the individual diodes are considered.

For short range performance 10 surface mounted diodes of type IR11-21C/TR8 are mounted in front of the lens, as close as possible to the receiver optical axis, see Figure 7. The middle range performance is handled by 6 diodes of type HIR204/HO that are placed three above and three beneath the lens. And finally, for long range performance a total of 12 diodes of type HIR204 are placed, six on each side of the lens. The complete placement information and the diode specifications can be found in Table 1.1.

The calculated optical power along the optical axis of the receiver for the complete flash system and the three different diode setups is shown in Figure 8. A plot of the optical power as a function of distance and angle to the beacon is shown in Figure 9.

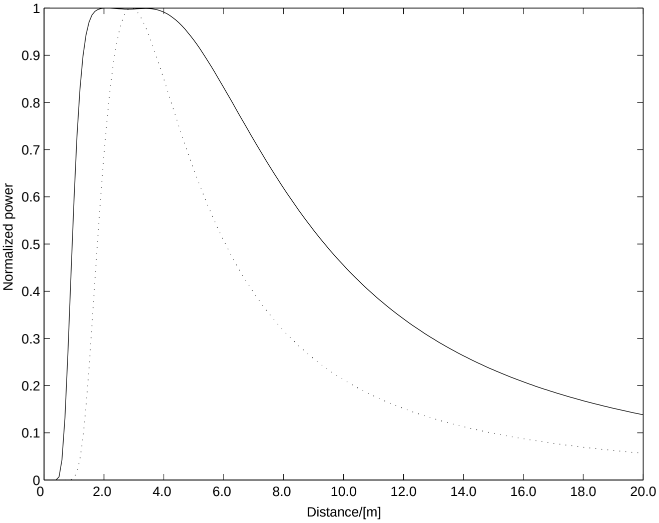

Figure 10: Normalised received optical power as function of distance to a reflective beacon along the optical axis of the camera when using a standard LED flash (dotted) and a LED flash (solid) improved by the methods presented in this paper. For the standard flash the received optical power is above 50% *between* $1.8m$ *and* $6m$*, and for the improved it stays over* 50% *between* $1m$ *and* $9.3m$*. The usable flash range for the improved flash is almost doubled compared to the standard LED flash.*

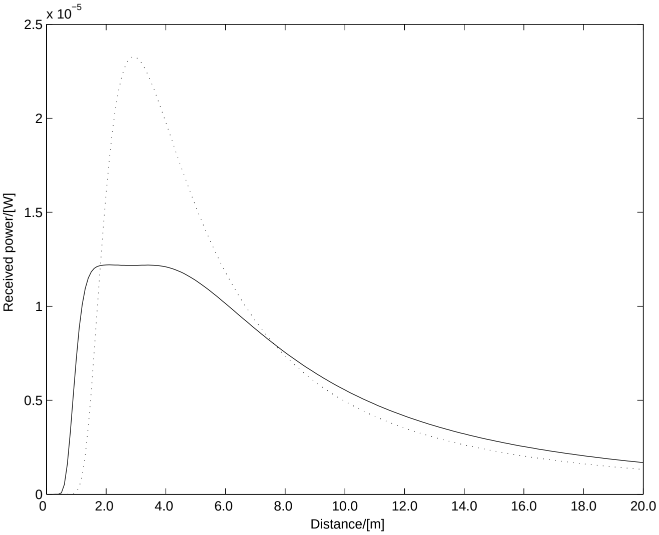

Figure 11: Comparison of received power between the ordinary and improved LED flash described in the text, along the optical axis of the camera. The standard flash has high peak power at 3m. The improved LED flash has lower but wider peak power.

No. Diodes	Type	θ_{S0}/[degree]	I_{S0}/[mW/sr]	r_i/[mm]	θ_{Si}/[degree]
2	IR11-21C	80	85	8	0
4	IR11-21C	80	85	9.7	0
4	IR11-21C	80	85	10.2	0
2	HIR204/HO	51	380	22	0
2+2	HIR204/HO	51	380	22	+20,−20
2+2	HIR204	25	900	36	+20,−20
2+2	HIR204	25	900	37	+20,−20
2+2	HIR204	25	900	39	+20,−20

Table 1.1: LED data and placement information for the improved flash.

4 Conclusions and discussion

We have in this paper shown the necessary equations and important design considerations when constructing a multi source flash for retroreflective beacon detection in a CMOS camera. We have built a LED flash according to these ideas and tested it with good results compared to an ordinary flash with all LEDs at equal distance from the receiver optics.

The calculations tells us that the received optical power for the standard flash is above 50% between 1.8m and 6m. For the improved flash the received optical power stays over 50% between 1m and 9.3m. This indicates that the usable flash range for the improved flash is increased to almost 200% of the standard LED flash.

One interesting thing to notice is the comparison of the received optical power seen in Figure 10 and Figure 11. These two figures show the same power curves normalised and not normalised. In the first plot it is clearly visible that the improved flash has a much wider range where the power is strong compared to the ordinary flash, but as seen in the second plot the maximum power is higher for the ordinary flash.

Since the camera system has a limited dynamic range a flat power curve is preferable when adjusting the parameters of the camera and lens system.

5 Future work

The tests done on the system so far indicates that the work presented in this article is correct. It is however necessary to do more measurements on the flash system to be able to really verify all the assumptions and formulae presented.

We plan to build a better test system which makes it easier to test different flashes. With a new system it would be possible to use more LEDs in the flash design, and in that way increase the performance of the flash.

References

[1] K. Hyyppä, *On a laser anglemeter for mobile robot navigation.* PhD thesis, Luleå University of Technology, Sweden, Apr 1993.

[2] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Surface reflections: Physical and geometrical perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-13, no. 7, pp. 611–634, 1991.

[3] V. V. Barun, "Imaging of retroreflective objects under highly nonuniform illumination," *Optical Engineering 35(07)*, 1996.

[4] J. Rennilson, "Specialized optical systems for measurement of retroreflective materials," in *Proc. SPIE Vol. 3140, p. 48-57, Photometric Engineering of Sources and Systems*, 1997.

[5] B. So, Y. Jung, and D. Lee, "Shape design of efficient retroreflective articles," *Materials Processing Technology, 130-131*, 2002.

[6] V. V. Barun, "Estimations for optimal angular retroreflectance scale of road-object retroreflective markers," in *Proc. SPIE Vol. 3207, p. 118-125, Intelligent Transportation Systems*, 1998.

[7] R. McCluney, *Introduction to radiometry and photometry.* Artech House, 1994.

[8] B. K. P. Horn, *Robot Vision.* McGraw-Hill Book Company, 1986.

Paper B

snowBOTs: A Mobile Robot on Snow Covered Ice

Authors:
Håkan Fredriksson, Sven Rönnbäck, Tomas Berglund, Åke Wernersson, Kalevi Hyyppä

Reformatted version of paper originally published in:
Proceeding of the Robotics and Applications and Telematics - 2007, Würzburg, Germany

snowBOTs: A Mobile Robot on Snow Covered Ice

Håkan Fredriksson, Sven Rönnbäck, Tomas Berglund, Åke Wernersson, Kalevi Hyyppä

Abstract

We introduce snowBOTs as a generic name for robots working in snow. This paper is a study on using scanning range measuring lasers towards an autonomous snow-cleaning robot, working in an environment consisting almost entirely of snow and ice. The problem addressed here is using lasers for detecting the edges generated by ”the snow meeting the road”.

First the laser data were filtered using histogram/median to discriminate against falling snowflakes and small objects. Then the road surface was extracted using the range weighted Hough/Radon transform. Finally the left and right edges of the road was detected by thresholding.

Tests have been made with a laser on top of a car driven in an automobile test range just south of the Arctic Circle. Moreover, in the campus area, the algorithms were tested in closed loop with the laser on board a robotized wheelchair.

1 Introduction

Detection of road boundaries are very important when running an autonomous vehicle on a road. During winter season when the landscape is covered with snow, plow machines remove snow from the road which create piles of snow on each side. These snow piles are natural boundaries for the road.

A snow covered road is constantly changing during the winter. Every time a snow plow passes the road to remove snow, the snow edges on the side of the road are moved a bit. The road surface is often rather rough, and it is expected to find small snow piles on the road, consisting of snow fallen from cars and trucks. During the winter one can also expect weather conditions that give limited visibility for a laser scanner, [1]. When developing sensing and algorithms for detection of the snow edges on the side of the road one has to consider all these conditions.

There are several different methods to detect the road boundaries using cameras and lasers, [2] and [3]. Some tests to navigate a robot in arctic conditions using a couple of different sensors have also been made, [4].

One contribution in this paper is an edge detection algorithm for snow edges. The robustness of the presented method is illustrated by the fact that we were able to run a robot, in the form of a wheelchair, on a road partly covered by snow. The robot was able to traverse a 200m long walking path several times. Even though the wheelchair was running on snow slush which made the wheels spin and the laser wiggle, it managed to correct its pose by aiming at a point a couple of meters ahead between the left and right snow edges. This test was made without a rate gyro. More information about this project is available at [5].

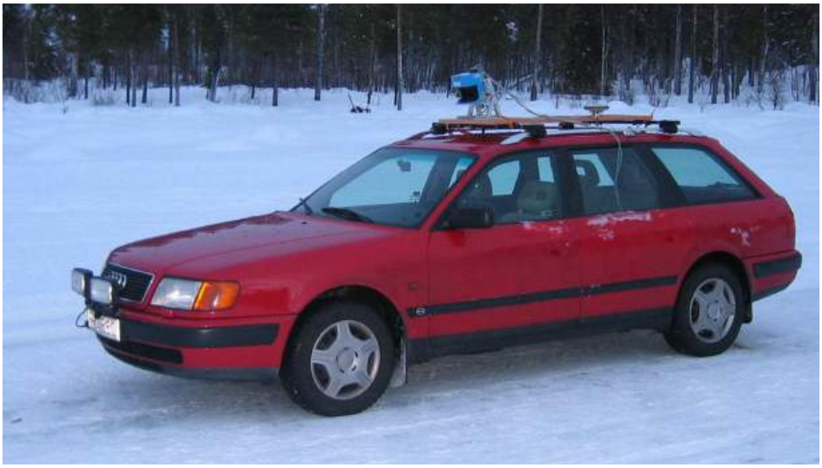

Figure 1: Testcar on a frozen lake used as an icetrack for automobile testing under arctic conditions. The laser sensor, tilted downwards, and a GPS antenna are mounted on the roofrack of the car.

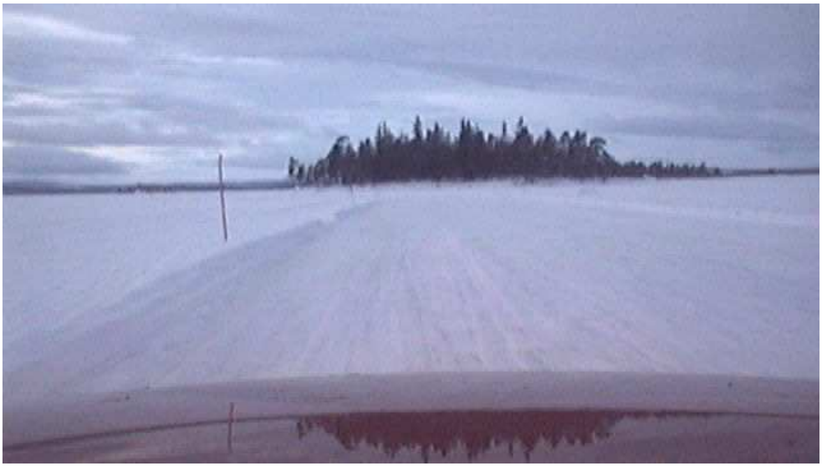

Figure 2: Part of the icetrack seen from the inside of the car. Note the vertical marker stick near the left boundary of the road.

1.1 Paper outline

Section 2 brings up the modelling and theoretical description for the edge detection method. Section 3 describes the test area and the equipment used in the development of the method. A description on how we closed the loop and made use of the edge detection algorithm to run a robot is given in Section 4. In section 5 we give a short description of laser measurements collected during a snowfall. Furthermore, Sections 6 and 7 contain our conclusions and proposed future work respectively.

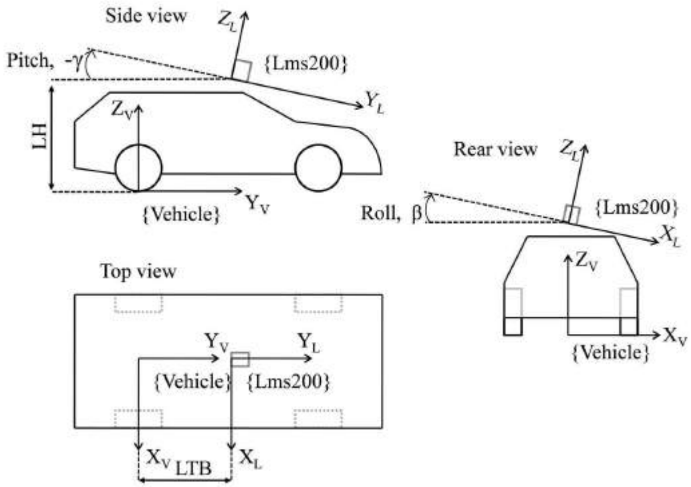

Figure 3: The coordinate systems on our test vehicle. LH is the mounting height of the laser, LTB is the distance from the laser to the rear wheel base on the vehicle. Pitch and roll angles are calculated around the X_L and Y_L axis respectively.

2 The snow edge detection method

The method presented in this section finds the snow edges on the sides of a road. This is done in a step by step process. Throughout the text we use the notation for coordinate systems described by Craig, [6].

First we detect the road surface. Then we calculate the roll and pitch angles of the laser. After that we transform the laser measurements from the laser coordinate system to vehicle coordinates, see Figure 3. Then the left and right edges on the road are found by thresholding. A block diagram describing these steps can be seen in Figure 4.

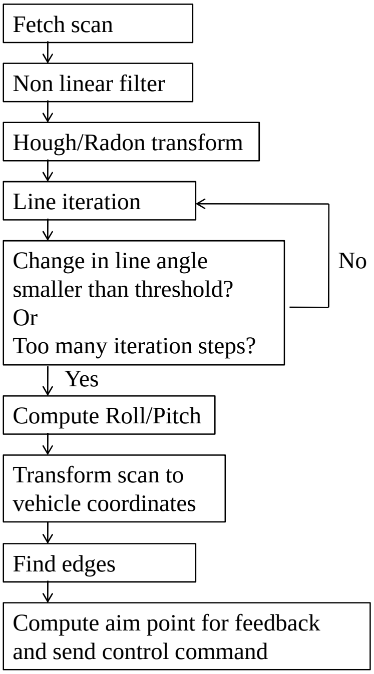
Fetch scan
Non linear filter
Hough/Radon transform
Line iteration
Change in line angle smaller than threshold? Or Too many iteration steps?
No
Yes
Compute Roll/Pitch
Transform scan to vehicle coordinates
Find edges
Compute aim point for feedback and send control command

Figure 4: The block diagram gives the different steps used in the snow edge detection algorithm, for the case of one laser scan. The last step is only used during closed loop testing of the algorithm.

2.1 Laser measurement on a vehicle, in general

The laser scanner Sick LMS200 was set to measure distance to surrounding objects in a sector of 180 degrees with an angle increment of 0.5 degrees. A complete scan consists of 361 measurements. The range span is $0 - 80$m with a resolution of 0.01m.

To be able to see the road and snow edges the laser scanner is mounted looking downwards with a small tilt/pitch angle. The angle has to be big enough so that the road surface is always visible in the laser scan. During testing an angle of at least $10°$ shows good results in both providing accurate road measurements, without limiting the range view in front of the vehicle too much.

The vehicle is modelled to run with all four wheels attached to a perfectly flat and horizontal surface, and the X_V and Y_V axis lying flat on that surface, see Figure 3. The Z_V axis is passing through the centre of, and the X_V axis is parallel to, the rear wheel axle. Relative position changes between {L} and {V} due to movements in the vehicle suspension are not considered, and thus the origins of the laser and the vehicle coordinate systems are assumed to be fixed to each other at distances LTB and LH. Pitch and roll angle changes of the laser are however considered and the angles are calculated around the laser X_L and Y_L axis respectively.

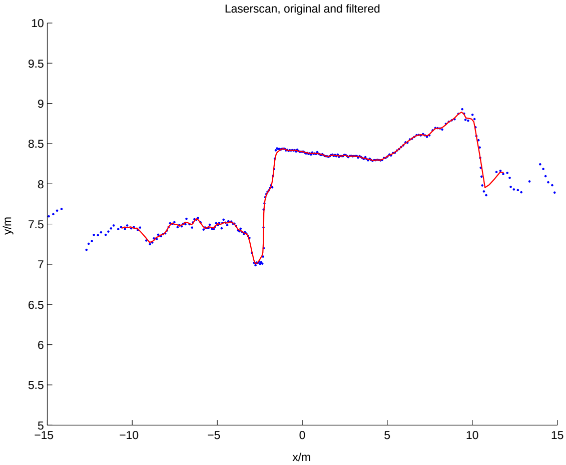

Figure 5: A example of a laserscan in laser coordinates from the icetrack. Note the different scales on the x and y axis. The dots are the actual measurements, and the line represent the filtered measurements used for ground and edge detection.

2.2 Laser measurement in a snowy environment

The surface of a snow covered road is rather rough. There are small piles of snow and other roughness on the road that disturb the laser readings. These disturbances are not of interest since these may also produce false snow edge detections and make it harder to detect the true snow edges. Therefore a filter was applied to smooth out the roughness and remove spurious measurements, for example falling snow. The drawback with this type of filter is that it may also remove real objects in the environment, like narrow marker poles on the side of the road and other small obstacles. This is not a serious problem in our application, since we are only interested in the snow edges.

First a median/histogram filter is used on the range data to remove spurious measurements from each scan. Then an averaging/mean value filter is used to smooth the scan. Both filters work on the range data in polar coordinates.

The median filter is of length 11, which means it can remove up to 5 consecutive deviant measurement in a window of 11 measurements.

The averaging filter is of length 3, i.e. it takes the average value of three consecutive measurements. To not affect large range jumps it uses a threshold; if the average value of the three measurements differ more than 0.5m from the median value of the same measurements, the actual data point is not filtered. In this way the average filter does not affect big range jumps in the laser scan. Figure 5 shows an example of how the filter affects a raw laser scan.

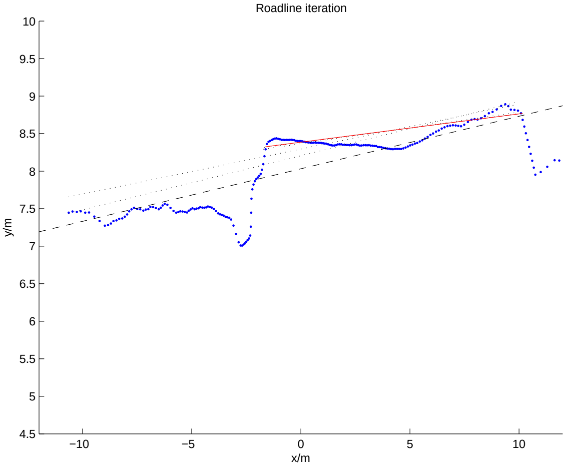

Figure 6: Roadline iteration in laser coordinates. The dashed line is the found Houghline, the dotted lines represent the iterative line steps. The solid line is the accepted roadline.

2.3 Road surface detection

To be able to detect the snow edges on the sides of the road we first need to find the actual road. The road is represented with a straight line, called the roadline. The search process for the roadline is done in the laser coordinate system.

The roadline is found in a iterative process. First the range weighted Hough/Radon transform [7] is applied to find a Houghline that is used as a first rough estimate of the roadline. Since we know where the road is expected to be, we can decrease the computation time of the Hough transform by limiting the distance and angle interval.

To find the angle of, and the distance to, the Houghline corresponding to the road, we first select all the peaks that lie within 90% of the maximum value in the Hough diagram. We then pick the distance to the peak that is furthest away as the distance to the line. If there are several peaks at that distance we select the angle from the highest, and if there is several peaks with the same height but at different angles, we take the average of those angles as the angle for the Houghline.

Measurements in a distance interval around the Houghline is selected, and a least square fit of a new roadline is made to those measurements. The roadline creation process is then repeated; measurements near the last found line are selected and a least square fit of a new line to those measurements is made, see Figure 6. When the difference in angle between the last line and the new line is smaller than a specified threshold, or the process has been repeated 10 times, the process is stopped and the roadline accepted.

The equation for the roadline in laser coordinates is

$$^{L}Y = A + B\ ^{L}X, \tag{1}$$

where A is the distance from the laser to the roadline along the Y_L axis, and B is the slope of the line. For illustrative purpose in the figures the lines fitted to a data set is only plotted between the projected end points in the data set.

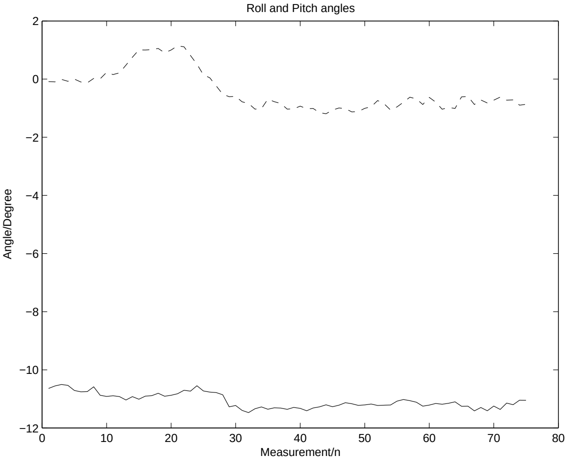

Figure 7: Calculated roll and pitch angles for 75 consecutive laser scans collected while the vehicle was moving from the right side of the road to the left side, see Figure 9. The dashed line represent the roll angle, and the solid line is the pitch angle. The big change in roll angle around scan 20 is due to the vehicle driving over the crest in the middle of the road, see Figure 8.

2.4 Pitch and roll calculations

To calculate the pitch angle of the laser we use the laser mounting height LH, and the distance A between the laser and the roadline along the Y_L axis. Since the vehicle is modelled to run on a perfectly flat surface the detected roadline should have a slope of zero degrees, if the slope differs from zero it must be due to a roll angle of the laser, the distance A to the line is not affected. We also assume the mounting height LH from the ground to the laser to be constant. The laser pitch angle, γ, is then calculated with

$$\gamma = -arcsin(\frac{LH}{A}), \tag{2}$$

and the roll angle, β, of the laser is given as

$$\beta = arcsin(B\ tan(\gamma)). \tag{3}$$

A example of calculated roll and pitch angles is seen in Figure 7.

2.5 Coordinate transformation

Each laser scan is transformed into the vehicle coordinate system using

$$^{V}X = \ ^{L}Xcos(\beta), \tag{4}$$

$$^{V}Y = \ ^{L}Xsin(\gamma)sin(\beta) + \ ^{L}Ycos(\gamma) + LTB, \tag{5}$$

and

$$^{V}Z = - \ ^{L}Xcos(\gamma)sin(\beta) + \ ^{L}Ysin(\gamma) + LH. \tag{6}$$

Where LTB is the distance from the laser to the wheel base of the vehicle, and LH is the mounting height of the laser, see Figure 3.

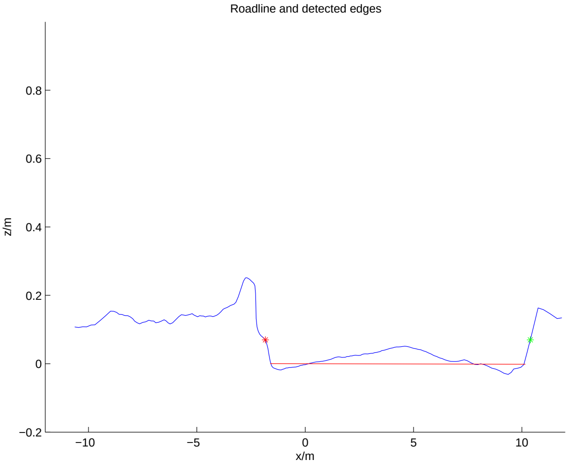

Figure 8: Roadline and detected edges in the X-Z plane seen from the vehicle. Observe the different scales in the figure. The road is about 12m wide while the snow edges on the side are approx. 0.2m high.

2.6 Edge detection

The edge detection algorithm looks at the laser data in the vehicle X_V and Z_V coordinates, see Figure 8.

The edges are detected where two consecutive measurements lies on separate sides of an imaginary threshold line, placed at certain distance above, and parallel to, the roadline. Measurements with a Z value below the threshold line are assumed to belong to the ground, and the ones above belong to the surrounding boundaries.

This algorithm may find several left and right edges. To identify the left and right bounding edges of the road we need to know where the actual road is. An estimate of the centre position of the road is computed by calculating the average position of all the measurements that the roadline is created from. When we have the centre position we can find the right and left edges corresponding to the road. This is done by selecting the closest left edge on the left side of the centre point as the detected left edge, and the closest right edge on the right side of the centre point as the detected right edge.

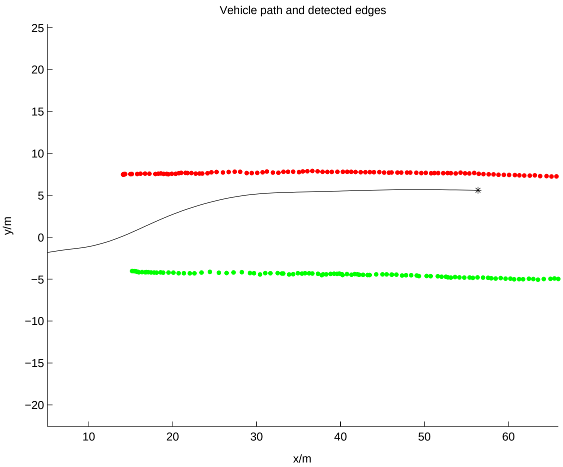

Figure 9: Detected edges when the vehicle is traversing an icetrack on a frozen lake. The curve represent the vehicle path, estimated using odometry and a low drift fiber optic gyro. The vehicle was travelling from the left to the right in the figure. The dots are the detected left and right edges on the road. Due to the tilt angle of the laser the detected edges is approx. 10m in front of the vehicle.

3 Test area and equipment

In this section we present a description of the test area and the test equipment used in the development of the algorithms.

3.1 Test area

The snow edge detection algorithm has been tested on data recorded at an icetrack located in the village Arjeplog in the northern part of Sweden. The icetrack is part of a large winter test facility for different car manufacturers. A part of the track is seen in Figure 2.

Since the test track is located on an ice covered lake, the assumption made that the vehicle is moving on a flat horizontal surface is fairly good. Another thing to notice is the

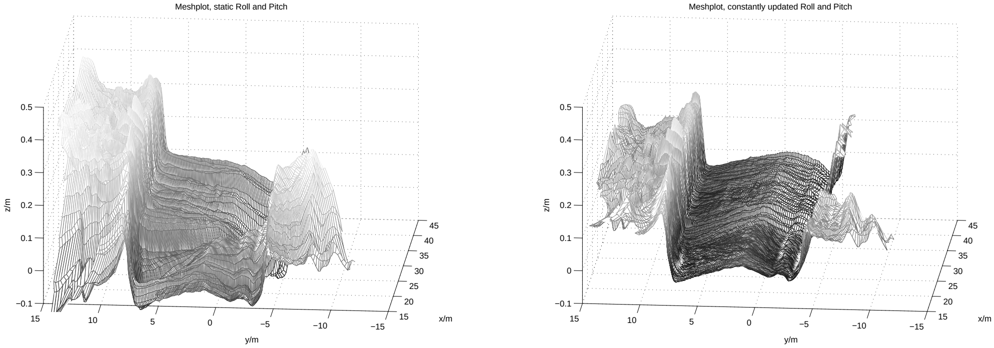

Figure 10: Two different meshplots of the same 25m part of the icetrack as seen in Figure 9, the first part where the vehicle travels from the right to the left side of the road. In the left figure the laser scans are rotated and translated using fixed roll and pitch angles. In the right figure all single scans are rotated and translated using the estimated roll and pitch angles, as described in the paper. The scans are put together using dead reckoning.

fact that the actual road the vehicle is moving on is the lowest part in the environment. There are no ditches or holes on or beside the road, only snow and ice.

3.2 Test equipment

The test vehicle was an ordinary car, equipped with some sensors, see Figure 1: GPS receiver, wheel encoder, web camera, fiber optic rate gyro, and a laser scanner. All connected to a computer that recorded data.

The laser scanner was mounted on the roofrack on top of the car at a height, $LH = 1.65$m, and a distance, $LTB = 1.5$m from the rear wheel base. The tilt/pitch angle of the laser was approx. 11^o.

3.3 Test description

All the figures described in this section is from the same part of the testrun. The laser scans are stitched together using dead reckoning based only on the low drift fiber optical rate gyro and the wheel encoder.

Figure 7 shows the calculated pitch and roll angles for the testrun. Figure 9 shows the detected left and right edges of the icetrack as the car was driving.

Figure 10, presenting a three dimensional meshplot made from laser scans rotated using fixed roll and pitch angles shows a good example on how the motion of the car is affecting the resulting laser measurements. The influence of the vehicle motion is noticeable reduced when using the estimated roll and pitch angles, see Figure 10.

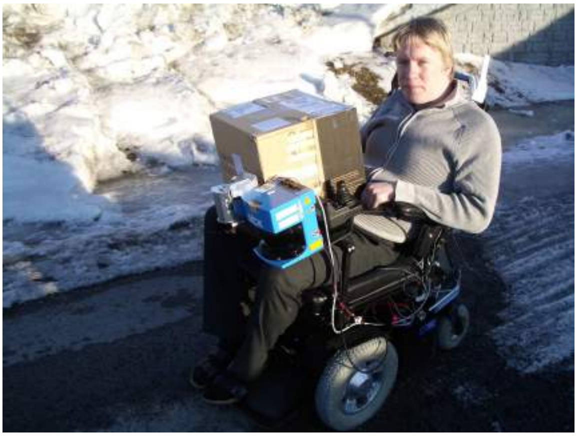

Figure 11: Robot used as platform during the closed loop testing. The LMS200 laser is seen in the front, the box behind it is a sunscreen for a laptop. The only function of the person is emergency stopping.

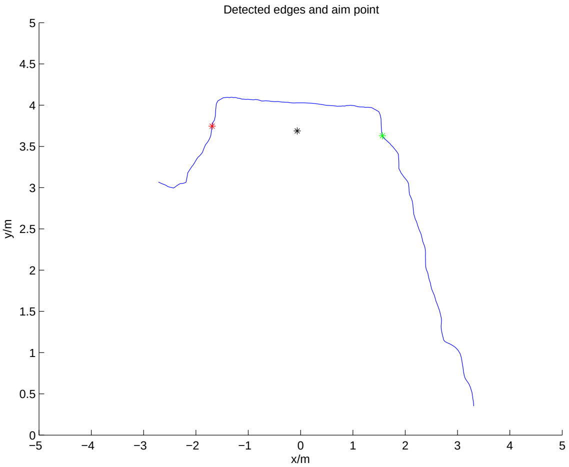

Figure 12: Detected left and right edges in one laser scan from the closed loop testrun. Note that the plot is in the X-Y plane in vehicle coordinates. The star in the middle is the aiming point for the robot, (the "rabbit"). It is located in the middle between the left and right edges.

4 Closing the loop

A closed loop test of the edge detection algorithm was done by letting a robot follow the snow edges as estimated from the laser. The robot used in the test was a wheelchair, equipped with the necessary sensors and a computer, see Figure 11. In this test the wheelchair was not stabilized by any gyro while driving, (gyro not functional). Under these conditions the dead reckoning is very poor, especially when driving in snow slush.

To test feedback from the edge detection algorithm we used a rather simple control law. The robot was driving with constant speed, aiming at a point a couple of meters ahead, in the middle between the estimated left and right edges, see Figure 12. If one of the edges was not found, the robot continued to drive straight forward. This control law made it necessary to always be prepared to manually stop the robot if the algorithm failed. Unwanted behaviour did occur when no snow edges where detected by the robot.

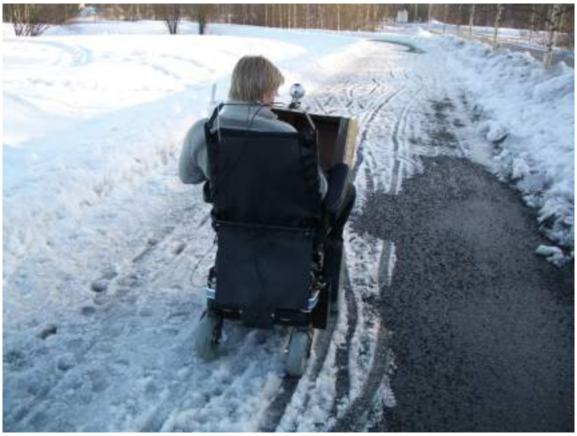

Figure 13: The robot driving autonomously through snow slush on a plowed walking path in the campus area.

The robot was able to traverse an approx. 200m long walking path forth and back several times. Even though the wheelchair was running through snow slush which made the laser wiggle and the wheels spin, see Figure 13, it managed to correct the position and heading just by driving towards the aiming point a couple of meters ahead between the left and right snow edges.

5 Laser measurements during snowfall

Work is in progress towards model based filtering to reduce the clutter from snowflakes for outdoor robots. Below we give a hint of properties using a LMS200 laser scanner during snowfall. A wall, see Figure 14, approximately 20m away from the laser was used to get range statistics in a sector of 13.5^o. From 13700 range registrations approximately 19% were from snowflakes, 81% fell on the wall behind, and about 0.2% were not detected i.e. absorbed.

The histogram in Figure 16 gives the range distribution to detected snowflakes. There is a significant peak close to 1 m and almost no snowflakes are detected beyond 8 m. In another part of the scene was a wall with range changing from 1 m to 10 m. For

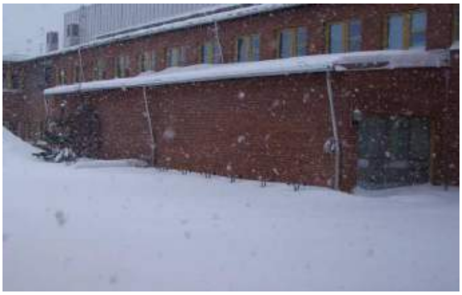

Figure 14: A part of this wall was used as reference object to get statistics about percentage of false detections by snowflakes, detections on the object behind, and no detections. It is possible to see falling snowflakes in the picture.

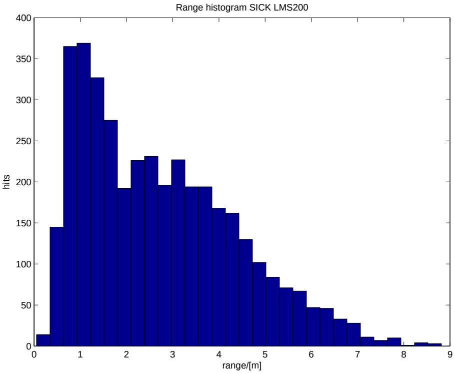

Figure 15: Histogram of detections of snowflakes from laser measurements for the scene in Figure 14. Note that there are few detections close to the laser. For clarity, the wall located approx. 20m away from the laser is left out of the histogram.

objects close to the laser the shape of the histogram changes significantly. This is due to properties of the detector electronics.

A second laser with a narrow modulated beam was also used during the tests. This short range narrow beam laser has a better penetration between the snowflakes and gives a histogram with very few detections triggered by snowflakes. The continuous wave detector is also different.

There are many relevant parameters to study including beam divergence, modulation and detector properties, snowflake size and wind motion.

Work in progress can be summarised as:

- Median filtering gives a substantial reduction in disturbing snowflake detections.
- Histograms on projections indicate that the Radon/Hough transform will give a robust extraction of plane surface segments such as roads.
- The gamma distribution describes fairly well the distribution of detected snowflakes, see Figure 15.

6 Conclusion and discussion

The methods presented in this paper can be used to detect snow edges on the sides of a road. It is also possible to continuously calculate and update roll and pitch angles of the laser, which is very useful when stitching together several laser scans.

We tested the algorithms by using the detected snow edges as references to run a robot, and it did work, even though the surface on which the robot was running was very rough and slippery.

7 Future work

An error analysis of the roadline detection needs to be done. How does an error in the roadline propagate into the estimated roll and pitch angles?

Another noticed thing; if the roadline detection fails to detect a proper roadline the whole edge detection algorithm breaks down. An extension to the presented methods could be to apply a Kalman filter to track the roadline and roll/pitch parameters, and also include IMU information in this filter. This would probably reduce the risk of total failure in edge detection.

The disturbances of falling snow should be studied in more detail.

An interesting method to test would be the circle sector expansion method, [8]. By removing the measurements that belong to the road surface, it is possible to use that method to find the free space and the road boundaries.

8 Acknowledgement

This work was funded by the Center for Automotive System Technologies and Testing, CASTT.

References

[1] N. Vandapel, S. Moorehead, W. Whittaker, R. Chatila, and R. Murrieta-Cid, "Preliminary results on the use of stereo, color cameras and laser sensors in antarctica," in *International Symposium on Experimental Robotics*, March 1999.

[2] W. Wijesoma, K. Kodagoda, and A. Balasuriya, “Laser and vision sensing for road detection and reconstruction,” in *The IEEE 5th Internationa Conference on Intelligent Transportation Systems, 2002. Proceedings.*, (Singapore), 2002.

[3] Z. Xu, “Laser rangefinder based road following,” in *Mechatronics and Automation, 2005 IEEE International Conference*, (Niagra Falls, Canada), July 2005.

[4] S. Moorehead, G. R. Simmons, D. Apostolopolous, and W. Whittaker, “Autonomous navigation field results of a planetary analog robot in antarctica,” in *ESA SP-440: Artificial Intelligence, Robotics and Automation in Space*, aug 1999.

[5] L. U. of Technology, “Mobile robots.” http://www.csee.ltu.se/∼hf/mobilerobots, Aug 2007.

[6] J. Craig, *Introduction to Robotics: Mechanics and Control.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[7] J. Forsberg, U. Larsson, and Å. Wernersson, “Mobile robot navigation using the range-weighted hough transform,” *Robotics and Automation Magazine, IEEE*, 1995.

[8] S. Rönnbäck, T. Berglund, H. Fredriksson, and K. Hyyppä, “On-line exploration by circle sector expansion,” in *IEEE International Conference on Robotics and Biometrics - ROBIO 2006*, 2006.

Paper C

Circle Sector Expansions for On-Line Exploration

Authors:
Sven Rönnbäck, Tomas Berglund, Håkan Fredriksson, Kalevi Hyyppä

Reformatted version of paper originally published in:
Proceedings of the IEEE International Conference on Robotics and Biomimetics 2006, Kunming, China

Circle Sector Expansions for On-Line Exploration

Sven Rönnbäck, Tomas Berglund, Håkan Fredriksson, Kalevi Hyyppä

Abstract

A novel and effective method denoted *circle sector expansion (CSE)* is presented that can be used to generate reduced Voronoi diagrams. It is intuitive and can be used to efficiently compute possible paths for a vehicle. The idea is to model free space instead of the features in the environment. It is easy to implement and can be used while a vehicle moves and collects new data of its surrounding. The method is directly applicable and has properties for fast computations of safety margins while at the same time having low complexity.

We have successfully implemented the algorithm and its methods and performed real-life tests using an autonomous wheelchair equipped with a range scanning laser, a rate gyro, and wheel-encoders. Tests showed good results for supporting the use of CSE. The results are applicable for example to improve assistive technology for wheelchair users.

1 Introduction

[1]The problem of creating a Voronoi diagram from a set of N points has been widely studied. In 1987 the Fortune sweep-line Voronoi algorithm was published. It computes the Voronoi diagram in $O(NlogN)$ time [1]. The sweep-line algorithm creates a complete Voronoi diagram and the diagram needs to be reduced for efficient use in mobile navigation [2, 3, 4]. The generalized form of Voronoi diagram calculation requires that the environment is parameterized in form of polygon lines [5, 6]. The problem of identifying polygon lines from range data has lots of association problems. Voronoi skeletons are interesting for navigation and can be used for localization [7, 8, 9, 10, 11].

The mentioned methods either require as input some pre-calculated parametrization of the environment, grid based methods, or they compute a full Voronoi diagram. The latter is a non-compact representation not always suitable nor easily used for practical purposes.

We present a novel and effective on-line method for exploring an environment. It is denoted *circle sector expansion* (CSE) and can be used while a vehicle is moving and at the same time gathers new information from its surrounding. It is [12] a wavefront based method and CSE is an iterative process where consecutive expanded circle sectors are bounded by collected range data, Figure 1. The polygonal connected origins of the circle sectors build an approximate reduced Voronoi diagram.

[1] This work is supported by the EU program Interreg IIIA Nord

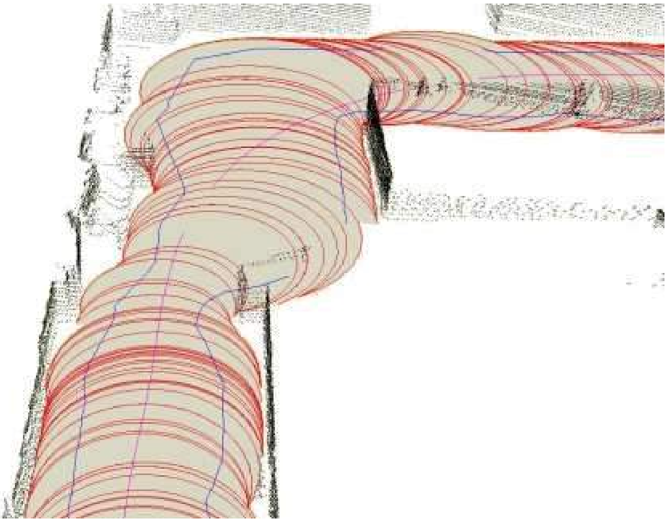

Figure 1: The circle sector expansion method (CSE) is used to find available free-space for an autonomous wheelchair equipped with a ladar. The consecutive circle sectors spans up the available free-space. Five interesting things are found in the figure: 1) Safety boundaries created from the circle sectors are the polygons that follows the contours of the walls. 2) An estimated trajectory of the wheelchair is visible as the smooth line plotted on the floor. 3) A vertical pipe is mounted on the wall near the lower right corner. 4) The places where the contour polygons are discontinuous are gateways which are wide enough for the wheelchair to traverse. 5) Near the upper left corner a bench and some boxes are visible in the laser data.

Instead of focusing on the geometry and properties of obstacles, the CSE method puts the focus on free space and places and directions that the vehicle is able to traverse.

It can be used in an unknown environment and can handle difficult passages as well as discover possible paths from a certain location based on vehicle width. The length of a circle sector chord decides whether it is possible or not for the vehicle to reach a certain position in front of it.

The consecutive circle sectors are also used to rapidly calculate safety margins such that the vehicle does not touch obstacles.

2 The Circle Sector Expansion Method

It is a novel approach and can be used for realtime obstacle avoidance and to compute safety margins for a robot in a two dimensional workspace.

The fundamental idea of the method is to find the bounding point for an expanding circle sector, visible in Figure 2.

A bounding point splits the current sector into two new circle sectors and the process is repeated.

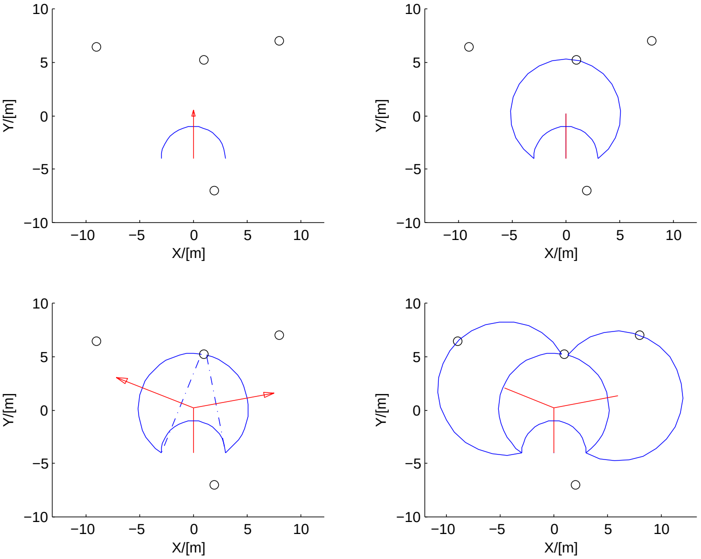

Figure 2: Example of an expansion sequence. Four points are represented as the small circles. 1) The initial condition before expansion where the arrow indicates the direction of expansion. 2) The forward expansion is bounded by a point. 3) The expanded sector is divided into two new sectors. The new expanding directions are marked with arrows. 4) The result after three sector expansions.

If no bounding point is found the radius is undefined; in such a case the radius is set to a fixed value.

Each circle sector indicates a direction to free space, see Figure 1. If the length of a circle sector chord is larger than the width of the vehicle plus the safety margin, it can be part of a possible path for the vehicle. The output from the circle sector expansion is consecutive circle sectors.

2.1 The CSE Expansion Process

Figure 3 shows more in detail an example of an expansion sequence where three points, marked as *3*,*4* and *5* are present in front of the starting sector. The expansion process starts at the sector denoted *c1*. The vehicle is represented as the thick circle. The thick arrows mark the expanding directions of the circle sectors. The dashed circle arcs are possible circle sectors candidates. Points are present inside the dashed circles and therefor they are disqualified as circle sectors for the expansion process. The largest expanded empty circle sector is always selected in each iteration. When the largest chord is selected we get the expansion sequence *c1*,*c2*,*c3* and *c4*. The connected chords generates the Delaunay triangles.

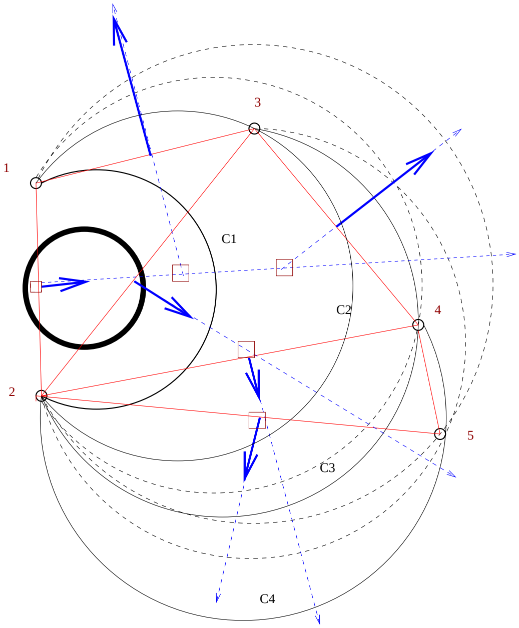

Figure 3: An illustration of an expansion sequence in an environment with three points present to the right of the starting chord defined by the points 1 and 2. The solid arrow marks the expanding directions of the circle sectors. Chord 4-5 is to short for the vehicle to traverse. The dashed straight lines build the Voronoi graph. The squares are origins for the expanding circle sectors.

2.2 Circle Sector Parameters

Circle sector $sec(k)$ is defined by five parameters, $sec(k) = \{x_c(k), y_c(k), r_c(k), \theta_c(k), \beta_c(k)\}$, where k is the iteration step. They are visualized in Figure 4, where $\theta_c(k)$ is the heading of the circle sector, $\beta_c(k)$ is the sector angle, $r_c(k)$ is the radius, and $z_c(k)$ is the origin of the circle.

The point that lies on the circle arc and bounds the circle sector is the complex number $z_2(k)$. To calculate the heading of the two new circle sectors we first perform a parametrization, called $\lambda_c(k)$, of point $z_2(k)$ along the sector arc,

$$\lambda_c(k) = \frac{1}{2} + \frac{\angle[(z_2(k) - z_c(k))] - \theta_c(k)}{\beta_c(k)}.$$

The notation $\angle z$ returns the angular value of the complex number z.

Since point $z_2(k)$ splits the circle sector into two new sectors, $\lambda_c(k) \in (0, 1)$. If $\lambda_c(k) \equiv 0.5$ the new sectors are equal in size. If $\lambda_c(k) < 0.5$ the left sector is larger than the right one. The sector angles for the new circle sectors are

$$\beta_{cR}(k) = \beta_c(k)\lambda_c(k),$$

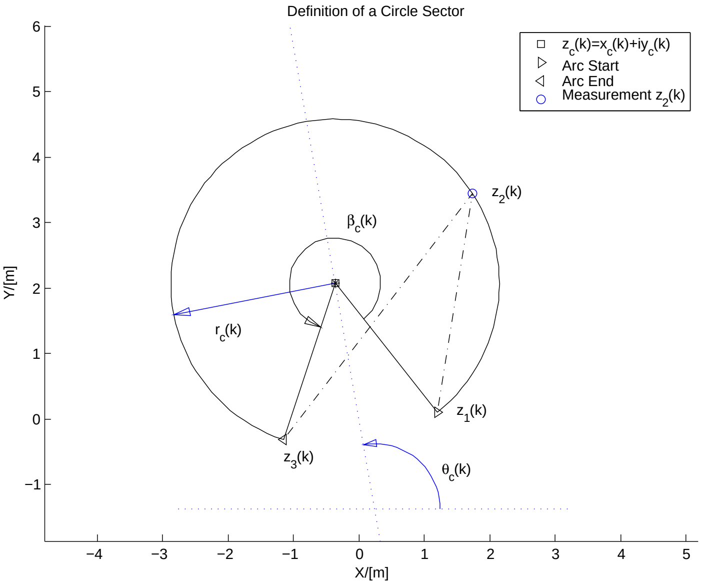

Figure 4: A circle sector with sector angle $\beta_c(k)$, radius $r_c(k)$, heading angle $\theta_c(k)$ and circle origin $z_c(k)$. The laser measurement $z_2(k)$ divides the sector into two new sectors. The dashed lines between $z_1(k), z_2(k)$ and $z_2(k), z_3(k)$ are the chords for the new circle sectors.

and

$$\beta_{cL}(k) = \beta_c(k)(1 - \lambda_c(k)).$$

The heading angles for the sectors expanding in the right and left direction are described by equation

$$\theta_{cR}(k) = \angle[z_2(k) - z_c(k)] - 0.5\beta_{cR}(k)$$

and

$$\theta_{cL}(k) = \angle[z_2 - z_c(k)] + 0.5\beta_{cL}(k),$$

respectively. The heading angles are expressed in vector form as $e^{i\theta_c(k)}$.

2.2.1 Given a Desired Target Position

Given a target position z_t the expansion process can proceed to the left or the right relative to split point $z_2(k)$. A bearing angle is calculated relative to the vector that points towards the bounding measurement $z_2(k)$,

$$\gamma_t(k) = \angle[(z_t - z_c(k))\left(\overline{z_2(k) - z_c(k)}\right)], \tag{1}$$

If $\gamma_t(k)$ is positive the left sector is selected.

2.3 Calculation of the Circle Sector Parameters

The output of the CSE method are consecutive circle sectors. Below follows a description on how to compute the parameters of a circle sector.

A circle sector with the parameters z_c, r_c, β_c and θ_c is bounded and defined by the complex numbers z_1, z_2 and z_3 shown in Figure 5.

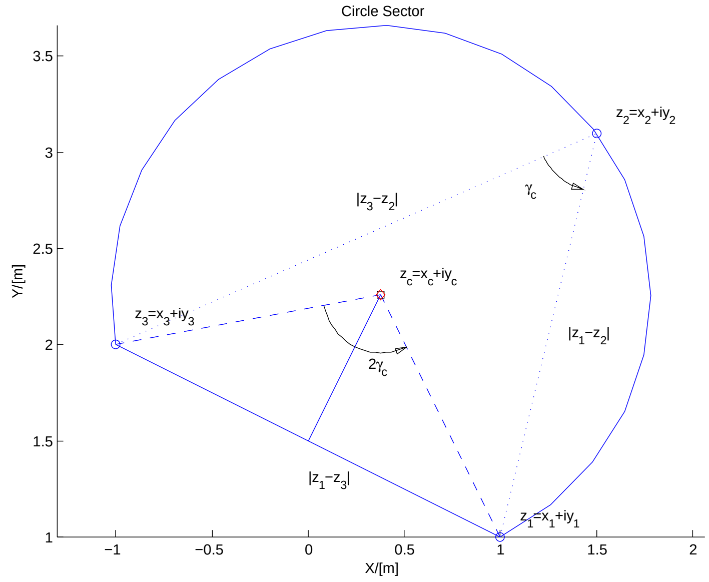

Figure 5: The origin of the circle sector, z_c, can be calculated if the three bounding points are known. The theorem of inscribed angles in a circle is applied to get an expression for z_c.

The origin, z_c, is calculated by translating the chord mid point, $(z_1 + z_3)/2$, along the vector that is orthogonal to the chord and points towards z_2.

The circle sector is undefined if z_1, z_2, and z_3 lie on a line. The inscribed angle γ_c is found by taking the argument of the complex number z_v, i.e. $\gamma_c = \angle z_v$, where the complex number z_v is calculated with

$$z_v = (z_1 - z_2)\overline{(z_3 - z_2)},$$

where $\overline{(z_3 - z_2)}$ is the complex conjugate of $(z_3 - z_2)$.

The expression for the circle sector origin z_c is

$$z_c = \frac{(z_1 + z_3)}{2} + \frac{(z_v + \overline{z_v})(z_1 - z_3)e^{i\pi/2}}{2(z_v - \overline{z_v})}.$$

The heading angle of the circle sector is $\theta_c = \angle[(z_1 - z_3)e^{i\pi/2}]$. In the implementation θ_c is computed using the *atan*2 function.

The sector angle β_c is calculated from the bisector angle, namely $\beta_c = 2\,(\pi - \gamma_c)$. The radius r_c for the circle bounded by z_2 is

$$r_c = \sqrt{(x_c - x_2)^2 + (y_c - y_2)^2}.$$

In the expansion process the length of the chord, $|z_1 - z_3|$, is compared to a threshold t_{ch}. If a chord is less than the threshold the expansion in corresponding direction is stopped.

2.4 Collision-Free Path and safety Margins

Given a set of $\{sec(1), sec(2), ..., sec(N_{sec})\}$ consecutive circle sectors, , has enough information to not only calculate a collision-free path, but also a left and a right safety margin for the vehicle. The set of circle sectors is used to calculate three polygons. One polygon is created by connecting the origins of the circle sectors,

$$path_C = \begin{bmatrix} x_c(1) & x_c(2) & \cdots & x_c(N_{sec}) \\ y_c(1) & y_c(2) & \cdots & y_c(N_{sec}) \end{bmatrix},$$

which forms a collision-free path for the vehicle. One polygon for the left safety margin,

$$margin_L = \begin{bmatrix} z_c(1) + (r_c(1) - m_w)e^{i(\theta_c(1)+0.5\beta_c(1))} \\ z_c(2) + (r_c(2) - m_w)e^{i(\theta_c(2)+0.5\beta_c(2))} \\ \vdots \end{bmatrix}^T,$$

and one safety margin to the right,

$$margin_R = \begin{bmatrix} z_c(1) + (r_c(1) - m_w)e^{i(\theta_c(1)-0.5\beta_c(1))} \\ z_c(2) + (r_c(2) - m_w)e^{i(\theta_c(2)-0.5\beta_c(2))} \\ \vdots \end{bmatrix}^T.$$

The desired safety margin, the distance to nearest obstacle, is denoted m_w. An example of calculated safety boundaries is visible in Figure 1

Where the gateways to free-space are wide enough, the circle sector expansion continues. In a place where the expansion can proceed in two directions, the circle sector is divided into two parts, see Figure 6. Each part form a new circle sector and the possibility to pick a desired direction, either to the left or to the right. The radii of the unbounded sectors are limited to a default value.

2.5 Special Cases of the CSE Method

There is a common interest to obtain obstacle avoidance in a realtime system [13]. The circle sector expansion has the advantage that it can perform obstacle avoidance from collected range data. The method returns a collision-free navigation area for the vehicle with safety margins included. In Figure 6 the safety margins are plotted as dashed lines.

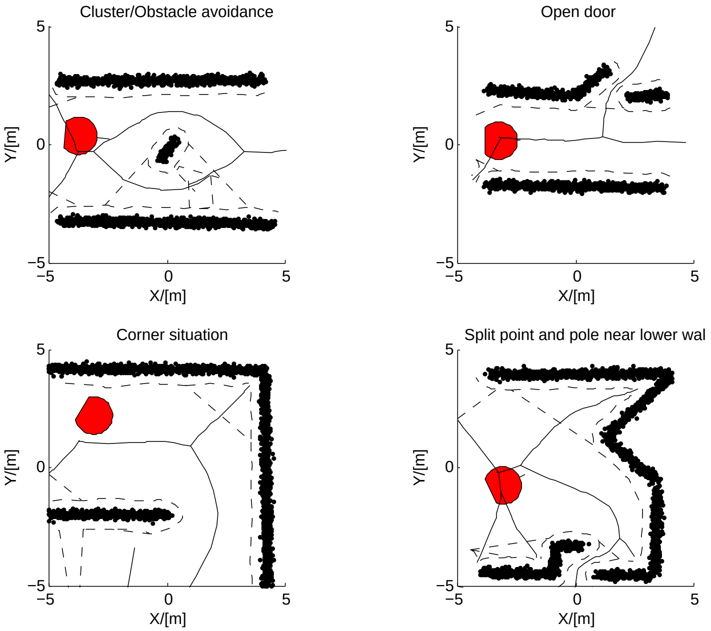

Figure 6: Four examples of approximate Voronoi diagrams created by the circle sector expansion method. 1) Shows an example where an obstacle is present in a corridor. The method finds two possible paths for the vehicle to take. 2) A corridor environment with an open door. The method finds a path through the doorway. 3) The Voronoi diagram generated in a typical corner situation. 4) Shows a typical dead-end situation.

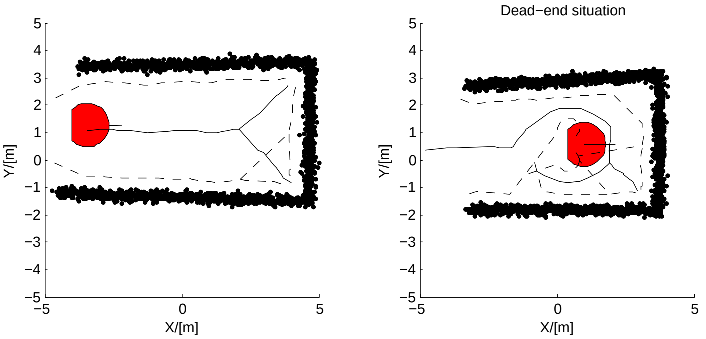

Figure 7: A typical dead-end situation. The vehicle will in a normal situation get trapped in one of the corners. To deal with this problem the circle expansion method is used. Two possible return paths are found. The dashed line is the calculated safety margin for the robot.

If the robot reaches a dead-end situation, which means that the vehicle either must reverse or turn 180^o, the solution is the following. As soon as a dead-end situation is identified (no further circle sector expansion is possible) the circle sector expansion restarts at the vehicle pose. The output is a possible path for the vehicle out of the trapped dead-end situation, as shown in Figure 7.

Some interesting properties of the CSE method;

- It is suitable for implementations on mobile robots.
- It reduces the number of points at the boundaries and the points needed to describe the available free space.
- The expansion process can start or continue at an arbitrary pose.
- It can be used to detect clusters of points. If a cycle is detected a cluster has been found and can be isolated. The points of a cluster can be sorted out by the surrounding polygon using the "Crossing Count" algorithm[2][14]. The "Crossing Count" algorithm works as follows; given a point and a polygon the question is to decide wether the point lies inside the closed polygon.
- It can be used to find a path through a narrow passage such as a doorway.
- It can be used to determine if paths leads to dead-ends.
- It can be used to find paths around obstacles.

3 Implementations and Tests

This section is devoted to describing our test environment, the system built for the tests, as well as showing how our tests have successfully been performed. As is emphasized earlier in this paper; in addition to the theoretical proposal for solving the problem of exploration, we have also implemented and tested the circle sector expansion method. The CSE implementation can together with the MICA software be used on-line and is what we denote a *soft* real-time system[15].

3.0.1 Speedup of Implementation

In the implementation the sector angle can be set to $\beta_c(1 - \epsilon_{\beta_c} \sin(\beta_c/2))$ where ϵ_{β_c} is a small number. This will make the method faster to traverse data where the sector angles are close to 180^o; such places are corridors.

3.0.2 Detection of a Loop

Check if any of the previous found circle sectors, $sec(1)$ to $sec(k-1)$, is build of any combination of $z_1(k)$, $z_2(k)$, and $z_3(k)$. Example; if the index numbers $673, 741, 981$ are found as combination $981, 673, 741$ a loop is detected. This will not work with the above described ϵ implementation.

[2] http://www.visibone.com/inpoly/

3.1 Hardware used in Test

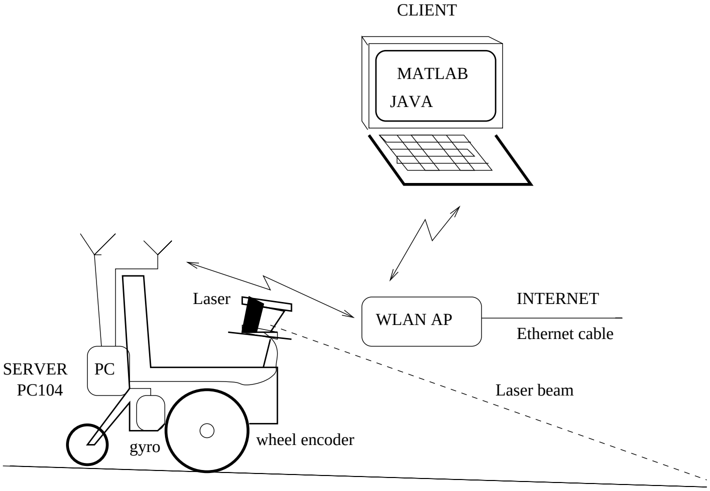

Figure 8: The vehicle (a wheelchair) has an onboard PC104 based computer running Linux. The PC104 collects data from the gyro, ladar, and wheel encoders. It is connected to a wireless access-point (AP). A client computer can through socket communication connect to the server software and request sensor readings. The client computer runs MATLAB *that process the sensor readings and computes control commands that are sent back the wheelchair. The laser is mounted on a pitched table. The laser beam hits the floor at a distance of approximately* 4 *m in front of the vehicle.*

The vehicle is a battery powered wheelchair which is equipped with sensors, a computer, and a wireless link to enable remote communication. It can either be locally, remotely controlled, or can work autonomously. It has wheel-encoders, one for each front wheel, for measuring the change in angle of the wheels. Its steering depends directly on the angular change of its front wheels, while its rear wheels are free and only follows the movement given by the front wheels. The vehicle can continuously change its velocity and angular velocity. The width of the vehicle is that of an ordinary wheelchair, approximately 0.65 m, which means that it can pass through regular doors.

It is also equipped with a rate gyro that, together with the encoders, is used for dead-reckoning of the vehicle pose.

The vehicle updates its position with a frequency of approximately 25Hz. The single-axis KVH-Ecore 1000 rate gyro delivers data with a frequency of 10Hz.

The wheelchair has a range scanning laser for environment detection. It is positioned at a height of approximately 0.75 m from the floor and it is pitched down approximately 10^o and measures a sector of 180^o, Figure 8.

Each scan consists of 361 range data points that are transformed into 3D cartesian coordinates since the laser is tilted down. Due to the pitch, the forward sight of the laser is limited to 4 m. Points within a distance 0.1 m from the ground are deleted in preprocessing of the range data.

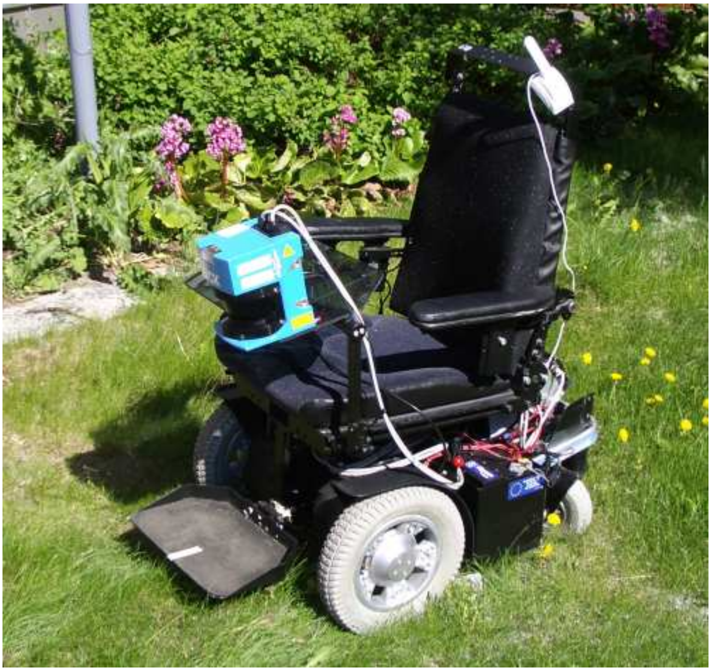

Figure 9: The wheelchair used in the tests. It is equipped with a pitched range scanning laser, a rate gyro and two wheel-encoders. It carries an embedded x86 computer running RedHat Linux 7.3 which is used to collect and time stamp sensor readings. It controls the wheelchair. Clients can connect to the on-board computer over a wireless link and request sensor readings and send control commands. Using the implemented circle sector expansion method, the wheelchair was able to explore an unknown environment using the MATLAB environment on a standard Laptop computer running Windows-XP.

3.2 Start Setting

We consider a two-dimensional environment and a circle shaped vehicle with fixed radius. The start-up setting is such that the vehicle is standing still for some seconds while it scans its surrounding.

The initial pose of the vehicle $[x_w(0), y_w(0), \theta_w(0)]^T$ is $[0, 0, 0]^T$. The starting condition for the first circle sector is the vehicle pose, $x_c(1) \equiv x_w, y_c(1) \equiv y_w, \theta_c(1) \equiv \theta_w$, and $\beta_c(1) \equiv 1.5\pi$. The start-up circle chord is defined by $z_1(1)$ and $z_3(1)$, $z_1(1) = z_c(1) +$

$r_c(1)e^{i(\theta_c(1)+\beta_c(1)*0.5)}$ and $z_3(1) = z_c(1) + r_c(1)e^{i(\theta_c(1)-\beta_c(1)*0.5)}$. Using collected range data, a sequence of circle sectors are computed. In the circle sector expansion process the length of the chords are checked. The chord threshold is set to $t_{ch} \approx 0.8$ m, which is vehicle specific and means that found passages and free space is large enough for the wheelchair to reach. If the chord length is above the threshold the sector expansion process continues. From the computed consecutive circle sectors, polygons for the safety boundaries are calculated.

The margin m_w is set to the 0.4m to get obstacle clearance.

3.3 The Test Environment

Our tests are done in a part of the underground basement and corridors of Luleå University of Technology. The environment is a fairly general surrounding. It contains geometries such as straight corridors, crossing, and different kinds of obstacles with various shape complexity.

3.4 Description of the On-Line Exploration

On the client computer a background process reads position estimates and compares it against a desired path for the vehicle. The implemented path following algorithm can be found in [16] and [17]. After receiving commands from the control thread the vehicle starts to move. The circle sector expansion guides the vehicle through the environment. If computations become so slow that the vehicle is on the verge of colliding with obstacles an implemented watchdog function on the onboard computer detects it and automatically halts the vehicle. If no new driving commands are received by the onboard computer the wheelchair will automatically stop. The circle origins form a polygon for the vehicle to follow. The circle expansion gives the possibility to navigate and make a selection, go left, go right, follow the widest route, or follow the chord that leads towards a selected targeted position.

In exploration mode the circle expansion method is used to build a representation of the navigation area.

An example of a 2D environment that has been explored using circle sector expansion is shown in Figure 10 which is grabbed from the visualization implementation that uses the Open Scene Graph 3D engine [18]. The upper left corner of Figure 10 is visible in Figure 1. The exit upwards is a passage that leads from the A-building to the α-building at Luleå University of Technology.

4 Conclusions

We address the problem of driving an autonomous wheelchair in an unknown environment using a method based on *circle sector expansions* (CSE) on data collected by a range scanning laser. We propose an algorithm based on expanding circle sectors that calculates a reduced Voronoi diagram which is used for detecting available free-space. The CSE)

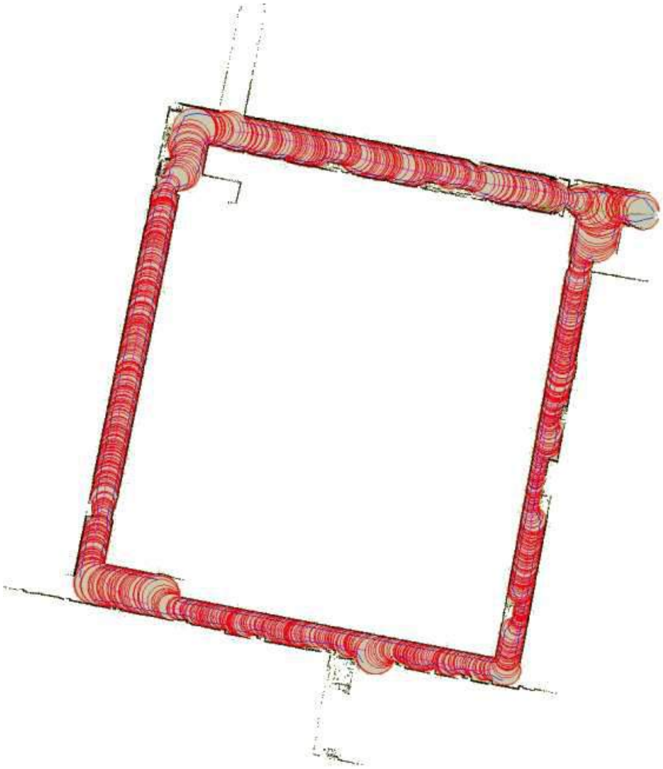

Figure 10: Explored part of basement in the A-building at Luleå University of Technology. The upper left corner is outside room A1501 and the unexplored exit upwards is the passage that leads to the α-building. The wheelchair moved in a clockwise direction. The circle sector expansion method was used to find a possible path and cover the present available free-space with consecutive circle sectors.

method can handle corridors, split-points, corners, obstacles, clusters, and dead-ends. It is general and is suitable for the exploration of different kinds of unknown environments and not only for the particular test environment described in this paper.

The algorithm and its connected methods are intuitive and based on simple algorithms suitable for a vehicle motion in two dimensions. The method can be used for on-line obstacle avoidance and computation of safety margins. The implementation shows that the CSE method successfully works in practice. The main problems when implementing were spurioses that occurred on the wireless-LAN and sometimes the Java virtual machine ran out of memory.

However the final conclusion, with the wheelchair and its hardware and sensors in mind, is that the CSE method can be used to support wheelchair users in both unknown and known environments.

Acknowledgments

Thanks to Patrik Emilsson, working at Permobil[3], for providing the wheelchair and his help with it.

References

[1] S. Fortune, "A sweepline algorithm for voronoi diagrams," *Algorithmica 2(2)*, pp. 153–174, 1987.

[2] P. Blaer, "Robot path planning using generalized voronoi diagrams." http://www.cs.columbia.edu/ pblaer/projects/path_planner/, Feb 2006.

[3] C. M. Gold and J. Snoeyink, "A one-step crust and skeleton extraction algorithm," *Algorithmica*, vol. 30, pp. 144–163, 2001.

[4] R. Ogniewicz and M. Ilg, "Voronoi skeletons: theory and applications," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR '92*, pp. 63–69, June 1992.

[5] N. Rao, "Robot navigation in unknown generalized polygonal terrains using vision sensors," *IEEE Transactions on Systems, Man and Cybernetics,*, vol. 25, pp. 947–962, June 1995.

[6] R. Mahkovic and T. Slivnik, "Constructing the generalized local voronoi diagram from laser range scanner data," in *IEEE Transactions on Systems, Man and Cybernetics, Part A*, pp. 710 – 719, 2000.

[7] T. Pendragon and L. While, "Path-planning by tessellation of obstacles," in *Proceedings of Conferences in Research and Practice in Information Technology(ACSC'03)* (M. Oudshoorn, ed.), vol. 16, (Adelaide, Australia), 2003.

[8] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *IEEE International Conference on Robotics and Automation (ICRA'05)*, 2005.

[9] E. Acar, H. Choset, and J. Lee, "Sensor-based coverage with extended range detectors," *IEEE Transactions on Robotics and Automation*, vol. 22, Feb 2006.

[10] O. Takahashi and R. Schilling, "Motion planning in a plane using generalized voronoi diagrams," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 143–150, April 1989.

[11] G. Sakellariou, M. Shanahan, and B. Kuipers, "Skeletonisation as mobile robot navigation," in *Towards Autonomic Robotic Systems (TAROS-04),*, 2004.

[3] Permobil AB, http://www.permobil.se, June 2006

[12] S. M. LaValle, *Planning Algorithms.* Cambridge University Press, 2006.

[13] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots in cluttered environments1,” in *IEEE International Conference on Robotics and Automation*, 1990.

[14] E. Haines, *Graphics Gems IV.* Academic Press, 1994.

[15] S. Rönnbäck, D. Rosendahl, and K. Hyyppä, “A matlab/java interface to the mica wheelchair,” *The 1st IFAC Symposium on Telematics Applications in Automation and Robotics*, pp. –, July 2004.

[16] J. A. da Cruz Pinto Gaspar, *Omnidirectional Vision for Mobile Robot Navigation.* PhD thesis, Universidade T´ecnica de Lisboa,Instituto superior T’ecnico, Dec 2002.

[17] C. udas de Wit, H.Khennouf, C.Samson, and O.J.Sordalen, *Nonlinear control design for mobile robots, Nonlinear control for mobile robots*, ch. 5. World Scientific series in Robotics and Intelligent Systems, 1993.

[18] O. Community, “Open scene graph.” www.openscenegraph.org, Dec 2005.

Paper D

Signature Graphs for Effective Localization

Authors:
Sven Rönnbäck, Tomas Berglund, Håkan Fredriksson, Kalevi Hyyppä

Reformatted version of paper originally published in:
Proceedings of the IEEE/RSJ International Conference on Robotics and Automation 2006, Beijing, China

Signature Graphs for Effective Localization

Sven Rönnbäck, Tomas Berglund, Håkan Fredriksson, Kalevi Hyyppä

Abstract

We present a novel method for the localization of an autonomous vehicle in two dimensions. Instead of focussing on properties of obstacles, we emphasize the description of free-space. We introduce the concept of signature graphs which is a compact data structure describing available free-space of an environment. It therefore also represents a map of the environment.

A signature is an edge in the graph and is defined by seven parameters. An edge indicates that it is possible for the vehicle to move between two graph nodes.

The signature graph is flexible and can easily be updated as new information about the environment is collected.

We have with an autonomous wheelchair implemented our methods and tests them with successful results.

1 Introduction

[1]This paper addresses the problem of localization of a vehicle in two dimensions. This problem is important in areas of research and applications such as assistive robotics and industrial mobile robots.

An efficient method for localization gives great benefits and can be used, for instance, for effective navigation [1, 2, 3]. Much research in the area of vehicle localization has already been made [4, 5, 6, 7]. One of the main drawbacks of many methods are the association problems involved. The modeling or representation of different geometric structures, with various complexity, such as points, circles, lines, etc is involved in many solutions [7][8].

We propose a novel and efficient method of solving the problem of efficient localization. Instead of trying to model obstacles, we model the free-space, i.e. space where the vehicle can go free from obstacles.

Given an environment, we build a data structure called a signature graph. The signature graph is built from signatures and it allows efficient computation; for instance the localization of a vehicle.

Voronoi based graphs can be used for representation of routes[9][10].

For improving the efficiency of using autonomous vehicles, such as the wheelchair, we propose methods for localization by introducing what we denote as a *signature*. We show

[1] *This work was supported in part by the EU program Interreg IIIA Nord

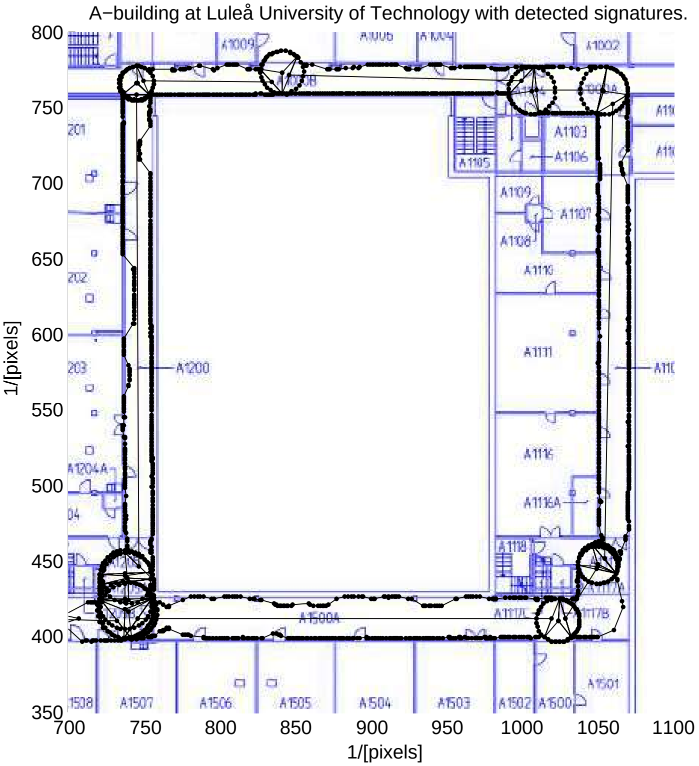

Figure 1: Signatures plotted on a construction plan. The circles represents nodes in the signature graph. Visible differences between the signatures and the map are caused by obstacles found in the corridors such as benches and tables. The signatures span up the free-space available for the wheelchair. It is possible to see that the door to room A1505 is partly opened, check with Figure 11.

how these signatures can be connected to build a *signature graph*, visible in Figure 1. We show that signatures and signature graphs give effective localization.

A signature is an edge in a signature graph and is matched against other signatures with seven parameters. These parameters are uniquely defined by circle sectors in the local free-space. An edge of the graph is telling that it is possible for the vehicle to move between the two nodes in the graph.

The main advantage of using a signature graph for localization compared to earlier presented localization methods is that it efficiently parameterizes the free-space available for navigation. Furthermore, the signature graph can be used for more purposes than efficient localization. Our proposed methods and data structures are suitable for different kinds of optimization and for visualization of data.

We present definitions of signatures and the signature graph. Different methods, benefits and, possibilities for the use of signature graphs are proposed. An implementation that produces signature graphs has been implemented.

The data structure and algorithms were successfully tested in a real-life environment using our semi-autonomous wheelchair.

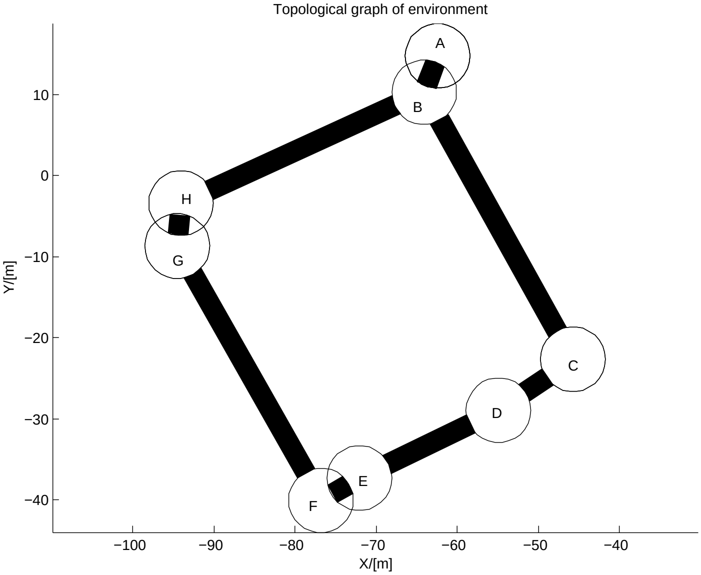

Figure 2: A topological metric graph of A-building at Luleå University of Technology. The graph has eight nodes,$\{A, ..., H\}$ and seven edges.

2 The Signature Graph

The underlying idea is to compactly represent available free-space for a vehicle.

They are detected as uniquely defined identifiers in two dimensions. These signatures are connections between two circle chords.

A signature marks the possibility to move between two graph nodes is used to define an edge in the signature graph.

A signature or node is taken to build a vertex of the graph G and free space between two nodes is taken to build an edge in the graph, see Figure 2 and compare it to Figure 1. Therefore, a signature graph $G = (V, E)$ consists of nodes or vertices $V = \{A, B, ..., H\}$ (circle sectors) and edges E (free-space).

Graph theory is a wide research area of its own [11]. There are numerous efficient ways of using, representing, and computing with graphs [12][13].

By representing an environment as a signature graph, we have the possibility to use powerful algorithms from graph theory research area.

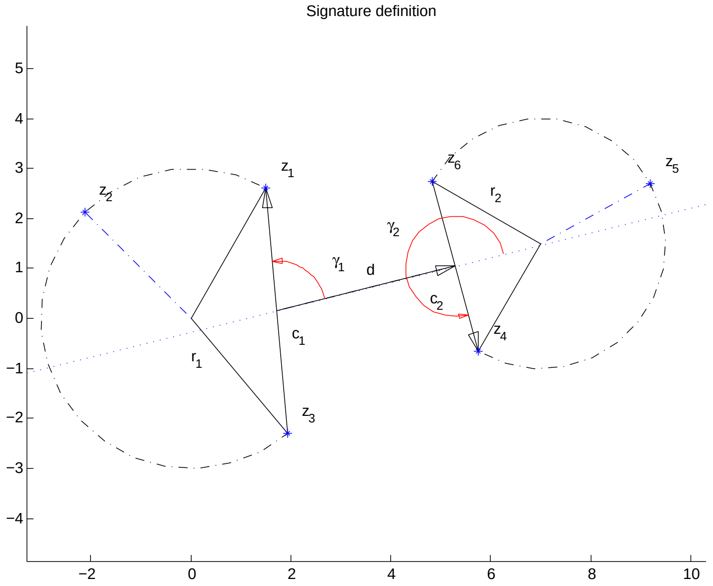

Figure 3: The definition of a signature is based on the angles between the chord vectors and the vector connecting the two circle sectors. The chord lengths and the radius of the circle sectors are also contained in the definition.

2.1 Definition of a Signature

A signature is defined by the seven parameters, see Figure 3.

$$signaure = \{r_1, r_2, d, \gamma_1, \gamma_2, c_1, c_2\}.$$

The radii of the two circles $\{r_1, r_2\}$, the two chord lengths $\{c_1, c_2\}$, the chord angles relative to the axis of the signature, $\{\gamma_1, \gamma_2\}$, the length $\{d = |z_d|\}$ and the length of the two circle chords c_1 and c_2,

$$c_1 = |z_1 - z_3|, \quad c_2 = |z_4 - z_6|.$$

The frame attached to the signature has its origin at $(z_1 + z_3)/2$. The length d of a signature is defined as the distance between the mid points of the circle chords,

$$z_d = \frac{(z_4 + z_6) - (z_1 + z_3)}{2}, \quad d = |z_d|.$$

The angles of the chords relative to the $\hat{X}$-axis of the signature are

$$\gamma_1 = \angle(z_1 - z_3)\bar{z}_d, \quad \gamma_2 = \angle(z_4 - z_6)\bar{z}_d.$$

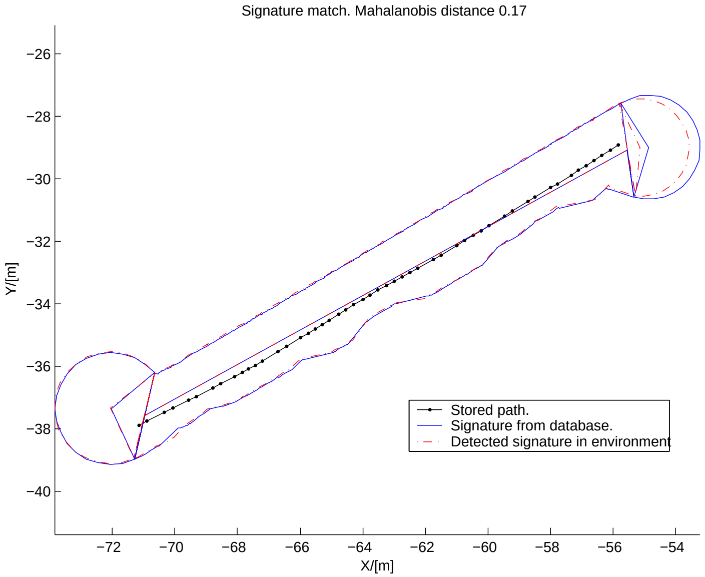

Figure 4: A signature, plotted with a solid line, was detected from collected range data. It was matched against reference signatures, stored in a signature graph, using the Mahalanobis distance. The reference signature is plotted dashed. Different trajectories can be stored in the signature object; one is plotted as the dotted solid line.

2.2 Matching of Signatures

When a signature is defined and a signature graph is built, it is necessary to have an efficient tool for the localization of a vehicle which detects new signatures from range data collected by the scanning laser.

Signatures are identified from environmental data collected by a range scanning laser. Figure 5 shows a detected signature together with the laser range data. A found signature is matched against reference signatures for effective localization. A found signature is matched to reference signature, stored in a database, by the Mahalanobis distance [14]. Let X_s be a vector with the seven signature parameters.

$$X_s = \left[\begin{array}{ccccccc} r_1 & r_2 & c_1 & c_2 & \gamma_1 & \gamma_2 & d \end{array} \right]$$

Let Z_s be the a vector that has the observed signature properties and $X_s(n)$ be the known properties of signature n. Innovation covariance $S_s(n)$ is calculated as

$$S_s(n) = E[X_s(n)X_s(n)^T] + E[Z_s Z_s^T].$$

The Mahalanobis distance,

$$\nu_{sig}(n) = (Z_s - X_s(n))^T S_s(n)^{-1}(Z_s - X_s(n)),$$

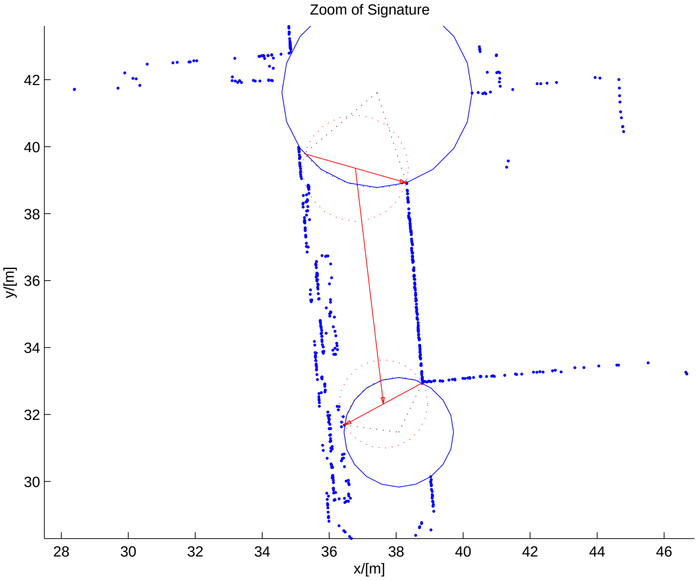

Figure 5: A detected signature plotted in its environment. It consists of two circle chords and a connecting line. A signature indicates that there is exists free-space for navigation.

, which is χ^2 distributed as it is assumed the noise involved are normally distributed. The distance variable $\nu_{sig}(n)$ is also dimensionless and compensates for interactions between the variables. We take the match, $\nu_{sig}(n)$, with highest probability and check against tables if the match was significant. The test has seven degrees of freedom and is equal to the number of observed signature parameters.

The confidence level of the χ^2 distribution to a given degrees of freedom is given in statistical tables.

2.3 The Circle Sector Expansion Method

To find signatures we use expanding circle sectors, see Figure 6. The circle sector expansion (CSE) method is intuitive and used to calculate a reduced Voronoi diagram based on two dimensional data [15]. The main idea with the method is to expand circle sectors until they are bounded by data.

It creates a chain of consecutive circle sectors that describe the available free-space for navigation. The centers of the consecutive circles form a path used as an initial path for the vehicle.

Figure 6 shows the first three steps of an circle sector expansion sequence in an environment with four measurements marked with small circles.

The circle sector in sub-figure one marks the initial circle sector. In each expansion step the sector is divided by a measurement into two new circle sectors. The arrows

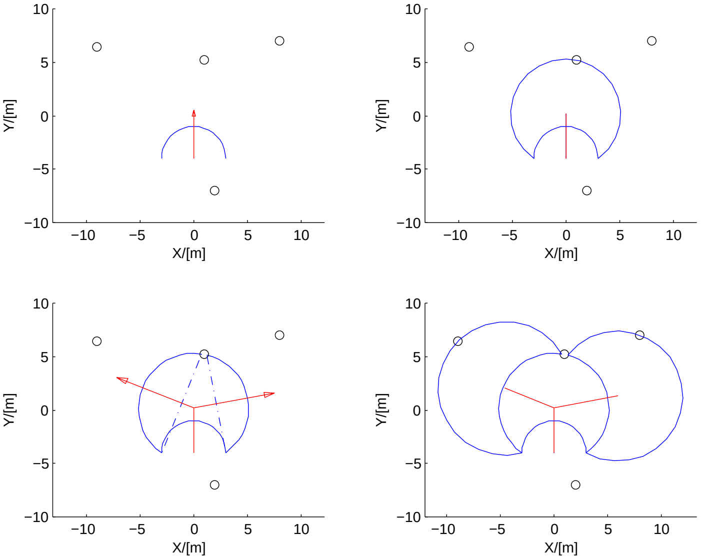

Figure 6: A thee step expansion sequence by the circle sector expansion method. The arrows marks are expanding directions. The connected circle centers form a Voronoi diagram.

attached to the chords show the expanding directions.

Each circle sector k defined by three measurements $z_1(k), z_2(k), z_3(k)$. Parameters $\theta_c(k)$ marks the heading of the circle sector, the sector angle $\beta(k)$ and the circle center $[x_c(k), y_c(k)]^T$. Circle k is has the parameters:

$$cir_c(k) = (x_c(k), y_c(k), r_c(k), \beta_c(k), \theta_c(k)).$$

The complex number z_v is used to calculate the circle center,

$$z_v(k) = [z_1(k) - z_2(k)]\overline{[z_3(k) - z_2(k)]},$$

where $\overline{[z_3(k) - z_2(k)]}$ is the complex conjugate of $[z_3(k) - z_2(k)]$. The derived analytic expression for the circle sector center $z_c(k)$ is

$$z_c(k) = \frac{z_1(k) + z_3(k)}{2} + \frac{[z_v(k) + \overline{z_v(k)}][z_1(k) - z_3(k)]e^{i\pi/2}}{2[z_v(k) - \overline{z_v(k)}]}.$$

The radius is $r_c(k) = |z_c(k) - z_1(k)|$. The circle sector arc is divided into two new sectors by measurement $z_2(k)$.

When a sector is bounded it can either, based on the chord length, stop the expansion or be divided into two new circle sectors.

It is possible for our wheelchair to drive in the environment if the chord length is more than 0.8 m; then the circle sector expansion will continue. If chord length is less

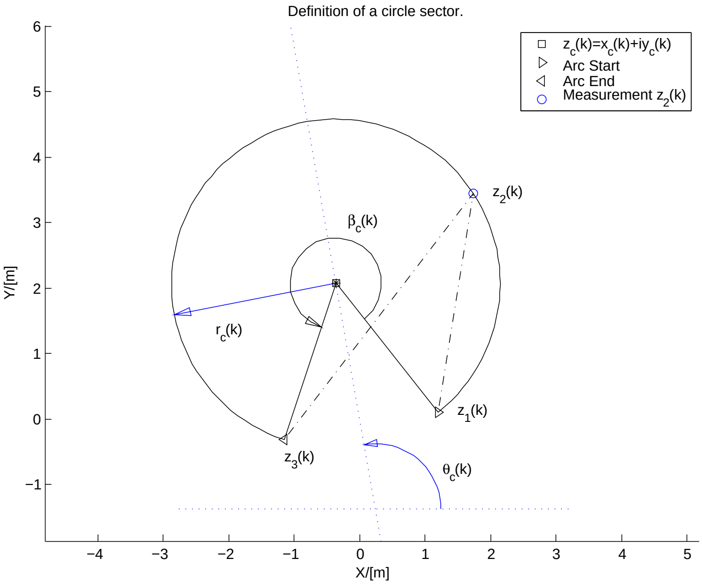

Figure 7: The circle sector is defined by the three complex numbers $z_1(k), z_2(k), z_3(k)$. The center of the circle sector is $z_c(k) = x_c + iy_c$. The radius is $r_c(k)$ and the sector angle $\beta_c(k)$. The heading of the sector is $\theta_c(k)$.

then 0.8 m the circle sector expansion will stop. If the circle is not bounded by any data the radius is set to 10 m.

Figure 8 shows a typical corridor environment with present obstacles. The circle sector expansion method is used to find a path through the environment.

The initial vehicle path is the dotted solid line created by connecting the circle centers. It is smoothed with a Bezier curve [16].

It is also possible to calculate a safety margin from the circle sectors. In Figure 8 they are plotted as dashed lines on each side of the polygon created by connecting the circle centers. The safety boundaries are useful to check if there is a risk for the vehicle to collide with the environment or any obstacle.

3 Implementation and Localization using Signature Graphs

The algorithms and data structures described in previous section are implemented in the Java programming language. The Java implementation is executed in MATLAB[17]. Our code is tested using a wheelchair for performing different tasks, namely moving between signatures.

To evaluate the implementation, the algorithms, and data structures, we use an au-

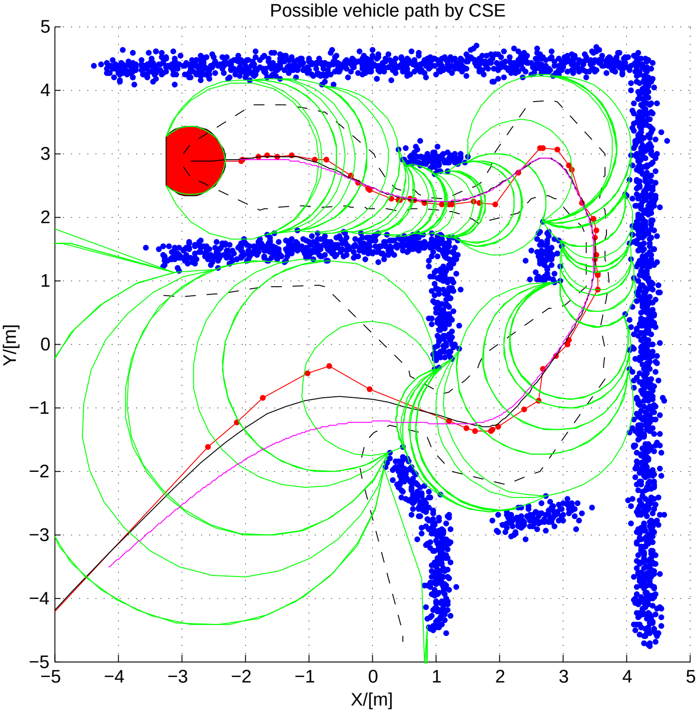

Figure 8: An example of the circle sector expansion in an artificial corridor environment with present obstacles. The CSE method finds a possible path pass the obstacles. In this case the strategy for the expansion process was to always pick the circle sector with the longest chord.

tonomous wheelchair, Figure 9. As range sensor we use a slightly pitch mounted range scanning laser of model SICK LMS200 [18]. The pitch angle is useful for covering the whole height span from floor to sensor placement.

With the laser, the vehicle is able to gather about 1700 two-dimensional data points per second (at 4.7 Hz) with a viewing angle of 180^o and within a range of about 4 m before the laser beam hits the floor. Points positioned less than 0.1 m from the the floor are filtered away and the remaining points are projected onto the floor.

By this, we assume that there are no obstacles, stairs etc, lower than 0.1 m from the floor. Furthermore, the vehicle carries a fiber optic rate gyro, KVH-Ecore 1000, and wheel-encoders. It can be driven semi-autonomously or fully autonomously. We consider it to be a good representative for a broad range of sensor equipped autonomous vehicles.

3.1 Selected Test Environment

Underground corridors at Luleå University of Technology are used test environment, Figure 1. This environment plotted by measured laser scans together with computed

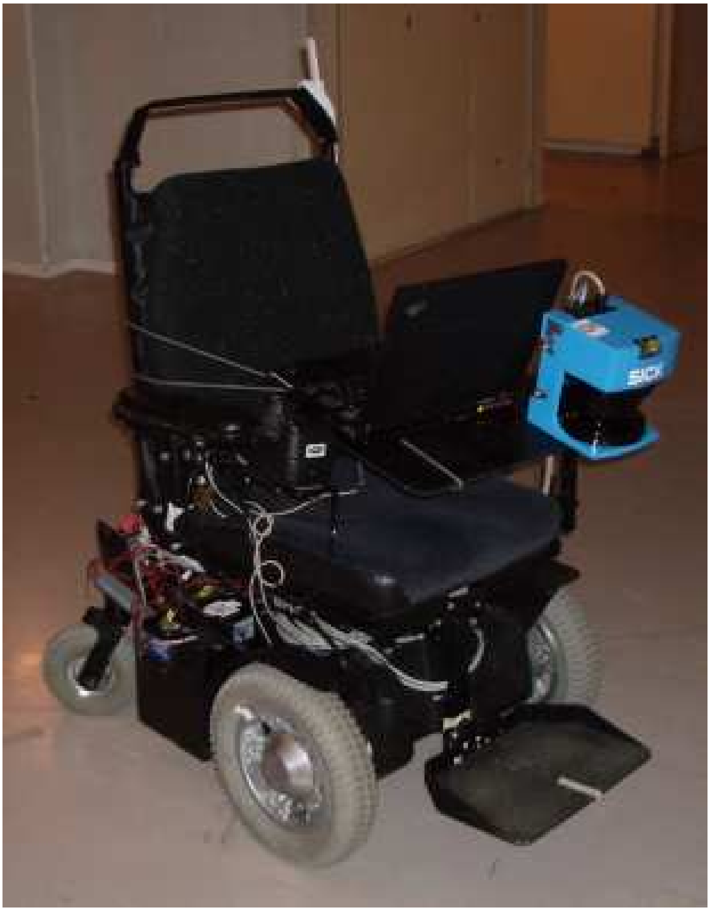

Figure 9: We used a modern wheelchair equipped with a SICK laser for the tests. The laser is pitched down approximately 4^o. It is also equipped with an embedded PC computer that runs a software that collects and time stamps sensor data readings. The computer is accessible over W-LAN, the antenna is visible on the top of the wheelchair. The used fiber optic gyro is visible under the seat between the front and back wheel.

signatures is shown in Figure 5. The chosen environment involves corridors, straight and non-straight crossings, and corners.

3.2 Navigation Between two Signatures

We have built a signature graph of the underground environment, Figure 1. We want to use the signature graph in order to move the vehicle between two signatures. Doing this, the vehicle in turn has to localize itself using the signature graph and its signatures.

Until the vehicle can localize itself at the next signature, the vehicle has to traverse

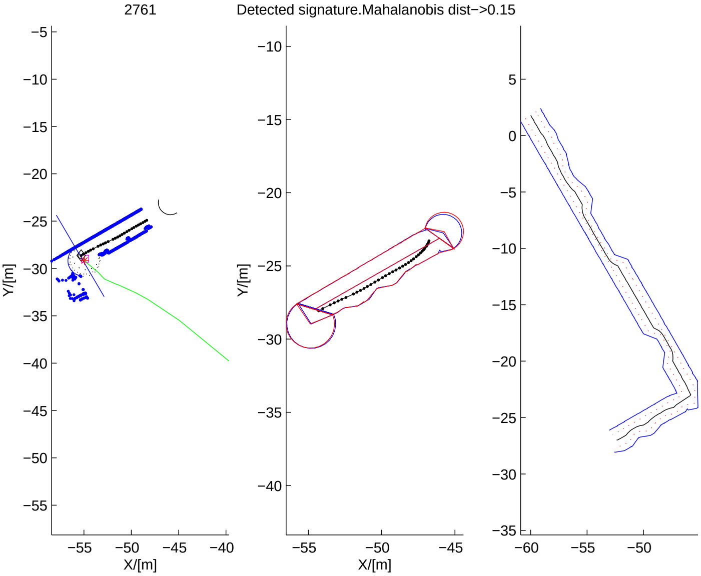

Figure 10: 1) The vehicle in exploring mode. The small half circle is the wheelchair and the dotted solid line behind it is the trajectory. 2) The last detected signature that was successfully matched against reference signatures. 3) Estimated free-space reconstructed from information stored in signatures.

edges, or free-space between consecutive signatures. If properties of the environment along these edges are unknown to the vehicle or if they are changing with time, the vehicle has to work with on-line methods while traversing free space. Figure 10 shows online exploration using the circle sector expansion method.

In sub figure 1 the wheelchair, represented by a small half circle, enters an intersection. Collected data by the range scanning laser are plotted near the vehicle. The most recent trajectory made by the vehicle is plotted as a dotted solid line.

The circle sector expansion method is used to find paths in the data. A new possible paths to the south east is plotted as solid line from the vehicle position. The circle expansion process is stopped if the center of a new circle sector passes the solid boundary line.

In sub figure 2 a signature detected in collected range data is plotted and successfully matched against known signatures stored in a signature graph. The value of the Mahalanobis distance $\nu_{sig}(n)$ is found in the title. The start of the detected signature is plotted as the half circle to the right in sub figure 1.

Free-space and safety boundaries calculated from the consecutive circle sectors expanded in collected range data are shown in sub figure 3.

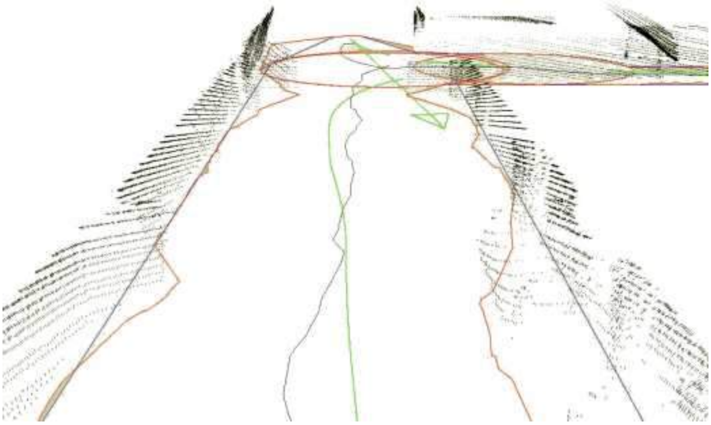

Figure 11: A picture grabbed from the 3D software with the signature environment plotted. The wheelchair trajectory is visible as the smooth line near the center. The centered line that follows the contours of the walls is created by connecting the centers of the consecutive circles found in the expansion process. On the left side a partly opened door to room A1505 is visible.

3.3 Visualization for Understanding of Data

The Open Scene Graph 3D engine[19] is used to visualize results from tests, Figure 11. The location of the grabbed scene is in the corridor between room A1504 and A1505, see Figure 1. The collected laser scanner data are visible as the dotted strips on the walls. A partly opened door to room A1505 is visible to the left and a bench to the right.

The trajectory made by the wheelchair is the smooth line plotted on floor level. The centered line that follows the contours of the walls is created by the circle sector expansion. The wall contours are calculated from the consecutive circle sectors. The solid straight lines are estimates of walls positions.

The arrow visible near the corner marks a used coordinate system.

4 Conclusions

We have presented a novel method for localization of an autonomous vehicle in two dimensions. Instead of focussing on properties of obstacles, we emphasize the description of free-space. We have introduced the concept free-space signatures using the method of circle sector expansion (CSE) to find the signatures and use them to build a signature graph. A signature graph is a data structure suitable for describing free-space of an environment.

The uniquely defined signatures are edges of the graph and the connection between signatures are the nodes. An edge announces that there is enough free space for a vehicle to move between two graph nodes. The graph can be used for fast computations such as finding the shortest or fastest path to a desired location.

We have successfully implemented our proposed methods and data structures. Our implementations and ideas are tested in real-life using an autonomous wheelchair equipped with a scanning laser. The wheelchair is originally constructed as an aid for elderly or impaired people with special needs. The test environment was chosen such that it contained general surroundings such as corridors, crossings and obstacles.

Using the wheelchair in the chosen environment, we are able to efficiently build and use a signature graph as a representation of the environment. The vehicle we able localizes itself using the signature graph.

5 Future work

There are lots of questions raised and interesting issues discovered during the work behind this paper.

First of all, we have presented a method of compacting the description of an environment into a signature graph. This is indeed a graph and graph theory is a wide and interesting research area.

It would be interesting to see whether and how it would be possible to use more than one vehicle at the same time for exploring an unknown environment and for building a signature graph. The vehicles must then be able to communicate and share signature and graph information.

One approach could then be to benefit wheelchair users in their daily life where the wheelchairs automatically exchange information as they meet. The information could consist of preferred vehicle paths, desired speed and angular velocity, and updates of a signature graphs used for localization and navigation.

Acknowledgement

Thanks Patrik Emilsson, at Permobil AB, for his support and his help with the wheelchair. This work was supported by EU Interreg IIIA Nord.

References

[1] T. Bailey, *Mobile robot localization and mapping in extensive outdoor environments.* PhD thesis, Australian Centre for Field Robotics. Department of Aerospace, Mechanical and Mechatronic Engineering. University of Sydney, 2002.

[2] B. Lisien, D. Morales, D. Silver, G. Kantor, I. M. Rekleitis, and H. Choset, “The hierarchical atlas,” *IEEE Transactions on Robotics and Automation*, vol. 21, pp. 473–481, June 2005.

[3] J. E. Guivant, *Efficient Simultaneous Localization and Mapping in Large Environments.* PhD thesis, University Of Sydney, May 2002.

[4] K. Choset, H.; Nagatani, "Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, 2001.

[5] E. Acar, H. Choset, and J. Lee, "Sensor-based coverage with extended range detectors," *IEEE Transactions on Robotics and Automation*, vol. 22, Feb 2006.

[6] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *IEEE International Conference on Robotics and Automation (ICRA'05)*, 2005.

[7] P. Newman, *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem.* PhD thesis, Australian Centre for Field Robotics, The University of Sydney, 1999.

[8] M. Betke and K. Gurvits, "Mobile robot localization using landmarks," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 135–142, May 1994.

[9] J. O. Wallgrun, *Spatial Coginition IV, Autonomous Construction of Hierarchical Voronoi-Based Route Graph Representations.* Springer-Verlag, Berlin, 2005.

[10] M. Garber and M. C. Lin, eds., *Algorithmic Foundation of Roboitca V; Constraint-Based Motion Planning Using Voronoi Diagrams.* Springer-Verlag, Berlin, 2004.

[11] Whiley, "Journal of graph theory." http://www3.interscience.wiley.com/cgi-bin/jhome/35334, Jan 2006.

[12] N. Rao, S. Kareti, W. Shi, and S. Iyenagar, "Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms," 1993.

[13] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A* search meets graph theory," in *16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05)*, (Vancouver, Canada), 2005.

[14] T. F. E. Wikipedia, "Mahalanobis distance." http://en.wikipedia.org/wiki/Mahalanobis_distance, Feb 2006.

[15] S. Rönnbäck, "The circle sector expansion method," July 2006.

[16] Wikipedia, "de casteljau's algorithm," May 2006.

[17] S. Rönnbäck, D. Rosendahl, and K. Hyyppä, "A matlab/java interface to the mica wheelchair," *The 1st IFAC Symposium on Telematics Applications in Automation and Robotics*, pp. –, July 2004.

[18] SICK, “Sick and lms200 and laser measurement system.” http://www.sick.com, Dec 2003.

[19] O. Community, “Open scene graph.” www.openscenegraph.org, Dec 2005.